GDCM

Generated by Doxygen 1.9.4

1 GDCM Documentation 1

2 Todo List 3

3 Deprecated List 5

4 Bug List 7

5 Namespace Index 9

5.1 Namespace List . . . 9

6 Hierarchical Index 11

6.1 Class Hierarchy . . . 11

7 Class Index 21

7.1 Class List . . . 21

8 File Index 35

8.1 File List . . . 35

9 Namespace Documentation 43

9.1 gdcm Namespace Reference . . . 43

9.1.1 Detailed Description . . . 58

9.1.2 Typedef Documentation . . . 58

9.1.2.1 AEComp . . . 58

9.1.2.2 ASComp . . . 58

9.1.2.3 BOOL_FUNCTION_PFILE_PFILE_POINTER . . . 58

9.1.2.4 CSComp . . . 59

9.1.2.5 DAComp . . . 59

9.1.2.6 DTComp . . . 59

9.1.2.7 FileList . . . 59

9.1.2.8 IconImage . . . 59

9.1.2.9 LOComp . . . 59

9.1.2.10 LTComp . . . 59

9.1.2.11 MacroEntry . . . 60

9.1.2.12 NestedMacroEntries . . . 60

9.1.2.13 PNComp . . . 60

9.1.2.14 SHComp . . . 60

9.1.2.15 STComp . . . 60

9.1.2.16 TMComp . . . 60

9.1.2.17 UCComp . . . 60

9.1.2.18 UIComp . . . 61

9.1.2.19 URComp . . . 61
9.1.2.20 UTComp . . . 61
9.1.3 Enumeration Type Documentation . . . 61
9.1.3.1 CompOperators . . . 61
9.1.3.2 ECharSet . . . 61
9.1.3.3 ENQueryType . . . 62
9.1.3.4 EQueryLevel . . . 62
9.1.3.5 EQueryType . . . 63
9.1.3.6 ERootType . . . 63
9.1.3.7 LodModeType . . . 63
9.1.4 Function Documentation . . . 63
9.1.4.1 add1() . . . 63
9.1.4.2 backslash() . . . 64
9.1.4.3 Clamp() . . . 64
9.1.4.4 clean() . . . 64
9.1.4.5 doround() . . . 64
9.1.4.6 GetVRFromTag() . . . 65
9.1.4.7 operator"!=() [1/2] . . . 65
9.1.4.8 operator"!=() [2/2] . . . 65
9.1.4.9 operator<<() [1/59] . . . 65
9.1.4.10 operator<<() [2/59] . . . 65
9.1.4.11 operator<<() [3/59] . . . 65
9.1.4.12 operator<<() [4/59] . . . 66
9.1.4.13 operator<<() [5/59] . . . 66
9.1.4.14 operator<<() [6/59] . . . 66
9.1.4.15 operator<<() [7/59] . . . 66
9.1.4.16 operator<<() [8/59] . . . 66
9.1.4.17 operator<<() [9/59] . . . 66
9.1.4.18 operator<<() [10/59] . . . 67
9.1.4.19 operator<<() [11/59] . . . 67
9.1.4.20 operator<<() [12/59] . . . 67
9.1.4.21 operator<<() [13/59] . . . 67
9.1.4.22 operator<<() [14/59] . . . 67
9.1.4.23 operator<<() [15/59] . . . 67
9.1.4.24 operator<<() [16/59] . . . 68
9.1.4.25 operator<<() [17/59] . . . 68
9.1.4.26 operator<<() [18/59] . . . 68
9.1.4.27 operator<<() [19/59] . . . 68
9.1.4.28 operator<<() [20/59] . . . 68

9.1.4.29 operator<<() [21/59] . . . 69
9.1.4.30 operator<<() [22/59] . . . 69
9.1.4.31 operator<<() [23/59] . . . 69
9.1.4.32 operator<<() [24/59] . . . 69
9.1.4.33 operator<<() [25/59] . . . 69
9.1.4.34 operator<<() [26/59] . . . 69
9.1.4.35 operator<<() [27/59] . . . 70
9.1.4.36 operator<<() [28/59] . . . 70
9.1.4.37 operator<<() [29/59] . . . 70
9.1.4.38 operator<<() [30/59] . . . 70
9.1.4.39 operator<<() [31/59] . . . 70
9.1.4.40 operator<<() [32/59] . . . 70
9.1.4.41 operator<<() [33/59] . . . 71
9.1.4.42 operator<<() [34/59] . . . 71
9.1.4.43 operator<<() [35/59] . . . 71
9.1.4.44 operator<<() [36/59] . . . 71
9.1.4.45 operator<<() [37/59] . . . 71
9.1.4.46 operator<<() [38/59] . . . 71
9.1.4.47 operator<<() [39/59] . . . 72
9.1.4.48 operator<<() [40/59] . . . 72
9.1.4.49 operator<<() [41/59] . . . 72
9.1.4.50 operator<<() [42/59] . . . 72
9.1.4.51 operator<<() [43/59] . . . 72
9.1.4.52 operator<<() [44/59] . . . 72
9.1.4.53 operator<<() [45/59] . . . 73
9.1.4.54 operator<<() [46/59] . . . 73
9.1.4.55 operator<<() [47/59] . . . 73
9.1.4.56 operator<<() [48/59] . . . 73
9.1.4.57 operator<<() [49/59] . . . 73
9.1.4.58 operator<<() [50/59] . . . 73
9.1.4.59 operator<<() [51/59] . . . 74
9.1.4.60 operator<<() [52/59] . . . 74
9.1.4.61 operator<<() [53/59] . . . 74
9.1.4.62 operator<<() [54/59] . . . 74
9.1.4.63 operator<<() [55/59] . . . 74
9.1.4.64 operator<<() [56/59] . . . 74
9.1.4.65 operator<<() [57/59] . . . 75
9.1.4.66 operator<<() [58/59] . . . 75
9.1.4.67 operator<<() [59/59] . . . 75

9.1.4.68 operator==() . . . 75
9.1.4.69 operator>>() [1/3] . . . 75
9.1.4.70 operator>>() [2/3] . . . 76
9.1.4.71 operator>>() [3/3] . . . 76
9.1.4.72 Round() . . . 76
9.1.4.73 roundat() . . . 76
9.1.4.74 TYPETOENCODING() . . . 76
9.1.4.75 x16printf() . . . 77
9.1.5 Variable Documentation . . . 77
9.1.5.1 GlobalInstance . . . 77
9.1.5.2 VRBINARY . . . 77
9.2 gdcm::network Namespace Reference . . . 77
9.2.1 Enumeration Type Documentation . . . 81
9.2.1.1 EEventID . . . 82
9.2.1.2 EStateID . . . 82
9.2.2 Function Documentation . . . 83
9.2.2.1 GetStateIndex() . . . 83
9.2.3 Variable Documentation . . . 83
9.2.3.1 cMaxEventID . . . 83
9.2.3.2 cMaxStateID . . . 83
9.3 gdcm::SegmentHelper Namespace Reference . . . 84
9.4 gdcm::terminal Namespace Reference . . . 84
9.4.1 Detailed Description . . . 84
9.4.2 Enumeration Type Documentation . . . 85
9.4.2.1 Attribute . . . 85
9.4.2.2 Color . . . 85
9.4.2.3 Mode . . . 85
9.4.3 Function Documentation . . . 86
9.4.3.1 setattribute() . . . 86
9.4.3.2 setbgcolor() . . . 86
9.4.3.3 setfgcolor() . . . 86
9.4.3.4 setmode() . . . 86

10 Class Documentation **87**

10.1 gdcm::network::AAbortPDU Class Reference . . . 87
10.1.1 Detailed Description . . . 88
10.1.2 Constructor & Destructor Documentation . . . 88
10.1.2.1 AAbortPDU() . . . 88
10.1.3 Member Function Documentation . . . 88

10.1.3.1 IsLastFragment() . . . 88
10.1.3.2 Print() . . . 88
10.1.3.3 Read() . . . 89
10.1.3.4 SetReason() . . . 89
10.1.3.5 SetSource() . . . 89
10.1.3.6 Size() . . . 89
10.1.3.7 Write() . . . 89
10.2 gdcm::network::AAssociateACPDU Class Reference . . . 90
10.2.1 Detailed Description . . . 91
10.2.2 Member Typedef Documentation . . . 91
10.2.2.1 SizeType . . . 91
10.2.3 Constructor & Destructor Documentation . . . 91
10.2.3.1 AAssociateACPDU() . . . 92
10.2.4 Member Function Documentation . . . 92
10.2.4.1 AddPresentationContextAC() . . . 92
10.2.4.2 GetNumberOfPresentationContextAC() . . . 92
10.2.4.3 GetPresentationContextAC() . . . 92
10.2.4.4 GetUserInformation() . . . 92
10.2.4.5 InitFromRQ() . . . 92
10.2.4.6 IsLastFragment() . . . 93
10.2.4.7 Print() . . . 93
10.2.4.8 Read() . . . 93
10.2.4.9 SetCalledAETitle() . . . 93
10.2.4.10 SetCallingAETitle() . . . 93
10.2.4.11 Size() . . . 93
10.2.4.12 Write() . . . 94
10.2.5 Friends And Related Function Documentation . . . 94
10.2.5.1 AAssociateRQPDU . . . 94
10.3 gdcm::network::AAssociateRJPDU Class Reference . . . 94
10.3.1 Detailed Description . . . 95
10.3.2 Constructor & Destructor Documentation . . . 95
10.3.2.1 AAssociateRJPDU() . . . 95
10.3.3 Member Function Documentation . . . 95
10.3.3.1 IsLastFragment() . . . 96
10.3.3.2 Print() . . . 96
10.3.3.3 Read() . . . 96
10.3.3.4 Size() . . . 96
10.3.3.5 Write() . . . 96
10.4 gdcm::network::AAssociateRQPDU Class Reference . . . 97

10.4.1 Detailed Description . . . 98
10.4.2 Member Typedef Documentation . . . 98
10.4.2.1 PresentationContextArrayType . . . 99
10.4.2.2 SizeType . . . 99
10.4.3 Constructor & Destructor Documentation . . . 99
10.4.3.1 AAssociateRQPDU() [1/2] . . . 99
10.4.3.2 AAssociateRQPDU() [2/2] . . . 99
10.4.4 Member Function Documentation . . . 99
10.4.4.1 AddPresentationContext() . . . 99
10.4.4.2 GetCalledAETitle() . . . 99
10.4.4.3 GetCallingAETitle() . . . 100
10.4.4.4 GetNumberOfPresentationContext() . . . 100
10.4.4.5 GetPresentationContext() . . . 100
10.4.4.6 GetPresentationContextByAbstractSyntax() . . . 100
10.4.4.7 GetPresentationContextByID() . . . 100
10.4.4.8 GetPresentationContexts() . . . 100
10.4.4.9 GetReserved43_74() . . . 101
10.4.4.10 GetUserInformation() . . . 101
10.4.4.11 IsAETitleValid() . . . 101
10.4.4.12 IsLastFragment() . . . 101
10.4.4.13 Print() . . . 101
10.4.4.14 Read() . . . 102
10.4.4.15 SetCalledAETitle() . . . 102
10.4.4.16 SetCallingAETitle() . . . 102
10.4.4.17 SetUserInformation() . . . 102
10.4.4.18 Size() . . . 102
10.4.4.19 Write() . . . 103
10.4.5 Friends And Related Function Documentation . . . 103
10.4.5.1 AAssociateACPDU . . . 103
10.5 gdcm::AbortEvent Class Reference . . . 103
10.6 gdcm::network::AbstractSyntax Class Reference . . . 104
10.6.1 Detailed Description . . . 105
10.6.2 Constructor & Destructor Documentation . . . 105
10.6.2.1 AbstractSyntax() . . . 105
10.6.3 Member Function Documentation . . . 105
10.6.3.1 GetAsDataElement() . . . 105
10.6.3.2 GetName() . . . 105
10.6.3.3 operator==() . . . 105
10.6.3.4 Print() . . . 105

10.6.3.5 Read() . . . 106
10.6.3.6 SetName() . . . 106
10.6.3.7 SetNameFromUID() . . . 106
10.6.3.8 Size() . . . 106
10.6.3.9 Write() . . . 106
10.7 gdcm::AnonymizeEvent Class Reference . . . 107
10.7.1 Detailed Description . . . 108
10.7.2 Member Typedef Documentation . . . 108
10.7.2.1 Self . . . 108
10.7.2.2 Superclass . . . 108
10.7.3 Constructor & Destructor Documentation . . . 108
10.7.3.1 AnonymizeEvent() [1/2] . . . 109
10.7.3.2 ∼AnonymizeEvent() . . . 109
10.7.3.3 AnonymizeEvent() [2/2] . . . 109
10.7.4 Member Function Documentation . . . 109
10.7.4.1 CheckEvent() . . . 109
10.7.4.2 GetEventName() . . . 109
10.7.4.3 GetTag() . . . 110
10.7.4.4 MakeObject() . . . 110
10.7.4.5 operator=() . . . 110
10.7.4.6 SetTag() . . . 110
10.8 gdcm::Anonymizer Class Reference . . . 111
10.8.1 Detailed Description . . . 113
10.8.2 Constructor & Destructor Documentation . . . 114
10.8.2.1 Anonymizer() . . . 114
10.8.2.2 ∼Anonymizer() . . . 114
10.8.3 Member Function Documentation . . . 114
10.8.3.1 BALCPProtect() . . . 114
10.8.3.2 BasicApplicationLevelConfidentialityProfile() . . . 114
10.8.3.3 CanEmptyTag() . . . 115
10.8.3.4 Clear() [1/2] . . . 115
10.8.3.5 Clear() [2/2] . . . 115
10.8.3.6 ClearInternalUIDs() . . . 115
10.8.3.7 Empty() [1/2] . . . 115
10.8.3.8 Empty() [2/2] . . . 116
10.8.3.9 GetBasicApplicationLevelConfidentialityProfileAttributes() . . . 116
10.8.3.10 GetCryptographicMessageSyntax() . . . 116
10.8.3.11 GetFile() . . . 116
10.8.3.12 New() . . . 117

10.8.3.13 RecurseDataSet() . . . 117
10.8.3.14 Remove() [1/2] . . . 117
10.8.3.15 Remove() [2/2] . . . 117
10.8.3.16 RemoveGroupLength() . . . 117
10.8.3.17 RemovePrivateTags() . . . 118
10.8.3.18 RemoveRetired() . . . 118
10.8.3.19 Replace() [1/4] . . . 118
10.8.3.20 Replace() [2/4] . . . 118
10.8.3.21 Replace() [3/4] . . . 118
10.8.3.22 Replace() [4/4] . . . 119
10.8.3.23 SetCryptographicMessageSyntax() . . . 119
10.8.3.24 SetFile() . . . 119
10.9 gdcm::AnyEvent Class Reference . . . 120
10.10 gdcm::network::ApplicationContext Class Reference . . . 121
10.10.1 Detailed Description . . . 121
10.10.2 Constructor & Destructor Documentation . . . 122
10.10.2.1 ApplicationContext() . . . 122
10.10.3 Member Function Documentation . . . 122
10.10.3.1 GetName() . . . 122
10.10.3.2 Print() . . . 122
10.10.3.3 Read() . . . 122
10.10.3.4 SetName() . . . 122
10.10.3.5 Size() . . . 123
10.10.3.6 Write() . . . 123
10.11 gdcm::ApplicationEntity Class Reference . . . 123
10.11.1 Detailed Description . . . 124
10.11.2 Member Function Documentation . . . 124
10.11.2.1 IsValid() . . . 124
10.11.2.2 Print() . . . 124
10.11.2.3 SetBlob() . . . 125
10.11.2.4 Squeeze() . . . 125
10.11.3 Member Data Documentation . . . 125
10.11.3.1 Internal . . . 125
10.11.3.2 MaxLength . . . 125
10.11.3.3 MaxNumberOfComponents . . . 125
10.11.3.4 Padding . . . 125
10.11.3.5 Separator . . . 126
10.12 gdcm::network::AReleaseRPPDU Class Reference . . . 126
10.12.1 Detailed Description . . . 127

10.12.2 Constructor & Destructor Documentation . . . 127
10.12.2.1 AReleaseRPPDU() . . . 127
10.12.3 Member Function Documentation . . . 127
10.12.3.1 IsLastFragment() . . . 127
10.12.3.2 Print() . . . 127
10.12.3.3 Read() . . . 128
10.12.3.4 Size() . . . 128
10.12.3.5 Write() . . . 128
10.13 gdcm::network::AReleaseRQPDU Class Reference . . . 128
10.13.1 Detailed Description . . . 129
10.13.2 Constructor & Destructor Documentation . . . 129
10.13.2.1 AReleaseRQPDU() . . . 129
10.13.3 Member Function Documentation . . . 129
10.13.3.1 IsLastFragment() . . . 130
10.13.3.2 Print() . . . 130
10.13.3.3 Read() . . . 130
10.13.3.4 Size() . . . 130
10.13.3.5 Write() . . . 130
10.14 gdcm::network::ARTIMTimer Class Reference . . . 131
10.14.1 Detailed Description . . . 131
10.14.2 Constructor & Destructor Documentation . . . 131
10.14.2.1 ARTIMTimer() . . . 131
10.14.3 Member Function Documentation . . . 131
10.14.3.1 GetElapsedTime() . . . 132
10.14.3.2 GetHasExpired() . . . 132
10.14.3.3 GetTimeout() . . . 132
10.14.3.4 SetTimeout() . . . 132
10.14.3.5 Start() . . . 132
10.14.3.6 Stop() . . . 132
10.15 gdcm::ASN1 Class Reference . . . 132
10.15.1 Detailed Description . . . 133
10.15.2 Constructor & Destructor Documentation . . . 133
10.15.2.1 ASN1() [1/2] . . . 133
10.15.2.2 ~ASN1() . . . 133
10.15.2.3 ASN1() [2/2] . . . 133
10.15.3 Member Function Documentation . . . 134
10.15.3.1 operator=() . . . 134
10.15.3.2 ParseDump() . . . 134
10.15.3.3 ParseDumpFile() . . . 134

10.15.3.4 TestPBKDF2() . . . 134
10.16 gdcm::network::AsynchronousOperationsWindowSub Class Reference . . . 134
10.16.1 Detailed Description . . . 135
10.16.2 Constructor & Destructor Documentation . . . 135
10.16.2.1 AsynchronousOperationsWindowSub() . . . 135
10.16.3 Member Function Documentation . . . 135
10.16.3.1 Print() . . . 135
10.16.3.2 Read() . . . 135
10.16.3.3 Size() . . . 136
10.16.3.4 Write() . . . 136
10.17 gdcm::Attribute< Group, Element, TVR, TVM > Class Template Reference . . . 136
10.17.1 Detailed Description . . . 138
10.17.2 Member Typedef Documentation . . . 138
10.17.2.1 ArrayType . . . 138
10.17.3 Member Enumeration Documentation . . . 138
10.17.3.1 anonymous enum . . . 138
10.17.4 Member Function Documentation . . . 139
10.17.4.1 GDCM_STATIC_ASSERT() [1/3] . . . 139
10.17.4.2 GDCM_STATIC_ASSERT() [2/3] . . . 139
10.17.4.3 GDCM_STATIC_ASSERT() [3/3] . . . 139
10.17.4.4 GetAsDataElement() . . . 139
10.17.4.5 GetDictVM() . . . 140
10.17.4.6 GetDictVR() . . . 140
10.17.4.7 GetNumberOfValues() . . . 140
10.17.4.8 GetTag() . . . 140
10.17.4.9 GetValue() [1/2] . . . 141
10.17.4.10 GetValue() [2/2] . . . 141
10.17.4.11 GetValues() . . . 141
10.17.4.12 GetVM() . . . 141
10.17.4.13 GetVR() . . . 142
10.17.4.14 operator"!=() . . . 142
10.17.4.15 operator<() . . . 142
10.17.4.16 operator==() . . . 142
10.17.4.17 operator[]() [1/2] . . . 142
10.17.4.18 operator[]() [2/2] . . . 143
10.17.4.19 Print() . . . 143
10.17.4.20 Set() . . . 143
10.17.4.21 SetByteValue() . . . 143
10.17.4.22 SetByteValueNoSwap() . . . 144

10.17.4.23 SetFromDataElement() . . . 144
10.17.4.24 SetFromDataSet() . . . 144
10.17.4.25 SetValue() . . . 145
10.17.4.26 SetValues() . . . 145
10.17.5 Member Data Documentation . . . 145
10.17.5.1 Internal . . . 145
10.18 gdcm::Attribute< Group, Element, TVR, VM::VM1 > Class Template Reference . . . 146
10.18.1 Member Typedef Documentation . . . 147
10.18.1.1 ArrayType . . . 147
10.18.2 Member Enumeration Documentation . . . 147
10.18.2.1 anonymous enum . . . 147
10.18.3 Member Function Documentation . . . 148
10.18.3.1 GDCM_STATIC_ASSERT() [1/4] . . . 148
10.18.3.2 GDCM_STATIC_ASSERT() [2/4] . . . 148
10.18.3.3 GDCM_STATIC_ASSERT() [3/4] . . . 148
10.18.3.4 GDCM_STATIC_ASSERT() [4/4] . . . 148
10.18.3.5 GetAsDataElement() . . . 149
10.18.3.6 GetDictVM() . . . 149
10.18.3.7 GetDictVR() . . . 149
10.18.3.8 GetNumberOfValues() . . . 149
10.18.3.9 GetTag() . . . 149
10.18.3.10 GetValue() [1/2] . . . 149
10.18.3.11 GetValue() [2/2] . . . 150
10.18.3.12 GetValues() . . . 150
10.18.3.13 GetVM() . . . 150
10.18.3.14 GetVR() . . . 150
10.18.3.15 operator"!=() . . . 150
10.18.3.16 operator<() . . . 150
10.18.3.17 operator==() . . . 151
10.18.3.18 Print() . . . 151
10.18.3.19 Set() . . . 151
10.18.3.20 SetByteValue() . . . 151
10.18.3.21 SetByteValueNoSwap() . . . 151
10.18.3.22 SetFromDataElement() . . . 152
10.18.3.23 SetFromDataSet() . . . 152
10.18.3.24 SetValue() . . . 152
10.18.4 Member Data Documentation . . . 152
10.18.4.1 Internal . . . 152
10.19 gdcm::Attribute< Group, Element, TVR, VM::VM1_3 > Class Template Reference . . . 153

10.19.1 Member Function Documentation . . . 153
10.19.1.1 GetVM() . . . 154
10.20 gdcm::Attribute< Group, Element, TVR, VM::VM1_8 > Class Template Reference . . . 154
10.20.1 Member Function Documentation . . . 155
10.20.1.1 GetVM() . . . 155
10.21 gdcm::Attribute< Group, Element, TVR, VM::VM1_n > Class Template Reference . . . 155
10.21.1 Member Typedef Documentation . . . 156
10.21.1.1 ArrayType . . . 156
10.21.2 Constructor & Destructor Documentation . . . 157
10.21.2.1 Attribute() . . . 157
10.21.2.2 ~Attribute() . . . 157
10.21.3 Member Function Documentation . . . 157
10.21.3.1 GDCM_STATIC_ASSERT() [1/3] . . . 157
10.21.3.2 GDCM_STATIC_ASSERT() [2/3] . . . 157
10.21.3.3 GDCM_STATIC_ASSERT() [3/3] . . . 157
10.21.3.4 GetAsDataElement() . . . 158
10.21.3.5 GetDictVM() . . . 158
10.21.3.6 GetDictVR() . . . 158
10.21.3.7 GetNumberOfValues() . . . 158
10.21.3.8 GetTag() . . . 158
10.21.3.9 GetValue() [1/2] . . . 158
10.21.3.10 GetValue() [2/2] . . . 159
10.21.3.11 GetValues() . . . 159
10.21.3.12 GetVM() . . . 159
10.21.3.13 GetVR() . . . 159
10.21.3.14 operator[]() [1/2] . . . 159
10.21.3.15 operator[]() [2/2] . . . 159
10.21.3.16 Print() . . . 160
10.21.3.17 Set() . . . 160
10.21.3.18 SetByteValue() . . . 160
10.21.3.19 SetFromDataElement() . . . 160
10.21.3.20 SetFromDataSet() . . . 160
10.21.3.21 SetNumberOfValues() . . . 161
10.21.3.22 SetValue() [1/2] . . . 161
10.21.3.23 SetValue() [2/2] . . . 161
10.21.3.24 SetValues() . . . 161
10.22 gdcm::Attribute< Group, Element, TVR, VM::VM2_2n > Class Template Reference . . . 162
10.22.1 Member Function Documentation . . . 163
10.22.1.1 GetVM() . . . 163

10.23 gdcm::Attribute< Group, Element, TVR, VM::VM2_n > Class Template Reference . . . 163
10.23.1 Member Function Documentation . . . 164
10.23.1.1 GetVM() . . . 164
10.24 gdcm::Attribute< Group, Element, TVR, VM::VM3_3n > Class Template Reference . . . 165
10.24.1 Member Function Documentation . . . 166
10.24.1.1 GetVM() . . . 166
10.25 gdcm::Attribute< Group, Element, TVR, VM::VM3_n > Class Template Reference . . . 166
10.25.1 Member Function Documentation . . . 167
10.25.1.1 GetVM() . . . 167
10.26 gdcm::AudioCodec Class Reference . . . 168
10.26.1 Detailed Description . . . 169
10.26.2 Constructor & Destructor Documentation . . . 169
10.26.2.1 AudioCodec() . . . 169
10.26.2.2 ~AudioCodec() . . . 169
10.26.3 Member Function Documentation . . . 169
10.26.3.1 CanCode() . . . 169
10.26.3.2 CanDecode() . . . 170
10.26.3.3 Decode() . . . 170
10.27 gdcm::Base64 Class Reference . . . 170
10.27.1 Detailed Description . . . 171
10.27.2 Constructor & Destructor Documentation . . . 171
10.27.2.1 Base64() . . . 171
10.27.3 Member Function Documentation . . . 171
10.27.3.1 Decode() . . . 171
10.27.3.2 Encode() . . . 172
10.27.3.3 GetDecodeLength() . . . 172
10.27.3.4 GetEncodeLength() . . . 172
10.27.3.5 operator=() . . . 173
10.28 gdcm::network::BaseCompositeMessage Class Reference . . . 173
10.28.1 Detailed Description . . . 174
10.28.2 Constructor & Destructor Documentation . . . 174
10.28.2.1 ~BaseCompositeMessage() . . . 174
10.28.3 Member Function Documentation . . . 174
10.28.3.1 ConstructPDV() . . . 174
10.29 gdcm::network::BaseNormalizedMessage Class Reference . . . 175
10.29.1 Detailed Description . . . 176
10.29.2 Constructor & Destructor Documentation . . . 176
10.29.2.1 ~BaseNormalizedMessage() . . . 176
10.29.3 Member Function Documentation . . . 176

10.29.3.1 ConstructPDV() . . . 176
10.30 gdcm::network::BasePDU Class Reference . . . 177
10.30.1 Detailed Description . . . 177
10.30.2 Constructor & Destructor Documentation . . . 178
10.30.2.1 ~BasePDU() . . . 178
10.30.3 Member Function Documentation . . . 178
10.30.3.1 IsLastFragment() . . . 178
10.30.3.2 Print() . . . 178
10.30.3.3 Read() . . . 179
10.30.3.4 Size() . . . 179
10.30.3.5 Write() . . . 179
10.31 gdcm::BaseQuery Class Reference . . . 179
10.31.1 Detailed Description . . . 181
10.31.2 Constructor & Destructor Documentation . . . 181
10.31.2.1 BaseQuery() . . . 181
10.31.2.2 ~BaseQuery() . . . 181
10.31.3 Member Function Documentation . . . 181
10.31.3.1 AddQueryDataSet() . . . 182
10.31.3.2 GetAbstractSyntaxUID() . . . 182
10.31.3.3 GetQueryDataSet() [1/2] . . . 182
10.31.3.4 GetQueryDataSet() [2/2] . . . 182
10.31.3.5 GetSOPInstanceUID() . . . 182
10.31.3.6 Print() . . . 182
10.31.3.7 SetSearchParameter() [1/3] . . . 183
10.31.3.8 SetSearchParameter() [2/3] . . . 183
10.31.3.9 SetSearchParameter() [3/3] . . . 183
10.31.3.10 SetSOPInstanceUID() . . . 183
10.31.3.11 ValidateQuery() . . . 183
10.31.3.12 ValidDataSet() . . . 184
10.31.3.13 WriteHelpFile() . . . 184
10.31.3.14 WriteQuery() . . . 184
10.31.4 Friends And Related Function Documentation . . . 184
10.31.4.1 QueryFactory . . . 184
10.31.5 Member Data Documentation . . . 184
10.31.5.1 mDataSet . . . 184
10.31.5.2 mSopInstanceUID . . . 185
10.32 gdcm::BaseRootQuery Class Reference . . . 185
10.32.1 Detailed Description . . . 186
10.32.2 Constructor & Destructor Documentation . . . 187

10.32.2.1 BaseRootQuery() . . . 187
10.32.2.2 ~BaseRootQuery() . . . 187
10.32.3 Member Function Documentation . . . 187
10.32.3.1 Construct() . . . 187
10.32.3.2 GetQueryLevelFromQueryRoot() . . . 187
10.32.3.3 GetQueryLevelFromString() . . . 187
10.32.3.4 GetQueryLevelString() . . . 188
10.32.3.5 GetTagListByLevel() . . . 188
10.32.3.6 InitializeDataSet() . . . 188
10.32.3.7 ValidateQuery() . . . 188
10.32.4 Friends And Related Function Documentation . . . 189
10.32.4.1 QueryFactory . . . 189
10.32.5 Member Data Documentation . . . 189
10.32.5.1 mHelpDescription . . . 189
10.32.5.2 mImage . . . 189
10.32.5.3 mPatient . . . 189
10.32.5.4 mRootType . . . 189
10.32.5.5 mSeries . . . 189
10.32.5.6 mStudy . . . 190
10.33 gdcm::SegmentHelper::BasicCodedEntry Struct Reference . . . 190
10.33.1 Detailed Description . . . 191
10.33.2 Constructor & Destructor Documentation . . . 191
10.33.2.1 BasicCodedEntry() [1/3] . . . 191
10.33.2.2 BasicCodedEntry() [2/3] . . . 191
10.33.2.3 BasicCodedEntry() [3/3] . . . 192
10.33.3 Member Function Documentation . . . 192
10.33.3.1 IsEmpty() . . . 192
10.33.4 Member Data Documentation . . . 192
10.33.4.1 CM . . . 192
10.33.4.2 CSD . . . 192
10.33.4.3 CSV . . . 193
10.33.4.4 CV . . . 193
10.34 gdcm::BasicOffsetTable Class Reference . . . 193
10.34.1 Detailed Description . . . 194
10.34.2 Constructor & Destructor Documentation . . . 194
10.34.2.1 BasicOffsetTable() . . . 195
10.34.3 Member Function Documentation . . . 195
10.34.3.1 Read() . . . 195
10.34.4 Friends And Related Function Documentation . . . 195

10.34.4.1 operator<< . . . 195
10.35 gdcm::Bitmap Class Reference . . . 196
10.35.1 Detailed Description . . . 198
10.35.2 Member Typedef Documentation . . . 199
10.35.2.1 LUTPtr . . . 199
10.35.3 Constructor & Destructor Documentation . . . 199
10.35.3.1 Bitmap() . . . 199
10.35.3.2 ~Bitmap() . . . 199
10.35.4 Member Function Documentation . . . 199
10.35.4.1 AreOverlaysInPixelData() . . . 199
10.35.4.2 Clear() . . . 199
10.35.4.3 ComputeLossyFlag() . . . 200
10.35.4.4 GetBuffer() . . . 200
10.35.4.5 GetBuffer2() . . . 200
10.35.4.6 GetBufferLength() . . . 200
10.35.4.7 GetColumns() . . . 200
10.35.4.8 GetDataElement() [1/2] . . . 201
10.35.4.9 GetDataElement() [2/2] . . . 201
10.35.4.10 GetDimension() . . . 201
10.35.4.11 GetDimensions() . . . 201
10.35.4.12 GetLUT() [1/2] . . . 201
10.35.4.13 GetLUT() [2/2] . . . 202
10.35.4.14 GetNeedByteSwap() . . . 202
10.35.4.15 GetNumberOfDimensions() . . . 202
10.35.4.16 GetPhotometricInterpretation() . . . 202
10.35.4.17 GetPixelFormat() [1/2] . . . 203
10.35.4.18 GetPixelFormat() [2/2] . . . 203
10.35.4.19 GetPlanarConfiguration() . . . 203
10.35.4.20 GetRows() . . . 203
10.35.4.21 GetTransferSyntax() . . . 203
10.35.4.22 IsEmpty() . . . 204
10.35.4.23 IsLossy() . . . 204
10.35.4.24 IsTransferSyntaxCompatible() . . . 204
10.35.4.25 Print() . . . 204
10.35.4.26 SetColumns() . . . 204
10.35.4.27 SetDataElement() . . . 205
10.35.4.28 SetDimension() . . . 205
10.35.4.29 SetDimensions() . . . 205
10.35.4.30 SetLossyFlag() . . . 205

10.35.4.31 SetLUT() . . . 206
10.35.4.32 SetNeedByteSwap() . . . 206
10.35.4.33 SetNumberOfDimensions() . . . 206
10.35.4.34 SetPhotometricInterpretation() . . . 206
10.35.4.35 SetPixelFormat() . . . 207
10.35.4.36 SetPlanarConfiguration() . . . 207
10.35.4.37 SetRows() . . . 207
10.35.4.38 SetTransferSyntax() . . . 207
10.35.4.39 TryJPEG2000Codec() . . . 208
10.35.4.40 TryJPEG2000Codec2() . . . 208
10.35.4.41 TryJPEGCodec() . . . 208
10.35.4.42 TryJPEGCodec2() . . . 208
10.35.4.43 TryJPEGLSCodec() . . . 208
10.35.4.44 TryKAKADUCodec() . . . 208
10.35.4.45 TryPVRGCodec() . . . 209
10.35.4.46 TryRAWCodec() . . . 209
10.35.4.47 TryRLECodec() . . . 209
10.35.4.48 UnusedBitsPresentInPixelData() . . . 209
10.35.5 Friends And Related Function Documentation . . . 209
10.35.5.1 ImageChangeTransferSyntax . . . 209
10.35.5.2 PixmapReader . . . 209
10.35.6 Member Data Documentation . . . 210
10.35.6.1 Dimensions . . . 210
10.35.6.2 LossyFlag . . . 210
10.35.6.3 LUT . . . 210
10.35.6.4 NeedByteSwap . . . 210
10.35.6.5 NumberOfDimensions . . . 210
10.35.6.6 PF . . . 210
10.35.6.7 PI . . . 211
10.35.6.8 PixelData . . . 211
10.35.6.9 PlanarConfiguration . . . 211
10.35.6.10 TS . . . 211
10.36 gdcm::BitmapToBitmapFilter Class Reference . . . 211
10.36.1 Detailed Description . . . 212
10.36.2 Constructor & Destructor Documentation . . . 213
10.36.2.1 BitmapToBitmapFilter() . . . 213
10.36.2.2 ∼BitmapToBitmapFilter() . . . 213
10.36.3 Member Function Documentation . . . 213
10.36.3.1 GetOutput() . . . 213

10.36.3.2 GetOutputAsBitmap() . . . 213
10.36.3.3 SetInput() . . . 213
10.36.4 Member Data Documentation . . . 214
10.36.4.1 Input . . . 214
10.36.4.2 Output . . . 214
10.37 gdcm::BoxRegion Class Reference . . . 214
10.37.1 Detailed Description . . . 216
10.37.2 Constructor & Destructor Documentation . . . 216
10.37.2.1 BoxRegion() [1/2] . . . 216
10.37.2.2 ∼BoxRegion() . . . 216
10.37.2.3 BoxRegion() [2/2] . . . 216
10.37.3 Member Function Documentation . . . 216
10.37.3.1 Area() . . . 216
10.37.3.2 BoundingBox() . . . 217
10.37.3.3 Clone() . . . 217
10.37.3.4 ComputeBoundingBox() . . . 217
10.37.3.5 Empty() . . . 217
10.37.3.6 GetXMax() . . . 217
10.37.3.7 GetXMin() . . . 218
10.37.3.8 GetYMax() . . . 218
10.37.3.9 GetYMin() . . . 218
10.37.3.10 GetZMax() . . . 218
10.37.3.11 GetZMin() . . . 218
10.37.3.12 IsValid() . . . 218
10.37.3.13 operator=() . . . 219
10.37.3.14 Print() . . . 219
10.37.3.15 SetDomain() . . . 219
10.38 gdcm::ByteBuffer Class Reference . . . 219
10.38.1 Detailed Description . . . 220
10.38.2 Constructor & Destructor Documentation . . . 220
10.38.2.1 ByteBuffer() . . . 220
10.38.3 Member Function Documentation . . . 220
10.38.3.1 Get() . . . 220
10.38.3.2 GetStart() . . . 220
10.38.3.3 ShiftEnd() . . . 221
10.38.3.4 UpdatePosition() . . . 221
10.39 gdcm::ByteSwap< T > Class Template Reference . . . 221
10.39.1 Detailed Description . . . 221
10.39.2 Member Function Documentation . . . 222

10.39.2.1 Swap() . . . 222
10.39.2.2 SwapFromSwapCodeIntoSystem() . . . 222
10.39.2.3 SwapRange() . . . 222
10.39.2.4 SwapRangeFromSwapCodeIntoSystem() . . . 222
10.39.2.5 SystemIsBigEndian() . . . 223
10.39.2.6 SystemIsLittleEndian() . . . 223
10.40 gdcm::ByteSwapFilter Class Reference . . . 223
10.40.1 Detailed Description . . . 223
10.40.2 Constructor & Destructor Documentation . . . 223
10.40.2.1 ByteSwapFilter() [1/2] . . . 224
10.40.2.2 ∼ByteSwapFilter() . . . 224
10.40.2.3 ByteSwapFilter() [2/2] . . . 224
10.40.3 Member Function Documentation . . . 224
10.40.3.1 ByteSwap() . . . 224
10.40.3.2 operator=() . . . 224
10.40.3.3 SetByteSwapTag() . . . 224
10.41 gdcm::ByteValue Class Reference . . . 225
10.41.1 Detailed Description . . . 227
10.41.2 Constructor & Destructor Documentation . . . 227
10.41.2.1 ByteValue() [1/2] . . . 227
10.41.2.2 ByteValue() [2/2] . . . 227
10.41.2.3 ∼ByteValue() . . . 227
10.41.3 Member Function Documentation . . . 228
10.41.3.1 Append() . . . 228
10.41.3.2 Clear() . . . 228
10.41.3.3 ComputeLength() . . . 228
10.41.3.4 Fill() . . . 228
10.41.3.5 GetBuffer() . . . 228
10.41.3.6 GetLength() . . . 229
10.41.3.7 GetPointer() . . . 229
10.41.3.8 GetVoidPointer() [1/2] . . . 229
10.41.3.9 GetVoidPointer() [2/2] . . . 230
10.41.3.10 IsEmpty() . . . 230
10.41.3.11 IsPrintable() . . . 230
10.41.3.12 operator const std::vector< char > &() . . . 230
10.41.3.13 operator=() . . . 230
10.41.3.14 operator==() [1/2] . . . 231
10.41.3.15 operator==() [2/2] . . . 231
10.41.3.16 Print() . . . 231

10.41.3.17 PrintASCII() . . . 231
10.41.3.18 PrintASCIIXML() . . . 231
10.41.3.19 PrintGroupLength() . . . 231
10.41.3.20 PrintHex() . . . 232
10.41.3.21 PrintHexXML() . . . 232
10.41.3.22 PrintPNXML() . . . 232
10.41.3.23 Read() [1/2] . . . 232
10.41.3.24 Read() [2/2] . . . 232
10.41.3.25 SetLength() . . . 232
10.41.3.26 SetLengthOnly() . . . 233
10.41.3.27 Write() [1/2] . . . 233
10.41.3.28 Write() [2/2] . . . 233
10.41.3.29 WriteBuffer() . . . 233
10.42 gdcm::CAPICryptoFactory Class Reference . . . 234
10.42.1 Constructor & Destructor Documentation . . . 234
10.42.1.1 CAPICryptoFactory() . . . 235
10.42.2 Member Function Documentation . . . 235
10.42.2.1 CreateCMSProvider() . . . 235
10.43 gdcm::CAPICryptographicMessageSyntax Class Reference . . . 235
10.43.1 Constructor & Destructor Documentation . . . 236
10.43.1.1 CAPICryptographicMessageSyntax() . . . 236
10.43.1.2 ∼CAPICryptographicMessageSyntax() . . . 237
10.43.2 Member Function Documentation . . . 237
10.43.2.1 Decrypt() . . . 237
10.43.2.2 Encrypt() . . . 237
10.43.2.3 GetCipherType() . . . 237
10.43.2.4 GetInitialized() . . . 238
10.43.2.5 ParseCertificateFile() . . . 238
10.43.2.6 ParseKeyFile() . . . 238
10.43.2.7 SetCipherType() . . . 238
10.43.2.8 SetPassword() . . . 238
10.44 gdcm::network::CEchoRQ Class Reference . . . 239
10.44.1 Detailed Description . . . 240
10.44.2 Member Function Documentation . . . 240
10.44.2.1 ConstructPDV() . . . 240
10.44.3 Member Data Documentation . . . 240
10.44.3.1 AffectedSOPClassUID . . . 240
10.44.3.2 MessageID . . . 240
10.45 gdcm::network::CEchoRSP Class Reference . . . 241

10.45.1 Detailed Description . . . 241
10.45.2 Member Function Documentation . . . 242
10.45.2.1 ConstructPDVByDataSet() . . . 242
10.46 gdcm::network::CFind Class Reference . . . 242
10.46.1 Detailed Description . . . 242
10.47 gdcm::network::CFindCancelRQ Class Reference . . . 242
10.47.1 Detailed Description . . . 243
10.47.2 Member Function Documentation . . . 243
10.47.2.1 ConstructPDVByDataSet() . . . 243
10.48 gdcm::network::CFindRQ Class Reference . . . 244
10.48.1 Detailed Description . . . 244
10.48.2 Member Function Documentation . . . 245
10.48.2.1 ConstructPDV() . . . 245
10.49 gdcm::network::CFindRSP Class Reference . . . 245
10.49.1 Detailed Description . . . 246
10.49.2 Member Function Documentation . . . 246
10.49.2.1 ConstructPDVByDataSet() . . . 246
10.50 gdcm::Cleaner Class Reference . . . 247
10.50.1 Detailed Description . . . 248
10.50.2 Constructor & Destructor Documentation . . . 249
10.50.2.1 Cleaner() . . . 249
10.50.2.2 ∼Cleaner() . . . 249
10.50.3 Member Function Documentation . . . 249
10.50.3.1 Clean() . . . 249
10.50.3.2 Empty() [1/4] . . . 249
10.50.3.3 Empty() [2/4] . . . 249
10.50.3.4 Empty() [3/4] . . . 250
10.50.3.5 Empty() [4/4] . . . 250
10.50.3.6 GetFile() . . . 250
10.50.3.7 New() . . . 250
10.50.3.8 Preserve() . . . 250
10.50.3.9 Remove() [1/4] . . . 250
10.50.3.10 Remove() [2/4] . . . 251
10.50.3.11 Remove() [3/4] . . . 251
10.50.3.12 Remove() [4/4] . . . 251
10.50.3.13 RemoveAllGroupLength() . . . 251
10.50.3.14 RemoveAllIllegal() . . . 251
10.50.3.15 RemoveAllMissingPrivateCreator() . . . 251
10.50.3.16 RemoveMissingPrivateCreator() . . . 252

10.50.3.17 Scrub() [1/4] . . . 252
10.50.3.18 Scrub() [2/4] . . . 252
10.50.3.19 Scrub() [3/4] . . . 252
10.50.3.20 Scrub() [4/4] . . . 252
10.50.3.21 SetFile() . . . 252
10.51 gdcm::network::CMoveCancelRq Class Reference . . . 253
10.51.1 Member Function Documentation . . . 253
10.51.1.1 ConstructPDVByDataSet() . . . 254
10.52 gdcm::network::CMoveRQ Class Reference . . . 254
10.52.1 Detailed Description . . . 255
10.52.2 Member Function Documentation . . . 255
10.52.2.1 ConstructPDV() . . . 255
10.53 gdcm::network::CMoveRSP Class Reference . . . 255
10.53.1 Detailed Description . . . 256
10.53.2 Member Function Documentation . . . 256
10.53.2.1 ConstructPDVByDataSet() . . . 256
10.54 gdcm::Codec Class Reference . . . 257
10.54.1 Detailed Description . . . 257
10.55 gdcm::Coder Class Reference . . . 258
10.55.1 Detailed Description . . . 258
10.55.2 Constructor & Destructor Documentation . . . 258
10.55.2.1 ∼Coder() . . . 259
10.55.3 Member Function Documentation . . . 259
10.55.3.1 CanCode() . . . 259
10.55.3.2 Code() . . . 259
10.55.3.3 InternalCode() . . . 259
10.56 gdcm::CodeString Class Reference . . . 260
10.56.1 Detailed Description . . . 261
10.56.2 Member Typedef Documentation . . . 261
10.56.2.1 const_iterator . . . 261
10.56.2.2 const_reference . . . 261
10.56.2.3 const_reverse_iterator . . . 261
10.56.2.4 difference_type . . . 261
10.56.2.5 iterator . . . 262
10.56.2.6 pointer . . . 262
10.56.2.7 reference . . . 262
10.56.2.8 reverse_iterator . . . 262
10.56.2.9 size_type . . . 262
10.56.2.10 value_type . . . 262

10.56.3 Constructor & Destructor Documentation . . . 262
10.56.3.1 CodeString() [1/4] . . . 263
10.56.3.2 CodeString() [2/4] . . . 263
10.56.3.3 CodeString() [3/4] . . . 263
10.56.3.4 CodeString() [4/4] . . . 263
10.56.4 Member Function Documentation . . . 263
10.56.4.1 GetAsString() . . . 263
10.56.4.2 IsValid() . . . 264
10.56.4.3 Size() . . . 264
10.56.4.4 TrimInternal() . . . 264
10.56.5 Friends And Related Function Documentation . . . 264
10.56.5.1 operator"!= . . . 264
10.56.5.2 operator<< . . . 264
10.56.5.3 operator== . . . 265
10.57 gdcm::Command Class Reference . . . 265
10.57.1 Detailed Description . . . 266
10.57.2 Constructor & Destructor Documentation . . . 266
10.57.2.1 Command() [1/2] . . . 267
10.57.2.2 Command() [2/2] . . . 267
10.57.2.3 ~Command() . . . 267
10.57.3 Member Function Documentation . . . 267
10.57.3.1 Execute() [1/2] . . . 267
10.57.3.2 Execute() [2/2] . . . 267
10.57.3.3 operator=() . . . 268
10.58 gdcm::CommandDataSet Class Reference . . . 268
10.58.1 Detailed Description . . . 269
10.58.2 Constructor & Destructor Documentation . . . 269
10.58.2.1 CommandDataSet() . . . 269
10.58.2.2 ~CommandDataSet() . . . 269
10.58.3 Member Function Documentation . . . 269
10.58.3.1 Insert() . . . 270
10.58.3.2 Read() . . . 270
10.58.3.3 Replace() . . . 270
10.58.3.4 Write() . . . 270
10.58.4 Friends And Related Function Documentation . . . 270
10.58.4.1 operator<< . . . 270
10.59 gdcm::network::CompositeMessageFactory Class Reference . . . 271
10.59.1 Detailed Description . . . 271
10.59.2 Member Function Documentation . . . 271

10.59.2.1 ConstructCEchoRQ() . . . 271
10.59.2.2 ConstructCFindRQ() . . . 271
10.59.2.3 ConstructCMoveRQ() . . . 272
10.59.2.4 ConstructCStoreRQ() . . . 272
10.59.2.5 ConstructCStoreRSP() . . . 272
10.60 gdcm::CompositeNetworkFunctions Class Reference . . . 272
10.60.1 Detailed Description . . . 273
10.60.2 Member Typedef Documentation . . . 273
10.60.2.1 KeyValuePairArrayType . . . 273
10.60.2.2 KeyValuePairType . . . 274
10.60.3 Member Function Documentation . . . 274
10.60.3.1 CEcho() . . . 274
10.60.3.2 CFind() . . . 274
10.60.3.3 CMove() . . . 276
10.60.3.4 ConstructQuery() [1/2] . . . 277
10.60.3.5 ConstructQuery() [2/2] . . . 277
10.60.3.6 CStore() . . . 277
10.61 gdcm::ConstCharWrapper Class Reference . . . 278
10.61.1 Detailed Description . . . 278
10.61.2 Constructor & Destructor Documentation . . . 278
10.61.2.1 ConstCharWrapper() . . . 279
10.61.3 Member Function Documentation . . . 279
10.61.3.1 operator const char *() . . . 279
10.62 gdcm::CP246ExplicitDataElement Class Reference . . . 279
10.62.1 Detailed Description . . . 280
10.62.2 Member Function Documentation . . . 280
10.62.2.1 GetLength() . . . 281
10.62.2.2 Read() . . . 281
10.62.2.3 ReadPreValue() . . . 281
10.62.2.4 ReadValue() . . . 281
10.62.2.5 ReadWithLength() . . . 281
10.63 gdcm::CryptoFactory Class Reference . . . 282
10.63.1 Detailed Description . . . 283
10.63.2 Member Enumeration Documentation . . . 283
10.63.2.1 CryptoLib . . . 283
10.63.3 Constructor & Destructor Documentation . . . 283
10.63.3.1 CryptoFactory() [1/2] . . . 283
10.63.3.2 CryptoFactory() [2/2] . . . 284
10.63.3.3 ~CryptoFactory() . . . 284

10.63.4 Member Function Documentation . . . 284
10.63.4.1 CreateCMSProvider() . . . 284
10.63.4.2 GetFactoryInstance() . . . 284
10.64 gdcm::CryptographicMessageSyntax Class Reference . . . 285
10.64.1 Detailed Description . . . 285
10.64.2 Member Enumeration Documentation . . . 286
10.64.2.1 CipherTypes . . . 286
10.64.3 Constructor & Destructor Documentation . . . 286
10.64.3.1 CryptographicMessageSyntax() [1/2] . . . 286
10.64.3.2 ∼CryptographicMessageSyntax() . . . 286
10.64.3.3 CryptographicMessageSyntax() [2/2] . . . 286
10.64.4 Member Function Documentation . . . 286
10.64.4.1 Decrypt() . . . 287
10.64.4.2 Encrypt() . . . 287
10.64.4.3 GetCipherType() . . . 287
10.64.4.4 operator=() . . . 287
10.64.4.5 ParseCertificateFile() . . . 288
10.64.4.6 ParseKeyFile() . . . 288
10.64.4.7 SetCipherType() . . . 288
10.64.4.8 SetPassword() . . . 288
10.65 gdcm::CSAElement Class Reference . . . 289
10.65.1 Detailed Description . . . 290
10.65.2 Member Typedef Documentation . . . 290
10.65.2.1 DataPtr . . . 291
10.65.3 Constructor & Destructor Documentation . . . 291
10.65.3.1 CSAElement() [1/2] . . . 291
10.65.3.2 CSAElement() [2/2] . . . 291
10.65.4 Member Function Documentation . . . 291
10.65.4.1 GetByteValue() . . . 291
10.65.4.2 GetKey() . . . 292
10.65.4.3 GetName() . . . 292
10.65.4.4 GetNoOfItems() . . . 292
10.65.4.5 GetSyngoDT() . . . 292
10.65.4.6 GetValue() [1/2] . . . 292
10.65.4.7 GetValue() [2/2] . . . 293
10.65.4.8 GetVM() . . . 293
10.65.4.9 GetVR() . . . 293
10.65.4.10 IsEmpty() . . . 293
10.65.4.11 operator<() . . . 293

10.65.4.12 operator=() . . . 294
10.65.4.13 operator==() . . . 294
10.65.4.14 SetByteValue() . . . 294
10.65.4.15 SetKey() . . . 294
10.65.4.16 SetName() . . . 294
10.65.4.17 SetNoOfItems() . . . 294
10.65.4.18 SetSyngoDT() . . . 295
10.65.4.19 SetValue() . . . 295
10.65.4.20 SetVM() . . . 295
10.65.4.21 SetVR() . . . 295
10.65.5 Friends And Related Function Documentation . . . 295
10.65.5.1 operator<< . . . 295
10.65.6 Member Data Documentation . . . 295
10.65.6.1 DataField . . . 296
10.65.6.2 KeyField . . . 296
10.65.6.3 NameField . . . 296
10.65.6.4 NoOfItemsField . . . 296
10.65.6.5 SyngoDTField . . . 296
10.65.6.6 ValueMultiplicityField . . . 296
10.65.6.7 VRField . . . 297
10.66 gdcm::CSAHeader Class Reference . . . 297
10.66.1 Detailed Description . . . 298
10.66.2 Member Enumeration Documentation . . . 298
10.66.2.1 CSAHeaderType . . . 298
10.66.3 Constructor & Destructor Documentation . . . 299
10.66.3.1 CSAHeader() . . . 299
10.66.3.2 ~CSAHeader() . . . 299
10.66.4 Member Function Documentation . . . 299
10.66.4.1 FindCSAElementByName() . . . 299
10.66.4.2 GetCSADataInfo() . . . 300
10.66.4.3 GetCSAEEnd() . . . 300
10.66.4.4 GetCSAElementByName() . . . 300
10.66.4.5 GetCSAImageHeaderInfoTag() . . . 300
10.66.4.6 GetCSASeriesHeaderInfoTag() . . . 301
10.66.4.7 GetDataSet() . . . 301
10.66.4.8 GetFormat() . . . 301
10.66.4.9 GetInterfile() . . . 301
10.66.4.10 GetMrProtocol() . . . 301
10.66.4.11 LoadFromDataElement() . . . 302

10.66.4.12 Print() . . . 302
10.66.5 Friends And Related Function Documentation . . . 302
10.66.5.1 operator<< . . . 302
10.67 gdcm::CSAHeaderDict Class Reference . . . 302
10.67.1 Detailed Description . . . 303
10.67.2 Member Typedef Documentation . . . 303
10.67.2.1 ConstIterator . . . 303
10.67.2.2 Iterator . . . 304
10.67.2.3 MapCSAHeaderDictEntry . . . 304
10.67.3 Constructor & Destructor Documentation . . . 304
10.67.3.1 CSAHeaderDict() [1/2] . . . 304
10.67.3.2 CSAHeaderDict() [2/2] . . . 304
10.67.4 Member Function Documentation . . . 304
10.67.4.1 AddCSAHeaderDictEntry() . . . 304
10.67.4.2 Begin() . . . 304
10.67.4.3 End() . . . 305
10.67.4.4 GetCSAHeaderDictEntry() . . . 305
10.67.4.5 IsEmpty() . . . 305
10.67.4.6 LoadDefault() . . . 305
10.67.4.7 operator=() . . . 305
10.67.5 Friends And Related Function Documentation . . . 305
10.67.5.1 Dicts . . . 305
10.67.5.2 operator<< . . . 306
10.68 gdcm::CSAHeaderDictEntry Class Reference . . . 306
10.68.1 Detailed Description . . . 307
10.68.2 Constructor & Destructor Documentation . . . 307
10.68.2.1 CSAHeaderDictEntry() . . . 307
10.68.3 Member Function Documentation . . . 307
10.68.3.1 GetDescription() . . . 307
10.68.3.2 GetName() . . . 308
10.68.3.3 GetVM() . . . 308
10.68.3.4 GetVR() . . . 308
10.68.3.5 operator<() . . . 308
10.68.3.6 SetDescription() . . . 308
10.68.3.7 SetName() . . . 309
10.68.3.8 SetVM() . . . 309
10.68.3.9 SetVR() . . . 309
10.68.4 Friends And Related Function Documentation . . . 309
10.68.4.1 operator<< . . . 309

10.69 gdcm::CSAHeaderDictException Class Reference . . . 310
10.70 gdcm::network::CStoreRQ Class Reference . . . 310
10.70.1 Detailed Description . . . 311
10.70.2 Member Function Documentation . . . 311
10.70.2.1 ConstructPDV() . . . 312
10.71 gdcm::network::CStoreRSP Class Reference . . . 312
10.71.1 Detailed Description . . . 313
10.71.2 Member Function Documentation . . . 313
10.71.2.1 ConstructPDV() . . . 313
10.72 gdcm::Curve Class Reference . . . 313
10.72.1 Detailed Description . . . 315
10.72.2 Constructor & Destructor Documentation . . . 315
10.72.2.1 Curve() [1/2] . . . 315
10.72.2.2 ~Curve() . . . 315
10.72.2.3 Curve() [2/2] . . . 315
10.72.3 Member Function Documentation . . . 315
10.72.3.1 Decode() . . . 316
10.72.3.2 GetAsPoints() . . . 316
10.72.3.3 GetCurveDataDescriptor() . . . 316
10.72.3.4 GetDataValueRepresentation() . . . 316
10.72.3.5 GetDimensions() . . . 316
10.72.3.6 GetGroup() . . . 316
10.72.3.7 GetNumberOfCurves() . . . 316
10.72.3.8 GetNumberOfPoints() . . . 317
10.72.3.9 GetTypeOfData() . . . 317
10.72.3.10 GetTypeOfDataDescription() . . . 317
10.72.3.11 IsEmpty() . . . 317
10.72.3.12 Print() . . . 317
10.72.3.13 SetCoordinateStartValue() . . . 317
10.72.3.14 SetCoordinateStepValue() . . . 318
10.72.3.15 SetCurve() . . . 318
10.72.3.16 SetCurveDataDescriptor() . . . 318
10.72.3.17 SetCurveDescription() . . . 318
10.72.3.18 SetDataValueRepresentation() . . . 318
10.72.3.19 SetDimensions() . . . 318
10.72.3.20 SetGroup() . . . 319
10.72.3.21 SetNumberOfPoints() . . . 319
10.72.3.22 SetTypeOfData() . . . 319
10.72.3.23 Update() . . . 319

10.73 gdcm::DataElement Class Reference . . . 319
10.73.1 Detailed Description . . . 322
10.73.2 Member Typedef Documentation . . . 323
10.73.2.1 ValuePtr . . . 323
10.73.3 Constructor & Destructor Documentation . . . 323
10.73.3.1 DataElement() [1/2] . . . 323
10.73.3.2 DataElement() [2/2] . . . 323
10.73.4 Member Function Documentation . . . 323
10.73.4.1 Clear() . . . 323
10.73.4.2 Empty() . . . 324
10.73.4.3 GetByteValue() . . . 324
10.73.4.4 GetLength() . . . 324
10.73.4.5 GetSequenceOfFragments() [1/2] . . . 324
10.73.4.6 GetSequenceOfFragments() [2/2] . . . 325
10.73.4.7 GetTag() [1/2] . . . 325
10.73.4.8 GetTag() [2/2] . . . 325
10.73.4.9 GetValue() [1/2] . . . 325
10.73.4.10 GetValue() [2/2] . . . 326
10.73.4.11 GetValueAsSQ() . . . 326
10.73.4.12 GetVL() [1/2] . . . 326
10.73.4.13 GetVL() [2/2] . . . 327
10.73.4.14 GetVR() . . . 327
10.73.4.15 IsEmpty() . . . 327
10.73.4.16 IsUndefinedLength() . . . 328
10.73.4.17 operator<() . . . 328
10.73.4.18 operator=() . . . 328
10.73.4.19 operator==() . . . 328
10.73.4.20 Read() . . . 328
10.73.4.21 ReadOrSkip() . . . 329
10.73.4.22 ReadPreValue() . . . 329
10.73.4.23 ReadValue() . . . 329
10.73.4.24 ReadValueWithLength() . . . 329
10.73.4.25 ReadWithLength() . . . 329
10.73.4.26 SetByteValue() . . . 330
10.73.4.27 SetTag() . . . 330
10.73.4.28 SetValue() . . . 331
10.73.4.29 SetValueFieldLength() . . . 331
10.73.4.30 SetVL() . . . 331
10.73.4.31 SetVLToUndefined() . . . 331

10.73.4.32 SetVR() . . . 332
10.73.4.33 Write() . . . 332
10.73.5 Friends And Related Function Documentation . . . 332
10.73.5.1 operator<< . . . 332
10.73.6 Member Data Documentation . . . 332
10.73.6.1 TagField . . . 333
10.73.6.2 ValueField . . . 333
10.73.6.3 ValueLengthField . . . 333
10.73.6.4 VRField . . . 333
10.74 gdcm::DataElementException Class Reference . . . 334
10.75 gdcm::DataEvent Class Reference . . . 334
10.75.1 Detailed Description . . . 336
10.75.2 Member Typedef Documentation . . . 336
10.75.2.1 Self . . . 336
10.75.2.2 Superclass . . . 336
10.75.3 Constructor & Destructor Documentation . . . 336
10.75.3.1 DataEvent() [1/2] . . . 336
10.75.3.2 ~DataEvent() . . . 337
10.75.3.3 DataEvent() [2/2] . . . 337
10.75.4 Member Function Documentation . . . 337
10.75.4.1 CheckEvent() . . . 337
10.75.4.2 GetData() . . . 337
10.75.4.3 GetDataLength() . . . 337
10.75.4.4 GetEventName() . . . 337
10.75.4.5 MakeObject() . . . 338
10.75.4.6 operator=() . . . 338
10.75.4.7 SetData() . . . 338
10.76 gdcm::DataSet Class Reference . . . 338
10.76.1 Detailed Description . . . 340
10.76.2 Member Typedef Documentation . . . 341
10.76.2.1 ConstIterator . . . 341
10.76.2.2 DataElementSet . . . 341
10.76.2.3 Iterator . . . 341
10.76.2.4 SizeType . . . 341
10.76.3 Member Function Documentation . . . 342
10.76.3.1 Begin() [1/2] . . . 342
10.76.3.2 Begin() [2/2] . . . 342
10.76.3.3 Clear() . . . 342
10.76.3.4 ComputeDataElement() . . . 342

10.76.3.5 ComputeGroupLength() . . . 342
10.76.3.6 End() [1/2] . . . 343
10.76.3.7 End() [2/2] . . . 343
10.76.3.8 FindDataElement() [1/2] . . . 343
10.76.3.9 FindDataElement() [2/2] . . . 343
10.76.3.10 FindNextDataElement() . . . 344
10.76.3.11 GetDataElement() [1/2] . . . 344
10.76.3.12 GetDataElement() [2/2] . . . 344
10.76.3.13 GetDEEnd() . . . 345
10.76.3.14 GetDES() [1/2] . . . 345
10.76.3.15 GetDES() [2/2] . . . 345
10.76.3.16 GetLength() . . . 345
10.76.3.17 GetMediaStorage() . . . 345
10.76.3.18 GetPrivateCreator() . . . 345
10.76.3.19 GetPrivateTag() . . . 346
10.76.3.20 Insert() . . . 346
10.76.3.21 InsertDataElement() . . . 346
10.76.3.22 IsEmpty() . . . 346
10.76.3.23 operator()() . . . 347
10.76.3.24 operator=() . . . 347
10.76.3.25 operator[]() . . . 347
10.76.3.26 Print() . . . 347
10.76.3.27 Read() . . . 347
10.76.3.28 ReadNested() . . . 348
10.76.3.29 ReadSelectedPrivateTags() . . . 348
10.76.3.30 ReadSelectedPrivateTagsWithLength() . . . 348
10.76.3.31 ReadSelectedTags() . . . 348
10.76.3.32 ReadSelectedTagsWithLength() . . . 348
10.76.3.33 ReadUpToTag() . . . 349
10.76.3.34 ReadUpToTagWithLength() . . . 349
10.76.3.35 ReadWithLength() . . . 349
10.76.3.36 Remove() . . . 349
10.76.3.37 Replace() . . . 350
10.76.3.38 ReplaceEmpty() . . . 350
10.76.3.39 Size() . . . 350
10.76.3.40 Write() . . . 351
10.76.4 Friends And Related Function Documentation . . . 351
10.76.4.1 CSAHeader . . . 351
10.76.4.2 operator<< . . . 351

10.77 gdcm::DataSetEvent Class Reference . . . 351
10.77.1 Detailed Description . . . 352
10.77.2 Member Typedef Documentation . . . 353
10.77.2.1 Self . . . 353
10.77.2.2 Superclass . . . 353
10.77.3 Constructor & Destructor Documentation . . . 353
10.77.3.1 DataSetEvent() [1/2] . . . 353
10.77.3.2 ~DataSetEvent() . . . 353
10.77.3.3 DataSetEvent() [2/2] . . . 353
10.77.4 Member Function Documentation . . . 353
10.77.4.1 CheckEvent() . . . 354
10.77.4.2 GetDataSet() . . . 354
10.77.4.3 GetEventName() . . . 354
10.77.4.4 MakeObject() . . . 354
10.77.4.5 operator=() . . . 354
10.77.5 Member Data Documentation . . . 354
10.77.5.1 m_DataSet . . . 355
10.78 gdcm::DataSetHelper Class Reference . . . 355
10.78.1 Detailed Description . . . 355
10.78.2 Member Function Documentation . . . 355
10.78.2.1 ComputeVR() . . . 355
10.79 gdcm::Decoder Class Reference . . . 356
10.79.1 Detailed Description . . . 356
10.79.2 Constructor & Destructor Documentation . . . 356
10.79.2.1 ~Decoder() . . . 357
10.79.3 Member Function Documentation . . . 357
10.79.3.1 CanDecode() . . . 357
10.79.3.2 Decode() . . . 357
10.79.3.3 DecodeByStreams() . . . 357
10.80 gdcm::DefinedTerms Class Reference . . . 358
10.80.1 Detailed Description . . . 358
10.80.2 Constructor & Destructor Documentation . . . 358
10.80.2.1 DefinedTerms() . . . 358
10.81 gdcm::Defs Class Reference . . . 358
10.81.1 Detailed Description . . . 359
10.81.2 Constructor & Destructor Documentation . . . 359
10.81.2.1 Defs() [1/2] . . . 360
10.81.2.2 ~Defs() . . . 360
10.81.2.3 Defs() [2/2] . . . 360

10.81.3 Member Function Documentation . . . 360
10.81.3.1 GetIODFromFile() . . . 360
10.81.3.2 GetIODNameFromMediaStorage() . . . 360
10.81.3.3 GetIODs() [1/2] . . . 360
10.81.3.4 GetIODs() [2/2] . . . 361
10.81.3.5 GetMacros() [1/2] . . . 361
10.81.3.6 GetMacros() [2/2] . . . 361
10.81.3.7 GetModules() [1/2] . . . 361
10.81.3.8 GetModules() [2/2] . . . 361
10.81.3.9 GetTypeFromTag() . . . 362
10.81.3.10 IsEmpty() . . . 362
10.81.3.11 LoadDefaults() . . . 362
10.81.3.12 LoadFromFile() . . . 362
10.81.3.13 operator=() . . . 362
10.81.3.14 Verify() [1/2] . . . 362
10.81.3.15 Verify() [2/2] . . . 363
10.81.4 Friends And Related Function Documentation . . . 363
10.81.4.1 Global . . . 363
10.82 gdcm::DeltaEncodingCodec Class Reference . . . 363
10.82.1 Detailed Description . . . 364
10.82.2 Constructor & Destructor Documentation . . . 364
10.82.2.1 DeltaEncodingCodec() . . . 364
10.82.2.2 ~DeltaEncodingCodec() . . . 364
10.82.3 Member Function Documentation . . . 365
10.82.3.1 CanDecode() . . . 365
10.82.3.2 Decode() [1/2] . . . 365
10.82.3.3 Decode() [2/2] . . . 365
10.83 gdcm::DICOMDIR Class Reference . . . 365
10.83.1 Detailed Description . . . 366
10.83.2 Constructor & Destructor Documentation . . . 366
10.83.2.1 DICOMDIR() [1/2] . . . 366
10.83.2.2 DICOMDIR() [2/2] . . . 366
10.84 gdcm::DICOMDIRGenerator Class Reference . . . 366
10.84.1 Detailed Description . . . 367
10.84.2 Member Typedef Documentation . . . 368
10.84.2.1 FilenamesType . . . 368
10.84.2.2 FilenameType . . . 368
10.84.3 Constructor & Destructor Documentation . . . 368
10.84.3.1 DICOMDIRGenerator() . . . 368

10.84.3.2 ~DICOMDIRGenerator() . . . 368
10.84.4 Member Function Documentation . . . 368
10.84.4.1 AddImageDirectoryRecord() . . . 368
10.84.4.2 AddPatientDirectoryRecord() . . . 369
10.84.4.3 AddSeriesDirectoryRecord() . . . 369
10.84.4.4 AddStudyDirectoryRecord() . . . 369
10.84.4.5 Generate() . . . 369
10.84.4.6 GetFile() . . . 369
10.84.4.7 GetScanner() . . . 369
10.84.4.8 SetDescriptor() . . . 370
10.84.4.9 SetFile() . . . 370
10.84.4.10 SetFilenames() . . . 370
10.84.4.11 SetRootDirectory() . . . 370
10.85 gdcm::Dict Class Reference . . . 371
10.85.1 Detailed Description . . . 371
10.85.2 Member Typedef Documentation . . . 372
10.85.2.1 ConstIterator . . . 372
10.85.2.2 Iterator . . . 372
10.85.2.3 MapDictEntry . . . 372
10.85.3 Constructor & Destructor Documentation . . . 372
10.85.3.1 Dict() [1/2] . . . 372
10.85.3.2 Dict() [2/2] . . . 372
10.85.4 Member Function Documentation . . . 372
10.85.4.1 AddDictEntry() . . . 373
10.85.4.2 Begin() . . . 373
10.85.4.3 End() . . . 373
10.85.4.4 GetDictEntry() . . . 373
10.85.4.5 GetDictEntryByKeyword() . . . 373
10.85.4.6 GetDictEntryByName() . . . 374
10.85.4.7 GetKeywordFromTag() . . . 374
10.85.4.8 IsEmpty() . . . 374
10.85.4.9 LoadDefault() . . . 374
10.85.4.10 operator=() . . . 374
10.85.5 Friends And Related Function Documentation . . . 374
10.85.5.1 Dicts . . . 375
10.85.5.2 operator<< . . . 375
10.86 gdcm::DictConverter Class Reference . . . 375
10.86.1 Detailed Description . . . 376
10.86.2 Member Enumeration Documentation . . . 376

10.86.2.1 OutputTypes . . . 376
10.86.3 Constructor & Destructor Documentation . . . 377
10.86.3.1 DictConverter() . . . 377
10.86.3.2 ~DictConverter() . . . 377
10.86.4 Member Function Documentation . . . 377
10.86.4.1 AddGroupLength() . . . 377
10.86.4.2 Convert() . . . 377
10.86.4.3 ConvertToCXX() . . . 377
10.86.4.4 ConvertToXML() . . . 378
10.86.4.5 GetDictName() . . . 378
10.86.4.6 GetInputFilename() . . . 378
10.86.4.7 GetOutputFilename() . . . 378
10.86.4.8 GetOutputType() . . . 378
10.86.4.9 Readuint16() . . . 378
10.86.4.10 ReadVM() . . . 379
10.86.4.11 ReadVR() . . . 379
10.86.4.12 SetDictName() . . . 379
10.86.4.13 SetInputFileName() . . . 379
10.86.4.14 SetOutputFileName() . . . 379
10.86.4.15 SetOutputType() . . . 379
10.86.4.16 WriteFooter() . . . 380
10.86.4.17 WriteHeader() . . . 380
10.87 gdcm::DictEntry Class Reference . . . 380
10.87.1 Detailed Description . . . 381
10.87.2 Constructor & Destructor Documentation . . . 381
10.87.2.1 DictEntry() . . . 381
10.87.3 Member Function Documentation . . . 381
10.87.3.1 GetKeyword() . . . 382
10.87.3.2 GetName() . . . 382
10.87.3.3 GetRetired() . . . 382
10.87.3.4 GetVM() . . . 382
10.87.3.5 GetVR() . . . 383
10.87.3.6 IsUnique() . . . 383
10.87.3.7 SetElementXX() . . . 383
10.87.3.8 SetGroupXX() . . . 383
10.87.3.9 SetKeyword() . . . 383
10.87.3.10 SetName() . . . 384
10.87.3.11 SetRetired() . . . 384
10.87.3.12 SetVM() . . . 384

10.87.3.13 SetVR() . . . 384
10.87.4 Friends And Related Function Documentation . . . 384
10.87.4.1 Dict . . . 384
10.87.4.2 operator<< . . . 385
10.88 gdcm::DictPrinter Class Reference . . . 385
10.88.1 Detailed Description . . . 386
10.88.2 Constructor & Destructor Documentation . . . 386
10.88.2.1 DictPrinter() . . . 387
10.88.2.2 ~DictPrinter() . . . 387
10.88.3 Member Function Documentation . . . 387
10.88.3.1 Print() . . . 387
10.88.3.2 PrintDataElement2() . . . 387
10.88.3.3 PrintDataSet2() . . . 387
10.89 gdcm::Dicts Class Reference . . . 388
10.89.1 Detailed Description . . . 389
10.89.2 Member Enumeration Documentation . . . 389
10.89.2.1 ConstructorType . . . 389
10.89.3 Constructor & Destructor Documentation . . . 389
10.89.3.1 Dicts() [1/2] . . . 389
10.89.3.2 ~Dicts() . . . 389
10.89.3.3 Dicts() [2/2] . . . 390
10.89.4 Member Function Documentation . . . 390
10.89.4.1 GetConstructorString() . . . 390
10.89.4.2 GetCSAHeaderDict() . . . 390
10.89.4.3 GetDictEntry() [1/2] . . . 390
10.89.4.4 GetDictEntry() [2/2] . . . 390
10.89.4.5 GetPrivateDict() [1/2] . . . 391
10.89.4.6 GetPrivateDict() [2/2] . . . 391
10.89.4.7 GetPublicDict() . . . 391
10.89.4.8 IsEmpty() . . . 391
10.89.4.9 LoadDefaults() . . . 391
10.89.4.10 operator=() . . . 391
10.89.5 Friends And Related Function Documentation . . . 391
10.89.5.1 Global . . . 392
10.89.5.2 operator<< . . . 392
10.90 gdcm::network::DIMSE Class Reference . . . 392
10.90.1 Detailed Description . . . 393
10.90.2 Member Enumeration Documentation . . . 393
10.90.2.1 CommandTypes . . . 393

10.91 gdcm::DirectionCosines Class Reference . . . 394
10.91.1 Detailed Description . . . 394
10.91.2 Constructor & Destructor Documentation . . . 395
10.91.2.1 DirectionCosines() [1/2] . . . 395
10.91.2.2 DirectionCosines() [2/2] . . . 395
10.91.2.3 ∼DirectionCosines() . . . 395
10.91.3 Member Function Documentation . . . 395
10.91.3.1 ComputeDistAlongNormal() . . . 395
10.91.3.2 Cross() . . . 395
10.91.3.3 CrossDot() . . . 396
10.91.3.4 Dot() [1/2] . . . 396
10.91.3.5 Dot() [2/2] . . . 396
10.91.3.6 IsValid() . . . 396
10.91.3.7 Normalize() [1/2] . . . 396
10.91.3.8 Normalize() [2/2] . . . 397
10.91.3.9 operator const double ∗() . . . 397
10.91.3.10 Print() . . . 397
10.91.3.11 SetFromString() . . . 397
10.92 gdcm::Directory Class Reference . . . 397
10.92.1 Detailed Description . . . 398
10.92.2 Member Typedef Documentation . . . 399
10.92.2.1 FilenamesType . . . 399
10.92.2.2 FilenameType . . . 399
10.92.3 Constructor & Destructor Documentation . . . 399
10.92.3.1 Directory() . . . 399
10.92.3.2 ∼Directory() . . . 399
10.92.4 Member Function Documentation . . . 399
10.92.4.1 Explore() . . . 400
10.92.4.2 GetDirectories() . . . 400
10.92.4.3 GetFilenames() . . . 400
10.92.4.4 GetToplevel() . . . 400
10.92.4.5 Load() . . . 401
10.92.4.6 Print() . . . 401
10.92.5 Friends And Related Function Documentation . . . 401
10.92.5.1 operator<< . . . 401
10.93 gdcm::DirectoryHelper Class Reference . . . 402
10.93.1 Detailed Description . . . 402
10.93.2 Member Function Documentation . . . 402
10.93.2.1 GetCTImageSeriesUIDs() . . . 402

10.93.2.2 GetFilenamesFromSeriesUIDs() . . . 403
10.93.2.3 GetFrameOfReference() . . . 403
10.93.2.4 GetMRImageSeriesUIDs() . . . 403
10.93.2.5 GetRTStructSeriesUIDs() . . . 403
10.93.2.6 GetSeriesUIDsBySOPClassUID() . . . 403
10.93.2.7 GetSOPClassUID() . . . 403
10.93.2.8 GetStringValueFromTag() . . . 404
10.93.2.9 LoadImageFromFiles() . . . 404
10.93.2.10 RetrieveSOPInstanceUIDFromIndex() . . . 404
10.93.2.11 RetrieveSOPInstanceUIDFromZPosition() . . . 404
10.94 gdcm::DPath Class Reference . . . 404
10.94.1 Detailed Description . . . 405
10.94.2 Constructor & Destructor Documentation . . . 405
10.94.2.1 DPath() . . . 405
10.94.2.2 ∼DPath() . . . 405
10.94.3 Member Function Documentation . . . 405
10.94.3.1 ConstructFromString() . . . 406
10.94.3.2 IsValid() . . . 406
10.94.3.3 Match() . . . 406
10.94.3.4 operator<() . . . 406
10.94.3.5 Print() . . . 406
10.94.4 Friends And Related Function Documentation . . . 406
10.94.4.1 operator<< . . . 407
10.95 gdcm::DummyValueGenerator Class Reference . . . 407
10.95.1 Detailed Description . . . 407
10.95.2 Member Function Documentation . . . 407
10.95.2.1 Generate() . . . 407
10.96 gdcm::Dumper Class Reference . . . 408
10.96.1 Detailed Description . . . 409
10.96.2 Constructor & Destructor Documentation . . . 409
10.96.2.1 Dumper() . . . 409
10.96.2.2 ∼Dumper() . . . 409
10.97 gdcm::Element< TVR, TVM > Class Template Reference . . . 409
10.97.1 Detailed Description . . . 411
10.97.2 Member Typedef Documentation . . . 412
10.97.2.1 Type . . . 412
10.97.3 Member Function Documentation . . . 412
10.97.3.1 GetAsDataElement() . . . 412
10.97.3.2 GetLength() . . . 412

10.97.3.3 GetValue() [1/2] . . . 412
10.97.3.4 GetValue() [2/2] . . . 413
10.97.3.5 GetValues() . . . 413
10.97.3.6 GetVM() . . . 413
10.97.3.7 GetVR() . . . 413
10.97.3.8 operator[]() . . . 413
10.97.3.9 Print() . . . 414
10.97.3.10 Read() . . . 414
10.97.3.11 Set() . . . 414
10.97.3.12 SetFromDataElement() . . . 414
10.97.3.13 SetNoSwap() . . . 415
10.97.3.14 SetValue() . . . 415
10.97.3.15 Write() . . . 415
10.97.4 Member Data Documentation . . . 415
10.97.4.1 Internal . . . 415
10.98 gdcm::Element< TVR, VM::VM1_2 > Class Template Reference . . . 416
10.98.1 Member Typedef Documentation . . . 417
10.98.1.1 Parent . . . 417
10.98.2 Member Function Documentation . . . 417
10.98.2.1 SetLength() . . . 417
10.99 gdcm::Element< TVR, VM::VM1_n > Class Template Reference . . . 417
10.99.1 Member Typedef Documentation . . . 418
10.99.1.1 Type . . . 418
10.99.2 Constructor & Destructor Documentation . . . 419
10.99.2.1 Element() [1/2] . . . 419
10.99.2.2 ~Element() . . . 419
10.99.2.3 Element() [2/2] . . . 419
10.99.3 Member Function Documentation . . . 419
10.99.3.1 GetAsDataElement() . . . 419
10.99.3.2 GetLength() . . . 419
10.99.3.3 GetValue() [1/2] . . . 420
10.99.3.4 GetValue() [2/2] . . . 420
10.99.3.5 GetVM() . . . 420
10.99.3.6 GetVR() . . . 420
10.99.3.7 operator=() . . . 420
10.99.3.8 operator[]() . . . 420
10.99.3.9 Print() . . . 421
10.99.3.10 Read() . . . 421
10.99.3.11 Set() . . . 421

10.99.3.12 SetArray() . . . 421
10.99.3.13 SetFromDataElement() . . . 421
10.99.3.14 SetLength() . . . 422
10.99.3.15 SetNoSwap() . . . 422
10.99.3.16 SetValue() . . . 422
10.99.3.17 Write() . . . 422
10.99.3.18 WriteASCII() . . . 422
10.100 gdcm::Element< TVR, VM::VM2_2n > Class Template Reference . . . 423
10.100.1 Member Typedef Documentation . . . 424
10.100.1.1 Parent . . . 424
10.100.2 Member Function Documentation . . . 424
10.100.2.1 SetLength() . . . 424
10.101 gdcm::Element< TVR, VM::VM2_n > Class Template Reference . . . 425
10.101.1 Member Typedef Documentation . . . 426
10.101.1.1 Parent . . . 426
10.101.2 Member Function Documentation . . . 426
10.101.2.1 SetLength() . . . 426
10.102 gdcm::Element< TVR, VM::VM3_3n > Class Template Reference . . . 426
10.102.1 Member Typedef Documentation . . . 428
10.102.1.1 Parent . . . 428
10.102.2 Member Function Documentation . . . 428
10.102.2.1 SetLength() . . . 428
10.103 gdcm::Element< TVR, VM::VM3_4 > Class Template Reference . . . 428
10.103.1 Member Typedef Documentation . . . 429
10.103.1.1 Parent . . . 430
10.103.2 Member Function Documentation . . . 430
10.103.2.1 SetLength() . . . 430
10.104 gdcm::Element< TVR, VM::VM3_n > Class Template Reference . . . 430
10.104.1 Member Typedef Documentation . . . 431
10.104.1.1 Parent . . . 431
10.104.2 Member Function Documentation . . . 431
10.104.2.1 SetLength() . . . 432
10.105 gdcm::Element< VR::AS, VM::VM5 > Class Reference . . . 432
10.105.1 Member Function Documentation . . . 432
10.105.1.1 GetLength() . . . 432
10.105.1.2 Print() . . . 432
10.105.2 Member Data Documentation . . . 432
10.105.2.1 Internal . . . 433
10.106 gdcm::Element< VR::OB, VM::VM1 > Class Reference . . . 433

10.107 gdcm::Element< VR::OW, VM::VM1 > Class Reference . . . 434
10.108 gdcm::ElementDisableCombinations< TVR, TVM > Class Template Reference . . . 436
10.108.1 Detailed Description . . . 437
10.109 gdcm::ElementDisableCombinations< VR::OB, VM::VM1_n > Class Reference . . . 437
10.110 gdcm::ElementDisableCombinations< VR::OW, VM::VM1_n > Class Reference . . . 437
10.111 gdcm::EmptyMaskGenerator Class Reference . . . 437
10.111.1 Detailed Description . . . 438
10.111.2 Member Enumeration Documentation . . . 439
10.111.2.1 SOPClassUIDMode . . . 439
10.111.3 Constructor & Destructor Documentation . . . 439
10.111.3.1 EmptyMaskGenerator() . . . 439
10.111.3.2 ~EmptyMaskGenerator() . . . 439
10.111.4 Member Function Documentation . . . 439
10.111.4.1 Execute() . . . 439
10.111.4.2 SetInputDirectory() . . . 440
10.111.4.3 SetOutputDirectory() . . . 440
10.111.4.4 SetSOPClassUIDMode() . . . 440
10.112 gdcm::EncapsulatedDocument Class Reference . . . 440
10.112.1 Detailed Description . . . 441
10.112.2 Constructor & Destructor Documentation . . . 441
10.112.2.1 EncapsulatedDocument() . . . 441
10.113 gdcm::EncodingImplementation< T > Class Template Reference . . . 441
10.113.1 Detailed Description . . . 441
10.114 gdcm::EncodingImplementation< VR::VRASCII > Class Reference . . . 442
10.114.1 Member Function Documentation . . . 442
10.114.1.1 Read() . . . 442
10.114.1.2 ReadComputeLength() . . . 442
10.114.1.3 ReadNoSwap() . . . 443
10.114.1.4 Write() [1/2] . . . 443
10.114.1.5 Write() [2/2] . . . 443
10.115 gdcm::EncodingImplementation< VR::VRBINARY > Class Reference . . . 443
10.115.1 Member Function Documentation . . . 444
10.115.1.1 Read() . . . 444
10.115.1.2 ReadComputeLength() . . . 444
10.115.1.3 ReadNoSwap() . . . 444
10.115.1.4 Write() . . . 444
10.116 gdcm::EndEvent Class Reference . . . 445
10.117 gdcm::EnumeratedValues Class Reference . . . 446
10.117.1 Detailed Description . . . 446

10.117.2 Constructor & Destructor Documentation . . . 446
10.117.2.1 EnumeratedValues() . . . 446
10.118 gdcm::EquipmentManufacturer Class Reference . . . 446
10.118.1 Detailed Description . . . 447
10.118.2 Member Enumeration Documentation . . . 447
10.118.2.1 Type . . . 447
10.118.3 Member Function Documentation . . . 448
10.118.3.1 Compute() . . . 448
10.118.3.2 TypeToString() . . . 448
10.119 gdcm::Event Class Reference . . . 448
10.119.1 Detailed Description . . . 450
10.119.2 Constructor & Destructor Documentation . . . 450
10.119.2.1 Event() [1/2] . . . 450
10.119.2.2 ~Event() . . . 450
10.119.2.3 Event() [2/2] . . . 450
10.119.3 Member Function Documentation . . . 450
10.119.3.1 CheckEvent() . . . 450
10.119.3.2 GetEventName() . . . 451
10.119.3.3 MakeObject() . . . 451
10.119.3.4 operator=() . . . 451
10.119.3.5 Print() . . . 451
10.120 gdcm::Exception Class Reference . . . 452
10.120.1 Detailed Description . . . 453
10.120.2 Constructor & Destructor Documentation . . . 453
10.120.2.1 Exception() . . . 453
10.120.2.2 ~Exception() . . . 453
10.120.3 Member Function Documentation . . . 453
10.120.3.1 GetDescription() . . . 454
10.120.3.2 what() . . . 454
10.121 gdcm::ExitEvent Class Reference . . . 454
10.122 gdcm::ExplicitDataElement Class Reference . . . 455
10.122.1 Detailed Description . . . 456
10.122.2 Member Function Documentation . . . 457
10.122.2.1 GetLength() . . . 457
10.122.2.2 Read() . . . 457
10.122.2.3 ReadPreValue() . . . 457
10.122.2.4 ReadValue() . . . 457
10.122.2.5 ReadWithLength() . . . 457
10.122.2.6 Write() . . . 458

10.123 gdcm::ExplicitImplicitDataElement Class Reference . . . 458
10.123.1 Detailed Description . . . 459
10.123.2 Member Function Documentation . . . 460
10.123.2.1 GetLength() . . . 460
10.123.2.2 Read() . . . 460
10.123.2.3 ReadPreValue() . . . 460
10.123.2.4 ReadValue() . . . 460
10.123.2.5 ReadWithLength() . . . 460
10.124 gdcm::Fiducials Class Reference . . . 461
10.124.1 Detailed Description . . . 461
10.124.2 Constructor & Destructor Documentation . . . 461
10.124.2.1 Fiducials() . . . 461
10.125 gdcm::File Class Reference . . . 461
10.125.1 Detailed Description . . . 463
10.125.2 Constructor & Destructor Documentation . . . 463
10.125.2.1 File() . . . 464
10.125.2.2 ∼File() . . . 464
10.125.3 Member Function Documentation . . . 464
10.125.3.1 GetDataSet() [1/2] . . . 464
10.125.3.2 GetDataSet() [2/2] . . . 464
10.125.3.3 GetHeader() [1/2] . . . 465
10.125.3.4 GetHeader() [2/2] . . . 465
10.125.3.5 Read() . . . 465
10.125.3.6 SetDataSet() . . . 465
10.125.3.7 SetHeader() . . . 465
10.125.3.8 Write() . . . 466
10.125.4 Friends And Related Function Documentation . . . 466
10.125.4.1 operator<< . . . 466
10.126 gdcm::FileAnonymizer Class Reference . . . 466
10.126.1 Detailed Description . . . 467
10.126.2 Constructor & Destructor Documentation . . . 468
10.126.2.1 FileAnonymizer() . . . 468
10.126.2.2 ∼FileAnonymizer() . . . 468
10.126.3 Member Function Documentation . . . 468
10.126.3.1 Empty() . . . 468
10.126.3.2 Remove() . . . 469
10.126.3.3 Replace() [1/2] . . . 469
10.126.3.4 Replace() [2/2] . . . 469
10.126.3.5 SetInputFileName() . . . 469

10.126.3.6 SetOutputFileName() . . . 470
10.126.3.7 Write() . . . 470
10.127 gdcm::FileChangeTransferSyntax Class Reference . . . 470
10.127.1 Detailed Description . . . 471
10.127.2 Constructor & Destructor Documentation . . . 472
10.127.2.1 FileChangeTransferSyntax() . . . 472
10.127.2.2 ~FileChangeTransferSyntax() . . . 472
10.127.3 Member Function Documentation . . . 472
10.127.3.1 Change() . . . 472
10.127.3.2 GetCodec() . . . 473
10.127.3.3 New() . . . 473
10.127.3.4 SetInputFileName() . . . 473
10.127.3.5 SetOutputFileName() . . . 473
10.127.3.6 SetTransferSyntax() . . . 474
10.128 gdcm::FileDecompressLookupTable Class Reference . . . 474
10.128.1 Detailed Description . . . 475
10.128.2 Constructor & Destructor Documentation . . . 475
10.128.2.1 FileDecompressLookupTable() . . . 476
10.128.2.2 ~FileDecompressLookupTable() . . . 476
10.128.3 Member Function Documentation . . . 476
10.128.3.1 Change() . . . 476
10.128.3.2 GetFile() . . . 476
10.128.3.3 GetPixmap() [1/2] . . . 476
10.128.3.4 GetPixmap() [2/2] . . . 476
10.128.3.5 SetFile() . . . 477
10.128.3.6 SetPixmap() . . . 477
10.129 gdcm::FileDerivation Class Reference . . . 477
10.129.1 Detailed Description . . . 478
10.129.2 Constructor & Destructor Documentation . . . 478
10.129.2.1 FileDerivation() . . . 478
10.129.2.2 ~FileDerivation() . . . 478
10.129.3 Member Function Documentation . . . 478
10.129.3.1 AddDerivationDescription() . . . 479
10.129.3.2 AddPurposeOfReferenceCodeSequence() . . . 479
10.129.3.3 AddReference() . . . 479
10.129.3.4 AddSourceImageSequence() . . . 479
10.129.3.5 Derive() . . . 479
10.129.3.6 GetFile() [1/2] . . . 480
10.129.3.7 GetFile() [2/2] . . . 480

10.129.3.8 SetAppendDerivationHistory() . . . 480
10.129.3.9 SetDerivationCodeSequenceCodeValue() . . . 480
10.129.3.10 SetDerivationDescription() . . . 480
10.129.3.11 SetFile() . . . 481
10.129.3.12 SetPurposeOfReferenceCodeSequenceCodeValue() . . . 481
10.130 gdcm::FileExplicitFilter Class Reference . . . 481
10.130.1 Detailed Description . . . 482
10.130.2 Constructor & Destructor Documentation . . . 482
10.130.2.1 FileExplicitFilter() . . . 482
10.130.2.2 ~FileExplicitFilter() . . . 482
10.130.3 Member Function Documentation . . . 482
10.130.3.1 Change() . . . 483
10.130.3.2 ChangeFMI() . . . 483
10.130.3.3 GetFile() . . . 483
10.130.3.4 ProcessDataSet() . . . 483
10.130.3.5 SetChangePrivateTags() . . . 483
10.130.3.6 SetFile() . . . 484
10.130.3.7 SetRecomputeItemLength() . . . 484
10.130.3.8 SetRecomputeSequenceLength() . . . 484
10.130.3.9 SetUseVRUN() . . . 484
10.131 gdcm::FileMetaInformation Class Reference . . . 485
10.131.1 Detailed Description . . . 487
10.131.2 Constructor & Destructor Documentation . . . 487
10.131.2.1 FileMetaInformation() [1/2] . . . 487
10.131.2.2 ~FileMetaInformation() . . . 488
10.131.2.3 FileMetaInformation() [2/2] . . . 488
10.131.3 Member Function Documentation . . . 488
10.131.3.1 AppendImplementationClassUID() . . . 488
10.131.3.2 ComputeDataSetMediaStorageSOPClass() . . . 488
10.131.3.3 ComputeDataSetTransferSyntax() . . . 488
10.131.3.4 Default() . . . 488
10.131.3.5 FillFromDataSet() . . . 489
10.131.3.6 GetDataSetTransferSyntax() . . . 489
10.131.3.7 GetFileMetaInformationVersion() . . . 489
10.131.3.8 GetFullLength() . . . 489
10.131.3.9 GetGDCMImplementationClassUID() . . . 489
10.131.3.10 GetGDCMImplementationVersionName() . . . 489
10.131.3.11 GetGDCMSourceApplicationEntityTitle() . . . 490
10.131.3.12 GetImplementationClassUID() . . . 490

10.131.3.13 GetImplementationVersionName() . . . 490
10.131.3.14 GetMediaStorage() . . . 490
10.131.3.15 GetMediaStorageAsString() . . . 490
10.131.3.16 GetMetaInformationTS() . . . 490
10.131.3.17 GetPreamble() [1/2] . . . 490
10.131.3.18 GetPreamble() [2/2] . . . 491
10.131.3.19 GetSourceApplicationEntityTitle() . . . 491
10.131.3.20 Insert() . . . 491
10.131.3.21 IsValid() . . . 491
10.131.3.22 operator=() . . . 491
10.131.3.23 Read() . . . 491
10.131.3.24 ReadCompat() . . . 492
10.131.3.25 ReadCompatInternal() . . . 492
10.131.3.26 Replace() . . . 492
10.131.3.27 SetDataSetTransferSyntax() . . . 492
10.131.3.28 SetImplementationClassUID() . . . 493
10.131.3.29 SetImplementationVersionName() . . . 493
10.131.3.30 SetPreamble() . . . 493
10.131.3.31 SetSourceApplicationEntityTitle() . . . 493
10.131.3.32 Write() . . . 493
10.131.4 Friends And Related Function Documentation . . . 493
10.131.4.1 operator<< . . . 494
10.131.5 Member Data Documentation . . . 494
10.131.5.1 DataSetMS . . . 494
10.131.5.2 DataSetTS . . . 494
10.131.5.3 MetaInformationTS . . . 494
10.132 gdcm::Filename Class Reference . . . 494
10.132.1 Detailed Description . . . 495
10.132.2 Constructor & Destructor Documentation . . . 495
10.132.2.1 Filename() . . . 495
10.132.3 Member Function Documentation . . . 496
10.132.3.1 EndWith() . . . 496
10.132.3.2 GetExtension() . . . 496
10.132.3.3 GetFileName() . . . 496
10.132.3.4 GetName() . . . 496
10.132.3.5 GetPath() . . . 496
10.132.3.6 IsEmpty() . . . 497
10.132.3.7 IsIdentical() . . . 497
10.132.3.8 Join() . . . 497

10.132.3.9 operator const char *() . . . 497
10.132.3.10 ToUnixSlashes() . . . 497
10.132.3.11 ToWindowsSlashes() . . . 498
10.133 gdcm::FileNameEvent Class Reference . . . 498
10.133.1 Detailed Description . . . 499
10.133.2 Member Typedef Documentation . . . 500
10.133.2.1 Self . . . 500
10.133.2.2 Superclass . . . 500
10.133.3 Constructor & Destructor Documentation . . . 500
10.133.3.1 FileNameEvent() [1/2] . . . 500
10.133.3.2 ~FileNameEvent() . . . 500
10.133.3.3 FileNameEvent() [2/2] . . . 500
10.133.4 Member Function Documentation . . . 500
10.133.4.1 CheckEvent() . . . 501
10.133.4.2 GetEventName() . . . 501
10.133.4.3 GetFileName() . . . 501
10.133.4.4 MakeObject() . . . 501
10.133.4.5 operator=() . . . 501
10.133.4.6 SetFileName() . . . 502
10.134 gdcm::FilenameGenerator Class Reference . . . 502
10.134.1 Detailed Description . . . 503
10.134.2 Member Typedef Documentation . . . 503
10.134.2.1 FilenamesType . . . 503
10.134.2.2 FilenameType . . . 503
10.134.2.3 SizeType . . . 503
10.134.3 Constructor & Destructor Documentation . . . 503
10.134.3.1 FilenameGenerator() . . . 504
10.134.3.2 ~FilenameGenerator() . . . 504
10.134.4 Member Function Documentation . . . 504
10.134.4.1 Generate() . . . 504
10.134.4.2 GetFilename() . . . 504
10.134.4.3 GetFilenames() . . . 504
10.134.4.4 GetNumberOfFilenames() . . . 505
10.134.4.5 GetPattern() . . . 505
10.134.4.6 GetPrefix() . . . 505
10.134.4.7 SetNumberOfFilenames() . . . 505
10.134.4.8 SetPattern() . . . 505
10.134.4.9 SetPrefix() . . . 506
10.135 gdcm::FileSet Class Reference . . . 506

10.135.1 Detailed Description . . . 506
10.135.2 Member Typedef Documentation . . . 506
10.135.2.1 FilesType . . . 507
10.135.2.2 FileType . . . 507
10.135.3 Constructor & Destructor Documentation . . . 507
10.135.3.1 FileSet() . . . 507
10.135.4 Member Function Documentation . . . 507
10.135.4.1 AddFile() [1/2] . . . 507
10.135.4.2 AddFile() [2/2] . . . 507
10.135.4.3 GetFiles() . . . 508
10.135.4.4 SetFiles() . . . 508
10.135.5 Friends And Related Function Documentation . . . 508
10.135.5.1 operator<< . . . 508
10.136 gdcm::FileStreamer Class Reference . . . 508
10.136.1 Detailed Description . . . 510
10.136.2 Constructor & Destructor Documentation . . . 510
10.136.2.1 FileStreamer() . . . 510
10.136.2.2 ~FileStreamer() . . . 510
10.136.3 Member Function Documentation . . . 510
10.136.3.1 AppendToDataElement() . . . 511
10.136.3.2 AppendToGroupDataElement() . . . 511
10.136.3.3 CheckDataElement() . . . 511
10.136.3.4 CheckTemplateFileName() . . . 511
10.136.3.5 New() . . . 512
10.136.3.6 ReserveDataElement() . . . 512
10.136.3.7 ReserveGroupDataElement() . . . 512
10.136.3.8 SetOutputFileName() . . . 512
10.136.3.9 SetTemplateFileName() . . . 512
10.136.3.10 StartDataElement() . . . 513
10.136.3.11 StartGroupDataElement() . . . 513
10.136.3.12 StopDataElement() . . . 513
10.136.3.13 StopGroupDataElement() . . . 513
10.137 gdcm::FileWithName Class Reference . . . 514
10.137.1 Detailed Description . . . 515
10.137.2 Constructor & Destructor Documentation . . . 515
10.137.2.1 FileWithName() . . . 515
10.137.3 Member Data Documentation . . . 515
10.137.3.1 filename . . . 515
10.138 gdcm::FindPatientRootQuery Class Reference . . . 516

10.138.1 Detailed Description . . . 517
10.138.2 Constructor & Destructor Documentation . . . 517
10.138.2.1 FindPatientRootQuery() . . . 517
10.138.3 Member Function Documentation . . . 517
10.138.3.1 GetAbstractSyntaxUID() . . . 517
10.138.3.2 GetTagListByLevel() . . . 517
10.138.3.3 InitializeDataSet() . . . 518
10.138.3.4 ValidateQuery() . . . 518
10.138.4 Friends And Related Function Documentation . . . 518
10.138.4.1 QueryFactory . . . 518
10.139 gdcm::FindStudyRootQuery Class Reference . . . 519
10.139.1 Detailed Description . . . 520
10.139.2 Constructor & Destructor Documentation . . . 520
10.139.2.1 FindStudyRootQuery() . . . 520
10.139.3 Member Function Documentation . . . 520
10.139.3.1 GetAbstractSyntaxUID() . . . 520
10.139.3.2 GetTagListByLevel() . . . 520
10.139.3.3 InitializeDataSet() . . . 521
10.139.3.4 ValidateQuery() . . . 521
10.139.4 Friends And Related Function Documentation . . . 521
10.139.4.1 QueryFactory . . . 521
10.140 gdcm::Fragment Class Reference . . . 522
10.140.1 Detailed Description . . . 523
10.140.2 Constructor & Destructor Documentation . . . 523
10.140.2.1 Fragment() . . . 523
10.140.3 Member Function Documentation . . . 523
10.140.3.1 ComputeLength() . . . 524
10.140.3.2 GetLength() . . . 524
10.140.3.3 Read() . . . 524
10.140.3.4 ReadBacktrack() . . . 524
10.140.3.5 ReadPreValue() . . . 524
10.140.3.6 ReadValue() . . . 525
10.140.3.7 Write() . . . 525
10.140.4 Friends And Related Function Documentation . . . 525
10.140.4.1 operator<< . . . 525
10.141 gdcm::Global Class Reference . . . 525
10.141.1 Detailed Description . . . 526
10.141.2 Constructor & Destructor Documentation . . . 526
10.141.2.1 Global() [1/2] . . . 527

10.141.2.2 ~Global() . . . 527
10.141.2.3 Global() [2/2] . . . 527
10.141.3 Member Function Documentation . . . 527
10.141.3.1 Append() . . . 527
10.141.3.2 GetDefs() . . . 527
10.141.3.3 GetDicts() [1/2] . . . 528
10.141.3.4 GetDicts() [2/2] . . . 528
10.141.3.5 GetInstance() . . . 528
10.141.3.6 LoadResourcesFiles() . . . 528
10.141.3.7 Locate() . . . 529
10.141.3.8 operator=() . . . 529
10.141.3.9 Prepend() . . . 529
10.141.4 Friends And Related Function Documentation . . . 529
10.141.4.1 operator<< . . . 529
10.142 gdcm::GroupDict Class Reference . . . 529
10.142.1 Detailed Description . . . 530
10.142.2 Member Typedef Documentation . . . 530
10.142.2.1 GroupStringVector . . . 530
10.142.3 Constructor & Destructor Documentation . . . 530
10.142.3.1 GroupDict() . . . 531
10.142.3.2 ~GroupDict() . . . 531
10.142.4 Member Function Documentation . . . 531
10.142.4.1 Add() . . . 531
10.142.4.2 GetAbbreviation() . . . 531
10.142.4.3 GetName() . . . 531
10.142.4.4 Insert() . . . 531
10.142.4.5 Size() . . . 532
10.142.5 Friends And Related Function Documentation . . . 532
10.142.5.1 operator<< . . . 532
10.143 gdcm::IconImageFilter Class Reference . . . 532
10.143.1 Detailed Description . . . 533
10.143.2 Constructor & Destructor Documentation . . . 533
10.143.2.1 IconImageFilter() . . . 533
10.143.2.2 ~IconImageFilter() . . . 533
10.143.3 Member Function Documentation . . . 534
10.143.3.1 Extract() . . . 534
10.143.3.2 ExtractIconImages() . . . 534
10.143.3.3 ExtractVeproIconImages() . . . 534
10.143.3.4 GetFile() [1/2] . . . 534

10.143.3.5 GetFile() [2/2] . . . 534
10.143.3.6 GetIconImage() . . . 535
10.143.3.7 GetNumberOfIconImages() . . . 535
10.143.3.8 SetFile() . . . 535
10.144 gdcm::IconImageGenerator Class Reference . . . 535
10.144.1 Detailed Description . . . 536
10.144.2 Constructor & Destructor Documentation . . . 536
10.144.2.1 IconImageGenerator() . . . 537
10.144.2.2 ~IconImageGenerator() . . . 537
10.144.3 Member Function Documentation . . . 537
10.144.3.1 AutoPixelMinMax() . . . 537
10.144.3.2 ConvertRGBToPaletteColor() . . . 537
10.144.3.3 Generate() . . . 537
10.144.3.4 GetIconImage() . . . 538
10.144.3.5 GetPixmap() [1/2] . . . 538
10.144.3.6 GetPixmap() [2/2] . . . 538
10.144.3.7 SetOutputDimensions() . . . 538
10.144.3.8 SetOutsideValuePixel() . . . 538
10.144.3.9 SetPixelMinMax() . . . 539
10.144.3.10 SetPixmap() . . . 539
10.145 gdcm::ignore_char Struct Reference . . . 539
10.145.1 Constructor & Destructor Documentation . . . 539
10.145.1.1 ignore_char() . . . 540
10.145.2 Member Data Documentation . . . 540
10.145.2.1 m_char . . . 540
10.146 gdcm::Image Class Reference . . . 540
10.146.1 Detailed Description . . . 542
10.146.2 Constructor & Destructor Documentation . . . 542
10.146.2.1 Image() . . . 542
10.146.2.2 ~Image() . . . 543
10.146.3 Member Function Documentation . . . 543
10.146.3.1 GetDirectionCosines() [1/2] . . . 543
10.146.3.2 GetDirectionCosines() [2/2] . . . 543
10.146.3.3 GetIntercept() . . . 543
10.146.3.4 GetOrigin() [1/2] . . . 543
10.146.3.5 GetOrigin() [2/2] . . . 544
10.146.3.6 GetSlope() . . . 544
10.146.3.7 GetSpacing() [1/2] . . . 544
10.146.3.8 GetSpacing() [2/2] . . . 544

10.146.3.9 Print() . . . 544
10.146.3.10 SetDirectionCosines() [1/3] . . . 545
10.146.3.11 SetDirectionCosines() [2/3] . . . 545
10.146.3.12 SetDirectionCosines() [3/3] . . . 545
10.146.3.13 SetIntercept() . . . 545
10.146.3.14 SetOrigin() [1/3] . . . 545
10.146.3.15 SetOrigin() [2/3] . . . 546
10.146.3.16 SetOrigin() [3/3] . . . 546
10.146.3.17 SetSlope() . . . 546
10.146.3.18 SetSpacing() [1/2] . . . 546
10.146.3.19 SetSpacing() [2/2] . . . 546
10.147 gdcm::ImageApplyLookupTable Class Reference . . . 547
10.147.1 Detailed Description . . . 549
10.147.2 Constructor & Destructor Documentation . . . 549
10.147.2.1 ImageApplyLookupTable() . . . 549
10.147.2.2 ~ImageApplyLookupTable() . . . 549
10.147.3 Member Function Documentation . . . 549
10.147.3.1 Apply() . . . 549
10.147.3.2 SetRGB8() . . . 549
10.148 gdcm::ImageChangePhotometricInterpretation Class Reference . . . 550
10.148.1 Detailed Description . . . 552
10.148.2 Constructor & Destructor Documentation . . . 552
10.148.2.1 ImageChangePhotometricInterpretation() . . . 552
10.148.2.2 ~ImageChangePhotometricInterpretation() . . . 552
10.148.3 Member Function Documentation . . . 552
10.148.3.1 Change() . . . 553
10.148.3.2 ChangeMonochrome() . . . 553
10.148.3.3 ChangeRGB2YBR() . . . 553
10.148.3.4 ChangeYBR2RGB() . . . 553
10.148.3.5 GetPhotometricInterpretation() . . . 553
10.148.3.6 RGB2YBR() . . . 553
10.148.3.7 SetPhotometricInterpretation() . . . 554
10.148.3.8 YBR2RGB() . . . 554
10.149 gdcm::ImageChangePlanarConfiguration Class Reference . . . 554
10.149.1 Detailed Description . . . 556
10.149.2 Constructor & Destructor Documentation . . . 556
10.149.2.1 ImageChangePlanarConfiguration() . . . 556
10.149.2.2 ~ImageChangePlanarConfiguration() . . . 556
10.149.3 Member Function Documentation . . . 556

10.149.3.1 Change() . . . 556
10.149.3.2 GetPlanarConfiguration() . . . 557
10.149.3.3 RGBPixelsToRGBPlanes() . . . 557
10.149.3.4 RGBPlanesToRGBPixels() . . . 557
10.149.3.5 SetPlanarConfiguration() . . . 557
10.150 gdcm::ImageChangeTransferSyntax Class Reference . . . 558
10.150.1 Detailed Description . . . 560
10.150.2 Constructor & Destructor Documentation . . . 560
10.150.2.1 ImageChangeTransferSyntax() . . . 560
10.150.2.2 ∼ImageChangeTransferSyntax() . . . 560
10.150.3 Member Function Documentation . . . 561
10.150.3.1 Change() . . . 561
10.150.3.2 GetTransferSyntax() . . . 561
10.150.3.3 SetCompressIconImage() . . . 561
10.150.3.4 SetForce() . . . 561
10.150.3.5 SetTransferSyntax() . . . 562
10.150.3.6 SetUserCodec() . . . 562
10.150.3.7 TryJPEG2000Codec() . . . 562
10.150.3.8 TryJPEGCodec() . . . 562
10.150.3.9 TryJPEGLSCodec() . . . 563
10.150.3.10 TryRAWCodec() . . . 563
10.150.3.11 TryRLECodec() . . . 563
10.151 gdcm::ImageCodec Class Reference . . . 563
10.151.1 Detailed Description . . . 566
10.151.2 Member Typedef Documentation . . . 566
10.151.2.1 LUTPtr . . . 566
10.151.3 Constructor & Destructor Documentation . . . 566
10.151.3.1 ImageCodec() . . . 566
10.151.3.2 ∼ImageCodec() . . . 566
10.151.4 Member Function Documentation . . . 566
10.151.4.1 AppendFrameEncode() . . . 567
10.151.4.2 AppendRowEncode() . . . 567
10.151.4.3 CanCode() . . . 567
10.151.4.4 CanDecode() . . . 567
10.151.4.5 CleanupUnusedBits() . . . 568
10.151.4.6 Clone() . . . 568
10.151.4.7 Decode() . . . 568
10.151.4.8 DecodeByStreams() . . . 568
10.151.4.9 DoByteSwap() . . . 569

10.151.4.10 DoInvertMonochrome() . . . 569
10.151.4.11 DoOverlayCleanup() . . . 569
10.151.4.12 DoPaddedCompositePixelCode() . . . 569
10.151.4.13 DoPlanarConfiguration() . . . 569
10.151.4.14 DoSimpleCopy() . . . 569
10.151.4.15 DoYBR() . . . 570
10.151.4.16 DoYBRFull422() . . . 570
10.151.4.17 GetDimensions() . . . 570
10.151.4.18 GetHeaderInfo() . . . 570
10.151.4.19 GetLossyFlag() . . . 570
10.151.4.20 GetLUT() . . . 570
10.151.4.21 GetNeedByteSwap() . . . 571
10.151.4.22 GetNumberOfDimensions() . . . 571
10.151.4.23 GetPhotometricInterpretation() . . . 571
10.151.4.24 GetPixelFormat() [1/2] . . . 571
10.151.4.25 GetPixelFormat() [2/2] . . . 571
10.151.4.26 GetPlanarConfiguration() . . . 571
10.151.4.27 IsFrameEncoder() . . . 572
10.151.4.28 IsLossy() . . . 572
10.151.4.29 IsRowEncoder() . . . 572
10.151.4.30 IsValid() . . . 572
10.151.4.31 SetDimensions() [1/2] . . . 572
10.151.4.32 SetDimensions() [2/2] . . . 572
10.151.4.33 SetLossyFlag() . . . 573
10.151.4.34 SetLUT() . . . 573
10.151.4.35 SetNeedByteSwap() . . . 573
10.151.4.36 SetNeedOverlayCleanup() . . . 573
10.151.4.37 SetNumberOfDimensions() . . . 573
10.151.4.38 SetPhotometricInterpretation() . . . 574
10.151.4.39 SetPixelFormat() . . . 574
10.151.4.40 SetPlanarConfiguration() . . . 574
10.151.4.41 StartEncode() . . . 574
10.151.4.42 StopEncode() . . . 574
10.151.5 Friends And Related Function Documentation . . . 575
10.151.5.1 FileChangeTransferSyntax . . . 575
10.151.5.2 ImageChangePhotometricInterpretation . . . 575
10.151.6 Member Data Documentation . . . 575
10.151.6.1 Dimensions . . . 575
10.151.6.2 LossyFlag . . . 575

10.151.6.3 LUT . . . 575
10.151.6.4 NeedByteSwap . . . 576
10.151.6.5 NeedOverlayCleanup . . . 576
10.151.6.6 NumberOfDimensions . . . 576
10.151.6.7 PF . . . 576
10.151.6.8 PI . . . 576
10.151.6.9 PlanarConfiguration . . . 576
10.151.6.10 RequestPaddedCompositePixelCode . . . 576
10.151.6.11 RequestPlanarConfiguration . . . 577
10.152 gdcm::ImageConverter Class Reference . . . 577
10.152.1 Detailed Description . . . 577
10.152.2 Constructor & Destructor Documentation . . . 577
10.152.2.1 ImageConverter() . . . 577
10.152.2.2 ∼ImageConverter() . . . 578
10.152.3 Member Function Documentation . . . 578
10.152.3.1 Convert() . . . 578
10.152.3.2 GetOuput() . . . 578
10.152.3.3 SetInput() . . . 578
10.153 gdcm::ImageFragmentSplitter Class Reference . . . 579
10.153.1 Detailed Description . . . 581
10.153.2 Constructor & Destructor Documentation . . . 581
10.153.2.1 ImageFragmentSplitter() . . . 581
10.153.2.2 ∼ImageFragmentSplitter() . . . 581
10.153.3 Member Function Documentation . . . 581
10.153.3.1 GetFragmentSizeMax() . . . 581
10.153.3.2 SetForce() . . . 581
10.153.3.3 SetFragmentSizeMax() . . . 582
10.153.3.4 Split() . . . 582
10.154 gdcm::ImageHelper Class Reference . . . 582
10.154.1 Detailed Description . . . 583
10.154.2 Member Function Documentation . . . 583
10.154.2.1 ComputeMediaStorageFromModality() . . . 584
10.154.2.2 ComputeSpacingFromImagePositionPatient() . . . 584
10.154.2.3 GetDimensionsValue() . . . 584
10.154.2.4 GetDirectionCosinesFromDataSet() . . . 584
10.154.2.5 GetDirectionCosinesValue() . . . 585
10.154.2.6 GetForcePixelSpacing() . . . 585
10.154.2.7 GetForceRescaleInterceptSlope() . . . 585
10.154.2.8 GetLUT() . . . 585

10.154.2.9 GetOriginValue() . . . 585
10.154.2.10 GetPhotometricInterpretationValue() . . . 585
10.154.2.11 GetPixelFormatValue() . . . 586
10.154.2.12 GetPlanarConfigurationValue() . . . 586
10.154.2.13 GetPMSRescaleInterceptSlope() . . . 586
10.154.2.14 GetPointerFromElement() . . . 586
10.154.2.15 GetRealWorldValueMappingContent() . . . 586
10.154.2.16 GetRescaleInterceptSlopeValue() . . . 586
10.154.2.17 GetSpacingTagFromMediaStorage() . . . 587
10.154.2.18 GetSpacingValue() . . . 587
10.154.2.19 GetZSpacingTagFromMediaStorage() . . . 587
10.154.2.20 SetDimensionsValue() . . . 587
10.154.2.21 SetDirectionCosinesValue() . . . 587
10.154.2.22 SetForcePixelSpacing() . . . 588
10.154.2.23 SetForceRescaleInterceptSlope() . . . 588
10.154.2.24 SetOriginValue() . . . 588
10.154.2.25 SetPMSRescaleInterceptSlope() . . . 588
10.154.2.26 SetRescaleInterceptSlopeValue() . . . 588
10.154.2.27 SetSpacingValue() . . . 589
10.155 gdcm::ImageReader Class Reference . . . 589
10.155.1 Detailed Description . . . 591
10.155.2 Constructor & Destructor Documentation . . . 591
10.155.2.1 ImageReader() . . . 591
10.155.2.2 ~ImageReader() . . . 591
10.155.3 Member Function Documentation . . . 591
10.155.3.1 GetImage() [1/2] . . . 591
10.155.3.2 GetImage() [2/2] . . . 592
10.155.3.3 Read() . . . 592
10.155.3.4 ReadACRNEMAImage() . . . 592
10.155.3.5 ReadImage() . . . 593
10.156 gdcm::ImageRegionReader Class Reference . . . 593
10.156.1 Detailed Description . . . 595
10.156.2 Constructor & Destructor Documentation . . . 595
10.156.2.1 ImageRegionReader() . . . 595
10.156.2.2 ~ImageRegionReader() . . . 595
10.156.3 Member Function Documentation . . . 595
10.156.3.1 ComputeBufferLength() . . . 596
10.156.3.2 GetRegion() . . . 596
10.156.3.3 Read() . . . 596

10.156.3.4 ReadInformation() . . . 596
10.156.3.5 ReadIntoBuffer() . . . 597
10.156.3.6 SetRegion() . . . 597
10.157 gdcm::ImageToImageFilter Class Reference . . . 597
10.157.1 Detailed Description . . . 599
10.157.2 Constructor & Destructor Documentation . . . 599
10.157.2.1 ImageToImageFilter() . . . 599
10.157.2.2 ∼ImageToImageFilter() . . . 599
10.157.3 Member Function Documentation . . . 599
10.157.3.1 GetInput() . . . 599
10.157.3.2 GetOutput() . . . 600
10.158 gdcm::ImageWriter Class Reference . . . 600
10.158.1 Detailed Description . . . 602
10.158.2 Constructor & Destructor Documentation . . . 602
10.158.2.1 ImageWriter() . . . 602
10.158.2.2 ∼ImageWriter() . . . 602
10.158.3 Member Function Documentation . . . 602
10.158.3.1 ComputeTargetMediaStorage() . . . 603
10.158.3.2 GetImage() [1/2] . . . 603
10.158.3.3 GetImage() [2/2] . . . 603
10.158.3.4 Write() . . . 603
10.159 gdcm::network::ImplementationClassUIDSub Class Reference . . . 604
10.159.1 Detailed Description . . . 604
10.159.2 Constructor & Destructor Documentation . . . 604
10.159.2.1 ImplementationClassUIDSub() . . . 604
10.159.3 Member Function Documentation . . . 604
10.159.3.1 Print() . . . 604
10.159.3.2 Read() . . . 605
10.159.3.3 Size() . . . 605
10.159.3.4 Write() . . . 605
10.160 gdcm::network::ImplementationUIDSub Class Reference . . . 605
10.160.1 Detailed Description . . . 605
10.160.2 Constructor & Destructor Documentation . . . 605
10.160.2.1 ImplementationUIDSub() . . . 606
10.160.3 Member Function Documentation . . . 606
10.160.3.1 Write() . . . 606
10.161 gdcm::network::ImplementationVersionNameSub Class Reference . . . 606
10.161.1 Detailed Description . . . 606
10.161.2 Constructor & Destructor Documentation . . . 606

10.161.2.1 ImplementationVersionNameSub() . . . 607
10.161.3 Member Function Documentation . . . 607
10.161.3.1 Print() . . . 607
10.161.3.2 Read() . . . 607
10.161.3.3 Size() . . . 607
10.161.3.4 Write() . . . 607
10.162 gdcm::ImplicitDataElement Class Reference . . . 608
10.162.1 Detailed Description . . . 609
10.162.2 Member Function Documentation . . . 609
10.162.2.1 GetLength() . . . 609
10.162.2.2 Read() . . . 609
10.162.2.3 ReadPreValue() . . . 610
10.162.2.4 ReadValue() . . . 610
10.162.2.5 ReadValueWithLength() . . . 610
10.162.2.6 ReadWithLength() . . . 610
10.162.2.7 Write() . . . 610
10.163 gdcm::InitializeEvent Class Reference . . . 611
10.164 gdcm::IOD Class Reference . . . 612
10.164.1 Detailed Description . . . 612
10.164.2 Member Typedef Documentation . . . 612
10.164.2.1 MapIODEntry . . . 613
10.164.2.2 SizeType . . . 613
10.164.3 Constructor & Destructor Documentation . . . 613
10.164.3.1 IOD() . . . 613
10.164.4 Member Function Documentation . . . 613
10.164.4.1 AddIODEntry() . . . 613
10.164.4.2 Clear() . . . 613
10.164.4.3 GetIODEntry() . . . 613
10.164.4.4 GetNumberOfIODs() . . . 614
10.164.4.5 GetTypeFromTag() . . . 614
10.164.5 Friends And Related Function Documentation . . . 614
10.164.5.1 operator<< . . . 614
10.165 gdcm::IODEntry Class Reference . . . 614
10.165.1 Detailed Description . . . 615
10.165.2 Constructor & Destructor Documentation . . . 615
10.165.2.1 IODEntry() . . . 615
10.165.3 Member Function Documentation . . . 616
10.165.3.1 GetIE() . . . 616
10.165.3.2 GetName() . . . 616

10.165.3.3 GetRef() . . . 616
10.165.3.4 GetUsage() . . . 616
10.165.3.5 GetUsageType() . . . 616
10.165.3.6 SetIE() . . . 616
10.165.3.7 SetName() . . . 617
10.165.3.8 SetRef() . . . 617
10.165.3.9 SetUsage() . . . 617
10.165.4 Friends And Related Function Documentation . . . 617
10.165.4.1 operator<< . . . 617
10.166 gdcm::IODs Class Reference . . . 617
10.166.1 Detailed Description . . . 618
10.166.2 Member Typedef Documentation . . . 618
10.166.2.1 IODMapType . . . 618
10.166.2.2 IODMapTypeConstIterator . . . 619
10.166.2.3 IODName . . . 619
10.166.3 Constructor & Destructor Documentation . . . 619
10.166.3.1 IODs() . . . 619
10.166.4 Member Function Documentation . . . 619
10.166.4.1 AddIOD() . . . 619
10.166.4.2 Begin() . . . 619
10.166.4.3 Clear() . . . 620
10.166.4.4 End() . . . 620
10.166.4.5 GetIOD() . . . 620
10.166.5 Friends And Related Function Documentation . . . 620
10.166.5.1 operator<< . . . 620
10.167 gdcm::IPPSorter Class Reference . . . 621
10.167.1 Detailed Description . . . 622
10.167.2 Constructor & Destructor Documentation . . . 622
10.167.2.1 IPPSorter() . . . 622
10.167.3 Member Function Documentation . . . 623
10.167.3.1 GetDirectionCosinesTolerance() . . . 623
10.167.3.2 GetZSpacing() . . . 623
10.167.3.3 GetZSpacingTolerance() . . . 623
10.167.3.4 SetComputeZSpacing() . . . 623
10.167.3.5 SetDirectionCosinesTolerance() . . . 624
10.167.3.6 SetDropDuplicatePositions() . . . 624
10.167.3.7 SetZSpacingTolerance() . . . 624
10.167.3.8 Sort() . . . 624
10.167.4 Member Data Documentation . . . 625

10.167.4.1 ComputeZSpacing . . . 625
10.167.4.2 DirCosTolerance . . . 625
10.167.4.3 DropDuplicatePositions . . . 625
10.167.4.4 ZSpacing . . . 625
10.167.4.5 ZTolerance . . . 625
10.168 gdcm::Item Class Reference . . . 626
10.168.1 Detailed Description . . . 627
10.168.2 Constructor & Destructor Documentation . . . 627
10.168.2.1 Item() [1/2] . . . 628
10.168.2.2 Item() [2/2] . . . 628
10.168.3 Member Function Documentation . . . 628
10.168.3.1 Clear() . . . 628
10.168.3.2 FindDataElement() . . . 628
10.168.3.3 GetDataElement() . . . 628
10.168.3.4 GetLength() . . . 628
10.168.3.5 GetNestedDataSet() [1/2] . . . 629
10.168.3.6 GetNestedDataSet() [2/2] . . . 629
10.168.3.7 InsertDataElement() . . . 629
10.168.3.8 Read() . . . 629
10.168.3.9 SetNestedDataSet() . . . 629
10.168.3.10 Write() . . . 630
10.168.4 Friends And Related Function Documentation . . . 630
10.168.4.1 operator<< . . . 630
10.169 gdcm::IterationEvent Class Reference . . . 630
10.170 gdcm::JPEG12Codec Class Reference . . . 631
10.170.1 Detailed Description . . . 633
10.170.2 Constructor & Destructor Documentation . . . 633
10.170.2.1 JPEG12Codec() . . . 633
10.170.2.2 ~JPEG12Codec() . . . 633
10.170.3 Member Function Documentation . . . 633
10.170.3.1 DecodeByStreams() . . . 633
10.170.3.2 EncodeBuffer() . . . 634
10.170.3.3 GetHeaderInfo() . . . 634
10.170.3.4 InternalCode() . . . 634
10.170.3.5 IsStateSuspension() . . . 634
10.171 gdcm::JPEG16Codec Class Reference . . . 635
10.171.1 Detailed Description . . . 636
10.171.2 Constructor & Destructor Documentation . . . 636
10.171.2.1 JPEG16Codec() . . . 636

10.171.2.2 ~JPEG16Codec() . . . 636
10.171.3 Member Function Documentation . . . 636
10.171.3.1 DecodeByStreams() . . . 637
10.171.3.2 EncodeBuffer() . . . 637
10.171.3.3 GetHeaderInfo() . . . 637
10.171.3.4 InternalCode() . . . 637
10.171.3.5 IsStateSuspension() . . . 637
10.172 gdcm::JPEG2000Codec Class Reference . . . 638
10.172.1 Detailed Description . . . 639
10.172.2 Constructor & Destructor Documentation . . . 640
10.172.2.1 JPEG2000Codec() . . . 640
10.172.2.2 ~JPEG2000Codec() . . . 640
10.172.3 Member Function Documentation . . . 640
10.172.3.1 AppendFrameEncode() . . . 640
10.172.3.2 AppendRowEncode() . . . 640
10.172.3.3 CanCode() . . . 641
10.172.3.4 CanDecode() . . . 641
10.172.3.5 Clone() . . . 641
10.172.3.6 Code() . . . 641
10.172.3.7 Decode() . . . 642
10.172.3.8 DecodeByStreams() . . . 642
10.172.3.9 DecodeExtent() . . . 642
10.172.3.10 GetHeaderInfo() . . . 642
10.172.3.11 GetQuality() . . . 643
10.172.3.12 GetRate() . . . 643
10.172.3.13 IsFrameEncoder() . . . 643
10.172.3.14 IsRowEncoder() . . . 643
10.172.3.15 SetMCT() . . . 643
10.172.3.16 SetNumberOfResolutions() . . . 643
10.172.3.17 SetNumberOfThreadsForDecompression() . . . 643
10.172.3.18 SetQuality() . . . 644
10.172.3.19 SetRate() . . . 644
10.172.3.20 SetReversible() . . . 644
10.172.3.21 SetTileSize() . . . 644
10.172.3.22 StartEncode() . . . 644
10.172.3.23 StopEncode() . . . 645
10.172.4 Friends And Related Function Documentation . . . 645
10.172.4.1 Bitmap . . . 645
10.172.4.2 ImageRegionReader . . . 645

10.173 gdcm::JPEG8Codec Class Reference . . . 645
10.173.1 Detailed Description . . . 647
10.173.2 Constructor & Destructor Documentation . . . 647
10.173.2.1 JPEG8Codec() . . . 647
10.173.2.2 ∼JPEG8Codec() . . . 647
10.173.3 Member Function Documentation . . . 647
10.173.3.1 DecodeByStreams() . . . 647
10.173.3.2 EncodeBuffer() . . . 648
10.173.3.3 GetHeaderInfo() . . . 648
10.173.3.4 InternalCode() . . . 648
10.173.3.5 IsStateSuspension() . . . 648
10.174 gdcm::JPEGCodec Class Reference . . . 649
10.174.1 Detailed Description . . . 651
10.174.2 Constructor & Destructor Documentation . . . 651
10.174.2.1 JPEGCodec() . . . 651
10.174.2.2 ∼JPEGCodec() . . . 651
10.174.3 Member Function Documentation . . . 651
10.174.3.1 AppendFrameEncode() . . . 652
10.174.3.2 AppendRowEncode() . . . 652
10.174.3.3 CanCode() . . . 652
10.174.3.4 CanDecode() . . . 652
10.174.3.5 Clone() . . . 653
10.174.3.6 Code() . . . 653
10.174.3.7 ComputeOffsetTable() . . . 653
10.174.3.8 Decode() . . . 653
10.174.3.9 DecodeByStreams() . . . 653
10.174.3.10 DecodeExtent() . . . 654
10.174.3.11 EncodeBuffer() . . . 654
10.174.3.12 GetHeaderInfo() . . . 654
10.174.3.13 GetLossless() . . . 654
10.174.3.14 GetQuality() . . . 655
10.174.3.15 IsFrameEncoder() . . . 655
10.174.3.16 IsRowEncoder() . . . 655
10.174.3.17 IsStateSuspension() . . . 655
10.174.3.18 IsValid() . . . 655
10.174.3.19 SetBitSample() . . . 655
10.174.3.20 SetLossless() . . . 656
10.174.3.21 SetPixelFormat() . . . 656
10.174.3.22 SetQuality() . . . 656

10.174.3.23 StartEncode() . . . 656
10.174.3.24 StopEncode() . . . 657
10.174.4 Friends And Related Function Documentation . . . 657
10.174.4.1 ImageRegionReader . . . 657
10.174.5 Member Data Documentation . . . 657
10.174.5.1 BitSample . . . 657
10.174.5.2 Quality . . . 657
10.175 gdcm::JPEGLSCodec Class Reference . . . 658
10.175.1 Detailed Description . . . 659
10.175.2 Constructor & Destructor Documentation . . . 660
10.175.2.1 JPEGLSCodec() . . . 660
10.175.2.2 ∼JPEGLSCodec() . . . 660
10.175.3 Member Function Documentation . . . 660
10.175.3.1 AppendFrameEncode() . . . 660
10.175.3.2 AppendRowEncode() . . . 660
10.175.3.3 CanCode() . . . 661
10.175.3.4 CanDecode() . . . 661
10.175.3.5 Clone() . . . 661
10.175.3.6 Code() . . . 661
10.175.3.7 Decode() [1/2] . . . 662
10.175.3.8 Decode() [2/2] . . . 662
10.175.3.9 DecodeExtent() . . . 662
10.175.3.10 GetBufferLength() . . . 662
10.175.3.11 GetHeaderInfo() . . . 663
10.175.3.12 GetLossless() . . . 663
10.175.3.13 IsFrameEncoder() . . . 663
10.175.3.14 IsRowEncoder() . . . 663
10.175.3.15 SetBufferLength() . . . 663
10.175.3.16 SetLossless() . . . 663
10.175.3.17 SetLossyError() . . . 664
10.175.3.18 StartEncode() . . . 664
10.175.3.19 StopEncode() . . . 664
10.175.4 Friends And Related Function Documentation . . . 664
10.175.4.1 ImageRegionReader . . . 664
10.176 gdcm::JSON Class Reference . . . 664
10.176.1 Detailed Description . . . 665
10.176.2 Constructor & Destructor Documentation . . . 665
10.176.2.1 JSON() . . . 665
10.176.2.2 ∼JSON() . . . 665

10.176.3 Member Function Documentation . . . 665
10.176.3.1 Code() . . . 665
10.176.3.2 Decode() . . . 666
10.176.3.3 GetPrettyPrint() . . . 666
10.176.3.4 PrettyPrintOff() . . . 666
10.176.3.5 PrettyPrintOn() . . . 666
10.176.3.6 SetPrettyPrint() . . . 666
10.177 gdcm::KAKADUCodec Class Reference . . . 667
10.177.1 Detailed Description . . . 668
10.177.2 Constructor & Destructor Documentation . . . 668
10.177.2.1 KAKADUCodec() . . . 668
10.177.2.2 ~KAKADUCodec() . . . 668
10.177.3 Member Function Documentation . . . 668
10.177.3.1 CanCode() . . . 668
10.177.3.2 CanDecode() . . . 669
10.177.3.3 Clone() . . . 669
10.177.3.4 Code() . . . 669
10.177.3.5 Decode() . . . 669
10.178 gdcm::LO Class Reference . . . 670
10.178.1 Detailed Description . . . 671
10.178.2 Member Typedef Documentation . . . 671
10.178.2.1 const_iterator . . . 671
10.178.2.2 const_reference . . . 671
10.178.2.3 const_reverse_iterator . . . 671
10.178.2.4 difference_type . . . 671
10.178.2.5 iterator . . . 672
10.178.2.6 pointer . . . 672
10.178.2.7 reference . . . 672
10.178.2.8 reverse_iterator . . . 672
10.178.2.9 size_type . . . 672
10.178.2.10 Superclass . . . 672
10.178.2.11 value_type . . . 672
10.178.3 Constructor & Destructor Documentation . . . 673
10.178.3.1 LO() [1/4] . . . 673
10.178.3.2 LO() [2/4] . . . 673
10.178.3.3 LO() [3/4] . . . 673
10.178.3.4 LO() [4/4] . . . 673
10.178.4 Member Function Documentation . . . 673
10.178.4.1 IsValid() . . . 673

10.179 gdcm::LookupTable Class Reference . . . 674
10.179.1 Detailed Description . . . 676
10.179.2 Member Enumeration Documentation . . . 676
10.179.2.1 LookupTableType . . . 676
10.179.3 Constructor & Destructor Documentation . . . 676
10.179.3.1 LookupTable() [1/2] . . . 676
10.179.3.2 ~LookupTable() . . . 676
10.179.3.3 LookupTable() [2/2] . . . 677
10.179.4 Member Function Documentation . . . 677
10.179.4.1 Allocate() . . . 677
10.179.4.2 Clear() . . . 677
10.179.4.3 Decode() [1/2] . . . 677
10.179.4.4 Decode() [2/2] . . . 677
10.179.4.5 Decode8() . . . 678
10.179.4.6 GetBitSample() . . . 678
10.179.4.7 GetBufferAsRGBA() . . . 678
10.179.4.8 GetLUT() . . . 678
10.179.4.9 GetLUTDescriptor() . . . 678
10.179.4.10 GetLUTLength() . . . 679
10.179.4.11 GetPointer() . . . 679
10.179.4.12 InitializeBlueLUT() . . . 679
10.179.4.13 Initialized() . . . 679
10.179.4.14 InitializeGreenLUT() . . . 679
10.179.4.15 InitializeLUT() . . . 680
10.179.4.16 InitializeRedLUT() . . . 680
10.179.4.17 IsRGB8() . . . 680
10.179.4.18 Print() . . . 680
10.179.4.19 SetBlueLUT() . . . 681
10.179.4.20 SetGreenLUT() . . . 681
10.179.4.21 SetLUT() . . . 681
10.179.4.22 SetRedLUT() . . . 681
10.179.4.23 WriteBufferAsRGBA() . . . 681
10.179.5 Member Data Documentation . . . 681
10.179.5.1 BitSample . . . 682
10.179.5.2 IncompleteLUT . . . 682
10.179.5.3 Internal . . . 682
10.180 gdcm::Scanner2::ltstr Struct Reference . . . 682
10.180.1 Member Function Documentation . . . 682
10.180.1.1 operator()() . . . 682

10.181 gdcm::Scanner::ltstr Struct Reference . . . 683
10.181.1 Member Function Documentation . . . 683
10.181.1.1 operator()() . . . 683
10.182 gdcm::StrictScanner2::ltstr Struct Reference . . . 683
10.182.1 Member Function Documentation . . . 683
10.182.1.1 operator()() . . . 683
10.183 gdcm::StrictScanner::ltstr Struct Reference . . . 684
10.183.1 Member Function Documentation . . . 684
10.183.1.1 operator()() . . . 684
10.184 gdcm::Macro Class Reference . . . 684
10.184.1 Detailed Description . . . 685
10.184.2 Member Typedef Documentation . . . 685
10.184.2.1 ArrayIncludeMacrosType . . . 685
10.184.2.2 MapModuleEntry . . . 685
10.184.3 Constructor & Destructor Documentation . . . 685
10.184.3.1 Macro() . . . 685
10.184.4 Member Function Documentation . . . 685
10.184.4.1 AddMacroEntry() . . . 686
10.184.4.2 Clear() . . . 686
10.184.4.3 FindMacroEntry() . . . 686
10.184.4.4 GetMacroEntry() . . . 686
10.184.4.5 GetName() . . . 686
10.184.4.6 SetName() . . . 686
10.184.4.7 Verify() . . . 687
10.184.5 Friends And Related Function Documentation . . . 687
10.184.5.1 operator<< . . . 687
10.185 gdcm::Macros Class Reference . . . 687
10.185.1 Detailed Description . . . 688
10.185.2 Member Typedef Documentation . . . 688
10.185.2.1 ModuleMapType . . . 688
10.185.3 Constructor & Destructor Documentation . . . 688
10.185.3.1 Macros() . . . 688
10.185.4 Member Function Documentation . . . 688
10.185.4.1 AddMacro() . . . 688
10.185.4.2 Clear() . . . 689
10.185.4.3 GetMacro() . . . 689
10.185.4.4 IsEmpty() . . . 689
10.185.5 Friends And Related Function Documentation . . . 689
10.185.5.1 operator<< . . . 689

10.186 gdcm::network::MaximumLengthSub Class Reference . . . 689
10.186.1 Detailed Description . . . 690
10.186.2 Constructor & Destructor Documentation . . . 690
10.186.2.1 MaximumLengthSub() . . . 690
10.186.3 Member Function Documentation . . . 690
10.186.3.1 GetMaximumLength() . . . 690
10.186.3.2 Print() . . . 690
10.186.3.3 Read() . . . 691
10.186.3.4 SetMaximumLength() . . . 691
10.186.3.5 Size() . . . 691
10.186.3.6 Write() . . . 691
10.187 gdcm::MD5 Class Reference . . . 691
10.187.1 Detailed Description . . . 692
10.187.2 Member Function Documentation . . . 692
10.187.2.1 Compute() . . . 692
10.187.2.2 ComputeFile() . . . 692
10.188 gdcm::MediaStorage Class Reference . . . 692
10.188.1 Detailed Description . . . 696
10.188.2 Member Enumeration Documentation . . . 696
10.188.2.1 MSType . . . 696
10.188.2.2 ObjectType . . . 699
10.188.3 Constructor & Destructor Documentation . . . 699
10.188.3.1 MediaStorage() . . . 699
10.188.4 Member Function Documentation . . . 699
10.188.4.1 GetModality() . . . 699
10.188.4.2 GetModalityDimension() . . . 700
10.188.4.3 GetMSString() . . . 700
10.188.4.4 GetMSType() . . . 700
10.188.4.5 GetNumberOfModality() . . . 700
10.188.4.6 GetNumberOfMSString() . . . 700
10.188.4.7 GetNumberOfMSType() . . . 701
10.188.4.8 GetString() . . . 701
10.188.4.9 GuessFromModality() . . . 701
10.188.4.10 IsImage() . . . 701
10.188.4.11 IsUndefined() . . . 702
10.188.4.12 operator MSType() . . . 702
10.188.4.13 SetFromDataSet() . . . 702
10.188.4.14 SetFromFile() . . . 702
10.188.4.15 SetFromHeader() . . . 702

10.188.4.16 SetFromModality() . . . 703
10.188.4.17 SetFromSourceImageSequence() . . . 703
10.188.5 Friends And Related Function Documentation . . . 703
10.188.5.1 operator<< . . . 703
10.189 gdcm::MemberCommand< T > Class Template Reference . . . 703
10.189.1 Detailed Description . . . 705
10.189.2 Member Typedef Documentation . . . 705
10.189.2.1 Self . . . 706
10.189.2.2 TConstMemberFunctionPointer . . . 706
10.189.2.3 TMemberFunctionPointer . . . 706
10.189.3 Constructor & Destructor Documentation . . . 706
10.189.3.1 MemberCommand() [1/2] . . . 706
10.189.3.2 MemberCommand() [2/2] . . . 706
10.189.3.3 ~MemberCommand() . . . 707
10.189.4 Member Function Documentation . . . 707
10.189.4.1 Execute() [1/2] . . . 707
10.189.4.2 Execute() [2/2] . . . 707
10.189.4.3 New() . . . 707
10.189.4.4 operator=() . . . 708
10.189.4.5 SetCallbackFunction() [1/2] . . . 708
10.189.4.6 SetCallbackFunction() [2/2] . . . 708
10.189.5 Member Data Documentation . . . 708
10.189.5.1 m_ConstMemberFunction . . . 708
10.189.5.2 m_MemberFunction . . . 709
10.189.5.3 m_This . . . 709
10.190 gdcm::MeshPrimitive Class Reference . . . 709
10.190.1 Detailed Description . . . 711
10.190.2 Member Typedef Documentation . . . 711
10.190.2.1 PrimitivesData . . . 711
10.190.3 Member Enumeration Documentation . . . 711
10.190.3.1 MPType . . . 711
10.190.4 Constructor & Destructor Documentation . . . 712
10.190.4.1 MeshPrimitive() . . . 712
10.190.4.2 ~MeshPrimitive() . . . 712
10.190.5 Member Function Documentation . . . 712
10.190.5.1 AddPrimitiveData() . . . 712
10.190.5.2 GetMPType() . . . 712
10.190.5.3 GetMPTypeString() . . . 713
10.190.5.4 GetNumberOfPrimitivesData() . . . 713

10.190.5.5 GetPrimitiveData() [1/4] . . . 713
10.190.5.6 GetPrimitiveData() [2/4] . . . 713
10.190.5.7 GetPrimitiveData() [3/4] . . . 713
10.190.5.8 GetPrimitiveData() [4/4] . . . 713
10.190.5.9 GetPrimitivesData() [1/2] . . . 713
10.190.5.10 GetPrimitivesData() [2/2] . . . 714
10.190.5.11 GetPrimitiveType() . . . 714
10.190.5.12 SetPrimitiveData() [1/2] . . . 714
10.190.5.13 SetPrimitiveData() [2/2] . . . 714
10.190.5.14 SetPrimitivesData() . . . 714
10.190.5.15 SetPrimitiveType() . . . 714
10.190.6 Member Data Documentation . . . 714
10.190.6.1 PrimitiveData . . . 715
10.190.6.2 PrimitiveType . . . 715
10.191 gdcm::ModalityPerformedProcedureStepCreateQuery Class Reference . . . 715
10.191.1 Detailed Description . . . 716
10.191.2 Constructor & Destructor Documentation . . . 717
10.191.2.1 ModalityPerformedProcedureStepCreateQuery() . . . 717
10.191.3 Member Function Documentation . . . 717
10.191.3.1 GetAbstractSyntaxUID() . . . 717
10.191.3.2 GetRequiredDataSet() . . . 717
10.191.3.3 ValidateQuery() . . . 717
10.191.4 Friends And Related Function Documentation . . . 717
10.191.4.1 QueryFactory . . . 718
10.192 gdcm::ModalityPerformedProcedureStepSetQuery Class Reference . . . 718
10.192.1 Detailed Description . . . 719
10.192.2 Constructor & Destructor Documentation . . . 720
10.192.2.1 ModalityPerformedProcedureStepSetQuery() . . . 720
10.192.3 Member Function Documentation . . . 720
10.192.3.1 GetAbstractSyntaxUID() . . . 720
10.192.3.2 GetRequiredDataSet() . . . 720
10.192.3.3 ValidateQuery() . . . 720
10.192.4 Friends And Related Function Documentation . . . 720
10.192.4.1 QueryFactory . . . 721
10.193 gdcm::ModifiedEvent Class Reference . . . 721
10.194 gdcm::Module Class Reference . . . 722
10.194.1 Detailed Description . . . 722
10.194.2 Member Typedef Documentation . . . 723
10.194.2.1 ArrayIncludeMacrosType . . . 723

10.194.2.2 MapModuleEntry . . . 723
10.194.3 Constructor & Destructor Documentation . . . 723
10.194.3.1 Module() . . . 723
10.194.4 Member Function Documentation . . . 723
10.194.4.1 AddMacro() . . . 723
10.194.4.2 AddModuleEntry() . . . 723
10.194.4.3 Clear() . . . 724
10.194.4.4 FindModuleEntryInMacros() . . . 724
10.194.4.5 GetModuleEntryInMacros() . . . 724
10.194.4.6 GetName() . . . 724
10.194.4.7 SetName() . . . 724
10.194.4.8 Verify() . . . 725
10.194.5 Friends And Related Function Documentation . . . 725
10.194.5.1 operator<< . . . 725
10.195 gdcm::ModuleEntry Class Reference . . . 725
10.195.1 Detailed Description . . . 727
10.195.2 Member Typedef Documentation . . . 727
10.195.2.1 Description . . . 727
10.195.3 Constructor & Destructor Documentation . . . 727
10.195.3.1 ModuleEntry() . . . 727
10.195.3.2 ~ModuleEntry() . . . 727
10.195.4 Member Function Documentation . . . 728
10.195.4.1 GetDescription() . . . 728
10.195.4.2 GetName() . . . 728
10.195.4.3 GetType() . . . 728
10.195.4.4 SetDescription() . . . 728
10.195.4.5 SetName() . . . 728
10.195.4.6 SetType() . . . 728
10.195.5 Friends And Related Function Documentation . . . 729
10.195.5.1 operator<< . . . 729
10.195.6 Member Data Documentation . . . 729
10.195.6.1 DataElementType . . . 729
10.195.6.2 DescriptionField . . . 729
10.195.6.3 Name . . . 729
10.196 gdcm::Modules Class Reference . . . 729
10.196.1 Detailed Description . . . 730
10.196.2 Member Typedef Documentation . . . 730
10.196.2.1 ModuleMapType . . . 730
10.196.3 Constructor & Destructor Documentation . . . 730

10.196.3.1 Modules() . . . 731
10.196.4 Member Function Documentation . . . 731
10.196.4.1 AddModule() . . . 731
10.196.4.2 Clear() . . . 731
10.196.4.3 GetModule() . . . 731
10.196.4.4 IsEmpty() . . . 731
10.196.5 Friends And Related Function Documentation . . . 731
10.196.5.1 operator<< . . . 732
10.197 gdcm::MovePatientRootQuery Class Reference . . . 732
10.197.1 Detailed Description . . . 733
10.197.2 Constructor & Destructor Documentation . . . 733
10.197.2.1 MovePatientRootQuery() . . . 733
10.197.3 Member Function Documentation . . . 734
10.197.3.1 GetAbstractSyntaxUID() . . . 734
10.197.3.2 GetTagListByLevel() . . . 734
10.197.3.3 InitializeDataSet() . . . 734
10.197.3.4 ValidateQuery() . . . 734
10.197.4 Friends And Related Function Documentation . . . 735
10.197.4.1 QueryFactory . . . 735
10.198 gdcm::MoveStudyRootQuery Class Reference . . . 735
10.198.1 Detailed Description . . . 736
10.198.2 Constructor & Destructor Documentation . . . 736
10.198.2.1 MoveStudyRootQuery() . . . 736
10.198.3 Member Function Documentation . . . 737
10.198.3.1 GetAbstractSyntaxUID() . . . 737
10.198.3.2 GetTagListByLevel() . . . 737
10.198.3.3 InitializeDataSet() . . . 737
10.198.3.4 ValidateQuery() . . . 737
10.198.4 Friends And Related Function Documentation . . . 738
10.198.4.1 QueryFactory . . . 738
10.199 gdcm::MrProtocol Class Reference . . . 738
10.199.1 Detailed Description . . . 739
10.199.2 Constructor & Destructor Documentation . . . 739
10.199.2.1 MrProtocol() . . . 739
10.199.2.2 ~MrProtocol() . . . 739
10.199.3 Member Function Documentation . . . 739
10.199.3.1 FindMrProtocolByName() . . . 739
10.199.3.2 GetMrProtocolByName() . . . 739
10.199.3.3 GetSliceArray() . . . 740

10.199.3.4 GetVersion() . . . 740
10.199.3.5 Load() . . . 740
10.199.3.6 Print() . . . 740
10.199.4 Friends And Related Function Documentation . . . 740
10.199.4.1 operator<< . . . 740
10.200 gdcm::network::NActionRQ Class Reference . . . 741
10.200.1 Detailed Description . . . 741
10.200.2 Member Function Documentation . . . 742
10.200.2.1 ConstructPDV() . . . 742
10.201 gdcm::network::NActionRSP Class Reference . . . 742
10.201.1 Detailed Description . . . 743
10.201.2 Member Function Documentation . . . 743
10.201.2.1 ConstructPDVByDataSet() . . . 743
10.202 gdcm::network::NCreateRQ Class Reference . . . 744
10.202.1 Detailed Description . . . 744
10.202.2 Member Function Documentation . . . 745
10.202.2.1 ConstructPDV() . . . 745
10.203 gdcm::network::NCreateRSP Class Reference . . . 745
10.203.1 Detailed Description . . . 746
10.203.2 Member Function Documentation . . . 746
10.203.2.1 ConstructPDVByDataSet() . . . 746
10.204 gdcm::network::NDeleteRQ Class Reference . . . 747
10.204.1 Detailed Description . . . 747
10.204.2 Member Function Documentation . . . 748
10.204.2.1 ConstructPDV() . . . 748
10.205 gdcm::network::NDeleteRSP Class Reference . . . 748
10.205.1 Detailed Description . . . 749
10.205.2 Member Function Documentation . . . 749
10.205.2.1 ConstructPDVByDataSet() . . . 749
10.206 gdcm::NestedModuleEntries Class Reference . . . 750
10.206.1 Detailed Description . . . 751
10.206.2 Member Typedef Documentation . . . 751
10.206.2.1 SizeType . . . 751
10.206.3 Constructor & Destructor Documentation . . . 751
10.206.3.1 NestedModuleEntries() . . . 752
10.206.4 Member Function Documentation . . . 752
10.206.4.1 AddModuleEntry() . . . 752
10.206.4.2 GetModuleEntry() [1/2] . . . 752
10.206.4.3 GetModuleEntry() [2/2] . . . 752

10.206.4.4 GetNumberOfModuleEntries() . . . 752
10.206.5 Friends And Related Function Documentation . . . 752
10.206.5.1 operator<< . . . 753
10.207 gdcm::network::NEventReportRQ Class Reference . . . 753
10.207.1 Detailed Description . . . 754
10.207.2 Member Function Documentation . . . 754
10.207.2.1 ConstructPDV() . . . 754
10.208 gdcm::network::NEventReportRSP Class Reference . . . 754
10.208.1 Detailed Description . . . 755
10.208.2 Member Function Documentation . . . 755
10.208.2.1 ConstructPDVByDataSet() . . . 755
10.209 gdcm::network::NGetRQ Class Reference . . . 756
10.209.1 Detailed Description . . . 756
10.209.2 Member Function Documentation . . . 757
10.209.2.1 ConstructPDV() . . . 757
10.210 gdcm::network::NGetRSP Class Reference . . . 757
10.210.1 Detailed Description . . . 758
10.210.2 Member Function Documentation . . . 758
10.210.2.1 ConstructPDVByDataSet() . . . 758
10.211 gdcm::NoEvent Class Reference . . . 759
10.211.1 Detailed Description . . . 759
10.212 gdcm::network::NormalizedMessageFactory Class Reference . . . 760
10.212.1 Member Function Documentation . . . 760
10.212.1.1 ConstructNAction() . . . 760
10.212.1.2 ConstructNCreate() . . . 760
10.212.1.3 ConstructNDelete() . . . 761
10.212.1.4 ConstructNEventReport() . . . 761
10.212.1.5 ConstructNGet() . . . 761
10.212.1.6 ConstructNSet() . . . 761
10.213 gdcm::NormalizedNetworkFunctions Class Reference . . . 761
10.213.1 Detailed Description . . . 762
10.213.2 Member Function Documentation . . . 762
10.213.2.1 ConstructQuery() . . . 762
10.213.2.2 NAction() . . . 763
10.213.2.3 NCreate() . . . 763
10.213.2.4 NDelete() . . . 763
10.213.2.5 NEventReport() . . . 763
10.213.2.6 NGet() . . . 764
10.213.2.7 NSet() . . . 764

10.214 gdcm::network::NSetRQ Class Reference . . . 764
10.214.1 Detailed Description . . . 765
10.214.2 Member Function Documentation . . . 765
10.214.2.1 ConstructPDV() . . . 765
10.215 gdcm::network::NSetRSP Class Reference . . . 766
10.215.1 Detailed Description . . . 766
10.215.2 Member Function Documentation . . . 767
10.215.2.1 ConstructPDVByDataSet() . . . 767
10.216 gdcm::Object Class Reference . . . 767
10.216.1 Detailed Description . . . 769
10.216.2 Constructor & Destructor Documentation . . . 769
10.216.2.1 Object() [1/2] . . . 769
10.216.2.2 ∼Object() . . . 769
10.216.2.3 Object() [2/2] . . . 769
10.216.3 Member Function Documentation . . . 770
10.216.3.1 operator=() . . . 770
10.216.3.2 Print() . . . 770
10.216.3.3 Register() . . . 770
10.216.3.4 UnRegister() . . . 770
10.216.4 Friends And Related Function Documentation . . . 770
10.216.4.1 operator<< . . . 771
10.216.4.2 SmartPointer . . . 771
10.217 gdcm::OpenSSLCryptoFactory Class Reference . . . 771
10.217.1 Constructor & Destructor Documentation . . . 772
10.217.1.1 OpenSSLCryptoFactory() . . . 772
10.217.2 Member Function Documentation . . . 772
10.217.2.1 CreateCMSProvider() . . . 772
10.217.2.2 InitOpenSSL() . . . 772
10.218 gdcm::OpenSSLCryptographicMessageSyntax Class Reference . . . 773
10.218.1 Constructor & Destructor Documentation . . . 774
10.218.1.1 OpenSSLCryptographicMessageSyntax() . . . 774
10.218.1.2 ∼OpenSSLCryptographicMessageSyntax() . . . 774
10.218.2 Member Function Documentation . . . 774
10.218.2.1 Decrypt() . . . 774
10.218.2.2 Encrypt() . . . 775
10.218.2.3 GetCipherType() . . . 775
10.218.2.4 ParseCertificateFile() . . . 775
10.218.2.5 ParseKeyFile() . . . 775
10.218.2.6 SetCipherType() . . . 775

10.218.2.7 SetPassword() . . . 776
10.219 gdcm::OpenSSLP7CryptoFactory Class Reference . . . 776
10.219.1 Constructor & Destructor Documentation . . . 777
10.219.1.1 OpenSSLP7CryptoFactory() . . . 777
10.219.2 Member Function Documentation . . . 777
10.219.2.1 CreateCMSProvider() . . . 777
10.220 gdcm::OpenSSLP7CryptographicMessageSyntax Class Reference . . . 778
10.220.1 Detailed Description . . . 779
10.220.2 Constructor & Destructor Documentation . . . 779
10.220.2.1 OpenSSLP7CryptographicMessageSyntax() . . . 779
10.220.2.2 ∼OpenSSLP7CryptographicMessageSyntax() . . . 779
10.220.3 Member Function Documentation . . . 779
10.220.3.1 Decrypt() . . . 779
10.220.3.2 Encrypt() . . . 780
10.220.3.3 GetCipherType() . . . 780
10.220.3.4 ParseCertificateFile() . . . 780
10.220.3.5 ParseKeyFile() . . . 780
10.220.3.6 SetCipherType() . . . 780
10.220.3.7 SetPassword() . . . 781
10.221 gdcm::Orientation Class Reference . . . 781
10.221.1 Detailed Description . . . 782
10.221.2 Member Enumeration Documentation . . . 782
10.221.2.1 OrientationType . . . 782
10.221.3 Constructor & Destructor Documentation . . . 782
10.221.3.1 Orientation() . . . 782
10.221.3.2 ∼Orientation() . . . 782
10.221.4 Member Function Documentation . . . 783
10.221.4.1 GetLabel() . . . 783
10.221.4.2 GetMajorAxisFromPatientRelativeDirectionCosine() . . . 783
10.221.4.3 GetObliquityThresholdCosineValue() . . . 783
10.221.4.4 GetType() . . . 783
10.221.4.5 Print() . . . 784
10.221.4.6 SetObliquityThresholdCosineValue() . . . 784
10.221.5 Friends And Related Function Documentation . . . 784
10.221.5.1 operator<< . . . 784
10.222 gdcm::Overlay Class Reference . . . 784
10.222.1 Detailed Description . . . 787
10.222.2 Member Enumeration Documentation . . . 787
10.222.2.1 OverlayType . . . 787

10.222.3 Constructor & Destructor Documentation . . . 788
10.222.3.1 Overlay() [1/2] . . . 788
10.222.3.2 ~Overlay() . . . 788
10.222.3.3 Overlay() [2/2] . . . 788
10.222.4 Member Function Documentation . . . 788
10.222.4.1 Decompress() . . . 788
10.222.4.2 GetBitPosition() . . . 788
10.222.4.3 GetBitsAllocated() . . . 789
10.222.4.4 GetColumns() . . . 789
10.222.4.5 GetDescription() . . . 789
10.222.4.6 GetGroup() . . . 789
10.222.4.7 GetOrigin() . . . 789
10.222.4.8 GetOverlayData() . . . 789
10.222.4.9 GetOverlayTypeAsString() . . . 790
10.222.4.10 GetOverlayTypeFromString() . . . 790
10.222.4.11 GetRows() . . . 790
10.222.4.12 GetType() . . . 790
10.222.4.13 GetTypeAsEnum() . . . 790
10.222.4.14 GetUnpackBuffer() . . . 790
10.222.4.15 GetUnpackBufferLength() . . . 791
10.222.4.16 GrabOverlayFromPixelData() . . . 791
10.222.4.17 IsEmpty() . . . 791
10.222.4.18 IsInPixelData() [1/2] . . . 791
10.222.4.19 IsInPixelData() [2/2] . . . 791
10.222.4.20 IsZero() . . . 791
10.222.4.21 operator=() . . . 792
10.222.4.22 Print() . . . 792
10.222.4.23 SetBitPosition() . . . 792
10.222.4.24 SetBitsAllocated() . . . 792
10.222.4.25 SetColumns() . . . 792
10.222.4.26 SetDescription() . . . 793
10.222.4.27 SetFrameOrigin() . . . 793
10.222.4.28 SetGroup() . . . 793
10.222.4.29 SetNumberOfFrames() . . . 793
10.222.4.30 SetOrigin() . . . 793
10.222.4.31 SetOverlay() . . . 794
10.222.4.32 SetRows() . . . 794
10.222.4.33 SetType() . . . 794
10.222.4.34 Update() . . . 794

10.223 gdcm::ParseException Class Reference . . . 795
10.223.1 Detailed Description . . . 796
10.223.2 Constructor & Destructor Documentation . . . 796
10.223.2.1 ParseException() [1/2] . . . 796
10.223.2.2 ~ParseException() . . . 796
10.223.2.3 ParseException() [2/2] . . . 796
10.223.3 Member Function Documentation . . . 796
10.223.3.1 GetLastElement() . . . 796
10.223.3.2 operator=() . . . 797
10.223.3.3 SetLastElement() . . . 797
10.224 gdcm::Parser Class Reference . . . 797
10.224.1 Detailed Description . . . 798
10.224.2 Member Typedef Documentation . . . 798
10.224.2.1 EndElementHandler . . . 798
10.224.2.2 StartElementHandler . . . 798
10.224.3 Member Enumeration Documentation . . . 799
10.224.3.1 ErrorType . . . 799
10.224.4 Constructor & Destructor Documentation . . . 800
10.224.4.1 Parser() . . . 800
10.224.4.2 ~Parser() . . . 800
10.224.5 Member Function Documentation . . . 800
10.224.5.1 GetBuffer() . . . 800
10.224.5.2 GetCurrentByteIndex() . . . 800
10.224.5.3 GetErrorCode() . . . 801
10.224.5.4 GetErrorString() . . . 801
10.224.5.5 GetUserData() . . . 801
10.224.5.6 Parse() . . . 801
10.224.5.7 ParseBuffer() . . . 801
10.224.5.8 Process() . . . 801
10.224.5.9 SetElementHandler() . . . 802
10.224.5.10 SetUserData() . . . 802
10.225 gdcm::Patient Class Reference . . . 802
10.225.1 Detailed Description . . . 802
10.225.2 Constructor & Destructor Documentation . . . 802
10.225.2.1 Patient() . . . 802
10.226 gdcm::network::PDataTFPDU Class Reference . . . 803
10.226.1 Detailed Description . . . 804
10.226.2 Member Typedef Documentation . . . 804
10.226.2.1 SizeType . . . 804

10.226.3 Constructor & Destructor Documentation . . . 804
10.226.3.1 PDataTFPDU() . . . 804
10.226.4 Member Function Documentation . . . 804
10.226.4.1 AddPresentationDataValue() . . . 804
10.226.4.2 GetNumberOfPresentationDataValues() . . . 805
10.226.4.3 GetPresentationDataValue() . . . 805
10.226.4.4 IsLastFragment() . . . 805
10.226.4.5 Print() . . . 805
10.226.4.6 Read() . . . 805
10.226.4.7 ReadInto() . . . 805
10.226.4.8 Size() . . . 806
10.226.4.9 Write() . . . 806
10.227 gdcm::PDBElement Class Reference . . . 806
10.227.1 Detailed Description . . . 807
10.227.2 Constructor & Destructor Documentation . . . 807
10.227.2.1 PDBElement() . . . 807
10.227.3 Member Function Documentation . . . 807
10.227.3.1 GetName() . . . 808
10.227.3.2 GetValue() . . . 808
10.227.3.3 operator==() . . . 808
10.227.3.4 SetName() . . . 808
10.227.3.5 SetValue() . . . 808
10.227.4 Friends And Related Function Documentation . . . 808
10.227.4.1 operator<< . . . 809
10.227.5 Member Data Documentation . . . 809
10.227.5.1 NameField . . . 809
10.227.5.2 ValueField . . . 809
10.228 gdcm::PDBHeader Class Reference . . . 809
10.228.1 Detailed Description . . . 810
10.228.2 Constructor & Destructor Documentation . . . 810
10.228.2.1 PDBHeader() . . . 810
10.228.2.2 ~PDBHeader() . . . 811
10.228.3 Member Function Documentation . . . 811
10.228.3.1 FindPDBElementByName() . . . 811
10.228.3.2 GetPDBEEnd() . . . 811
10.228.3.3 GetPDBElementByName() . . . 811
10.228.3.4 GetPDBInfoTag() . . . 811
10.228.3.5 LoadFromDataElement() . . . 812
10.228.3.6 Print() . . . 812

10.228.4 Friends And Related Function Documentation . . . 812
10.228.4.1 operator<< . . . 812
10.229 gdcm::PDFCodec Class Reference . . . 812
10.229.1 Detailed Description . . . 814
10.229.2 Constructor & Destructor Documentation . . . 814
10.229.2.1 PDFCodec() . . . 814
10.229.2.2 ~PDFCodec() . . . 814
10.229.3 Member Function Documentation . . . 814
10.229.3.1 CanCode() . . . 814
10.229.3.2 CanDecode() . . . 814
10.229.3.3 Decode() . . . 815
10.230 gdcm::network::PDUFactory Class Reference . . . 815
10.230.1 Detailed Description . . . 816
10.230.2 Member Function Documentation . . . 816
10.230.2.1 ConstructAbortPDU() . . . 816
10.230.2.2 ConstructPDU() . . . 816
10.230.2.3 ConstructReleasePDU() . . . 816
10.230.2.4 CreateCEchoPDU() . . . 816
10.230.2.5 CreateCFindPDU() . . . 816
10.230.2.6 CreateCMovePDU() . . . 817
10.230.2.7 CreateCStoreRQPDU() . . . 817
10.230.2.8 CreateCStoreRSPPDU() . . . 817
10.230.2.9 CreateNActionPDU() . . . 817
10.230.2.10 CreateNCreatePDU() . . . 817
10.230.2.11 CreateNDeletePDU() . . . 817
10.230.2.12 CreateNEventReportPDU() . . . 818
10.230.2.13 CreateNGetPDU() . . . 818
10.230.2.14 CreateNSetPDU() . . . 818
10.230.2.15 DetermineEventByPDU() . . . 818
10.230.2.16 GetPDVs() . . . 818
10.231 gdcm::PersonName Class Reference . . . 818
10.231.1 Detailed Description . . . 819
10.231.2 Member Function Documentation . . . 819
10.231.2.1 GetMaxLength() . . . 819
10.231.2.2 GetNumberOfComponents() . . . 819
10.231.2.3 Print() . . . 820
10.231.2.4 SetBlob() . . . 820
10.231.2.5 SetComponents() [1/2] . . . 820
10.231.2.6 SetComponents() [2/2] . . . 820

10.231.3 Member Data Documentation . . . 820
10.231.3.1 Component . . . 820
10.231.3.2 MaxLength . . . 820
10.231.3.3 MaxNumberOfComponents . . . 821
10.231.3.4 Padding . . . 821
10.231.3.5 Separator . . . 821
10.232 gdcm::PGXCodec Class Reference . . . 821
10.232.1 Detailed Description . . . 822
10.232.2 Constructor & Destructor Documentation . . . 822
10.232.2.1 PGXCodec() . . . 822
10.232.2.2 ∼PGXCodec() . . . 823
10.232.3 Member Function Documentation . . . 823
10.232.3.1 CanCode() . . . 823
10.232.3.2 CanDecode() . . . 823
10.232.3.3 Clone() . . . 823
10.232.3.4 GetHeaderInfo() . . . 823
10.232.3.5 Read() . . . 824
10.232.3.6 Write() . . . 824
10.233 gdcm::PhotometricInterpretation Class Reference . . . 824
10.233.1 Detailed Description . . . 825
10.233.2 Member Enumeration Documentation . . . 825
10.233.2.1 PIType . . . 826
10.233.3 Constructor & Destructor Documentation . . . 826
10.233.3.1 PhotometricInterpretation() . . . 826
10.233.4 Member Function Documentation . . . 826
10.233.4.1 GetPIString() . . . 827
10.233.4.2 GetPIType() . . . 827
10.233.4.3 GetSamplesPerPixel() . . . 827
10.233.4.4 GetString() . . . 827
10.233.4.5 GetType() . . . 827
10.233.4.6 IsLossless() . . . 827
10.233.4.7 IsLossy() . . . 827
10.233.4.8 IsRetired() . . . 828
10.233.4.9 IsSameColorSpace() . . . 828
10.233.4.10 operator PIType() . . . 828
10.233.5 Friends And Related Function Documentation . . . 828
10.233.5.1 operator<< . . . 828
10.234 gdcm::PixelFormat Class Reference . . . 828
10.234.1 Detailed Description . . . 830

10.234.2 Member Enumeration Documentation . . . 830
10.234.2.1 ScalarType . . . 830
10.234.3 Constructor & Destructor Documentation . . . 831
10.234.3.1 PixelFormat() [1/3] . . . 831
10.234.3.2 PixelFormat() [2/3] . . . 831
10.234.3.3 PixelFormat() [3/3] . . . 832
10.234.4 Member Function Documentation . . . 832
10.234.4.1 GetBitsAllocated() . . . 832
10.234.4.2 GetBitsStored() . . . 832
10.234.4.3 GetHighBit() . . . 832
10.234.4.4 GetMax() . . . 833
10.234.4.5 GetMin() . . . 833
10.234.4.6 GetPixelRepresentation() . . . 833
10.234.4.7 GetPixelSize() . . . 833
10.234.4.8 GetSamplesPerPixel() . . . 834
10.234.4.9 GetScalarType() . . . 834
10.234.4.10 GetScalarTypeAsString() . . . 834
10.234.4.11 IsCompatible() . . . 834
10.234.4.12 IsValid() . . . 835
10.234.4.13 operator ScalarType() . . . 835
10.234.4.14 operator"!=() [1/2] . . . 835
10.234.4.15 operator"!=() [2/2] . . . 835
10.234.4.16 operator==() [1/2] . . . 835
10.234.4.17 operator==() [2/2] . . . 835
10.234.4.18 Print() . . . 836
10.234.4.19 SetBitsAllocated() . . . 836
10.234.4.20 SetBitsStored() . . . 836
10.234.4.21 SetHighBit() . . . 836
10.234.4.22 SetPixelRepresentation() . . . 836
10.234.4.23 SetSamplesPerPixel() . . . 837
10.234.4.24 SetScalarType() . . . 837
10.234.4.25 Validate() . . . 837
10.234.5 Friends And Related Function Documentation . . . 837
10.234.5.1 Bitmap . . . 837
10.234.5.2 operator<< . . . 838
10.235 gdcm::Pixmap Class Reference . . . 838
10.235.1 Detailed Description . . . 840
10.235.2 Constructor & Destructor Documentation . . . 840
10.235.2.1 Pixmap() . . . 840

10.235.2.2 ~Pixmap() . . . 840
10.235.3 Member Function Documentation . . . 840
10.235.3.1 AreOverlaysInPixelData() . . . 840
10.235.3.2 GetCurve() [1/2] . . . 841
10.235.3.3 GetCurve() [2/2] . . . 841
10.235.3.4 GetIconImage() [1/2] . . . 841
10.235.3.5 GetIconImage() [2/2] . . . 841
10.235.3.6 GetNumberOfCurves() . . . 841
10.235.3.7 GetNumberOfOverlays() . . . 841
10.235.3.8 GetOverlay() [1/2] . . . 842
10.235.3.9 GetOverlay() [2/2] . . . 842
10.235.3.10 Print() . . . 842
10.235.3.11 RemoveOverlay() . . . 842
10.235.3.12 SetIconImage() . . . 842
10.235.3.13 SetNumberOfCurves() . . . 842
10.235.3.14 SetNumberOfOverlays() . . . 843
10.235.3.15 UnusedBitsPresentInPixelData() . . . 843
10.235.4 Member Data Documentation . . . 843
10.235.4.1 Curves . . . 843
10.235.4.2 Icon . . . 843
10.235.4.3 Overlays . . . 843
10.236 gdcm::PixmapReader Class Reference . . . 844
10.236.1 Detailed Description . . . 846
10.236.2 Constructor & Destructor Documentation . . . 846
10.236.2.1 PixmapReader() . . . 846
10.236.2.2 ~PixmapReader() . . . 846
10.236.3 Member Function Documentation . . . 846
10.236.3.1 GetPixmap() [1/2] . . . 847
10.236.3.2 GetPixmap() [2/2] . . . 847
10.236.3.3 Read() . . . 847
10.236.3.4 ReadACRNEMAImage() . . . 847
10.236.3.5 ReadImage() . . . 848
10.236.3.6 ReadImageInternal() . . . 848
10.236.4 Member Data Documentation . . . 848
10.236.4.1 PixelData . . . 848
10.237 gdcm::PixmapToPixmapFilter Class Reference . . . 848
10.237.1 Detailed Description . . . 849
10.237.2 Constructor & Destructor Documentation . . . 850
10.237.2.1 PixmapToPixmapFilter() . . . 850

10.237.2.2 ~PixmapToPixmapFilter() . . . 850
10.237.3 Member Function Documentation . . . 850
10.237.3.1 GetInput() . . . 850
10.237.3.2 GetOutput() . . . 850
10.237.3.3 GetOutputAsPixmap() . . . 850
10.238 gdcm::PixmapWriter Class Reference . . . 851
10.238.1 Detailed Description . . . 853
10.238.2 Constructor & Destructor Documentation . . . 853
10.238.2.1 PixmapWriter() . . . 853
10.238.2.2 ~PixmapWriter() . . . 853
10.238.3 Member Function Documentation . . . 853
10.238.3.1 DoIconImage() . . . 854
10.238.3.2 GetImage() [1/2] . . . 854
10.238.3.3 GetImage() [2/2] . . . 854
10.238.3.4 GetPixmap() [1/2] . . . 854
10.238.3.5 GetPixmap() [2/2] . . . 854
10.238.3.6 PrepareWrite() . . . 854
10.238.3.7 SetImage() . . . 855
10.238.3.8 SetPixmap() . . . 855
10.238.3.9 Write() . . . 855
10.238.4 Member Data Documentation . . . 855
10.238.4.1 PixelData . . . 855
10.239 gdcm::PNMCodec Class Reference . . . 856
10.239.1 Detailed Description . . . 857
10.239.2 Constructor & Destructor Documentation . . . 857
10.239.2.1 PNMCodec() . . . 857
10.239.2.2 ~PNMCodec() . . . 857
10.239.3 Member Function Documentation . . . 857
10.239.3.1 CanCode() . . . 858
10.239.3.2 CanDecode() . . . 858
10.239.3.3 Clone() . . . 858
10.239.3.4 GetBufferLength() . . . 858
10.239.3.5 GetHeaderInfo() . . . 858
10.239.3.6 Read() . . . 859
10.239.3.7 SetBufferLength() . . . 859
10.239.3.8 Write() . . . 859
10.240 gdcm::Preamble Class Reference . . . 859
10.240.1 Detailed Description . . . 860
10.240.2 Constructor & Destructor Documentation . . . 860

10.240.2.1 Preamble() [1/2] . . . 861
10.240.2.2 ~Preamble() . . . 861
10.240.2.3 Preamble() [2/2] . . . 861
10.240.3 Member Function Documentation . . . 861
10.240.3.1 Clear() . . . 861
10.240.3.2 Create() . . . 861
10.240.3.3 GetInternal() . . . 861
10.240.3.4 GetLength() . . . 862
10.240.3.5 IsEmpty() . . . 862
10.240.3.6 IsValid() . . . 862
10.240.3.7 operator=() . . . 862
10.240.3.8 Print() . . . 862
10.240.3.9 Read() . . . 862
10.240.3.10 Remove() . . . 863
10.240.3.11 Valid() . . . 863
10.240.3.12 Write() . . . 863
10.240.4 Friends And Related Function Documentation . . . 863
10.240.4.1 operator<< . . . 863
10.241 gdcm::PresentationContext Class Reference . . . 864
10.241.1 Detailed Description . . . 865
10.241.2 Member Typedef Documentation . . . 865
10.241.2.1 SizeType . . . 865
10.241.2.2 TransferSyntaxArrayType . . . 865
10.241.3 Constructor & Destructor Documentation . . . 865
10.241.3.1 PresentationContext() [1/2] . . . 865
10.241.3.2 PresentationContext() [2/2] . . . 866
10.241.4 Member Function Documentation . . . 866
10.241.4.1 AddTransferSyntax() . . . 866
10.241.4.2 GetAbstractSyntax() . . . 866
10.241.4.3 GetNumberOfTransferSyntaxes() . . . 866
10.241.4.4 GetPresentationContextID() . . . 866
10.241.4.5 GetTransferSyntax() . . . 866
10.241.4.6 operator==() . . . 867
10.241.4.7 Print() . . . 867
10.241.4.8 SetAbstractSyntax() . . . 867
10.241.4.9 SetPresentationContextID() . . . 867
10.241.5 Member Data Documentation . . . 867
10.241.5.1 AbstractSyntax . . . 867
10.241.5.2 ID . . . 867

10.241.5.3 TransferSyntaxes . . . 868
10.242 gdcm::network::PresentationContextAC Class Reference . . . 868
10.242.1 Detailed Description . . . 868
10.242.2 Constructor & Destructor Documentation . . . 868
10.242.2.1 PresentationContextAC() . . . 869
10.242.3 Member Function Documentation . . . 869
10.242.3.1 GetPresentationContextID() . . . 869
10.242.3.2 GetReason() . . . 869
10.242.3.3 GetTransferSyntax() . . . 869
10.242.3.4 Print() . . . 869
10.242.3.5 Read() . . . 869
10.242.3.6 SetPresentationContextID() . . . 870
10.242.3.7 SetReason() . . . 870
10.242.3.8 SetTransferSyntax() . . . 870
10.242.3.9 Size() . . . 870
10.242.3.10 Write() . . . 870
10.243 gdcm::PresentationContextGenerator Class Reference . . . 870
10.243.1 Detailed Description . . . 871
10.243.2 Member Typedef Documentation . . . 872
10.243.2.1 PresentationContextArrayType . . . 872
10.243.2.2 SizeType . . . 872
10.243.3 Constructor & Destructor Documentation . . . 872
10.243.3.1 PresentationContextGenerator() . . . 872
10.243.4 Member Function Documentation . . . 872
10.243.4.1 AddFromFile() . . . 872
10.243.4.2 AddPresentationContext() . . . 872
10.243.4.3 GenerateFromFilenames() . . . 873
10.243.4.4 GenerateFromUID() . . . 873
10.243.4.5 GetDefaultTransferSyntax() . . . 873
10.243.4.6 GetPresentationContexts() . . . 873
10.243.4.7 SetDefaultTransferSyntax() . . . 873
10.243.4.8 SetMergeModeToAbstractSyntax() . . . 874
10.243.4.9 SetMergeModeToTransferSyntax() . . . 874
10.244 gdcm::network::PresentationContextRQ Class Reference . . . 874
10.244.1 Detailed Description . . . 875
10.244.2 Member Typedef Documentation . . . 875
10.244.2.1 SizeType . . . 875
10.244.3 Constructor & Destructor Documentation . . . 875
10.244.3.1 PresentationContextRQ() [1/3] . . . 875

10.244.3.2 PresentationContextRQ() [2/3] . . . 875
10.244.3.3 PresentationContextRQ() [3/3] . . . 875
10.244.4 Member Function Documentation . . . 876
10.244.4.1 AddTransferSyntax() . . . 876
10.244.4.2 GetAbstractSyntax() [1/2] . . . 876
10.244.4.3 GetAbstractSyntax() [2/2] . . . 876
10.244.4.4 GetNumberOfTransferSyntaxes() . . . 876
10.244.4.5 GetPresentationContextID() . . . 876
10.244.4.6 GetTransferSyntax() [1/2] . . . 876
10.244.4.7 GetTransferSyntax() [2/2] . . . 877
10.244.4.8 GetTransferSyntaxes() . . . 877
10.244.4.9 operator==() . . . 877
10.244.4.10 Print() . . . 877
10.244.4.11 Read() . . . 877
10.244.4.12 SetAbstractSyntax() . . . 877
10.244.4.13 SetPresentationContextID() . . . 878
10.244.4.14 Size() . . . 878
10.244.4.15 Write() . . . 878
10.245 gdcm::network::PresentationDataValue Class Reference . . . 878
10.245.1 Detailed Description . . . 879
10.245.2 Constructor & Destructor Documentation . . . 879
10.245.2.1 PresentationDataValue() . . . 879
10.245.3 Member Function Documentation . . . 879
10.245.3.1 ConcatenatePDVBlobs() . . . 879
10.245.3.2 ConcatenatePDVBlobsAsExplicit() . . . 879
10.245.3.3 GetBlob() . . . 880
10.245.3.4 GetIsCommand() . . . 880
10.245.3.5 GetIsLastFragment() . . . 880
10.245.3.6 GetMessageHeader() . . . 880
10.245.3.7 GetPresentationContextID() . . . 880
10.245.3.8 Print() . . . 880
10.245.3.9 Read() . . . 880
10.245.3.10 ReadInto() . . . 881
10.245.3.11 SetBlob() . . . 881
10.245.3.12 SetCommand() . . . 881
10.245.3.13 SetDataSet() . . . 881
10.245.3.14 SetLastFragment() . . . 881
10.245.3.15 SetMessageHeader() . . . 882
10.245.3.16 SetPresentationContextID() . . . 882

10.245.3.17 Size() . . . 882
10.245.3.18 Write() . . . 882
10.246 gdcm::Printer Class Reference . . . 882
10.246.1 Detailed Description . . . 884
10.246.2 Member Enumeration Documentation . . . 884
10.246.2.1 PrintStyles . . . 884
10.246.3 Constructor & Destructor Documentation . . . 884
10.246.3.1 Printer() . . . 884
10.246.3.2 ∼Printer() . . . 885
10.246.4 Member Function Documentation . . . 885
10.246.4.1 GetPrintStyle() . . . 885
10.246.4.2 Print() . . . 885
10.246.4.3 PrintDataElement() . . . 885
10.246.4.4 PrintDataSet() . . . 886
10.246.4.5 PrintSQ() . . . 886
10.246.4.6 SetColor() . . . 886
10.246.4.7 SetFile() . . . 886
10.246.4.8 SetStyle() . . . 886
10.246.5 Member Data Documentation . . . 887
10.246.5.1 F . . . 887
10.246.5.2 MaxPrintLength . . . 887
10.246.5.3 PrintStyle . . . 887
10.247 gdcm::PrivateDict Class Reference . . . 887
10.247.1 Detailed Description . . . 888
10.247.2 Constructor & Destructor Documentation . . . 888
10.247.2.1 PrivateDict() . . . 888
10.247.2.2 ∼PrivateDict() . . . 888
10.247.3 Member Function Documentation . . . 888
10.247.3.1 AddDictEntry() . . . 888
10.247.3.2 FindDictEntry() . . . 889
10.247.3.3 GetDictEntry() . . . 889
10.247.3.4 IsEmpty() . . . 889
10.247.3.5 LoadDefault() . . . 889
10.247.3.6 PrintXML() . . . 889
10.247.3.7 RemoveDictEntry() . . . 889
10.247.4 Friends And Related Function Documentation . . . 890
10.247.4.1 Dicts . . . 890
10.247.4.2 operator<< . . . 890
10.248 gdcm::PrivateTag Class Reference . . . 890

10.248.1 Detailed Description . . . 891
10.248.2 Constructor & Destructor Documentation . . . 892
10.248.2.1 PrivateTag() [1/2] . . . 892
10.248.2.2 PrivateTag() [2/2] . . . 892
10.248.3 Member Function Documentation . . . 892
10.248.3.1 GetAsDataElement() . . . 892
10.248.3.2 GetOwner() . . . 892
10.248.3.3 operator"!=() [1/2] . . . 893
10.248.3.4 operator"!=() [2/2] . . . 893
10.248.3.5 operator<() . . . 893
10.248.3.6 operator=() . . . 893
10.248.3.7 operator==() [1/2] . . . 893
10.248.3.8 operator==() [2/2] . . . 894
10.248.3.9 ReadFromCommaSeparatedString() . . . 894
10.248.3.10 SetOwner() . . . 894
10.248.4 Friends And Related Function Documentation . . . 894
10.248.4.1 operator<< . . . 894
10.249 gdcm::ProgressEvent Class Reference . . . 895
10.249.1 Detailed Description . . . 896
10.249.2 Member Typedef Documentation . . . 896
10.249.2.1 Self . . . 896
10.249.2.2 Superclass . . . 896
10.249.3 Constructor & Destructor Documentation . . . 896
10.249.3.1 ProgressEvent() [1/2] . . . 897
10.249.3.2 ~ProgressEvent() . . . 897
10.249.3.3 ProgressEvent() [2/2] . . . 897
10.249.4 Member Function Documentation . . . 897
10.249.4.1 CheckEvent() . . . 897
10.249.4.2 GetEventName() . . . 897
10.249.4.3 GetProgress() . . . 898
10.249.4.4 MakeObject() . . . 898
10.249.4.5 operator=() . . . 898
10.249.4.6 SetProgress() . . . 898
10.250 gdcm::PVRGCodec Class Reference . . . 899
10.250.1 Detailed Description . . . 900
10.250.2 Constructor & Destructor Documentation . . . 900
10.250.2.1 PVRGCodec() . . . 900
10.250.2.2 ~PVRGCodec() . . . 900
10.250.3 Member Function Documentation . . . 900

10.250.3.1 CanCode() . . . 901
10.250.3.2 CanDecode() . . . 901
10.250.3.3 Clone() . . . 901
10.250.3.4 Code() . . . 901
10.250.3.5 Decode() . . . 902
10.250.3.6 SetLossyFlag() . . . 902
10.251 gdcm::PythonFilter Class Reference . . . 902
10.251.1 Detailed Description . . . 902
10.251.2 Constructor & Destructor Documentation . . . 903
10.251.2.1 PythonFilter() . . . 903
10.251.2.2 ∼PythonFilter() . . . 903
10.251.3 Member Function Documentation . . . 903
10.251.3.1 GetFile() [1/2] . . . 903
10.251.3.2 GetFile() [2/2] . . . 903
10.251.3.3 SetDicts() . . . 903
10.251.3.4 SetFile() . . . 903
10.251.3.5 ToPyObject() . . . 904
10.251.3.6 UseDictAlways() . . . 904
10.252 gdcm::QueryBase Class Reference . . . 904
10.252.1 Detailed Description . . . 905
10.252.2 Constructor & Destructor Documentation . . . 905
10.252.2.1 ∼QueryBase() . . . 905
10.252.3 Member Function Documentation . . . 905
10.252.3.1 GetAllRequiredTags() . . . 905
10.252.3.2 GetAllTags() . . . 906
10.252.3.3 GetHierachicalSearchTags() . . . 906
10.252.3.4 GetName() . . . 906
10.252.3.5 GetOptionalTags() . . . 906
10.252.3.6 GetQueryLevel() . . . 906
10.252.3.7 GetRequiredTags() . . . 907
10.252.3.8 GetUniqueTags() . . . 907
10.253 gdcm::QueryFactory Class Reference . . . 907
10.253.1 Detailed Description . . . 907
10.253.2 Member Function Documentation . . . 908
10.253.2.1 GetCharacterFromCurrentLocale() . . . 908
10.253.2.2 ListCharSets() . . . 908
10.253.2.3 ProduceCharacterSetDataElement() . . . 908
10.253.2.4 ProduceQuery() [1/2] . . . 908
10.253.2.5 ProduceQuery() [2/2] . . . 909

10.254 gdcm::QueryImage Class Reference . . . 909
10.254.1 Detailed Description . . . 910
10.254.2 Member Function Documentation . . . 910
10.254.2.1 GetHierachicalSearchTags() . . . 910
10.254.2.2 GetName() . . . 910
10.254.2.3 GetOptionalTags() . . . 910
10.254.2.4 GetQueryLevel() . . . 911
10.254.2.5 GetRequiredTags() . . . 911
10.254.2.6 GetUniqueTags() . . . 911
10.255 gdcm::QueryPatient Class Reference . . . 911
10.255.1 Detailed Description . . . 912
10.255.2 Member Function Documentation . . . 912
10.255.2.1 GetHierachicalSearchTags() . . . 912
10.255.2.2 GetName() . . . 913
10.255.2.3 GetOptionalTags() . . . 913
10.255.2.4 GetQueryLevel() . . . 913
10.255.2.5 GetRequiredTags() . . . 913
10.255.2.6 GetUniqueTags() . . . 913
10.256 gdcm::QuerySeries Class Reference . . . 914
10.256.1 Detailed Description . . . 915
10.256.2 Member Function Documentation . . . 915
10.256.2.1 GetHierachicalSearchTags() . . . 915
10.256.2.2 GetName() . . . 915
10.256.2.3 GetOptionalTags() . . . 915
10.256.2.4 GetQueryLevel() . . . 915
10.256.2.5 GetRequiredTags() . . . 916
10.256.2.6 GetUniqueTags() . . . 916
10.257 gdcm::QueryStudy Class Reference . . . 916
10.257.1 Detailed Description . . . 917
10.257.2 Member Function Documentation . . . 917
10.257.2.1 GetHierachicalSearchTags() . . . 917
10.257.2.2 GetName() . . . 918
10.257.2.3 GetOptionalTags() . . . 918
10.257.2.4 GetQueryLevel() . . . 918
10.257.2.5 GetRequiredTags() . . . 918
10.257.2.6 GetUniqueTags() . . . 918
10.258 gdcm::RAWCodec Class Reference . . . 919
10.258.1 Detailed Description . . . 920
10.258.2 Constructor & Destructor Documentation . . . 920

10.258.2.1 RAWCodec() . . . 920
10.258.2.2 ~RAWCodec() . . . 920
10.258.3 Member Function Documentation . . . 920
10.258.3.1 CanCode() . . . 921
10.258.3.2 CanDecode() . . . 921
10.258.3.3 Clone() . . . 921
10.258.3.4 Code() . . . 921
10.258.3.5 Decode() . . . 922
10.258.3.6 DecodeByStreams() . . . 922
10.258.3.7 DecodeBytes() . . . 922
10.258.3.8 GetHeaderInfo() . . . 922
10.259 gdcm::Reader Class Reference . . . 923
10.259.1 Detailed Description . . . 925
10.259.2 Constructor & Destructor Documentation . . . 926
10.259.2.1 Reader() . . . 926
10.259.2.2 ~Reader() . . . 926
10.259.3 Member Function Documentation . . . 926
10.259.3.1 CanRead() . . . 926
10.259.3.2 GetFile() [1/2] . . . 926
10.259.3.3 GetFile() [2/2] . . . 927
10.259.3.4 GetStreamCurrentPosition() . . . 927
10.259.3.5 GetStreamPtr() . . . 927
10.259.3.6 Read() . . . 928
10.259.3.7 ReadDataSet() . . . 928
10.259.3.8 ReadMetaInformation() . . . 928
10.259.3.9 ReadPreamble() . . . 928
10.259.3.10 ReadSelectedPrivateTags() . . . 929
10.259.3.11 ReadSelectedTags() . . . 929
10.259.3.12 ReadUpToTag() . . . 929
10.259.3.13 SetFile() . . . 929
10.259.3.14 SetFileName() . . . 930
10.259.3.15 SetStream() . . . 930
10.259.4 Friends And Related Function Documentation . . . 930
10.259.4.1 StreamImageReader . . . 930
10.259.5 Member Data Documentation . . . 931
10.259.5.1 F . . . 931
10.260 gdcm::RealWorldValueMappingContent Struct Reference . . . 931
10.260.1 Member Data Documentation . . . 932
10.260.1.1 CodeMeaning . . . 932

10.260.1.2 CodeValue . . . 932
10.260.1.3 RealWorldValueIntercept . . . 932
10.260.1.4 RealWorldValueSlope . . . 932
10.261 gdcm::Region Class Reference . . . 932
10.261.1 Detailed Description . . . 933
10.261.2 Constructor & Destructor Documentation . . . 933
10.261.2.1 Region() . . . 933
10.261.2.2 ∼Region() . . . 933
10.261.3 Member Function Documentation . . . 933
10.261.3.1 Area() . . . 934
10.261.3.2 Clone() . . . 934
10.261.3.3 ComputeBoundingBox() . . . 934
10.261.3.4 Empty() . . . 934
10.261.3.5 IsValid() . . . 934
10.261.3.6 Print() . . . 935
10.262 gdcm::Rescaler Class Reference . . . 935
10.262.1 Detailed Description . . . 936
10.262.2 Constructor & Destructor Documentation . . . 937
10.262.2.1 Rescaler() . . . 937
10.262.2.2 ∼Rescaler() . . . 937
10.262.3 Member Function Documentation . . . 937
10.262.3.1 ComputeInterceptSlopePixelType() . . . 937
10.262.3.2 ComputePixelTypeFromMinMax() . . . 937
10.262.3.3 GetIntercept() . . . 937
10.262.3.4 GetSlope() . . . 938
10.262.3.5 InverseRescale() . . . 938
10.262.3.6 InverseRescaleFunctionIntoBestFit() . . . 938
10.262.3.7 Rescale() . . . 938
10.262.3.8 RescaleFunctionIntoBestFit() . . . 939
10.262.3.9 SetIntercept() . . . 939
10.262.3.10 SetMinMaxForPixelType() . . . 939
10.262.3.11 SetPixelFormat() . . . 939
10.262.3.12 SetSlope() . . . 940
10.262.3.13 SetTargetPixelType() . . . 940
10.262.3.14 SetUseTargetPixelType() . . . 940
10.263 gdcm::RLECodec Class Reference . . . 940
10.263.1 Detailed Description . . . 942
10.263.2 Constructor & Destructor Documentation . . . 942
10.263.2.1 RLECodec() . . . 942

10.263.2.2 ~RLECodec() . . . 943
10.263.3 Member Function Documentation . . . 943
10.263.3.1 AppendFrameEncode() . . . 943
10.263.3.2 AppendRowEncode() . . . 943
10.263.3.3 CanCode() . . . 943
10.263.3.4 CanDecode() . . . 944
10.263.3.5 Clone() . . . 944
10.263.3.6 Code() . . . 944
10.263.3.7 Decode() . . . 944
10.263.3.8 DecodeByStreams() . . . 945
10.263.3.9 DecodeExtent() . . . 945
10.263.3.10 GetBufferLength() . . . 945
10.263.3.11 GetHeaderInfo() . . . 945
10.263.3.12 IsFrameEncoder() . . . 945
10.263.3.13 IsRowEncoder() . . . 946
10.263.3.14 SetBufferLength() . . . 946
10.263.3.15 SetLength() . . . 946
10.263.3.16 StartEncode() . . . 946
10.263.3.17 StopEncode() . . . 946
10.263.4 Friends And Related Function Documentation . . . 946
10.263.4.1 ImageRegionReader . . . 947
10.264 gdcm::network::RoleSelectionSub Class Reference . . . 947
10.264.1 Detailed Description . . . 947
10.264.2 Constructor & Destructor Documentation . . . 947
10.264.2.1 RoleSelectionSub() . . . 947
10.264.3 Member Function Documentation . . . 947
10.264.3.1 Print() . . . 948
10.264.3.2 Read() . . . 948
10.264.3.3 SetTuple() . . . 948
10.264.3.4 Size() . . . 948
10.264.3.5 Write() . . . 948
10.265 gdcm::Scanner Class Reference . . . 949
10.265.1 Detailed Description . . . 951
10.265.2 Member Typedef Documentation . . . 951
10.265.2.1 ConstIterator . . . 951
10.265.2.2 MappingType . . . 952
10.265.2.3 TagToValue . . . 952
10.265.2.4 TagToValueValueType . . . 952
10.265.2.5 ValuesType . . . 952

10.265.3 Constructor & Destructor Documentation . . . 952
10.265.3.1 Scanner() . . . 952
10.265.3.2 ∼Scanner() . . . 952
10.265.4 Member Function Documentation . . . 952
10.265.4.1 AddPrivateTag() . . . 953
10.265.4.2 AddSkipTag() . . . 953
10.265.4.3 AddTag() . . . 953
10.265.4.4 Begin() . . . 953
10.265.4.5 ClearSkipTags() . . . 953
10.265.4.6 ClearTags() . . . 953
10.265.4.7 End() . . . 954
10.265.4.8 GetAllFilenamesFromTagToValue() . . . 954
10.265.4.9 GetFilenameFromTagToValue() . . . 954
10.265.4.10 GetFilenames() . . . 954
10.265.4.11 GetKeys() . . . 954
10.265.4.12 GetMapping() . . . 955
10.265.4.13 GetMappingFromTagToValue() . . . 955
10.265.4.14 GetMappings() . . . 955
10.265.4.15 GetOrderedValues() . . . 955
10.265.4.16 GetValue() . . . 955
10.265.4.17 GetValues() [1/2] . . . 956
10.265.4.18 GetValues() [2/2] . . . 956
10.265.4.19 IsKey() . . . 956
10.265.4.20 New() . . . 956
10.265.4.21 Print() . . . 957
10.265.4.22 PrintTable() . . . 957
10.265.4.23 ProcessPublicTag() . . . 957
10.265.4.24 Scan() . . . 957
10.265.5 Friends And Related Function Documentation . . . 957
10.265.5.1 operator<< . . . 958
10.266 gdcm::Scanner2 Class Reference . . . 958
10.266.1 Detailed Description . . . 961
10.266.2 Member Typedef Documentation . . . 961
10.266.2.1 PrivateConstIterator . . . 961
10.266.2.2 PrivateMappingType . . . 961
10.266.2.3 PrivateTagToValue . . . 962
10.266.2.4 PrivateTagToValueValueType . . . 962
10.266.2.5 PublicConstIterator . . . 962
10.266.2.6 PublicMappingType . . . 962

10.266.2.7 PublicTagToValue . . . 962
10.266.2.8 PublicTagToValueValueType . . . 962
10.266.2.9 ValuesType . . . 962
10.266.3 Constructor & Destructor Documentation . . . 963
10.266.3.1 Scanner2() . . . 963
10.266.3.2 ~Scanner2() . . . 963
10.266.4 Member Function Documentation . . . 963
10.266.4.1 AddPrivateTag() . . . 963
10.266.4.2 AddPublicTag() . . . 963
10.266.4.3 AddSkipTag() . . . 963
10.266.4.4 Begin() . . . 964
10.266.4.5 ClearPrivateTags() . . . 964
10.266.4.6 ClearPublicTags() . . . 964
10.266.4.7 ClearSkipTags() . . . 964
10.266.4.8 End() . . . 964
10.266.4.9 GetAllFilenamesFromPrivateTagToValue() . . . 964
10.266.4.10 GetAllFilenamesFromPublicTagToValue() . . . 964
10.266.4.11 GetFilenameFromPrivateTagToValue() . . . 965
10.266.4.12 GetFilenameFromPublicTagToValue() . . . 965
10.266.4.13 GetFilenames() . . . 965
10.266.4.14 GetKeys() . . . 965
10.266.4.15 GetMappingFromPrivateTagToValue() . . . 965
10.266.4.16 GetMappingFromPublicTagToValue() . . . 965
10.266.4.17 GetPrivateMapping() . . . 966
10.266.4.18 GetPrivateMappings() . . . 966
10.266.4.19 GetPrivateOrderedValues() . . . 966
10.266.4.20 GetPrivateValue() . . . 966
10.266.4.21 GetPrivateValues() . . . 966
10.266.4.22 GetPublicMapping() . . . 966
10.266.4.23 GetPublicMappings() . . . 967
10.266.4.24 GetPublicOrderedValues() . . . 967
10.266.4.25 GetPublicValue() . . . 967
10.266.4.26 GetPublicValues() . . . 967
10.266.4.27 GetValues() . . . 967
10.266.4.28 IsKey() . . . 968
10.266.4.29 New() . . . 968
10.266.4.30 Print() . . . 968
10.266.4.31 PrintTable() . . . 968
10.266.4.32 PrivateBegin() . . . 968

10.266.4.33 PrivateEnd() . . . 969
10.266.4.34 ProcessPrivateTag() . . . 969
10.266.4.35 ProcessPublicTag() . . . 969
10.266.4.36 Scan() . . . 969
10.266.5 Friends And Related Function Documentation . . . 969
10.266.5.1 operator<< . . . 969
10.267 gdcm::Segment Class Reference . . . 970
10.267.1 Detailed Description . . . 972
10.267.2 Member Typedef Documentation . . . 972
10.267.2.1 BasicCodedEntryVector . . . 972
10.267.2.2 SurfaceVector . . . 972
10.267.3 Member Enumeration Documentation . . . 972
10.267.3.1 ALGOType . . . 972
10.267.4 Constructor & Destructor Documentation . . . 973
10.267.4.1 Segment() . . . 973
10.267.4.2 ∼Segment() . . . 973
10.267.5 Member Function Documentation . . . 973
10.267.5.1 AddSurface() . . . 973
10.267.5.2 GetALGOType() . . . 973
10.267.5.3 GetALGOTypeString() . . . 974
10.267.5.4 GetAnatomicRegion() [1/2] . . . 974
10.267.5.5 GetAnatomicRegion() [2/2] . . . 974
10.267.5.6 GetAnatomicRegionModifiers() [1/2] . . . 974
10.267.5.7 GetAnatomicRegionModifiers() [2/2] . . . 974
10.267.5.8 GetPropertyCategory() [1/2] . . . 974
10.267.5.9 GetPropertyCategory() [2/2] . . . 974
10.267.5.10 GetPropertyType() [1/2] . . . 975
10.267.5.11 GetPropertyType() [2/2] . . . 975
10.267.5.12 GetPropertyTypeModifiers() [1/2] . . . 975
10.267.5.13 GetPropertyTypeModifiers() [2/2] . . . 975
10.267.5.14 GetSegmentAlgorithmName() . . . 975
10.267.5.15 GetSegmentAlgorithmType() . . . 975
10.267.5.16 GetSegmentDescription() . . . 975
10.267.5.17 GetSegmentLabel() . . . 976
10.267.5.18 GetSegmentNumber() . . . 976
10.267.5.19 GetSurface() . . . 976
10.267.5.20 GetSurfaceCount() . . . 976
10.267.5.21 GetSurfaces() [1/2] . . . 976
10.267.5.22 GetSurfaces() [2/2] . . . 976

10.267.5.23 SetAnatomicRegion() . . . 976
10.267.5.24 SetAnatomicRegionModifiers() . . . 977
10.267.5.25 SetPropertyCategory() . . . 977
10.267.5.26 SetPropertyType() . . . 977
10.267.5.27 SetPropertyTypeModifiers() . . . 977
10.267.5.28 SetSegmentAlgorithmName() . . . 977
10.267.5.29 SetSegmentAlgorithmType() [1/2] . . . 977
10.267.5.30 SetSegmentAlgorithmType() [2/2] . . . 978
10.267.5.31 SetSegmentDescription() . . . 978
10.267.5.32 SetSegmentLabel() . . . 978
10.267.5.33 SetSegmentNumber() . . . 978
10.267.5.34 SetSurfaceCount() . . . 978
10.267.6 Member Data Documentation . . . 978
10.267.6.1 AnatomicRegion . . . 978
10.267.6.2 AnatomicRegionModifiers . . . 979
10.267.6.3 PropertyCategory . . . 979
10.267.6.4 PropertyType . . . 979
10.267.6.5 PropertyTypeModifiers . . . 979
10.267.6.6 SegmentAlgorithmName . . . 979
10.267.6.7 SegmentAlgorithmType . . . 979
10.267.6.8 SegmentDescription . . . 979
10.267.6.9 SegmentLabel . . . 980
10.267.6.10 SegmentNumber . . . 980
10.267.6.11 SurfaceCount . . . 980
10.267.6.12 Surfaces . . . 980
10.268 gdcm::SegmentedPaletteColorLookupTable Class Reference . . . 980
10.268.1 Detailed Description . . . 981
10.268.2 Constructor & Destructor Documentation . . . 981
10.268.2.1 SegmentedPaletteColorLookupTable() . . . 981
10.268.2.2 ~SegmentedPaletteColorLookupTable() . . . 982
10.268.3 Member Function Documentation . . . 982
10.268.3.1 Print() . . . 982
10.268.3.2 SetLUT() . . . 982
10.269 gdcm::SegmentReader Class Reference . . . 983
10.269.1 Detailed Description . . . 984
10.269.2 Member Typedef Documentation . . . 984
10.269.2.1 SegmentMap . . . 985
10.269.2.2 SegmentVector . . . 985
10.269.3 Constructor & Destructor Documentation . . . 985

10.269.3.1 SegmentReader() . . . 985
10.269.3.2 ~SegmentReader() . . . 985
10.269.4 Member Function Documentation . . . 985
10.269.4.1 GetSegments() [1/2] . . . 985
10.269.4.2 GetSegments() [2/2] . . . 985
10.269.4.3 Read() . . . 986
10.269.4.4 ReadSegment() . . . 986
10.269.4.5 ReadSegments() . . . 986
10.269.5 Member Data Documentation . . . 986
10.269.5.1 Segments . . . 986
10.270 gdcm::SegmentWriter Class Reference . . . 987
10.270.1 Detailed Description . . . 988
10.270.2 Member Typedef Documentation . . . 988
10.270.2.1 SegmentVector . . . 988
10.270.3 Constructor & Destructor Documentation . . . 988
10.270.3.1 SegmentWriter() . . . 988
10.270.3.2 ~SegmentWriter() . . . 988
10.270.4 Member Function Documentation . . . 989
10.270.4.1 AddSegment() . . . 989
10.270.4.2 GetNumberOfSegments() . . . 989
10.270.4.3 GetSegment() . . . 989
10.270.4.4 GetSegments() [1/2] . . . 989
10.270.4.5 GetSegments() [2/2] . . . 989
10.270.4.6 PrepareWrite() . . . 989
10.270.4.7 SetNumberOfSegments() . . . 990
10.270.4.8 SetSegments() . . . 990
10.270.4.9 Write() . . . 990
10.270.5 Member Data Documentation . . . 990
10.270.5.1 Segments . . . 990
10.271 gdcm::SequenceOfFragments Class Reference . . . 991
10.271.1 Detailed Description . . . 992
10.271.2 Member Typedef Documentation . . . 993
10.271.2.1 ConstIterator . . . 993
10.271.2.2 FragmentVector . . . 993
10.271.2.3 Iterator . . . 993
10.271.2.4 SizeType . . . 993
10.271.3 Constructor & Destructor Documentation . . . 993
10.271.3.1 SequenceOfFragments() . . . 993
10.271.4 Member Function Documentation . . . 993

10.271.4.1 AddFragment() . . . 994
10.271.4.2 Begin() [1/2] . . . 994
10.271.4.3 Begin() [2/2] . . . 994
10.271.4.4 Clear() . . . 994
10.271.4.5 ComputeByteLength() . . . 994
10.271.4.6 ComputeLength() . . . 994
10.271.4.7 End() [1/2] . . . 995
10.271.4.8 End() [2/2] . . . 995
10.271.4.9 GetBuffer() . . . 995
10.271.4.10 GetFragBuffer() . . . 995
10.271.4.11 GetFragment() . . . 995
10.271.4.12 GetLength() . . . 996
10.271.4.13 GetNumberOfFragments() . . . 996
10.271.4.14 GetTable() [1/2] . . . 996
10.271.4.15 GetTable() [2/2] . . . 996
10.271.4.16 New() . . . 996
10.271.4.17 operator==() . . . 997
10.271.4.18 Print() . . . 997
10.271.4.19 Read() . . . 997
10.271.4.20 ReadPreValue() . . . 997
10.271.4.21 ReadValue() . . . 997
10.271.4.22 SetLength() . . . 998
10.271.4.23 Write() . . . 998
10.271.4.24 WriteBuffer() . . . 998
10.272 gdcm::SequenceOfItems Class Reference . . . 999
10.272.1 Detailed Description . . . 1001
10.272.2 Member Typedef Documentation . . . 1001
10.272.2.1 ConstIterator . . . 1001
10.272.2.2 ItemVector . . . 1002
10.272.2.3 Iterator . . . 1002
10.272.2.4 SizeType . . . 1002
10.272.3 Constructor & Destructor Documentation . . . 1002
10.272.3.1 SequenceOfItems() . . . 1002
10.272.4 Member Function Documentation . . . 1002
10.272.4.1 AddItem() . . . 1002
10.272.4.2 AddNewUndefinedLengthItem() . . . 1003
10.272.4.3 Begin() [1/2] . . . 1003
10.272.4.4 Begin() [2/2] . . . 1003
10.272.4.5 Clear() . . . 1003

10.272.4.6 ComputeLength() . . . 1003
10.272.4.7 End() [1/2] . . . 1003
10.272.4.8 End() [2/2] . . . 1004
10.272.4.9 FindDataElement() . . . 1004
10.272.4.10 GetItem() [1/2] . . . 1004
10.272.4.11 GetItem() [2/2] . . . 1004
10.272.4.12 GetLength() . . . 1004
10.272.4.13 GetNumberOfItems() . . . 1005
10.272.4.14 IsEmpty() . . . 1005
10.272.4.15 IsUndefinedLength() . . . 1005
10.272.4.16 New() . . . 1005
10.272.4.17 operator=() . . . 1005
10.272.4.18 operator==() . . . 1006
10.272.4.19 Print() . . . 1006
10.272.4.20 Read() . . . 1006
10.272.4.21 RemoveItemByIndex() . . . 1006
10.272.4.22 SetLength() . . . 1006
10.272.4.23 SetLengthToUndefined() . . . 1007
10.272.4.24 SetNumberOfItems() . . . 1007
10.272.4.25 Write() . . . 1007
10.272.5 Member Data Documentation . . . 1007
10.272.5.1 Items . . . 1007
10.272.5.2 SequenceLengthField . . . 1007
10.273 gdcm::SerieHelper Class Reference . . . 1008
10.273.1 Detailed Description . . . 1009
10.273.2 Member Typedef Documentation . . . 1009
10.273.2.1 Rule . . . 1009
10.273.2.2 SerieRestrictions . . . 1010
10.273.2.3 SingleSerieUIDFileSetmap . . . 1010
10.273.3 Constructor & Destructor Documentation . . . 1010
10.273.3.1 SerieHelper() . . . 1010
10.273.3.2 ~SerieHelper() . . . 1010
10.273.4 Member Function Documentation . . . 1010
10.273.4.1 AddFile() . . . 1010
10.273.4.2 AddFileName() . . . 1010
10.273.4.3 AddRestriction() [1/3] . . . 1011
10.273.4.4 AddRestriction() [2/3] . . . 1011
10.273.4.5 AddRestriction() [3/3] . . . 1011
10.273.4.6 Clear() . . . 1011

10.273.4.7 CreateDefaultUniqueSeriesIdentifier() . . . 1011
10.273.4.8 CreateUniqueSeriesIdentifier() . . . 1011
10.273.4.9 FileNameOrdering() . . . 1012
10.273.4.10 GetFirstSingleSerieUIDFileSet() . . . 1012
10.273.4.11 GetNextSingleSerieUIDFileSet() . . . 1012
10.273.4.12 ImageNumberOrdering() . . . 1012
10.273.4.13 ImagePositionPatientOrdering() . . . 1012
10.273.4.14 OrderFileList() . . . 1012
10.273.4.15 SetDirectory() . . . 1013
10.273.4.16 SetLoadMode() . . . 1013
10.273.4.17 SetUseSeriesDetails() . . . 1013
10.273.4.18 UserOrdering() . . . 1013
10.273.5 Member Data Documentation . . . 1013
10.273.5.1 elem . . . 1013
10.273.5.2 ItFileSetHt . . . 1013
10.273.5.3 op . . . 1014
10.273.5.4 SingleSerieUIDFileSetHT . . . 1014
10.273.5.5 value . . . 1014
10.274 gdcm::Series Class Reference . . . 1014
10.274.1 Detailed Description . . . 1014
10.274.2 Constructor & Destructor Documentation . . . 1014
10.274.2.1 Series() . . . 1015
10.275 gdcm::network::ServiceClassApplicationInformation Class Reference . . . 1015
10.275.1 Detailed Description . . . 1015
10.275.2 Constructor & Destructor Documentation . . . 1015
10.275.2.1 ServiceClassApplicationInformation() . . . 1015
10.275.3 Member Function Documentation . . . 1015
10.275.3.1 Print() . . . 1016
10.275.3.2 Read() . . . 1016
10.275.3.3 SetTuple() . . . 1016
10.275.3.4 Size() . . . 1016
10.275.3.5 Write() . . . 1016
10.276 gdcm::ServiceClassUser Class Reference . . . 1017
10.276.1 Detailed Description . . . 1019
10.276.2 Constructor & Destructor Documentation . . . 1019
10.276.2.1 ServiceClassUser() [1/2] . . . 1019
10.276.2.2 ~ServiceClassUser() . . . 1019
10.276.2.3 ServiceClassUser() [2/2] . . . 1019
10.276.3 Member Function Documentation . . . 1020

10.276.3.1 GetAETitle() . . . 1020
10.276.3.2 GetCalledAETitle() . . . 1020
10.276.3.3 GetTimeout() . . . 1020
10.276.3.4 InitializeConnection() . . . 1020
10.276.3.5 IsPresentationContextAccepted() . . . 1020
10.276.3.6 New() . . . 1021
10.276.3.7 operator=() . . . 1021
10.276.3.8 SendEcho() . . . 1021
10.276.3.9 SendFind() . . . 1021
10.276.3.10 SendMove() [1/3] . . . 1021
10.276.3.11 SendMove() [2/3] . . . 1022
10.276.3.12 SendMove() [3/3] . . . 1022
10.276.3.13 SendStore() [1/3] . . . 1022
10.276.3.14 SendStore() [2/3] . . . 1022
10.276.3.15 SendStore() [3/3] . . . 1022
10.276.3.16 SetAETitle() . . . 1023
10.276.3.17 SetCalledAETitle() . . . 1023
10.276.3.18 SetHostname() . . . 1023
10.276.3.19 SetPort() . . . 1023
10.276.3.20 SetPortSCP() . . . 1024
10.276.3.21 SetPresentationContexts() . . . 1024
10.276.3.22 SetTimeout() . . . 1024
10.276.3.23 StartAssociation() . . . 1024
10.276.3.24 StopAssociation() . . . 1025
10.277 gdcm::SHA1 Class Reference . . . 1025
10.277.1 Detailed Description . . . 1025
10.277.2 Constructor & Destructor Documentation . . . 1026
10.277.2.1 SHA1() [1/2] . . . 1026
10.277.2.2 ∼SHA1() . . . 1026
10.277.2.3 SHA1() [2/2] . . . 1026
10.277.3 Member Function Documentation . . . 1026
10.277.3.1 Compute() . . . 1026
10.277.3.2 ComputeFile() . . . 1026
10.277.3.3 operator=() . . . 1027
10.278 gdcm::SimpleMemberCommand< T > Class Template Reference . . . 1027
10.278.1 Detailed Description . . . 1029
10.278.2 Member Typedef Documentation . . . 1029
10.278.2.1 Self . . . 1029
10.278.2.2 TMemberFunctionPointer . . . 1029

10.278.3 Constructor & Destructor Documentation . . . 1029
10.278.3.1 SimpleMemberCommand() [1/2] . . . 1029
10.278.3.2 SimpleMemberCommand() [2/2] . . . 1030
10.278.3.3 ~SimpleMemberCommand() . . . 1030
10.278.4 Member Function Documentation . . . 1030
10.278.4.1 Execute() [1/2] . . . 1030
10.278.4.2 Execute() [2/2] . . . 1030
10.278.4.3 New() . . . 1031
10.278.4.4 operator=() . . . 1031
10.278.4.5 SetCallbackFunction() . . . 1031
10.278.5 Member Data Documentation . . . 1031
10.278.5.1 m_MemberFunction . . . 1031
10.278.5.2 m_This . . . 1032
10.279 gdcm::SimpleSubjectWatcher Class Reference . . . 1032
10.279.1 Detailed Description . . . 1032
10.279.2 Constructor & Destructor Documentation . . . 1033
10.279.2.1 SimpleSubjectWatcher() [1/2] . . . 1033
10.279.2.2 ~SimpleSubjectWatcher() . . . 1033
10.279.2.3 SimpleSubjectWatcher() [2/2] . . . 1033
10.279.3 Member Function Documentation . . . 1033
10.279.3.1 EndFilter() . . . 1033
10.279.3.2 operator=() . . . 1033
10.279.3.3 ShowAbort() . . . 1034
10.279.3.4 ShowAnonymization() . . . 1034
10.279.3.5 ShowData() . . . 1034
10.279.3.6 ShowDataSet() . . . 1034
10.279.3.7 ShowFileName() . . . 1034
10.279.3.8 ShowIteration() . . . 1035
10.279.3.9 ShowProgress() . . . 1035
10.279.3.10 StartFilter() . . . 1035
10.279.3.11 TestAbortOff() . . . 1035
10.279.3.12 TestAbortOn() . . . 1035
10.280 gdcm::MrProtocol::Slice Struct Reference . . . 1036
10.280.1 Member Data Documentation . . . 1036
10.280.1.1 Normal . . . 1036
10.280.1.2 Position . . . 1036
10.281 gdcm::MrProtocol::SliceArray Struct Reference . . . 1037
10.281.1 Member Data Documentation . . . 1037
10.281.1.1 Slices . . . 1037

10.282 gdcm::SmartPointer< ObjectType > Class Template Reference . . . 1038
10.282.1 Detailed Description . . . 1039
10.282.2 Constructor & Destructor Documentation . . . 1039
10.282.2.1 SmartPointer() [1/4] . . . 1040
10.282.2.2 SmartPointer() [2/4] . . . 1040
10.282.2.3 SmartPointer() [3/4] . . . 1040
10.282.2.4 SmartPointer() [4/4] . . . 1040
10.282.2.5 ~SmartPointer() . . . 1040
10.282.3 Member Function Documentation . . . 1040
10.282.3.1 GetPointer() . . . 1041
10.282.3.2 operator ObjectType *() . . . 1041
10.282.3.3 operator*() . . . 1041
10.282.3.4 operator->() . . . 1041
10.282.3.5 operator=() [1/3] . . . 1041
10.282.3.6 operator=() [2/3] . . . 1042
10.282.3.7 operator=() [3/3] . . . 1042
10.283 gdcm::network::SOPClassExtendedNegociationSub Class Reference . . . 1042
10.283.1 Detailed Description . . . 1043
10.283.2 Constructor & Destructor Documentation . . . 1043
10.283.2.1 SOPClassExtendedNegociationSub() . . . 1043
10.283.3 Member Function Documentation . . . 1043
10.283.3.1 Print() . . . 1043
10.283.3.2 Read() . . . 1043
10.283.3.3 SetTuple() . . . 1043
10.283.3.4 Size() . . . 1044
10.283.3.5 Write() . . . 1044
10.284 gdcm::SOPClassUIDToIOD Class Reference . . . 1044
10.284.1 Detailed Description . . . 1044
10.284.2 Member Typedef Documentation . . . 1045
10.284.2.1 const . . . 1045
10.284.3 Member Function Documentation . . . 1045
10.284.3.1 GetIOD() . . . 1045
10.284.3.2 GetIODFromSOPClassUID() . . . 1045
10.284.3.3 GetNumberOfSOPClassToIOD() . . . 1045
10.284.3.4 GetSOPClassUIDFromIOD() . . . 1046
10.284.3.5 GetSOPClassUIDToIOD() . . . 1046
10.284.3.6 GetSOPClassUIDToIODs() . . . 1046
10.285 gdcm::Sorter Class Reference . . . 1046
10.285.1 Detailed Description . . . 1048

10.285.2 Member Typedef Documentation . . . 1048
10.285.2.1 SelectionMap . . . 1048
10.285.2.2 SortFunction . . . 1048
10.285.3 Constructor & Destructor Documentation . . . 1048
10.285.3.1 Sorter() . . . 1049
10.285.3.2 ∼Sorter() . . . 1049
10.285.4 Member Function Documentation . . . 1049
10.285.4.1 AddSelect() . . . 1049
10.285.4.2 GetFilenames() . . . 1049
10.285.4.3 Print() . . . 1049
10.285.4.4 SetSortFunction() . . . 1050
10.285.4.5 SetTagsToRead() . . . 1050
10.285.4.6 Sort() . . . 1050
10.285.4.7 StableSort() . . . 1050
10.285.5 Friends And Related Function Documentation . . . 1050
10.285.5.1 operator<< . . . 1051
10.285.6 Member Data Documentation . . . 1051
10.285.6.1 Filenames . . . 1051
10.285.6.2 Selection . . . 1051
10.285.6.3 SortFunc . . . 1051
10.285.6.4 TagsToRead . . . 1051
10.286 gdcm::Spacing Class Reference . . . 1051
10.286.1 Detailed Description . . . 1052
10.286.2 Member Enumeration Documentation . . . 1053
10.286.2.1 SpacingType . . . 1053
10.286.3 Constructor & Destructor Documentation . . . 1053
10.286.3.1 Spacing() . . . 1054
10.286.3.2 ∼Spacing() . . . 1054
10.286.4 Member Function Documentation . . . 1054
10.286.4.1 ComputePixelAspectRatioFromPixelSpacing() . . . 1054
10.287 gdcm::Spectroscopy Class Reference . . . 1054
10.287.1 Detailed Description . . . 1054
10.287.2 Constructor & Destructor Documentation . . . 1054
10.287.2.1 Spectroscopy() . . . 1055
10.288 gdcm::SplitMosaicFilter Class Reference . . . 1055
10.288.1 Detailed Description . . . 1056
10.288.2 Constructor & Destructor Documentation . . . 1056
10.288.2.1 SplitMosaicFilter() . . . 1056
10.288.2.2 ∼SplitMosaicFilter() . . . 1056

10.288.3 Member Function Documentation . . . 1056
10.288.3.1 ComputeMOSAICDimensions() . . . 1056
10.288.3.2 ComputeMOSAICSliceNormal() . . . 1057
10.288.3.3 ComputeMOSAICSlicePosition() . . . 1057
10.288.3.4 GetAcquisitionSize() . . . 1057
10.288.3.5 GetFile() [1/2] . . . 1057
10.288.3.6 GetFile() [2/2] . . . 1057
10.288.3.7 GetImage() [1/2] . . . 1057
10.288.3.8 GetImage() [2/2] . . . 1058
10.288.3.9 GetNumberOfImagesInMosaic() . . . 1058
10.288.3.10 SetFile() . . . 1058
10.288.3.11 SetImage() . . . 1058
10.288.3.12 Split() . . . 1058
10.289 gdcm::StartEvent Class Reference . . . 1059
10.290 gdcm::static_assert_test< x > Struct Template Reference . . . 1060
10.291 gdcm::STATIC_ASSERTION_FAILURE< x > Struct Template Reference . . . 1060
10.292 gdcm::STATIC_ASSERTION_FAILURE< true > Struct Reference . . . 1060
10.292.1 Member Enumeration Documentation . . . 1060
10.292.1.1 anonymous enum . . . 1060
10.293 gdcm::StreamImageReader Class Reference . . . 1061
10.293.1 Detailed Description . . . 1061
10.293.2 Constructor & Destructor Documentation . . . 1061
10.293.2.1 StreamImageReader() . . . 1062
10.293.2.2 ~StreamImageReader() . . . 1062
10.293.3 Member Function Documentation . . . 1062
10.293.3.1 CanReadImage() . . . 1062
10.293.3.2 DefinePixelExtent() . . . 1062
10.293.3.3 DefineProperBufferLength() . . . 1063
10.293.3.4 GetDimensionsValueForResolution() . . . 1063
10.293.3.5 GetFile() . . . 1063
10.293.3.6 Read() . . . 1063
10.293.3.7 ReadImageInformation() . . . 1064
10.293.3.8 SetFileName() . . . 1064
10.293.3.9 SetStream() . . . 1064
10.294 gdcm::StreamImageWriter Class Reference . . . 1065
10.294.1 Detailed Description . . . 1066
10.294.2 Constructor & Destructor Documentation . . . 1066
10.294.2.1 StreamImageWriter() . . . 1067
10.294.2.2 ~StreamImageWriter() . . . 1067

10.294.3 Member Function Documentation . . . 1067
10.294.3.1 CanWriteFile() . . . 1067
10.294.3.2 DefinePixelExtent() . . . 1067
10.294.3.3 DefineProperBufferLength() . . . 1068
10.294.3.4 SetFile() . . . 1068
10.294.3.5 SetFileName() . . . 1068
10.294.3.6 SetStream() . . . 1068
10.294.3.7 Write() . . . 1069
10.294.3.8 WriteImageInformation() . . . 1069
10.294.3.9 WriteImageSubregionRAW() . . . 1069
10.294.3.10 WriteRawHeader() . . . 1070
10.294.4 Member Data Documentation . . . 1070
10.294.4.1 mElementOffsets . . . 1070
10.294.4.2 mElementOffsets1 . . . 1070
10.294.4.3 mspFile . . . 1070
10.294.4.4 mWriter . . . 1070
10.294.4.5 mXMax . . . 1071
10.294.4.6 mXMin . . . 1071
10.294.4.7 mYMax . . . 1071
10.294.4.8 mYMin . . . 1071
10.294.4.9 mZMax . . . 1071
10.294.4.10 mZMin . . . 1071
10.295 gdcm::StrictScanner Class Reference . . . 1072
10.295.1 Detailed Description . . . 1074
10.295.2 Member Typedef Documentation . . . 1074
10.295.2.1 ConstIterator . . . 1074
10.295.2.2 MappingType . . . 1075
10.295.2.3 TagToValue . . . 1075
10.295.2.4 TagToValueValueType . . . 1075
10.295.2.5 ValuesType . . . 1075
10.295.3 Constructor & Destructor Documentation . . . 1075
10.295.3.1 StrictScanner() . . . 1075
10.295.3.2 ~StrictScanner() . . . 1075
10.295.4 Member Function Documentation . . . 1075
10.295.4.1 AddPrivateTag() . . . 1076
10.295.4.2 AddSkipTag() . . . 1076
10.295.4.3 AddTag() . . . 1076
10.295.4.4 Begin() . . . 1076
10.295.4.5 ClearSkipTags() . . . 1076

10.295.4.6 ClearTags() . . . 1076
10.295.4.7 End() . . . 1077
10.295.4.8 GetAllFilenamesFromTagToValue() . . . 1077
10.295.4.9 GetFilenameFromTagToValue() . . . 1077
10.295.4.10 GetFilenames() . . . 1077
10.295.4.11 GetKeys() . . . 1077
10.295.4.12 GetMapping() . . . 1077
10.295.4.13 GetMappingFromTagToValue() . . . 1078
10.295.4.14 GetMappings() . . . 1078
10.295.4.15 GetOrderedValues() . . . 1078
10.295.4.16 GetValue() . . . 1078
10.295.4.17 GetValues() [1/2] . . . 1078
10.295.4.18 GetValues() [2/2] . . . 1079
10.295.4.19 IsKey() . . . 1079
10.295.4.20 New() . . . 1079
10.295.4.21 Print() . . . 1079
10.295.4.22 PrintTable() . . . 1080
10.295.4.23 ProcessPublicTag() . . . 1080
10.295.4.24 Scan() . . . 1080
10.295.5 Friends And Related Function Documentation . . . 1080
10.295.5.1 operator<< . . . 1080
10.296 gdcm::StrictScanner2 Class Reference . . . 1081
10.296.1 Detailed Description . . . 1083
10.296.2 Member Typedef Documentation . . . 1084
10.296.2.1 PrivateConstIterator . . . 1084
10.296.2.2 PrivateMappingType . . . 1084
10.296.2.3 PrivateTagToValue . . . 1084
10.296.2.4 PrivateTagToValueValueType . . . 1084
10.296.2.5 PublicConstIterator . . . 1084
10.296.2.6 PublicMappingType . . . 1084
10.296.2.7 PublicTagToValue . . . 1085
10.296.2.8 PublicTagToValueValueType . . . 1085
10.296.2.9 ValuesType . . . 1085
10.296.3 Constructor & Destructor Documentation . . . 1085
10.296.3.1 StrictScanner2() . . . 1085
10.296.3.2 ~StrictScanner2() . . . 1085
10.296.4 Member Function Documentation . . . 1085
10.296.4.1 AddPrivateTag() . . . 1085
10.296.4.2 AddPublicTag() . . . 1086

10.296.4.3 AddSkipTag() . . . 1086
10.296.4.4 Begin() . . . 1086
10.296.4.5 ClearPrivateTags() . . . 1086
10.296.4.6 ClearPublicTags() . . . 1086
10.296.4.7 ClearSkipTags() . . . 1086
10.296.4.8 End() . . . 1087
10.296.4.9 GetAllFilenamesFromPrivateTagToValue() . . . 1087
10.296.4.10 GetAllFilenamesFromPublicTagToValue() . . . 1087
10.296.4.11 GetFilenameFromPrivateTagToValue() . . . 1087
10.296.4.12 GetFilenameFromPublicTagToValue() . . . 1087
10.296.4.13 GetFilenames() . . . 1087
10.296.4.14 GetKeys() . . . 1088
10.296.4.15 GetMappingFromPrivateTagToValue() . . . 1088
10.296.4.16 GetMappingFromPublicTagToValue() . . . 1088
10.296.4.17 GetPrivateMapping() . . . 1088
10.296.4.18 GetPrivateMappings() . . . 1088
10.296.4.19 GetPrivateOrderedValues() . . . 1088
10.296.4.20 GetPrivateValue() . . . 1089
10.296.4.21 GetPrivateValues() . . . 1089
10.296.4.22 GetPublicMapping() . . . 1089
10.296.4.23 GetPublicMappings() . . . 1089
10.296.4.24 GetPublicOrderedValues() . . . 1089
10.296.4.25 GetPublicValue() . . . 1089
10.296.4.26 GetPublicValues() . . . 1090
10.296.4.27 GetValues() . . . 1090
10.296.4.28 IsKey() . . . 1090
10.296.4.29 New() . . . 1090
10.296.4.30 Print() . . . 1090
10.296.4.31 PrintTable() . . . 1091
10.296.4.32 PrivateBegin() . . . 1091
10.296.4.33 PrivateEnd() . . . 1091
10.296.4.34 ProcessPrivateTag() . . . 1091
10.296.4.35 ProcessPublicTag() . . . 1091
10.296.4.36 Scan() . . . 1091
10.296.5 Friends And Related Function Documentation . . . 1092
10.296.5.1 operator<< . . . 1092
10.297 gdcm::String< TDelimiter, TMaxLength, TPadChar > Class Template Reference . . . 1092
10.297.1 Detailed Description . . . 1094
10.297.2 Member Typedef Documentation . . . 1094

10.297.2.1 const_iterator . . . 1094
10.297.2.2 const_reference . . . 1094
10.297.2.3 const_reverse_iterator . . . 1094
10.297.2.4 difference_type . . . 1095
10.297.2.5 iterator . . . 1095
10.297.2.6 pointer . . . 1095
10.297.2.7 reference . . . 1095
10.297.2.8 reverse_iterator . . . 1095
10.297.2.9 size_type . . . 1095
10.297.2.10 value_type . . . 1096
10.297.3 Constructor & Destructor Documentation . . . 1096
10.297.3.1 String() [1/4] . . . 1096
10.297.3.2 String() [2/4] . . . 1096
10.297.3.3 String() [3/4] . . . 1096
10.297.3.4 String() [4/4] . . . 1096
10.297.4 Member Function Documentation . . . 1097
10.297.4.1 IsValid() . . . 1097
10.297.4.2 operator const char *() . . . 1097
10.297.4.3 Trim() [1/2] . . . 1097
10.297.4.4 Trim() [2/2] . . . 1097
10.297.4.5 Truncate() . . . 1098
10.298 gdcm::StringFilter Class Reference . . . 1098
10.298.1 Detailed Description . . . 1099
10.298.2 Constructor & Destructor Documentation . . . 1099
10.298.2.1 StringFilter() . . . 1099
10.298.2.2 ~StringFilter() . . . 1099
10.298.3 Member Function Documentation . . . 1099
10.298.3.1 ExecuteQuery() [1/2] . . . 1099
10.298.3.2 ExecuteQuery() [2/2] . . . 1099
10.298.3.3 FromString() . . . 1100
10.298.3.4 GetFile() [1/2] . . . 1100
10.298.3.5 GetFile() [2/2] . . . 1100
10.298.3.6 SetDicts() . . . 1100
10.298.3.7 SetFile() . . . 1100
10.298.3.8 ToString() [1/3] . . . 1101
10.298.3.9 ToString() [2/3] . . . 1101
10.298.3.10 ToString() [3/3] . . . 1101
10.298.3.11 ToStringPair() [1/3] . . . 1101
10.298.3.12 ToStringPair() [2/3] . . . 1102

10.298.3.13 ToStringPair() [3/3] . . . 1102
10.298.3.14 UseDictAlways() . . . 1102
10.299 gdcm::Study Class Reference . . . 1102
10.299.1 Detailed Description . . . 1102
10.299.2 Constructor & Destructor Documentation . . . 1102
10.299.2.1 Study() . . . 1103
10.300 gdcm::Subject Class Reference . . . 1103
10.300.1 Detailed Description . . . 1104
10.300.2 Constructor & Destructor Documentation . . . 1104
10.300.2.1 Subject() . . . 1105
10.300.2.2 ∼Subject() . . . 1105
10.300.3 Member Function Documentation . . . 1105
10.300.3.1 AddObserver() [1/2] . . . 1105
10.300.3.2 AddObserver() [2/2] . . . 1105
10.300.3.3 GetCommand() . . . 1105
10.300.3.4 HasObserver() . . . 1106
10.300.3.5 InvokeEvent() [1/2] . . . 1106
10.300.3.6 InvokeEvent() [2/2] . . . 1106
10.300.3.7 RemoveAllObservers() . . . 1106
10.300.3.8 RemoveObserver() . . . 1106
10.301 gdcm::Surface Class Reference . . . 1107
10.301.1 Detailed Description . . . 1109
10.301.2 Member Enumeration Documentation . . . 1109
10.301.2.1 STATES . . . 1109
10.301.2.2 VIEWType . . . 1110
10.301.3 Constructor & Destructor Documentation . . . 1110
10.301.3.1 Surface() . . . 1110
10.301.3.2 ∼Surface() . . . 1110
10.301.4 Member Function Documentation . . . 1111
10.301.4.1 GetAlgorithmFamily() [1/2] . . . 1111
10.301.4.2 GetAlgorithmFamily() [2/2] . . . 1111
10.301.4.3 GetAlgorithmName() . . . 1111
10.301.4.4 GetAlgorithmVersion() . . . 1111
10.301.4.5 GetAxisOfRotation() . . . 1111
10.301.4.6 GetCenterOfRotation() . . . 1112
10.301.4.7 GetFiniteVolume() . . . 1112
10.301.4.8 GetManifold() . . . 1112
10.301.4.9 GetMaximumPointDistance() . . . 1112
10.301.4.10 GetMeanPointDistance() . . . 1112

10.301.4.11 GetMeshPrimitive() [1/2] . . . 1112
10.301.4.12 GetMeshPrimitive() [2/2] . . . 1113
10.301.4.13 GetNumberOfSurfacePoints() . . . 1113
10.301.4.14 GetNumberOfVectors() . . . 1113
10.301.4.15 GetPointCoordinatesData() [1/2] . . . 1113
10.301.4.16 GetPointCoordinatesData() [2/2] . . . 1113
10.301.4.17 GetPointPositionAccuracy() . . . 1113
10.301.4.18 GetPointsBoundingBoxCoordinates() . . . 1114
10.301.4.19 GetProcessingAlgorithm() [1/2] . . . 1114
10.301.4.20 GetProcessingAlgorithm() [2/2] . . . 1114
10.301.4.21 GetRecommendedDisplayCIELabValue() [1/2] . . . 1114
10.301.4.22 GetRecommendedDisplayCIELabValue() [2/2] . . . 1114
10.301.4.23 GetRecommendedDisplayGrayscaleValue() . . . 1114
10.301.4.24 GetRecommendedPresentationOpacity() . . . 1115
10.301.4.25 GetRecommendedPresentationType() . . . 1115
10.301.4.26 GetSTATES() . . . 1115
10.301.4.27 GetSTATESString() . . . 1115
10.301.4.28 GetSurfaceComments() . . . 1115
10.301.4.29 GetSurfaceNumber() . . . 1115
10.301.4.30 GetSurfaceProcessing() . . . 1115
10.301.4.31 GetSurfaceProcessingDescription() . . . 1116
10.301.4.32 GetSurfaceProcessingRatio() . . . 1116
10.301.4.33 GetVectorAccuracy() . . . 1116
10.301.4.34 GetVectorCoordinateData() [1/2] . . . 1116
10.301.4.35 GetVectorCoordinateData() [2/2] . . . 1116
10.301.4.36 GetVectorDimensionality() . . . 1116
10.301.4.37 GetVIEWType() . . . 1116
10.301.4.38 GetVIEWTypeString() . . . 1117
10.301.4.39 SetAlgorithmFamily() . . . 1117
10.301.4.40 SetAlgorithmName() . . . 1117
10.301.4.41 SetAlgorithmVersion() . . . 1117
10.301.4.42 SetAxisOfRotation() . . . 1117
10.301.4.43 SetCenterOfRotation() . . . 1117
10.301.4.44 SetFiniteVolume() . . . 1118
10.301.4.45 SetManifold() . . . 1118
10.301.4.46 SetMaximumPointDistance() . . . 1118
10.301.4.47 SetMeanPointDistance() . . . 1118
10.301.4.48 SetMeshPrimitive() . . . 1118
10.301.4.49 SetNumberOfSurfacePoints() . . . 1118

10.301.4.50 SetNumberOfVectors() . . . 1119
10.301.4.51 SetPointCoordinatesData() . . . 1119
10.301.4.52 SetPointPositionAccuracy() . . . 1119
10.301.4.53 SetPointsBoundingBoxCoordinates() . . . 1119
10.301.4.54 SetProcessingAlgorithm() . . . 1119
10.301.4.55 SetRecommendedDisplayCIELabValue() [1/3] . . . 1119
10.301.4.56 SetRecommendedDisplayCIELabValue() [2/3] . . . 1120
10.301.4.57 SetRecommendedDisplayCIELabValue() [3/3] . . . 1120
10.301.4.58 SetRecommendedDisplayGrayscaleValue() . . . 1120
10.301.4.59 SetRecommendedPresentationOpacity() . . . 1120
10.301.4.60 SetRecommendedPresentationType() . . . 1120
10.301.4.61 SetSurfaceComments() . . . 1120
10.301.4.62 SetSurfaceNumber() . . . 1121
10.301.4.63 SetSurfaceProcessing() . . . 1121
10.301.4.64 SetSurfaceProcessingDescription() . . . 1121
10.301.4.65 SetSurfaceProcessingRatio() . . . 1121
10.301.4.66 SetVectorAccuracy() . . . 1121
10.301.4.67 SetVectorCoordinateData() . . . 1121
10.301.4.68 SetVectorDimensionality() . . . 1122
10.302 gdcm::SurfaceHelper Class Reference . . . 1122
10.302.1 Detailed Description . . . 1122
10.302.2 Member Typedef Documentation . . . 1123
10.302.2.1 ColorArray . . . 1123
10.302.3 Member Function Documentation . . . 1123
10.302.3.1 RecommendedDisplayCIELabToRGB() [1/2] . . . 1123
10.302.3.2 RecommendedDisplayCIELabToRGB() [2/2] . . . 1123
10.302.3.3 RGBToRecommendedDisplayCIELab() . . . 1124
10.302.3.4 RGBToRecommendedDisplayGrayscale() . . . 1125
10.303 gdcm::SurfaceReader Class Reference . . . 1125
10.303.1 Detailed Description . . . 1127
10.303.2 Constructor & Destructor Documentation . . . 1127
10.303.2.1 SurfaceReader() . . . 1127
10.303.2.2 ~SurfaceReader() . . . 1127
10.303.3 Member Function Documentation . . . 1127
10.303.3.1 GetNumberOfSurfaces() . . . 1128
10.303.3.2 Read() . . . 1128
10.303.3.3 ReadPointMacro() . . . 1128
10.303.3.4 ReadSurface() . . . 1128
10.303.3.5 ReadSurfaces() . . . 1128

10.304 gdcm::SurfaceWriter Class Reference . . . 1129
10.304.1 Detailed Description . . . 1130
10.304.2 Constructor & Destructor Documentation . . . 1130
10.304.2.1 SurfaceWriter() . . . 1130
10.304.2.2 ∼SurfaceWriter() . . . 1130
10.304.3 Member Function Documentation . . . 1130
10.304.3.1 ComputeNumberOfSurfaces() . . . 1130
10.304.3.2 GetNumberOfSurfaces() . . . 1131
10.304.3.3 PrepareWrite() . . . 1131
10.304.3.4 PrepareWritePointMacro() . . . 1131
10.304.3.5 SetNumberOfSurfaces() . . . 1131
10.304.3.6 Write() . . . 1131
10.304.4 Member Data Documentation . . . 1131
10.304.4.1 NumberOfSurfaces . . . 1132
10.305 gdcm::SwapCode Class Reference . . . 1132
10.305.1 Detailed Description . . . 1133
10.305.2 Member Enumeration Documentation . . . 1133
10.305.2.1 SwapCodeType . . . 1133
10.305.3 Constructor & Destructor Documentation . . . 1133
10.305.3.1 SwapCode() . . . 1133
10.305.4 Member Function Documentation . . . 1133
10.305.4.1 GetIndex() . . . 1133
10.305.4.2 GetSwapCodeString() . . . 1134
10.305.4.3 operator SwapCode::SwapCodeType() . . . 1134
10.305.5 Friends And Related Function Documentation . . . 1134
10.305.5.1 operator<< . . . 1134
10.306 gdcm::SwapperDoOp Class Reference . . . 1134
10.306.1 Member Function Documentation . . . 1134
10.306.1.1 Swap() . . . 1135
10.306.1.2 SwapArray() . . . 1135
10.307 gdcm::SwapperNoOp Class Reference . . . 1135
10.307.1 Detailed Description . . . 1135
10.307.2 Member Function Documentation . . . 1135
10.307.2.1 Swap() . . . 1136
10.307.2.2 SwapArray() . . . 1136
10.308 gdcm::System Class Reference . . . 1136
10.308.1 Detailed Description . . . 1137
10.308.2 Member Function Documentation . . . 1137
10.308.2.1 ConvertToUNC() . . . 1138

10.308.2.2 DeleteDirectory() . . . 1138
10.308.2.3 EncodeBytes() . . . 1138
10.308.2.4 FileExists() . . . 1138
10.308.2.5 FileIsDirectory() . . . 1138
10.308.2.6 FileIsSymlink() . . . 1139
10.308.2.7 FileSize() . . . 1139
10.308.2.8 FileTime() . . . 1139
10.308.2.9 FormatDateTime() . . . 1139
10.308.2.10 GetCurrentDateTime() . . . 1140
10.308.2.11 GetCurrentModuleFileName() . . . 1140
10.308.2.12 GetCurrentProcessFileName() . . . 1140
10.308.2.13 GetCurrentResourcesDirectory() . . . 1140
10.308.2.14 GetCWD() . . . 1140
10.308.2.15 GetHostName() . . . 1140
10.308.2.16 GetLastSystemError() . . . 1141
10.308.2.17 GetLocaleCharset() . . . 1141
10.308.2.18 GetPermissions() . . . 1141
10.308.2.19 GetTimezoneOffsetFromUTC() . . . 1141
10.308.2.20 MakeDirectory() . . . 1141
10.308.2.21 ParseDateTime() [1/2] . . . 1142
10.308.2.22 ParseDateTime() [2/2] . . . 1142
10.308.2.23 RemoveFile() . . . 1142
10.308.2.24 SetPermissions() . . . 1142
10.308.2.25 StrCaseCmp() . . . 1143
10.308.2.26 StrNCaseCmp() . . . 1143
10.308.2.27 StrSep() . . . 1143
10.308.2.28 StrTokR() . . . 1143
10.309 gdcm::Table Class Reference . . . 1144
10.309.1 Detailed Description . . . 1145
10.309.2 Member Typedef Documentation . . . 1145
10.309.2.1 MapTableEntry . . . 1145
10.309.3 Constructor & Destructor Documentation . . . 1145
10.309.3.1 Table() [1/2] . . . 1145
10.309.3.2 ~Table() . . . 1145
10.309.3.3 Table() [2/2] . . . 1145
10.309.4 Member Function Documentation . . . 1145
10.309.4.1 GetTableEntry() . . . 1146
10.309.4.2 InsertEntry() . . . 1146
10.309.4.3 operator=() . . . 1146

10.309.5 Friends And Related Function Documentation . . . 1146
10.309.5.1 operator<< . . . 1146
10.309.6 Member Data Documentation . . . 1146
10.309.6.1 TableInternal . . . 1147
10.310 gdcm::TableEntry Class Reference . . . 1147
10.310.1 Detailed Description . . . 1147
10.310.2 Constructor & Destructor Documentation . . . 1147
10.310.2.1 TableEntry() . . . 1147
10.310.2.2 ~TableEntry() . . . 1148
10.311 gdcm::TableReader Class Reference . . . 1148
10.311.1 Detailed Description . . . 1149
10.311.2 Constructor & Destructor Documentation . . . 1149
10.311.2.1 TableReader() . . . 1149
10.311.2.2 ~TableReader() . . . 1149
10.311.3 Member Function Documentation . . . 1149
10.311.3.1 CharacterDataHandler() . . . 1149
10.311.3.2 EndElement() . . . 1149
10.311.3.3 GetDefs() . . . 1150
10.311.3.4 GetFilename() . . . 1150
10.311.3.5 HandleIOD() . . . 1150
10.311.3.6 HandleIODEntry() . . . 1150
10.311.3.7 HandleMacro() . . . 1150
10.311.3.8 HandleMacroEntry() . . . 1150
10.311.3.9 HandleMacroEntryDescription() . . . 1150
10.311.3.10 HandleModule() . . . 1151
10.311.3.11 HandleModuleEntry() . . . 1151
10.311.3.12 HandleModuleEntryDescription() . . . 1151
10.311.3.13 HandleModuleInclude() . . . 1151
10.311.3.14 Read() . . . 1151
10.311.3.15 SetFilename() . . . 1151
10.311.3.16 StartElement() . . . 1152
10.312 gdcm::network::TableRow Class Reference . . . 1152
10.312.1 Constructor & Destructor Documentation . . . 1153
10.312.1.1 TableRow() . . . 1153
10.312.1.2 ~TableRow() . . . 1153
10.312.2 Member Data Documentation . . . 1153
10.312.2.1 transitions . . . 1153
10.313 gdcm::Tag Class Reference . . . 1153
10.313.1 Detailed Description . . . 1155

10.313.2 Constructor & Destructor Documentation . . . 1155
10.313.2.1 Tag() [1/3] . . . 1156
10.313.2.2 Tag() [2/3] . . . 1156
10.313.2.3 Tag() [3/3] . . . 1156
10.313.3 Member Function Documentation . . . 1156
10.313.3.1 GetElement() . . . 1156
10.313.3.2 GetElementTag() . . . 1157
10.313.3.3 GetGroup() . . . 1157
10.313.3.4 GetLength() . . . 1157
10.313.3.5 GetPrivateCreator() . . . 1157
10.313.3.6 IsGroupLength() . . . 1158
10.313.3.7 IsGroupXX() . . . 1158
10.313.3.8 IsIllegal() . . . 1158
10.313.3.9 IsPrivate() . . . 1158
10.313.3.10 IsPrivateCreator() . . . 1159
10.313.3.11 IsPublic() . . . 1159
10.313.3.12 operator"!=() . . . 1159
10.313.3.13 operator<() . . . 1159
10.313.3.14 operator<=() . . . 1159
10.313.3.15 operator=() . . . 1160
10.313.3.16 operator==() . . . 1160
10.313.3.17 operator[]() [1/2] . . . 1160
10.313.3.18 operator[]() [2/2] . . . 1160
10.313.3.19 PrintAsContinuousString() . . . 1160
10.313.3.20 PrintAsContinuousUpperCaseString() . . . 1161
10.313.3.21 PrintAsPipeSeparatedString() . . . 1161
10.313.3.22 Read() . . . 1161
10.313.3.23 ReadFromCommaSeparatedString() . . . 1161
10.313.3.24 ReadFromContinuousString() . . . 1161
10.313.3.25 ReadFromPipeSeparatedString() . . . 1162
10.313.3.26 SetElement() . . . 1162
10.313.3.27 SetElementTag() [1/2] . . . 1162
10.313.3.28 SetElementTag() [2/2] . . . 1162
10.313.3.29 SetGroup() . . . 1163
10.313.3.30 SetPrivateCreator() . . . 1163
10.313.3.31 Write() . . . 1163
10.313.4 Friends And Related Function Documentation . . . 1163
10.313.4.1 operator<< . . . 1163
10.313.4.2 operator>> . . . 1164

10.313.5 Member Data Documentation . . . 1164
10.313.5.1 bytes . . . 1164
10.313.5.2 tag . . . 1164
10.313.5.3 tags . . . 1164
10.314 gdcm::TagPath Class Reference . . . 1164
10.314.1 Detailed Description . . . 1165
10.314.2 Constructor & Destructor Documentation . . . 1165
10.314.2.1 TagPath() . . . 1165
10.314.2.2 ∼TagPath() . . . 1165
10.314.3 Member Function Documentation . . . 1165
10.314.3.1 ConstructFromString() . . . 1166
10.314.3.2 ConstructFromTagList() . . . 1166
10.314.3.3 IsValid() . . . 1166
10.314.3.4 Print() . . . 1166
10.314.3.5 Push() [1/2] . . . 1166
10.314.3.6 Push() [2/2] . . . 1167
10.315 gdcm::Testing Class Reference . . . 1167
10.315.1 Detailed Description . . . 1168
10.315.2 Member Typedef Documentation . . . 1168
10.315.2.1 MD5DataImagesType . . . 1168
10.315.2.2 MediaStorageDataFilesType . . . 1168
10.315.3 Constructor & Destructor Documentation . . . 1169
10.315.3.1 Testing() . . . 1169
10.315.3.2 ∼Testing() . . . 1169
10.315.4 Member Function Documentation . . . 1169
10.315.4.1 ComputeFileMD5() . . . 1169
10.315.4.2 ComputeMD5() . . . 1169
10.315.4.3 GetDataExtraRoot() . . . 1170
10.315.4.4 GetDataRoot() . . . 1170
10.315.4.5 GetFileName() . . . 1170
10.315.4.6 GetFileNames() . . . 1170
10.315.4.7 GetLossyFlagFromFile() . . . 1171
10.315.4.8 GetMD5DataImage() . . . 1171
10.315.4.9 GetMD5DataImages() . . . 1171
10.315.4.10 GetMD5FromBrokenFile() . . . 1171
10.315.4.11 GetMD5FromFile() . . . 1171
10.315.4.12 GetMediaStorageDataFile() . . . 1171
10.315.4.13 GetMediaStorageDataFiles() . . . 1172
10.315.4.14 GetMediaStorageFromFile() . . . 1172

10.315.4.15 GetNumberOfFileNames() . . . 1172
10.315.4.16 GetNumberOfMD5DataImages() . . . 1172
10.315.4.17 GetNumberOfMediaStorageDataFiles() . . . 1172
10.315.4.18 GetPixelSpacingDataRoot() . . . 1172
10.315.4.19 GetSelectedPrivateGroupOffsetFromFile() . . . 1173
10.315.4.20 GetSelectedTagsOffsetFromFile() . . . 1173
10.315.4.21 GetSourceDirectory() . . . 1173
10.315.4.22 GetStreamOffsetFromFile() . . . 1173
10.315.4.23 GetTempDirectory() . . . 1173
10.315.4.24 GetTempDirectoryW() . . . 1174
10.315.4.25 GetTempFilename() . . . 1174
10.315.4.26 GetTempFilenameW() . . . 1174
10.315.4.27 Print() . . . 1174
10.316 gdcm::Trace Class Reference . . . 1175
10.316.1 Detailed Description . . . 1176
10.316.2 Constructor & Destructor Documentation . . . 1176
10.316.2.1 Trace() . . . 1176
10.316.2.2 ∼Trace() . . . 1176
10.316.3 Member Function Documentation . . . 1176
10.316.3.1 DebugOff() . . . 1176
10.316.3.2 DebugOn() . . . 1177
10.316.3.3 ErrorOff() . . . 1177
10.316.3.4 ErrorOn() . . . 1177
10.316.3.5 GetDebugFlag() . . . 1177
10.316.3.6 GetDebugStream() . . . 1177
10.316.3.7 GetErrorFlag() . . . 1177
10.316.3.8 GetErrorStream() . . . 1178
10.316.3.9 GetStream() . . . 1178
10.316.3.10 GetWarningFlag() . . . 1178
10.316.3.11 GetWarningStream() . . . 1178
10.316.3.12 SetDebug() . . . 1178
10.316.3.13 SetDebugStream() . . . 1178
10.316.3.14 SetError() . . . 1179
10.316.3.15 SetErrorStream() . . . 1179
10.316.3.16 SetStream() . . . 1179
10.316.3.17 SetStreamToFile() . . . 1179
10.316.3.18 SetWarning() . . . 1179
10.316.3.19 SetWarningStream() . . . 1180
10.316.3.20 WarningOff() . . . 1180

10.316.3.21 WarningOn() . . . 1180
10.317 gdcm::TransferSyntax Class Reference . . . 1180
10.317.1 Detailed Description . . . 1182
10.317.2 Member Enumeration Documentation . . . 1182
10.317.2.1 NegociatedType . . . 1182
10.317.2.2 TSType . . . 1183
10.317.3 Constructor & Destructor Documentation . . . 1183
10.317.3.1 TransferSyntax() . . . 1184
10.317.4 Member Function Documentation . . . 1184
10.317.4.1 CanStoreLossy() . . . 1184
10.317.4.2 GetNegociatedType() . . . 1184
10.317.4.3 GetString() . . . 1184
10.317.4.4 GetSwapCode() . . . 1184
10.317.4.5 GetTSString() . . . 1185
10.317.4.6 GetTSType() . . . 1185
10.317.4.7 IsEncapsulated() . . . 1185
10.317.4.8 IsEncoded() . . . 1185
10.317.4.9 IsExplicit() . . . 1185
10.317.4.10 IsImplicit() . . . 1186
10.317.4.11 IsLossless() . . . 1186
10.317.4.12 IsLossy() . . . 1186
10.317.4.13 IsValid() . . . 1186
10.317.4.14 operator TSType() . . . 1186
10.317.5 Friends And Related Function Documentation . . . 1186
10.317.5.1 operator<< . . . 1186
10.318 gdcm::network::TransferSyntaxSub Class Reference . . . 1187
10.318.1 Detailed Description . . . 1187
10.318.2 Constructor & Destructor Documentation . . . 1187
10.318.2.1 TransferSyntaxSub() . . . 1187
10.318.3 Member Function Documentation . . . 1187
10.318.3.1 GetName() . . . 1188
10.318.3.2 operator==() . . . 1188
10.318.3.3 Print() . . . 1188
10.318.3.4 Read() . . . 1188
10.318.3.5 SetName() . . . 1188
10.318.3.6 SetNameFromUID() . . . 1188
10.318.3.7 Size() . . . 1189
10.318.3.8 Write() . . . 1189
10.319 gdcm::network::Transition Struct Reference . . . 1189

10.319.1 Constructor & Destructor Documentation . . . 1190
10.319.1.1 Transition() [1/2] . . . 1190
10.319.1.2 ~Transition() . . . 1190
10.319.1.3 Transition() [2/2] . . . 1190
10.319.2 Member Function Documentation . . . 1190
10.319.2.1 MakeNew() . . . 1190
10.319.3 Member Data Documentation . . . 1191
10.319.3.1 mAction . . . 1191
10.319.3.2 mEnd . . . 1191
10.320 gdcm::Type Class Reference . . . 1191
10.320.1 Detailed Description . . . 1192
10.320.2 Member Enumeration Documentation . . . 1192
10.320.2.1 TypeType . . . 1192
10.320.3 Constructor & Destructor Documentation . . . 1193
10.320.3.1 Type() . . . 1193
10.320.4 Member Function Documentation . . . 1193
10.320.4.1 GetTypeString() . . . 1193
10.320.4.2 GetTypeType() . . . 1193
10.320.4.3 operator TypeType() . . . 1193
10.320.5 Friends And Related Function Documentation . . . 1193
10.320.5.1 operator<< . . . 1194
10.321 gdcm::UI Struct Reference . . . 1194
10.321.1 Friends And Related Function Documentation . . . 1194
10.321.1.1 operator<< . . . 1194
10.321.2 Member Data Documentation . . . 1194
10.321.2.1 Internal . . . 1194
10.322 gdcm::UIDGenerator Class Reference . . . 1195
10.322.1 Detailed Description . . . 1195
10.322.2 Constructor & Destructor Documentation . . . 1195
10.322.2.1 UIDGenerator() . . . 1196
10.322.3 Member Function Documentation . . . 1196
10.322.3.1 Generate() . . . 1196
10.322.3.2 GenerateUUID() . . . 1196
10.322.3.3 GetGDCMUID() . . . 1196
10.322.3.4 GetRoot() . . . 1197
10.322.3.5 IsValid() . . . 1197
10.322.3.6 SetRoot() . . . 1197
10.323 gdcm::UIDs Class Reference . . . 1197
10.323.1 Detailed Description . . . 1213

10.323.2 Member Typedef Documentation . . . 1213
10.323.2.1 TransferSyntaxStringsType . . . 1213
10.323.3 Member Enumeration Documentation . . . 1214
10.323.3.1 TSName . . . 1214
10.323.3.2 TSType . . . 1223
10.323.4 Member Function Documentation . . . 1233
10.323.4.1 GetName() . . . 1233
10.323.4.2 GetNumberOfTransferSyntaxStrings() . . . 1233
10.323.4.3 GetString() . . . 1233
10.323.4.4 GetTransferSyntaxString() . . . 1233
10.323.4.5 GetTransferSyntaxStrings() . . . 1234
10.323.4.6 GetUIDName() . . . 1234
10.323.4.7 GetUIDString() . . . 1234
10.323.4.8 operator TSType() . . . 1234
10.323.4.9 SetFromUID() . . . 1234
10.324 gdcm::network::ULAction Class Reference . . . 1235
10.324.1 Detailed Description . . . 1236
10.324.2 Constructor & Destructor Documentation . . . 1236
10.324.2.1 ULAction() [1/2] . . . 1236
10.324.2.2 ∼ULAction() . . . 1237
10.324.2.3 ULAction() [2/2] . . . 1237
10.324.3 Member Function Documentation . . . 1237
10.324.3.1 operator=() . . . 1237
10.324.3.2 PerformAction() . . . 1237
10.325 gdcm::network::ULActionAA1 Class Reference . . . 1238
10.325.1 Member Function Documentation . . . 1238
10.325.1.1 PerformAction() . . . 1239
10.326 gdcm::network::ULActionAA2 Class Reference . . . 1239
10.326.1 Member Function Documentation . . . 1240
10.326.1.1 PerformAction() . . . 1240
10.327 gdcm::network::ULActionAA3 Class Reference . . . 1240
10.327.1 Member Function Documentation . . . 1241
10.327.1.1 PerformAction() . . . 1241
10.328 gdcm::network::ULActionAA4 Class Reference . . . 1242
10.328.1 Member Function Documentation . . . 1242
10.328.1.1 PerformAction() . . . 1243
10.329 gdcm::network::ULActionAA5 Class Reference . . . 1243
10.329.1 Member Function Documentation . . . 1244
10.329.1.1 PerformAction() . . . 1244

10.330 gdcm::network::ULActionAA6 Class Reference 1244
10.330.1 Member Function Documentation 1245
10.330.1.1 PerformAction() 1245
10.331 gdcm::network::ULActionAA7 Class Reference 1246
10.331.1 Member Function Documentation 1246
10.331.1.1 PerformAction() 1247
10.332 gdcm::network::ULActionAA8 Class Reference 1247
10.332.1 Member Function Documentation 1248
10.332.1.1 PerformAction() 1248
10.333 gdcm::network::ULActionAE1 Class Reference 1248
10.333.1 Member Function Documentation 1249
10.333.1.1 PerformAction() 1249
10.334 gdcm::network::ULActionAE2 Class Reference 1250
10.334.1 Member Function Documentation 1250
10.334.1.1 PerformAction() 1251
10.335 gdcm::network::ULActionAE3 Class Reference 1251
10.335.1 Member Function Documentation 1252
10.335.1.1 PerformAction() 1252
10.336 gdcm::network::ULActionAE4 Class Reference 1252
10.336.1 Member Function Documentation 1253
10.336.1.1 PerformAction() 1253
10.337 gdcm::network::ULActionAE5 Class Reference 1254
10.337.1 Member Function Documentation 1254
10.337.1.1 PerformAction() 1255
10.338 gdcm::network::ULActionAE6 Class Reference 1255
10.338.1 Member Function Documentation 1256
10.338.1.1 PerformAction() 1256
10.339 gdcm::network::ULActionAE7 Class Reference 1256
10.339.1 Member Function Documentation 1257
10.339.1.1 PerformAction() 1257
10.340 gdcm::network::ULActionAE8 Class Reference 1258
10.340.1 Member Function Documentation 1258
10.340.1.1 PerformAction() 1259
10.341 gdcm::network::ULActionAR1 Class Reference 1259
10.341.1 Member Function Documentation 1260
10.341.1.1 PerformAction() 1260
10.342 gdcm::network::ULActionAR10 Class Reference 1260
10.342.1 Member Function Documentation 1261
10.342.1.1 PerformAction() 1261

10.343 gdcm::network::ULActionAR2 Class Reference 1262
10.343.1 Member Function Documentation 1262
10.343.1.1 PerformAction() 1263
10.344 gdcm::network::ULActionAR3 Class Reference 1263
10.344.1 Member Function Documentation 1264
10.344.1.1 PerformAction() 1264
10.345 gdcm::network::ULActionAR4 Class Reference 1264
10.345.1 Member Function Documentation 1265
10.345.1.1 PerformAction() 1265
10.346 gdcm::network::ULActionAR5 Class Reference 1266
10.346.1 Member Function Documentation 1266
10.346.1.1 PerformAction() 1267
10.347 gdcm::network::ULActionAR6 Class Reference 1267
10.347.1 Member Function Documentation 1268
10.347.1.1 PerformAction() 1268
10.348 gdcm::network::ULActionAR7 Class Reference 1268
10.348.1 Member Function Documentation 1269
10.348.1.1 PerformAction() 1269
10.349 gdcm::network::ULActionAR8 Class Reference 1270
10.349.1 Member Function Documentation 1270
10.349.1.1 PerformAction() 1271
10.350 gdcm::network::ULActionAR9 Class Reference 1271
10.350.1 Member Function Documentation 1272
10.350.1.1 PerformAction() 1272
10.351 gdcm::network::ULActionDT1 Class Reference 1272
10.351.1 Member Function Documentation 1273
10.351.1.1 PerformAction() 1273
10.352 gdcm::network::ULActionDT2 Class Reference 1274
10.352.1 Member Function Documentation 1274
10.352.1.1 PerformAction() 1275
10.353 gdcm::network::ULBasicCallback Class Reference 1275
10.353.1 Detailed Description 1276
10.353.2 Constructor & Destructor Documentation 1276
10.353.2.1 ULBasicCallback() 1276
10.353.2.2 ~ULBasicCallback() 1276
10.353.3 Member Function Documentation 1276
10.353.3.1 GetDataSets() 1276
10.353.3.2 GetResponses() 1277
10.353.3.3 HandleDataSet() 1277

10.353.3.4 HandleResponse() . . . 1277
10.354 gdcm::network::ULConnection Class Reference . . . 1277
10.354.1 Detailed Description . . . 1278
10.354.2 Constructor & Destructor Documentation . . . 1279
10.354.2.1 ULConnection() [1/2] . . . 1279
10.354.2.2 ~ULConnection() . . . 1279
10.354.2.3 ULConnection() [2/2] . . . 1279
10.354.3 Member Function Documentation . . . 1279
10.354.3.1 AddAcceptedPresentationContext() . . . 1279
10.354.3.2 FindContext() . . . 1279
10.354.3.3 GetAcceptedPresentationContexts() [1/2] . . . 1280
10.354.3.4 GetAcceptedPresentationContexts() [2/2] . . . 1280
10.354.3.5 GetConnectionInfo() . . . 1280
10.354.3.6 GetMaxPDUSize() . . . 1280
10.354.3.7 GetPresentationContextACByID() . . . 1280
10.354.3.8 GetPresentationContextIDFromPresentationContext() . . . 1280
10.354.3.9 GetPresentationContextRQByID() . . . 1281
10.354.3.10 GetPresentationContexts() . . . 1281
10.354.3.11 GetProtocol() . . . 1281
10.354.3.12 GetState() . . . 1281
10.354.3.13 GetTimer() . . . 1281
10.354.3.14 InitializeConnection() . . . 1281
10.354.3.15 InitializeIncomingConnection() . . . 1282
10.354.3.16 operator=() . . . 1282
10.354.3.17 SetMaxPDUSize() . . . 1282
10.354.3.18 SetPresentationContexts() [1/2] . . . 1282
10.354.3.19 SetPresentationContexts() [2/2] . . . 1282
10.354.3.20 SetState() . . . 1282
10.354.3.21 StopProtocol() . . . 1283
10.354.4 Friends And Related Function Documentation . . . 1283
10.354.4.1 ULActionAE6 . . . 1283
10.354.4.2 ULConnectionManager . . . 1283
10.355 gdcm::network::ULConnectionCallback Class Reference . . . 1283
10.355.1 Detailed Description . . . 1284
10.355.2 Constructor & Destructor Documentation . . . 1284
10.355.2.1 ULConnectionCallback() . . . 1284
10.355.2.2 ~ULConnectionCallback() . . . 1284
10.355.3 Member Function Documentation . . . 1285
10.355.3.1 DataSetHandled() . . . 1285

10.355.3.2 DataSetHandles() . . . 1285
10.355.3.3 HandleDataSet() . . . 1285
10.355.3.4 HandleResponse() . . . 1285
10.355.3.5 ResetHandledDataSet() . . . 1285
10.355.3.6 SetImplicitFlag() . . . 1285
10.355.4 Member Data Documentation . . . 1286
10.355.4.1 mImplicit . . . 1286
10.356 gdcm::network::ULConnectionInfo Class Reference . . . 1286
10.356.1 Detailed Description . . . 1286
10.356.2 Constructor & Destructor Documentation . . . 1286
10.356.2.1 ULConnectionInfo() . . . 1287
10.356.3 Member Function Documentation . . . 1287
10.356.3.1 GetCalledAETitle() . . . 1287
10.356.3.2 GetCalledComputerName() . . . 1287
10.356.3.3 GetCalledIPAddress() . . . 1287
10.356.3.4 GetCalledIPPort() . . . 1287
10.356.3.5 GetCallingAETitle() . . . 1287
10.356.3.6 GetMaxPDULength() . . . 1287
10.356.3.7 Initialize() . . . 1288
10.356.3.8 SetMaxPDULength() . . . 1288
10.357 gdcm::network::ULConnectionManager Class Reference . . . 1288
10.357.1 Detailed Description . . . 1290
10.357.2 Constructor & Destructor Documentation . . . 1290
10.357.2.1 ULConnectionManager() [1/2] . . . 1290
10.357.2.2 ULConnectionManager() [2/2] . . . 1290
10.357.2.3 ~ULConnectionManager() . . . 1290
10.357.3 Member Function Documentation . . . 1291
10.357.3.1 BreakConnection() . . . 1291
10.357.3.2 BreakConnectionNow() . . . 1291
10.357.3.3 EstablishConnection() . . . 1291
10.357.3.4 EstablishConnectionMove() . . . 1291
10.357.3.5 RunEventLoop() . . . 1292
10.357.3.6 RunMoveEventLoop() . . . 1292
10.357.3.7 SendEcho() . . . 1292
10.357.3.8 SendFind() [1/2] . . . 1292
10.357.3.9 SendFind() [2/2] . . . 1292
10.357.3.10 SendMove() [1/2] . . . 1292
10.357.3.11 SendMove() [2/2] . . . 1293
10.357.3.12 SendNAction() [1/2] . . . 1293

10.357.3.13 SendNAction() [2/2] . . . 1293
10.357.3.14 SendNCreate() [1/2] . . . 1293
10.357.3.15 SendNCreate() [2/2] . . . 1293
10.357.3.16 SendNDelete() [1/2] . . . 1293
10.357.3.17 SendNDelete() [2/2] . . . 1294
10.357.3.18 SendNEventReport() [1/2] . . . 1294
10.357.3.19 SendNEventReport() [2/2] . . . 1294
10.357.3.20 SendNGet() [1/2] . . . 1294
10.357.3.21 SendNGet() [2/2] . . . 1294
10.357.3.22 SendNSet() [1/2] . . . 1294
10.357.3.23 SendNSet() [2/2] . . . 1295
10.357.3.24 SendStore() [1/2] . . . 1295
10.357.3.25 SendStore() [2/2] . . . 1295
10.357.4 Member Data Documentation . . . 1295
10.357.4.1 mConnection . . . 1295
10.357.4.2 mSecondaryConnection . . . 1295
10.357.4.3 mTransitions . . . 1296
10.358 gdcm::network::ULEvent Class Reference . . . 1296
10.358.1 Detailed Description . . . 1296
10.358.2 Constructor & Destructor Documentation . . . 1296
10.358.2.1 ULEvent() [1/2] . . . 1297
10.358.2.2 ULEvent() [2/2] . . . 1297
10.358.2.3 ∼ULEvent() . . . 1297
10.358.3 Member Function Documentation . . . 1297
10.358.3.1 GetDataSetPos() . . . 1297
10.358.3.2 GetEvent() . . . 1297
10.358.3.3 GetIStream() . . . 1297
10.358.3.4 GetPDUs() . . . 1298
10.358.3.5 SetEvent() . . . 1298
10.358.3.6 SetPDU() . . . 1298
10.359 gdcm::network::ULTransitionTable Class Reference . . . 1298
10.359.1 Detailed Description . . . 1298
10.359.2 Constructor & Destructor Documentation . . . 1299
10.359.2.1 ULTransitionTable() . . . 1299
10.359.3 Member Function Documentation . . . 1299
10.359.3.1 HandleEvent() . . . 1299
10.359.3.2 PrintTable() . . . 1299
10.360 gdcm::network::ULWritingCallback Class Reference . . . 1300
10.360.1 Constructor & Destructor Documentation . . . 1301

10.360.1.1 ULWritingCallback() . . . 1301
10.360.1.2 ~ULWritingCallback() . . . 1301
10.360.2 Member Function Documentation . . . 1301
10.360.2.1 HandleDataSet() . . . 1301
10.360.2.2 HandleResponse() . . . 1301
10.360.2.3 SetDirectory() . . . 1302
10.361 gdcm::UNExplicitDataElement Class Reference . . . 1302
10.361.1 Detailed Description . . . 1303
10.361.2 Member Function Documentation . . . 1303
10.361.2.1 GetLength() . . . 1304
10.361.2.2 Read() . . . 1304
10.361.2.3 ReadPreValue() . . . 1304
10.361.2.4 ReadValue() . . . 1304
10.361.2.5 ReadWithLength() . . . 1304
10.362 gdcm::UNExplicitImplicitDataElement Class Reference . . . 1305
10.362.1 Detailed Description . . . 1306
10.362.2 Member Function Documentation . . . 1306
10.362.2.1 GetLength() . . . 1306
10.362.2.2 Read() . . . 1306
10.362.2.3 ReadPreValue() . . . 1306
10.362.2.4 ReadValue() . . . 1307
10.363 gdcm::Unpacker12Bits Class Reference . . . 1307
10.363.1 Detailed Description . . . 1307
10.363.2 Member Function Documentation . . . 1307
10.363.2.1 Pack() . . . 1308
10.363.2.2 Unpack() . . . 1308
10.364 gdcm::Usage Class Reference . . . 1308
10.364.1 Detailed Description . . . 1309
10.364.2 Member Enumeration Documentation . . . 1309
10.364.2.1 UsageType . . . 1309
10.364.3 Constructor & Destructor Documentation . . . 1310
10.364.3.1 Usage() . . . 1310
10.364.4 Member Function Documentation . . . 1310
10.364.4.1 GetUsageString() . . . 1310
10.364.4.2 GetUsageType() . . . 1310
10.364.4.3 operator UsageType() . . . 1310
10.364.5 Friends And Related Function Documentation . . . 1310
10.364.5.1 operator<< . . . 1310
10.365 gdcm::UserEvent Class Reference . . . 1311

10.366 gdcm::network::UserInformation Class Reference . . . 1312
10.366.1 Detailed Description . . . 1312
10.366.2 Constructor & Destructor Documentation . . . 1312
10.366.2.1 UserInformation() [1/2] . . . 1312
10.366.2.2 ~UserInformation() . . . 1312
10.366.2.3 UserInformation() [2/2] . . . 1313
10.366.3 Member Function Documentation . . . 1313
10.366.3.1 AddRoleSelectionSub() . . . 1313
10.366.3.2 AddSOPClassExtendedNegociationSub() . . . 1313
10.366.3.3 GetMaximumLengthSub() [1/2] . . . 1313
10.366.3.4 GetMaximumLengthSub() [2/2] . . . 1313
10.366.3.5 operator=() . . . 1313
10.366.3.6 Print() . . . 1314
10.366.3.7 Read() . . . 1314
10.366.3.8 Size() . . . 1314
10.366.3.9 Write() . . . 1314
10.367 gdcm::UUIDGenerator Class Reference . . . 1314
10.367.1 Detailed Description . . . 1315
10.367.2 Member Function Documentation . . . 1315
10.367.2.1 Generate() . . . 1315
10.367.2.2 IsValid() . . . 1315
10.368 gdcm::Validate Class Reference . . . 1315
10.368.1 Detailed Description . . . 1316
10.368.2 Constructor & Destructor Documentation . . . 1316
10.368.2.1 Validate() . . . 1316
10.368.2.2 ~Validate() . . . 1316
10.368.3 Member Function Documentation . . . 1316
10.368.3.1 GetValidatedFile() . . . 1316
10.368.3.2 SetFile() . . . 1317
10.368.3.3 Validation() . . . 1317
10.368.4 Member Data Documentation . . . 1317
10.368.4.1 F . . . 1317
10.368.4.2 V . . . 1317
10.369 gdcm::Value Class Reference . . . 1318
10.369.1 Detailed Description . . . 1319
10.369.2 Constructor & Destructor Documentation . . . 1319
10.369.2.1 Value() . . . 1319
10.369.2.2 ~Value() . . . 1319
10.369.3 Member Function Documentation . . . 1319

10.369.3.1 Clear() . . . 1319
10.369.3.2 GetLength() . . . 1320
10.369.3.3 operator==() . . . 1320
10.369.3.4 SetLength() . . . 1320
10.369.3.5 SetLengthOnly() . . . 1320
10.369.4 Friends And Related Function Documentation . . . 1320
10.369.4.1 DataElement . . . 1320
10.370 gdcm::ValueIO< TDE, TSwap, TType > Class Template Reference . . . 1321
10.370.1 Detailed Description . . . 1321
10.370.2 Member Function Documentation . . . 1321
10.370.2.1 Read() . . . 1321
10.370.2.2 Write() . . . 1321
10.371 gdcm::MrProtocol::Vector3 Struct Reference . . . 1322
10.371.1 Member Data Documentation . . . 1322
10.371.1.1 dCor . . . 1322
10.371.1.2 dSag . . . 1322
10.371.1.3 dTra . . . 1322
10.372 gdcm::Version Class Reference . . . 1322
10.372.1 Detailed Description . . . 1323
10.372.2 Constructor & Destructor Documentation . . . 1323
10.372.2.1 Version() . . . 1323
10.372.2.2 ~Version() . . . 1323
10.372.3 Member Function Documentation . . . 1323
10.372.3.1 GetBuildVersion() . . . 1324
10.372.3.2 GetMajorVersion() . . . 1324
10.372.3.3 GetMinorVersion() . . . 1324
10.372.3.4 GetVersion() . . . 1324
10.372.3.5 Print() . . . 1324
10.372.4 Friends And Related Function Documentation . . . 1324
10.372.4.1 operator<< . . . 1324
10.373 gdcm::VL Class Reference . . . 1325
10.373.1 Detailed Description . . . 1326
10.373.2 Member Typedef Documentation . . . 1326
10.373.2.1 Type . . . 1326
10.373.3 Constructor & Destructor Documentation . . . 1326
10.373.3.1 VL() . . . 1326
10.373.4 Member Function Documentation . . . 1326
10.373.4.1 GetLength() . . . 1326
10.373.4.2 GetVL16Max() . . . 1327

10.373.4.3 GetVL32Max() . . . 1327
10.373.4.4 IsOdd() . . . 1327
10.373.4.5 IsUndefined() . . . 1327
10.373.4.6 operator uint32_t() . . . 1327
10.373.4.7 operator++() [1/2] . . . 1327
10.373.4.8 operator++() [2/2] . . . 1327
10.373.4.9 operator+=() . . . 1328
10.373.4.10 Read() . . . 1328
10.373.4.11 Read16() . . . 1328
10.373.4.12 SetToUndefined() . . . 1328
10.373.4.13 Write() . . . 1328
10.373.4.14 Write16() . . . 1328
10.373.5 Friends And Related Function Documentation . . . 1329
10.373.5.1 operator<< . . . 1329
10.374 gdcm::VM Class Reference . . . 1329
10.374.1 Detailed Description . . . 1330
10.374.2 Member Enumeration Documentation . . . 1331
10.374.2.1 VMType . . . 1331
10.374.3 Constructor & Destructor Documentation . . . 1332
10.374.3.1 VM() . . . 1332
10.374.4 Member Function Documentation . . . 1332
10.374.4.1 Compatible() . . . 1332
10.374.4.2 GetIndex() . . . 1332
10.374.4.3 GetLength() . . . 1332
10.374.4.4 GetNumberOfElementsFromArray() . . . 1332
10.374.4.5 GetVMString() . . . 1333
10.374.4.6 GetVMType() . . . 1333
10.374.4.7 GetVMTypeFromLength() . . . 1333
10.374.4.8 IsValid() . . . 1333
10.374.4.9 operator VMType() . . . 1333
10.374.5 Friends And Related Function Documentation . . . 1333
10.374.5.1 operator<< . . . 1333
10.375 gdcm::VMToLength< T > Struct Template Reference . . . 1334
10.376 gdcm::VR Class Reference . . . 1334
10.376.1 Detailed Description . . . 1336
10.376.2 Member Enumeration Documentation . . . 1336
10.376.2.1 VRType . . . 1336
10.376.3 Constructor & Destructor Documentation . . . 1337
10.376.3.1 VR() . . . 1337

10.376.4 Member Function Documentation . . . 1337
10.376.4.1 CanDisplay() . . . 1338
10.376.4.2 Compatible() . . . 1338
10.376.4.3 GetLength() [1/2] . . . 1338
10.376.4.4 GetLength() [2/2] . . . 1338
10.376.4.5 GetSize() . . . 1338
10.376.4.6 GetSizeof() . . . 1339
10.376.4.7 GetVRString() . . . 1339
10.376.4.8 GetVRStringFromFile() . . . 1339
10.376.4.9 GetVRType() . . . 1339
10.376.4.10 GetVRTypeFromFile() . . . 1339
10.376.4.11 IsASCII() . . . 1339
10.376.4.12 IsASCII2() . . . 1340
10.376.4.13 IsBinary() . . . 1340
10.376.4.14 IsBinary2() . . . 1340
10.376.4.15 IsDual() . . . 1340
10.376.4.16 IsSwap() . . . 1340
10.376.4.17 IsValid() [1/2] . . . 1340
10.376.4.18 IsValid() [2/2] . . . 1341
10.376.4.19 IsVRFile() . . . 1341
10.376.4.20 operator VRType() . . . 1341
10.376.4.21 Read() . . . 1341
10.376.4.22 Write() . . . 1341
10.376.5 Friends And Related Function Documentation . . . 1341
10.376.5.1 operator<< . . . 1342
10.377 gdcm::VR16ExplicitDataElement Class Reference . . . 1342
10.377.1 Detailed Description . . . 1343
10.377.2 Member Function Documentation . . . 1343
10.377.2.1 GetLength() . . . 1344
10.377.2.2 Read() . . . 1344
10.377.2.3 ReadPreValue() . . . 1344
10.377.2.4 ReadValue() . . . 1344
10.377.2.5 ReadWithLength() . . . 1344
10.378 gdcm::VRToEncoding< T > Struct Template Reference . . . 1345
10.379 gdcm::VRToType< T > Struct Template Reference . . . 1345
10.379.1 Detailed Description . . . 1345
10.380 gdcm::VRVLSize< T > Class Template Reference . . . 1345
10.381 gdcm::VRVLSize< 0 > Class Reference . . . 1346
10.381.1 Member Function Documentation . . . 1346

10.381.1.1 Read() . . . 1346
10.381.1.2 Write() . . . 1346
10.382 gdcm::VRVLSize< 1 > Class Reference . . . 1346
10.382.1 Member Function Documentation . . . 1346
10.382.1.1 Read() . . . 1347
10.382.1.2 Write() . . . 1347
10.383 vtkGDCMImageReader Class Reference . . . 1347
10.383.1 Detailed Description . . . 1350
10.383.2 Constructor & Destructor Documentation . . . 1350
10.383.2.1 vtkGDCMImageReader() . . . 1350
10.383.2.2 ~vtkGDCMImageReader() . . . 1350
10.383.3 Member Function Documentation . . . 1350
10.383.3.1 CanReadFile() . . . 1350
10.383.3.2 ExecuteData() . . . 1351
10.383.3.3 ExecuteInformation() . . . 1351
10.383.3.4 FillMedicalImageInformation() . . . 1351
10.383.3.5 GetDescriptiveName() . . . 1351
10.383.3.6 GetFileExtensions() . . . 1351
10.383.3.7 GetIconImage() . . . 1351
10.383.3.8 GetOverlay() . . . 1351
10.383.3.9 LoadSingleFile() . . . 1352
10.383.3.10 New() . . . 1352
10.383.3.11 PrintSelf() . . . 1352
10.383.3.12 RequestDataCompat() . . . 1352
10.383.3.13 RequestInformationCompat() . . . 1352
10.383.3.14 SetCurve() . . . 1353
10.383.3.15 SetFileNames() . . . 1353
10.383.3.16 SetFilePattern() . . . 1353
10.383.3.17 SetFilePrefix() . . . 1353
10.383.3.18 SetMedicalImageProperties() . . . 1353
10.383.3.19 vtkBooleanMacro() [1/5] . . . 1353
10.383.3.20 vtkBooleanMacro() [2/5] . . . 1354
10.383.3.21 vtkBooleanMacro() [3/5] . . . 1354
10.383.3.22 vtkBooleanMacro() [4/5] . . . 1354
10.383.3.23 vtkBooleanMacro() [5/5] . . . 1354
10.383.3.24 vtkGetMacro() [1/11] . . . 1354
10.383.3.25 vtkGetMacro() [2/11] . . . 1354
10.383.3.26 vtkGetMacro() [3/11] . . . 1355
10.383.3.27 vtkGetMacro() [4/11] . . . 1355

10.383.3.28 vtkGetMacro() [5/11] . . . 1355
10.383.3.29 vtkGetMacro() [6/11] . . . 1355
10.383.3.30 vtkGetMacro() [7/11] . . . 1355
10.383.3.31 vtkGetMacro() [8/11] . . . 1355
10.383.3.32 vtkGetMacro() [9/11] . . . 1356
10.383.3.33 vtkGetMacro() [10/11] . . . 1356
10.383.3.34 vtkGetMacro() [11/11] . . . 1356
10.383.3.35 vtkGetObjectMacro() [1/4] . . . 1356
10.383.3.36 vtkGetObjectMacro() [2/4] . . . 1356
10.383.3.37 vtkGetObjectMacro() [3/4] . . . 1356
10.383.3.38 vtkGetObjectMacro() [4/4] . . . 1357
10.383.3.39 vtkGetStringMacro() [1/2] . . . 1357
10.383.3.40 vtkGetStringMacro() [2/2] . . . 1357
10.383.3.41 vtkGetVector3Macro() . . . 1357
10.383.3.42 vtkGetVector6Macro() . . . 1357
10.383.3.43 vtkSetMacro() [1/4] . . . 1357
10.383.3.44 vtkSetMacro() [2/4] . . . 1358
10.383.3.45 vtkSetMacro() [3/4] . . . 1358
10.383.3.46 vtkSetMacro() [4/4] . . . 1358
10.383.3.47 vtkSetVector6Macro() . . . 1358
10.383.3.48 vtkTypeMacro() . . . 1358
10.383.4 Member Data Documentation . . . 1358
10.383.4.1 ApplyInverseVideo . . . 1359
10.383.4.2 ApplyLookupTable . . . 1359
10.383.4.3 ApplyPlanarConfiguration . . . 1359
10.383.4.4 ApplyShiftScale . . . 1359
10.383.4.5 ApplyYBRToRGB . . . 1359
10.383.4.6 Curve . . . 1359
10.383.4.7 DirectionCosines . . . 1359
10.383.4.8 FileNames . . . 1360
10.383.4.9 ForceRescale . . . 1360
10.383.4.10 IconDataScalarType . . . 1360
10.383.4.11 IconImageDataExtent . . . 1360
10.383.4.12 IconNumberOfScalarComponents . . . 1360
10.383.4.13 ImageFormat . . . 1360
10.383.4.14 ImageOrientationPatient . . . 1360
10.383.4.15 ImagePositionPatient . . . 1361
10.383.4.16 LoadIconImage . . . 1361
10.383.4.17 LoadOverlays . . . 1361

10.383.4.18 LossyFlag . . . 1361
10.383.4.19 MedicalImageProperties . . . 1361
10.383.4.20 NumberOfIconImages . . . 1361
10.383.4.21 NumberOfOverlays . . . 1361
10.383.4.22 PlanarConfiguration . . . 1362
10.383.4.23 Scale . . . 1362
10.383.4.24 Shift . . . 1362
10.384 vtkGDCMImageReader2 Class Reference . . . 1362
10.384.1 Detailed Description . . . 1365
10.384.2 Constructor & Destructor Documentation . . . 1365
10.384.2.1 vtkGDCMImageReader2() . . . 1365
10.384.2.2 ~vtkGDCMImageReader2() . . . 1365
10.384.3 Member Function Documentation . . . 1365
10.384.3.1 CanReadFile() . . . 1365
10.384.3.2 FillMedicalImageInformation() . . . 1365
10.384.3.3 GetDescriptiveName() . . . 1365
10.384.3.4 GetFileExtensions() . . . 1366
10.384.3.5 GetIconImage() . . . 1366
10.384.3.6 GetIconImagePort() . . . 1366
10.384.3.7 GetOverlay() . . . 1366
10.384.3.8 GetOverlayPort() . . . 1366
10.384.3.9 LoadSingleFile() . . . 1366
10.384.3.10 New() . . . 1367
10.384.3.11 PrintSelf() . . . 1367
10.384.3.12 ProcessRequest() . . . 1367
10.384.3.13 RequestData() . . . 1367
10.384.3.14 RequestDataCompat() . . . 1367
10.384.3.15 RequestInformation() . . . 1368
10.384.3.16 RequestInformationCompat() . . . 1368
10.384.3.17 SetCurve() . . . 1368
10.384.3.18 SetFilePattern() . . . 1368
10.384.3.19 SetFilePrefix() . . . 1368
10.384.3.20 SetMedicalImageProperties() . . . 1368
10.384.3.21 vtkBooleanMacro() [1/5] . . . 1369
10.384.3.22 vtkBooleanMacro() [2/5] . . . 1369
10.384.3.23 vtkBooleanMacro() [3/5] . . . 1369
10.384.3.24 vtkBooleanMacro() [4/5] . . . 1369
10.384.3.25 vtkBooleanMacro() [5/5] . . . 1369
10.384.3.26 vtkGetMacro() [1/11] . . . 1369

10.384.3.27 vtkGetMacro() [2/11] . . . 1370
10.384.3.28 vtkGetMacro() [3/11] . . . 1370
10.384.3.29 vtkGetMacro() [4/11] . . . 1370
10.384.3.30 vtkGetMacro() [5/11] . . . 1370
10.384.3.31 vtkGetMacro() [6/11] . . . 1370
10.384.3.32 vtkGetMacro() [7/11] . . . 1370
10.384.3.33 vtkGetMacro() [8/11] . . . 1371
10.384.3.34 vtkGetMacro() [9/11] . . . 1371
10.384.3.35 vtkGetMacro() [10/11] . . . 1371
10.384.3.36 vtkGetMacro() [11/11] . . . 1371
10.384.3.37 vtkGetObjectMacro() [1/2] . . . 1371
10.384.3.38 vtkGetObjectMacro() [2/2] . . . 1371
10.384.3.39 vtkGetStringMacro() [1/2] . . . 1372
10.384.3.40 vtkGetStringMacro() [2/2] . . . 1372
10.384.3.41 vtkGetVector3Macro() . . . 1372
10.384.3.42 vtkGetVector6Macro() . . . 1372
10.384.3.43 vtkSetMacro() [1/4] . . . 1372
10.384.3.44 vtkSetMacro() [2/4] . . . 1372
10.384.3.45 vtkSetMacro() [3/4] . . . 1373
10.384.3.46 vtkSetMacro() [4/4] . . . 1373
10.384.3.47 vtkSetVector6Macro() . . . 1373
10.384.3.48 vtkTypeMacro() . . . 1373
10.384.4 Member Data Documentation . . . 1373
10.384.4.1 ApplyInverseVideo . . . 1373
10.384.4.2 ApplyLookupTable . . . 1373
10.384.4.3 ApplyPlanarConfiguration . . . 1374
10.384.4.4 ApplyShiftScale . . . 1374
10.384.4.5 ApplyYBRToRGB . . . 1374
10.384.4.6 Curve . . . 1374
10.384.4.7 DirectionCosines . . . 1374
10.384.4.8 ForceRescale . . . 1374
10.384.4.9 IconDataScalarType . . . 1374
10.384.4.10 IconImageDataExtent . . . 1375
10.384.4.11 IconNumberOfScalarComponents . . . 1375
10.384.4.12 ImageFormat . . . 1375
10.384.4.13 ImageOrientationPatient . . . 1375
10.384.4.14 ImagePositionPatient . . . 1375
10.384.4.15 LoadIconImage . . . 1375
10.384.4.16 LoadOverlays . . . 1375

10.384.4.17 LossyFlag . . . 1376
10.384.4.18 NumberOfIconImages . . . 1376
10.384.4.19 NumberOfOverlays . . . 1376
10.384.4.20 PlanarConfiguration . . . 1376
10.384.4.21 Scale . . . 1376
10.384.4.22 Shift . . . 1376
10.385 vtkGDCMImageWriter Class Reference . . . 1377
10.385.1 Detailed Description . . . 1379
10.385.2 Member Enumeration Documentation . . . 1379
10.385.2.1 CompressionTypes . . . 1379
10.385.3 Constructor & Destructor Documentation . . . 1379
10.385.3.1 vtkGDCMImageWriter() . . . 1379
10.385.3.2 ~vtkGDCMImageWriter() . . . 1379
10.385.4 Member Function Documentation . . . 1379
10.385.4.1 GetDescriptiveName() . . . 1380
10.385.4.2 GetFileExtensions() . . . 1380
10.385.4.3 GetFileName() . . . 1380
10.385.4.4 New() . . . 1380
10.385.4.5 PrintSelf() . . . 1380
10.385.4.6 SetDirectionCosines() . . . 1380
10.385.4.7 SetDirectionCosinesFromImageOrientationPatient() . . . 1381
10.385.4.8 SetFileNames() . . . 1381
10.385.4.9 SetMedicalImageProperties() . . . 1381
10.385.4.10 vtkBooleanMacro() [1/2] . . . 1381
10.385.4.11 vtkBooleanMacro() [2/2] . . . 1381
10.385.4.12 vtkGetMacro() [1/7] . . . 1382
10.385.4.13 vtkGetMacro() [2/7] . . . 1382
10.385.4.14 vtkGetMacro() [3/7] . . . 1382
10.385.4.15 vtkGetMacro() [4/7] . . . 1382
10.385.4.16 vtkGetMacro() [5/7] . . . 1382
10.385.4.17 vtkGetMacro() [6/7] . . . 1382
10.385.4.18 vtkGetMacro() [7/7] . . . 1383
10.385.4.19 vtkGetObjectMacro() [1/3] . . . 1383
10.385.4.20 vtkGetObjectMacro() [2/3] . . . 1383
10.385.4.21 vtkGetObjectMacro() [3/3] . . . 1383
10.385.4.22 vtkGetStringMacro() [1/2] . . . 1383
10.385.4.23 vtkGetStringMacro() [2/2] . . . 1383
10.385.4.24 vtkSetMacro() [1/7] . . . 1384
10.385.4.25 vtkSetMacro() [2/7] . . . 1384

10.385.4.26 vtkSetMacro() [3/7] . . . 1384
10.385.4.27 vtkSetMacro() [4/7] . . . 1384
10.385.4.28 vtkSetMacro() [5/7] . . . 1384
10.385.4.29 vtkSetMacro() [6/7] . . . 1384
10.385.4.30 vtkSetMacro() [7/7] . . . 1385
10.385.4.31 vtkSetStringMacro() [1/2] . . . 1385
10.385.4.32 vtkSetStringMacro() [2/2] . . . 1385
10.385.4.33 vtkTypeMacro() . . . 1385
10.385.4.34 Write() . . . 1385
10.385.4.35 WriteGDCMData() . . . 1386
10.385.4.36 WriteSlice() . . . 1386
10.386 vtkGDCMMedicalImageProperties Class Reference . . . 1386
10.386.1 Constructor & Destructor Documentation . . . 1387
10.386.1.1 vtkGDCMMedicalImageProperties() . . . 1387
10.386.1.2 ∼vtkGDCMMedicalImageProperties() . . . 1387
10.386.2 Member Function Documentation . . . 1387
10.386.2.1 Clear() . . . 1388
10.386.2.2 GetFile() . . . 1388
10.386.2.3 New() . . . 1388
10.386.2.4 PrintSelf() . . . 1388
10.386.2.5 PushBackFile() . . . 1388
10.386.2.6 vtkTypeMacro() . . . 1388
10.386.3 Friends And Related Function Documentation . . . 1388
10.386.3.1 vtkGDCMImageReader . . . 1389
10.386.3.2 vtkGDCMImageReader2 . . . 1389
10.386.3.3 vtkGDCMImageWriter . . . 1389
10.387 vtkGDCMPolyDataReader Class Reference . . . 1389
10.387.1 Detailed Description . . . 1391
10.387.2 Constructor & Destructor Documentation . . . 1391
10.387.2.1 vtkGDCMPolyDataReader() . . . 1391
10.387.2.2 ∼vtkGDCMPolyDataReader() . . . 1391
10.387.3 Member Function Documentation . . . 1391
10.387.3.1 FillMedicalImageInformation() . . . 1391
10.387.3.2 New() . . . 1391
10.387.3.3 PrintSelf() . . . 1392
10.387.3.4 RequestData() . . . 1392
10.387.3.5 RequestData_HemodynamicWaveformStorage() . . . 1392
10.387.3.6 RequestData_RTStructureSetStorage() . . . 1392
10.387.3.7 RequestInformation() . . . 1392

10.387.3.8 RequestInformation_HemodynamicWaveformStorage() . . . 1392
10.387.3.9 RequestInformation_RTStructureSetStorage() . . . 1393
10.387.3.10 vtkGetObjectMacro() [1/2] . . . 1393
10.387.3.11 vtkGetObjectMacro() [2/2] . . . 1393
10.387.3.12 vtkGetStringMacro() . . . 1393
10.387.3.13 vtkSetStringMacro() . . . 1393
10.387.3.14 vtkTypeMacro() . . . 1393
10.387.4 Member Data Documentation . . . 1394
10.387.4.1 FileName . . . 1394
10.387.4.2 MedicalImageProperties . . . 1394
10.387.4.3 RTStructSetProperties . . . 1394
10.388 vtkGDCMPolyDataWriter Class Reference . . . 1394
10.388.1 Detailed Description . . . 1396
10.388.2 Constructor & Destructor Documentation . . . 1396
10.388.2.1 vtkGDCMPolyDataWriter() . . . 1396
10.388.2.2 ~vtkGDCMPolyDataWriter() . . . 1396
10.388.3 Member Function Documentation . . . 1396
10.388.3.1 InitializeRTStructSet() . . . 1396
10.388.3.2 New() . . . 1397
10.388.3.3 PrintSelf() . . . 1397
10.388.3.4 SetMedicalImageProperties() . . . 1397
10.388.3.5 SetNumberOfInputPorts() . . . 1397
10.388.3.6 SetRTStructSetProperties() . . . 1398
10.388.3.7 vtkTypeMacro() . . . 1398
10.388.3.8 WriteData() . . . 1398
10.388.3.9 WriteRTSTRUCTData() . . . 1398
10.388.3.10 WriteRTSTRUCTInfo() . . . 1398
10.388.4 Member Data Documentation . . . 1398
10.388.4.1 MedicalImageProperties . . . 1399
10.388.4.2 RTStructSetProperties . . . 1399
10.389 vtkGDCMTesting Class Reference . . . 1399
10.389.1 Detailed Description . . . 1400
10.389.2 Member Typedef Documentation . . . 1400
10.389.2.1 MD5MetaImagesType . . . 1400
10.389.3 Constructor & Destructor Documentation . . . 1400
10.389.3.1 vtkGDCMTesting() . . . 1401
10.389.3.2 ~vtkGDCMTesting() . . . 1401
10.389.4 Member Function Documentation . . . 1401
10.389.4.1 GetGDCMDataRoot() . . . 1401

10.389.4.2 GetMD5MetaImage() . . . 1401
10.389.4.3 GetMHDMD5FromFile() . . . 1401
10.389.4.4 GetNumberOfMD5MetaImages() . . . 1402
10.389.4.5 GetRAWMD5FromFile() . . . 1402
10.389.4.6 GetVTKDataRoot() . . . 1402
10.389.4.7 New() . . . 1402
10.389.4.8 PrintSelf() . . . 1402
10.389.4.9 vtkTypeMacro() . . . 1403
10.390 vtkGDCMThreadedImageReader Class Reference . . . 1403
10.390.1 Constructor & Destructor Documentation . . . 1404
10.390.1.1 vtkGDCMThreadedImageReader() . . . 1405
10.390.1.2 ∼vtkGDCMThreadedImageReader() . . . 1405
10.390.2 Member Function Documentation . . . 1405
10.390.2.1 ExecuteData() . . . 1405
10.390.2.2 ExecuteInformation() . . . 1405
10.390.2.3 New() . . . 1405
10.390.2.4 PrintSelf() . . . 1405
10.390.2.5 ReadFiles() . . . 1406
10.390.2.6 RequestDataCompat() . . . 1406
10.390.2.7 vtkBooleanMacro() . . . 1406
10.390.2.8 vtkGetMacro() . . . 1406
10.390.2.9 vtkSetMacro() [1/3] . . . 1406
10.390.2.10 vtkSetMacro() [2/3] . . . 1406
10.390.2.11 vtkSetMacro() [3/3] . . . 1407
10.390.2.12 vtkTypeMacro() . . . 1407
10.391 vtkGDCMThreadedImageReader2 Class Reference . . . 1407
10.391.1 Constructor & Destructor Documentation . . . 1409
10.391.1.1 vtkGDCMThreadedImageReader2() . . . 1409
10.391.1.2 ∼vtkGDCMThreadedImageReader2() . . . 1409
10.391.2 Member Function Documentation . . . 1409
10.391.2.1 GetFileName() . . . 1409
10.391.2.2 New() . . . 1409
10.391.2.3 PrintSelf() . . . 1410
10.391.2.4 RequestInformation() . . . 1410
10.391.2.5 SetFileName() . . . 1410
10.391.2.6 SetFileNames() . . . 1410
10.391.2.7 SplitExtent() . . . 1410
10.391.2.8 ThreadedRequestData() . . . 1411
10.391.2.9 vtkBooleanMacro() [1/3] . . . 1411

10.391.2.10 vtkBooleanMacro() [2/3] . . . 1411
10.391.2.11 vtkBooleanMacro() [3/3] . . . 1411
10.391.2.12 vtkGetMacro() [1/8] . . . 1411
10.391.2.13 vtkGetMacro() [2/8] . . . 1412
10.391.2.14 vtkGetMacro() [3/8] . . . 1412
10.391.2.15 vtkGetMacro() [4/8] . . . 1412
10.391.2.16 vtkGetMacro() [5/8] . . . 1412
10.391.2.17 vtkGetMacro() [6/8] . . . 1412
10.391.2.18 vtkGetMacro() [7/8] . . . 1412
10.391.2.19 vtkGetMacro() [8/8] . . . 1413
10.391.2.20 vtkGetObjectMacro() . . . 1413
10.391.2.21 vtkGetVector3Macro() [1/2] . . . 1413
10.391.2.22 vtkGetVector3Macro() [2/2] . . . 1413
10.391.2.23 vtkGetVector6Macro() . . . 1413
10.391.2.24 vtkSetMacro() [1/7] . . . 1413
10.391.2.25 vtkSetMacro() [2/7] . . . 1414
10.391.2.26 vtkSetMacro() [3/7] . . . 1414
10.391.2.27 vtkSetMacro() [4/7] . . . 1414
10.391.2.28 vtkSetMacro() [5/7] . . . 1414
10.391.2.29 vtkSetMacro() [6/7] . . . 1414
10.391.2.30 vtkSetMacro() [7/7] . . . 1414
10.391.2.31 vtkSetVector3Macro() [1/2] . . . 1415
10.391.2.32 vtkSetVector3Macro() [2/2] . . . 1415
10.391.2.33 vtkSetVector6Macro() . . . 1415
10.391.2.34 vtkTypeMacro() . . . 1415
10.392 vtkImageColorViewer Class Reference . . . 1416
10.392.1 Detailed Description . . . 1418
10.392.2 Member Enumeration Documentation . . . 1418
10.392.2.1 anonymous enum . . . 1418
10.392.3 Constructor & Destructor Documentation . . . 1419
10.392.3.1 vtkImageColorViewer() . . . 1419
10.392.3.2 ~vtkImageColorViewer() . . . 1419
10.392.4 Member Function Documentation . . . 1419
10.392.4.1 AddInput() . . . 1419
10.392.4.2 AddInputConnection() . . . 1419
10.392.4.3 GetColorLevel() . . . 1420
10.392.4.4 GetColorWindow() . . . 1420
10.392.4.5 GetInput() . . . 1420
10.392.4.6 GetOffScreenRendering() . . . 1420

10.392.4.7 GetOverlayVisibility() . . . 1420
10.392.4.8 GetPosition() . . . 1420
10.392.4.9 GetSize() . . . 1420
10.392.4.10 GetSliceMax() . . . 1421
10.392.4.11 GetSliceMin() . . . 1421
10.392.4.12 GetSliceRange() [1/3] . . . 1421
10.392.4.13 GetSliceRange() [2/3] . . . 1421
10.392.4.14 GetSliceRange() [3/3] . . . 1421
10.392.4.15 GetWindowName() . . . 1421
10.392.4.16 InstallPipeline() . . . 1421
10.392.4.17 New() . . . 1422
10.392.4.18 PrintSelf() . . . 1422
10.392.4.19 Render() . . . 1422
10.392.4.20 SetColorLevel() . . . 1422
10.392.4.21 SetColorWindow() . . . 1422
10.392.4.22 SetDisplayId() . . . 1423
10.392.4.23 SetInput() . . . 1423
10.392.4.24 SetInputConnection() . . . 1423
10.392.4.25 SetOffScreenRendering() . . . 1423
10.392.4.26 SetOverlayVisibility() . . . 1423
10.392.4.27 SetParentId() . . . 1423
10.392.4.28 SetPosition() [1/2] . . . 1424
10.392.4.29 SetPosition() [2/2] . . . 1424
10.392.4.30 SetRenderer() . . . 1424
10.392.4.31 SetRenderWindow() . . . 1424
10.392.4.32 SetSize() [1/2] . . . 1424
10.392.4.33 SetSize() [2/2] . . . 1425
10.392.4.34 SetSlice() . . . 1425
10.392.4.35 SetSliceOrientation() . . . 1425
10.392.4.36 SetSliceOrientationToXY() . . . 1425
10.392.4.37 SetSliceOrientationToXZ() . . . 1425
10.392.4.38 SetSliceOrientationToYZ() . . . 1425
10.392.4.39 SetupInteractor() . . . 1426
10.392.4.40 SetWindowId() . . . 1426
10.392.4.41 UnInstallPipeline() . . . 1426
10.392.4.42 UpdateDisplayExtent() . . . 1426
10.392.4.43 UpdateOrientation() . . . 1426
10.392.4.44 VTK_LEGACY() [1/4] . . . 1426
10.392.4.45 VTK_LEGACY() [2/4] . . . 1427

10.392.4.46 VTK_LEGACY() [3/4] . . . 1427
10.392.4.47 VTK_LEGACY() [4/4] . . . 1427
10.392.4.48 vtkBooleanMacro() . . . 1427
10.392.4.49 vtkGetMacro() [1/2] . . . 1427
10.392.4.50 vtkGetMacro() [2/2] . . . 1427
10.392.4.51 vtkGetObjectMacro() [1/5] . . . 1428
10.392.4.52 vtkGetObjectMacro() [2/5] . . . 1428
10.392.4.53 vtkGetObjectMacro() [3/5] . . . 1428
10.392.4.54 vtkGetObjectMacro() [4/5] . . . 1428
10.392.4.55 vtkGetObjectMacro() [5/5] . . . 1428
10.392.4.56 vtkTypeMacro() . . . 1428
10.392.5 Friends And Related Function Documentation . . . 1429
10.392.5.1 vtkImageColorViewerCallback . . . 1429
10.392.6 Member Data Documentation . . . 1429
10.392.6.1 FirstRender . . . 1429
10.392.6.2 ImageActor . . . 1429
10.392.6.3 Interactor . . . 1429
10.392.6.4 InteractorStyle . . . 1429
10.392.6.5 OverlayImageActor . . . 1429
10.392.6.6 Renderer . . . 1430
10.392.6.7 RenderWindow . . . 1430
10.392.6.8 Slice . . . 1430
10.392.6.9 SliceOrientation . . . 1430
10.392.6.10 WindowLevel . . . 1430
10.393 vtkImageMapToColors16 Class Reference . . . 1431
10.393.1 Constructor & Destructor Documentation . . . 1432
10.393.1.1 vtkImageMapToColors16() . . . 1432
10.393.1.2 ~vtkImageMapToColors16() . . . 1432
10.393.2 Member Function Documentation . . . 1433
10.393.2.1 GetMTime() . . . 1433
10.393.2.2 New() . . . 1433
10.393.2.3 PrintSelf() . . . 1433
10.393.2.4 RequestData() . . . 1433
10.393.2.5 RequestInformation() . . . 1433
10.393.2.6 SetLookupTable() . . . 1434
10.393.2.7 SetOutputFormatToLuminance() . . . 1434
10.393.2.8 SetOutputFormatToLuminanceAlpha() . . . 1434
10.393.2.9 SetOutputFormatToRGB() . . . 1434
10.393.2.10 SetOutputFormatToRGBA() . . . 1434

10.393.2.11 ThreadedRequestData() . . . 1434
10.393.2.12 vtkBooleanMacro() . . . 1435
10.393.2.13 vtkGetMacro() [1/3] . . . 1435
10.393.2.14 vtkGetMacro() [2/3] . . . 1435
10.393.2.15 vtkGetMacro() [3/3] . . . 1435
10.393.2.16 vtkGetObjectMacro() . . . 1435
10.393.2.17 vtkSetMacro() [1/3] . . . 1435
10.393.2.18 vtkSetMacro() [2/3] . . . 1436
10.393.2.19 vtkSetMacro() [3/3] . . . 1436
10.393.2.20 vtkTypeMacro() . . . 1436
10.393.3 Member Data Documentation . . . 1436
10.393.3.1 ActiveComponent . . . 1436
10.393.3.2 DataWasPassed . . . 1436
10.393.3.3 LookupTable . . . 1436
10.393.3.4 OutputFormat . . . 1437
10.393.3.5 PassAlphaToOutput . . . 1437
10.394 vtkImageMapToWindowLevelColors2 Class Reference . . . 1437
10.394.1 Constructor & Destructor Documentation . . . 1438
10.394.1.1 vtkImageMapToWindowLevelColors2() . . . 1438
10.394.1.2 ∼vtkImageMapToWindowLevelColors2() . . . 1438
10.394.2 Member Function Documentation . . . 1439
10.394.2.1 New() . . . 1439
10.394.2.2 PrintSelf() . . . 1439
10.394.2.3 RequestData() . . . 1439
10.394.2.4 RequestInformation() . . . 1439
10.394.2.5 ThreadedRequestData() . . . 1439
10.394.2.6 vtkGetMacro() [1/2] . . . 1440
10.394.2.7 vtkGetMacro() [2/2] . . . 1440
10.394.2.8 vtkSetMacro() [1/2] . . . 1440
10.394.2.9 vtkSetMacro() [2/2] . . . 1440
10.394.2.10 vtkTypeMacro() . . . 1440
10.394.3 Member Data Documentation . . . 1440
10.394.3.1 Level . . . 1441
10.394.3.2 Window . . . 1441
10.395 vtkImagePlanarComponentsToComponents Class Reference . . . 1441
10.395.1 Constructor & Destructor Documentation . . . 1442
10.395.1.1 vtkImagePlanarComponentsToComponents() . . . 1442
10.395.1.2 ∼vtkImagePlanarComponentsToComponents() . . . 1442
10.395.2 Member Function Documentation . . . 1442

10.395.2.1 New() . . . 1442
10.395.2.2 PrintSelf() . . . 1443
10.395.2.3 RequestData() . . . 1443
10.395.2.4 vtkTypeMacro() . . . 1443
10.396 vtkImageRGBToYBR Class Reference . . . 1443
10.396.1 Constructor & Destructor Documentation . . . 1444
10.396.1.1 vtkImageRGBToYBR() . . . 1444
10.396.1.2 ∼vtkImageRGBToYBR() . . . 1444
10.396.2 Member Function Documentation . . . 1445
10.396.2.1 New() . . . 1445
10.396.2.2 PrintSelf() . . . 1445
10.396.2.3 ThreadedExecute() . . . 1445
10.396.2.4 vtkTypeMacro() . . . 1445
10.397 vtkImageYBRToRGB Class Reference . . . 1446
10.397.1 Constructor & Destructor Documentation . . . 1447
10.397.1.1 vtkImageYBRToRGB() . . . 1447
10.397.1.2 ∼vtkImageYBRToRGB() . . . 1447
10.397.2 Member Function Documentation . . . 1447
10.397.2.1 New() . . . 1447
10.397.2.2 PrintSelf() . . . 1447
10.397.2.3 ThreadedExecute() . . . 1448
10.397.2.4 vtkTypeMacro() . . . 1448
10.398 vtkLookupTable16 Class Reference . . . 1448
10.398.1 Constructor & Destructor Documentation . . . 1449
10.398.1.1 vtkLookupTable16() . . . 1450
10.398.1.2 ∼vtkLookupTable16() . . . 1450
10.398.2 Member Function Documentation . . . 1450
10.398.2.1 Build() . . . 1450
10.398.2.2 GetPointer() . . . 1450
10.398.2.3 MapScalarsThroughTable2() . . . 1450
10.398.2.4 New() . . . 1451
10.398.2.5 PrintSelf() . . . 1451
10.398.2.6 SetNumberOfTableValues() . . . 1451
10.398.2.7 vtkTypeMacro() . . . 1451
10.398.2.8 WritePointer() . . . 1451
10.398.3 Member Data Documentation . . . 1451
10.398.3.1 Table16 . . . 1452
10.399 vtkRTStructSetProperties Class Reference . . . 1452
10.399.1 Detailed Description . . . 1454

10.399.2 Constructor & Destructor Documentation . . . 1454
10.399.2.1 vtkRTStructSetProperties() . . . 1454
10.399.2.2 ~vtkRTStructSetProperties() . . . 1454
10.399.3 Member Function Documentation . . . 1454
10.399.3.1 AddContourReferencedFrameOfReference() . . . 1455
10.399.3.2 AddReferencedFrameOfReference() . . . 1455
10.399.3.3 AddStructureSetROI() . . . 1455
10.399.3.4 AddStructureSetROIObservation() . . . 1455
10.399.3.5 Clear() . . . 1455
10.399.3.6 DeepCopy() . . . 1456
10.399.3.7 GetContourReferencedFrameOfReferenceClassUID() . . . 1456
10.399.3.8 GetContourReferencedFrameOfReferenceInstanceUID() . . . 1456
10.399.3.9 GetNumberOfContourReferencedFrameOfReferences() [1/2] . . . 1456
10.399.3.10 GetNumberOfContourReferencedFrameOfReferences() [2/2] . . . 1456
10.399.3.11 GetNumberOfReferencedFrameOfReferences() . . . 1456
10.399.3.12 GetNumberOfStructureSetROIs() . . . 1457
10.399.3.13 GetReferencedFrameOfReferenceClassUID() . . . 1457
10.399.3.14 GetReferencedFrameOfReferenceInstanceUID() . . . 1457
10.399.3.15 GetStructureSetObservationNumber() . . . 1457
10.399.3.16 GetStructureSetROIDescription() . . . 1457
10.399.3.17 GetStructureSetROIGenerationAlgorithm() . . . 1457
10.399.3.18 GetStructureSetROIName() . . . 1458
10.399.3.19 GetStructureSetROINumber() . . . 1458
10.399.3.20 GetStructureSetROIObservationLabel() . . . 1458
10.399.3.21 GetStructureSetROIRefFrameRefUID() . . . 1458
10.399.3.22 GetStructureSetRTROIInterpretedType() . . . 1458
10.399.3.23 New() . . . 1458
10.399.3.24 PrintSelf() . . . 1459
10.399.3.25 vtkGetStringMacro() [1/9] . . . 1459
10.399.3.26 vtkGetStringMacro() [2/9] . . . 1459
10.399.3.27 vtkGetStringMacro() [3/9] . . . 1459
10.399.3.28 vtkGetStringMacro() [4/9] . . . 1459
10.399.3.29 vtkGetStringMacro() [5/9] . . . 1459
10.399.3.30 vtkGetStringMacro() [6/9] . . . 1460
10.399.3.31 vtkGetStringMacro() [7/9] . . . 1460
10.399.3.32 vtkGetStringMacro() [8/9] . . . 1460
10.399.3.33 vtkGetStringMacro() [9/9] . . . 1460
10.399.3.34 vtkSetStringMacro() [1/9] . . . 1460
10.399.3.35 vtkSetStringMacro() [2/9] . . . 1460

10.399.3.36 vtkSetStringMacro() [3/9] . . . 1461
10.399.3.37 vtkSetStringMacro() [4/9] . . . 1461
10.399.3.38 vtkSetStringMacro() [5/9] . . . 1461
10.399.3.39 vtkSetStringMacro() [6/9] . . . 1461
10.399.3.40 vtkSetStringMacro() [7/9] . . . 1461
10.399.3.41 vtkSetStringMacro() [8/9] . . . 1461
10.399.3.42 vtkSetStringMacro() [9/9] . . . 1462
10.399.3.43 vtkTypeMacro() . . . 1462
10.399.4 Member Data Documentation . . . 1462
10.399.4.1 Internals . . . 1462
10.399.4.2 ReferenceFrameOfReferenceUID . . . 1462
10.399.4.3 ReferenceSeriesInstanceUID . . . 1462
10.399.4.4 SeriesInstanceUID . . . 1462
10.399.4.5 SOPInstanceUID . . . 1463
10.399.4.6 StructureSetDate . . . 1463
10.399.4.7 StructureSetLabel . . . 1463
10.399.4.8 StructureSetName . . . 1463
10.399.4.9 StructureSetTime . . . 1463
10.399.4.10 StudyInstanceUID . . . 1463
10.400 gdcm::Waveform Class Reference . . . 1463
10.400.1 Detailed Description . . . 1464
10.400.2 Constructor & Destructor Documentation . . . 1464
10.400.2.1 Waveform() . . . 1464
10.401 gdcm::WLMFindQuery Class Reference . . . 1464
10.401.1 Detailed Description . . . 1465
10.401.2 Constructor & Destructor Documentation . . . 1465
10.401.2.1 WLMFindQuery() . . . 1466
10.401.3 Member Function Documentation . . . 1466
10.401.3.1 GetAbstractSyntaxUID() . . . 1466
10.401.3.2 GetTagListByLevel() . . . 1466
10.401.3.3 GetValidDataSet() . . . 1466
10.401.3.4 InitializeDataSet() . . . 1466
10.401.3.5 ValidateQuery() . . . 1467
10.401.4 Friends And Related Function Documentation . . . 1467
10.401.4.1 QueryFactory . . . 1467
10.402 gdcm::Writer Class Reference . . . 1467
10.402.1 Detailed Description . . . 1469
10.402.2 Constructor & Destructor Documentation . . . 1470
10.402.2.1 Writer() . . . 1470

10.402.2.2 ~Writer() . . . 1470
10.402.3 Member Function Documentation . . . 1470
10.402.3.1 CheckFileMetaInformationOff() . . . 1470
10.402.3.2 CheckFileMetaInformationOn() . . . 1471
10.402.3.3 GetCheckFileMetaInformation() . . . 1471
10.402.3.4 GetFile() . . . 1471
10.402.3.5 GetStreamPtr() . . . 1471
10.402.3.6 SetCheckFileMetaInformation() . . . 1471
10.402.3.7 SetFile() . . . 1472
10.402.3.8 SetFileName() . . . 1472
10.402.3.9 SetStream() . . . 1472
10.402.3.10 SetWriteDataSetOnly() . . . 1473
10.402.3.11 Write() . . . 1473
10.402.4 Friends And Related Function Documentation . . . 1473
10.402.4.1 StreamImageWriter . . . 1473
10.402.5 Member Data Documentation . . . 1473
10.402.5.1 Ofstream . . . 1473
10.402.5.2 Stream . . . 1474
10.403 gdcm::XMLDictReader Class Reference . . . 1474
10.403.1 Detailed Description . . . 1475
10.403.2 Constructor & Destructor Documentation . . . 1475
10.403.2.1 XMLDictReader() . . . 1475
10.403.2.2 ~XMLDictReader() . . . 1475
10.403.3 Member Function Documentation . . . 1475
10.403.3.1 CharacterDataHandler() . . . 1476
10.403.3.2 EndElement() . . . 1476
10.403.3.3 GetDict() . . . 1476
10.403.3.4 HandleDescription() . . . 1476
10.403.3.5 HandleEntry() . . . 1476
10.403.3.6 StartElement() . . . 1476
10.404 gdcm::XMLPrinter Class Reference . . . 1477
10.404.1 Member Enumeration Documentation . . . 1478
10.404.1.1 PrintStyles . . . 1478
10.404.2 Constructor & Destructor Documentation . . . 1478
10.404.2.1 XMLPrinter() . . . 1478
10.404.2.2 ~XMLPrinter() . . . 1478
10.404.3 Member Function Documentation . . . 1478
10.404.3.1 GetPrintStyle() . . . 1478
10.404.3.2 HandleBulkData() . . . 1479

10.404.3.3 Print() . . . 1479
10.404.3.4 PrintDataElement() . . . 1479
10.404.3.5 PrintDataSet() . . . 1479
10.404.3.6 PrintSQ() . . . 1479
10.404.3.7 SetFile() . . . 1480
10.404.3.8 SetStyle() . . . 1480
10.404.4 Member Data Documentation . . . 1480
10.404.4.1 F . . . 1480
10.404.4.2 PrintStyle . . . 1480
10.405 gdcm::XMLPrivateDictReader Class Reference . . . 1481
10.405.1 Detailed Description . . . 1482
10.405.2 Constructor & Destructor Documentation . . . 1482
10.405.2.1 XMLPrivateDictReader() . . . 1482
10.405.2.2 ∼XMLPrivateDictReader() . . . 1482
10.405.3 Member Function Documentation . . . 1482
10.405.3.1 CharacterDataHandler() . . . 1482
10.405.3.2 EndElement() . . . 1483
10.405.3.3 GetPrivateDict() . . . 1483
10.405.3.4 HandleDescription() . . . 1483
10.405.3.5 HandleEntry() . . . 1483
10.405.3.6 StartElement() . . . 1483

11 File Documentation **1485**

11.1 README.txt File Reference . . . 1485
11.2 TestsList.txt File Reference . . . 1485
11.3 gdcmASN1.h File Reference . . . 1485
11.4 gdcmASN1.h . . . 1486
11.5 gdcmBase64.h File Reference . . . 1487
11.6 gdcmBase64.h . . . 1487
11.7 gdcmBoxRegion.h File Reference . . . 1488
11.8 gdcmBoxRegion.h . . . 1489
11.9 gdcmByteSwap.h File Reference . . . 1489
11.10 gdcmByteSwap.h . . . 1490
11.11 gdcmCAPICryptoFactory.h File Reference . . . 1491
11.12 gdcmCAPICryptoFactory.h . . . 1492
11.13 gdcmCAPICryptographicMessageSyntax.h File Reference . . . 1492
11.14 gdcmCAPICryptographicMessageSyntax.h . . . 1493
11.15 gdcmCommand.h File Reference . . . 1494
11.16 gdcmCommand.h . . . 1495

11.17 gdcmCryptoFactory.h File Reference . . . 1497
11.18 gdcmCryptoFactory.h . . . 1498
11.19 gdcmCryptographicMessageSyntax.h File Reference . . . 1499
11.20 gdcmCryptographicMessageSyntax.h . . . 1500
11.21 gdcmDataEvent.h File Reference . . . 1501
11.22 gdcmDataEvent.h . . . 1502
11.23 gdcmDeflateStream.h File Reference . . . 1503
11.24 gdcmDeflateStream.h . . . 1503
11.25 gdcmDirectory.h File Reference . . . 1503
11.26 gdcmDirectory.h . . . 1504
11.27 gdcmDummyValueGenerator.h File Reference . . . 1506
11.28 gdcmDummyValueGenerator.h . . . 1506
11.29 gdcmEvent.h File Reference . . . 1507
11.29.1 Macro Definition Documentation . . . 1508
11.29.1.1 gdcmEventMacro . . . 1508
11.30 gdcmEvent.h . . . 1509
11.31 gdcmException.h File Reference . . . 1510
11.32 gdcmException.h . . . 1511
11.33 gdcmFilename.h File Reference . . . 1512
11.34 gdcmFilename.h . . . 1513
11.35 gdcmFileNameEvent.h File Reference . . . 1513
11.36 gdcmFileNameEvent.h . . . 1514
11.37 gdcmFilenameGenerator.h File Reference . . . 1515
11.38 gdcmFilenameGenerator.h . . . 1516
11.39 gdcmLegacyMacro.h File Reference . . . 1516
11.39.1 Macro Definition Documentation . . . 1517
11.39.1.1 GDCM_LEGACY . . . 1517
11.39.1.2 GDCM_LEGACY_BODY . . . 1517
11.39.1.3 GDCM_LEGACY_REPLACED_BODY . . . 1518
11.39.1.4 GDCM_NOOP_STATEMENT . . . 1518
11.40 gdcmLegacyMacro.h . . . 1518
11.41 gdcmMD5.h File Reference . . . 1519
11.42 gdcmMD5.h . . . 1520
11.43 gdcmObject.h File Reference . . . 1520
11.44 gdcmObject.h . . . 1521
11.45 gdcmOpenSSLCryptoFactory.h File Reference . . . 1523
11.46 gdcmOpenSSLCryptoFactory.h . . . 1523
11.47 gdcmOpenSSLCryptographicMessageSyntax.h File Reference . . . 1524
11.48 gdcmOpenSSLCryptographicMessageSyntax.h . . . 1525

11.49 gdcmOpenSSLP7CryptoFactory.h File Reference . . . 1526
11.50 gdcmOpenSSLP7CryptoFactory.h . . . 1527
11.51 gdcmOpenSSLP7CryptographicMessageSyntax.h File Reference . . . 1527
11.52 gdcmOpenSSLP7CryptographicMessageSyntax.h . . . 1529
11.53 gdcmProgressEvent.h File Reference . . . 1529
11.54 gdcmProgressEvent.h . . . 1530
11.55 gdcmRegion.h File Reference . . . 1531
11.56 gdcmRegion.h . . . 1532
11.57 gdcmSHA1.h File Reference . . . 1533
11.58 gdcmSHA1.h . . . 1534
11.59 gdcmSmartPointer.h File Reference . . . 1535
11.60 gdcmSmartPointer.h . . . 1535
11.61 gdcmStaticAssert.h File Reference . . . 1537
11.61.1 Macro Definition Documentation . . . 1537
11.61.1.1 GDCM_DO_JOIN . . . 1537
11.61.1.2 GDCM_DO_JOIN2 . . . 1538
11.61.1.3 GDCM_JOIN . . . 1538
11.61.1.4 GDCM_STATIC_ASSERT . . . 1538
11.62 gdcmStaticAssert.h . . . 1538
11.63 gdcmString.h File Reference . . . 1539
11.64 gdcmString.h . . . 1540
11.65 gdcmSubject.h File Reference . . . 1542
11.66 gdcmSubject.h . . . 1542
11.67 gdcmSwapCode.h File Reference . . . 1543
11.68 gdcmSwapCode.h . . . 1544
11.69 gdcmSwapper.h File Reference . . . 1545
11.70 gdcmSwapper.h . . . 1546
11.71 gdcmSystem.h File Reference . . . 1548
11.72 gdcmSystem.h . . . 1548
11.73 gdcmTerminal.h File Reference . . . 1550
11.74 gdcmTerminal.h . . . 1551
11.75 gdcmTestDriver.h File Reference . . . 1552
11.76 gdcmTestDriver.h . . . 1552
11.77 gdcmTesting.h File Reference . . . 1553
11.78 gdcmTesting.h . . . 1553
11.79 gdcmTrace.h File Reference . . . 1554
11.79.1 Macro Definition Documentation . . . 1556
11.79.1.1 GDCM_FUNCTION . . . 1556
11.79.1.2 gdcmAssertAlwaysMacro . . . 1556

11.79.1.3 gdcmAssertMacro . . . 1556
11.79.1.4 gdcmDebugMacro . . . 1557
11.79.1.5 gdcmErrorMacro . . . 1557
11.79.1.6 gdcmWarningMacro . . . 1558
11.80 gdcmTrace.h . . . 1558
11.81 gdcmTypes.h File Reference . . . 1560
11.82 gdcmTypes.h . . . 1561
11.83 gdcmUnpacker12Bits.h File Reference . . . 1562
11.84 gdcmUnpacker12Bits.h . . . 1562
11.85 gdcmVersion.h File Reference . . . 1563
11.86 gdcmVersion.h . . . 1564
11.87 gdcmWin32.h File Reference . . . 1564
11.87.1 Macro Definition Documentation . . . 1564
11.87.1.1 GDCM_EXPORT . . . 1565
11.88 gdcmWin32.h . . . 1565
11.89 gdcmCSAHeaderDict.h File Reference . . . 1566
11.90 gdcmCSAHeaderDict.h . . . 1567
11.91 gdcmCSAHeaderDictEntry.h File Reference . . . 1569
11.92 gdcmCSAHeaderDictEntry.h . . . 1570
11.93 gdcmDict.h File Reference . . . 1572
11.94 gdcmDict.h . . . 1573
11.95 gdcmDictConverter.h File Reference . . . 1577
11.96 gdcmDictConverter.h . . . 1578
11.97 gdcmDictEntry.h File Reference . . . 1579
11.98 gdcmDictEntry.h . . . 1580
11.99 gdcmDicts.h File Reference . . . 1582
11.100 gdcmDicts.h . . . 1583
11.101 gdcmGlobal.h File Reference . . . 1584
11.102 gdcmGlobal.h . . . 1585
11.103 gdcmGroupDict.h File Reference . . . 1586
11.104 gdcmGroupDict.h . . . 1587
11.105 gdcmSOPClassUIDToIOD.h File Reference . . . 1588
11.106 gdcmSOPClassUIDToIOD.h . . . 1588
11.107 gdcmUIDs.h File Reference . . . 1589
11.108 gdcmUIDs.h . . . 1590
11.109 gdcmAttribute.h File Reference . . . 1603
11.110 gdcmAttribute.h . . . 1604
11.111 gdcmBasicOffsetTable.h File Reference . . . 1617
11.112 gdcmBasicOffsetTable.h . . . 1618

11.113 gdcmByteBuffer.h File Reference . . . 1620
11.114 gdcmByteBuffer.h . . . 1621
11.115 gdcmByteSwapFilter.h File Reference . . . 1622
11.116 gdcmByteSwapFilter.h . . . 1623
11.117 gdcmByteValue.h File Reference . . . 1623
11.118 gdcmByteValue.h . . . 1624
11.119 gdcmCodeString.h File Reference . . . 1628
11.120 gdcmCodeString.h . . . 1628
11.121 gdcmCP246ExplicitDataElement.h File Reference . . . 1630
11.122 gdcmCP246ExplicitDataElement.h . . . 1630
11.123 gdcmCSAElement.h File Reference . . . 1631
11.124 gdcmCSAElement.h . . . 1632
11.125 gdcmCSAHeader.h File Reference . . . 1634
11.126 gdcmCSAHeader.h . . . 1635
11.127 gdcmDataElement.h File Reference . . . 1637
11.128 gdcmDataElement.h . . . 1638
11.129 gdcmDataSet.h File Reference . . . 1640
11.130 gdcmDataSet.h . . . 1642
11.131 gdcmDataSetEvent.h File Reference . . . 1645
11.132 gdcmDataSetEvent.h . . . 1646
11.133 gdcmElement.h File Reference . . . 1647
11.134 gdcmElement.h . . . 1648
11.135 gdcmExplicitDataElement.h File Reference . . . 1659
11.136 gdcmExplicitDataElement.h . . . 1660
11.137 gdcmExplicitImplicitDataElement.h File Reference . . . 1661
11.138 gdcmExplicitImplicitDataElement.h . . . 1662
11.139 gdcmFile.h File Reference . . . 1662
11.140 gdcmFile.h . . . 1663
11.141 gdcmFileMetaInformation.h File Reference . . . 1664
11.142 gdcmFileMetaInformation.h . . . 1665
11.143 gdcmFileSet.h File Reference . . . 1667
11.144 gdcmFileSet.h . . . 1669
11.145 gdcmFragment.h File Reference . . . 1669
11.146 gdcmFragment.h . . . 1671
11.147 gdcmImplicitDataElement.h File Reference . . . 1674
11.148 gdcmImplicitDataElement.h . . . 1674
11.149 gdcmItem.h File Reference . . . 1675
11.150 gdcmItem.h . . . 1676
11.151 gdcmLO.h File Reference . . . 1681

11.152 gdcmLO.h . . . 1681
11.153 gdcmMediaStorage.h File Reference . . . 1682
11.154 gdcmMediaStorage.h . . . 1683
11.155 gdcmMrProtocol.h File Reference . . . 1686
11.156 gdcmMrProtocol.h . . . 1687
11.157 gdcmParseException.h File Reference . . . 1688
11.158 gdcmParseException.h . . . 1689
11.159 gdcmParser.h File Reference . . . 1690
11.160 gdcmParser.h . . . 1691
11.161 gdcmPDBElement.h File Reference . . . 1693
11.162 gdcmPDBElement.h . . . 1694
11.163 gdcmPDBHeader.h File Reference . . . 1695
11.164 gdcmPDBHeader.h . . . 1695
11.165 gdcmPreamble.h File Reference . . . 1696
11.166 gdcmPreamble.h . . . 1698
11.167 gdcmPrivateTag.h File Reference . . . 1699
11.168 gdcmPrivateTag.h . . . 1700
11.169 gdcmReader.h File Reference . . . 1701
11.170 gdcmReader.h . . . 1702
11.171 gdcmSequenceOfFragments.h File Reference . . . 1703
11.172 gdcmSequenceOfFragments.h . . . 1704
11.173 gdcmSequenceOfItems.h File Reference . . . 1708
11.174 gdcmSequenceOfItems.h . . . 1709
11.175 gdcmTag.h File Reference . . . 1712
11.176 gdcmTag.h . . . 1713
11.177 gdcmTagToVR.h File Reference . . . 1717
11.178 gdcmTagToVR.h . . . 1717
11.179 gdcmTransferSyntax.h File Reference . . . 1718
11.180 gdcmTransferSyntax.h . . . 1719
11.181 gdcmUNExplicitDataElement.h File Reference . . . 1720
11.182 gdcmUNExplicitDataElement.h . . . 1721
11.183 gdcmUNExplicitImplicitDataElement.h File Reference . . . 1722
11.184 gdcmUNExplicitImplicitDataElement.h . . . 1723
11.185 gdcmValue.h File Reference . . . 1723
11.186 gdcmValue.h . . . 1724
11.187 gdcmValueIO.h File Reference . . . 1725
11.188 gdcmValueIO.h . . . 1726
11.189 gdcmVL.h File Reference . . . 1726
11.190 gdcmVL.h . . . 1727

11.191 gdcmVM.h File Reference . . . 1729
11.191.1 Macro Definition Documentation . . . 1730
11.191.1.1 TYPETOLENGTH . . . 1730
11.192 gdcmVM.h . . . 1730
11.193 gdcmVR.h File Reference . . . 1732
11.193.1 Macro Definition Documentation . . . 1734
11.193.1.1 TYPETOENCODING . . . 1734
11.193.1.2 VRTypeTemplateCase . . . 1734
11.194 gdcmVR.h . . . 1734
11.195 gdcmVR16ExplicitDataElement.h File Reference . . . 1739
11.196 gdcmVR16ExplicitDataElement.h . . . 1740
11.197 gdcmWriter.h File Reference . . . 1741
11.198 gdcmWriter.h . . . 1742
11.199 gdcmDefinedTerms.h File Reference . . . 1743
11.200 gdcmDefinedTerms.h . . . 1743
11.201 gdcmDefs.h File Reference . . . 1744
11.202 gdcmDefs.h . . . 1745
11.203 gdcmEnumeratedValues.h File Reference . . . 1747
11.204 gdcmEnumeratedValues.h . . . 1747
11.205 gdcmIOD.h File Reference . . . 1748
11.206 gdcmIOD.h . . . 1749
11.207 gdcmIODEntry.h File Reference . . . 1750
11.208 gdcmIODEntry.h . . . 1752
11.209 gdcmIODs.h File Reference . . . 1753
11.210 gdcmIODs.h . . . 1754
11.211 gdcmMacro.h File Reference . . . 1755
11.212 gdcmMacro.h . . . 1757
11.213 gdcmMacroEntry.h File Reference . . . 1758
11.213.1 Macro Definition Documentation . . . 1759
11.213.1.1 GDCMMACROENTRY_H . . . 1759
11.214 gdcmMacroEntry.h . . . 1760
11.215 gdcmMacros.h File Reference . . . 1761
11.216 gdcmMacros.h . . . 1762
11.217 gdcmModule.h File Reference . . . 1763
11.218 gdcmModule.h . . . 1765
11.219 gdcmModuleEntry.h File Reference . . . 1766
11.220 gdcmModuleEntry.h . . . 1768
11.221 gdcmModules.h File Reference . . . 1769
11.222 gdcmModules.h . . . 1770

11.223 gdcmNestedModuleEntries.h File Reference . . . 1771
11.224 gdcmNestedModuleEntries.h . . . 1772
11.225 gdcmPatient.h File Reference . . . 1773
11.226 gdcmPatient.h . . . 1773
11.227 gdcmSeries.h File Reference . . . 1774
11.228 gdcmSeries.h . . . 1775
11.229 gdcmStudy.h File Reference . . . 1776
11.230 gdcmStudy.h . . . 1777
11.231 gdcmTable.h File Reference . . . 1777
11.232 gdcmTable.h . . . 1778
11.233 gdcmTableEntry.h File Reference . . . 1779
11.234 gdcmTableEntry.h . . . 1780
11.235 gdcmTableReader.h File Reference . . . 1781
11.236 gdcmTableReader.h . . . 1782
11.237 gdcmType.h File Reference . . . 1783
11.238 gdcmType.h . . . 1784
11.239 gdcmUsage.h File Reference . . . 1785
11.240 gdcmUsage.h . . . 1787
11.241 gdcmXMLDictReader.h File Reference . . . 1788
11.242 gdcmXMLDictReader.h . . . 1788
11.243 gdcmXMLPrivateDictReader.h File Reference . . . 1789
11.244 gdcmXMLPrivateDictReader.h . . . 1790
11.245 gdcmAnonymizeEvent.h File Reference . . . 1790
11.246 gdcmAnonymizeEvent.h . . . 1792
11.247 gdcmAnonymizer.h File Reference . . . 1792
11.248 gdcmAnonymizer.h . . . 1793
11.249 gdcmApplicationEntity.h File Reference . . . 1794
11.250 gdcmApplicationEntity.h . . . 1795
11.251 gdcmAudioCodec.h File Reference . . . 1796
11.252 gdcmAudioCodec.h . . . 1797
11.253 gdcmBitmap.h File Reference . . . 1797
11.254 gdcmBitmap.h . . . 1798
11.255 gdcmBitmapToBitmapFilter.h File Reference . . . 1801
11.256 gdcmBitmapToBitmapFilter.h . . . 1801
11.257 gdcmCleaner.h File Reference . . . 1802
11.258 gdcmCleaner.h . . . 1803
11.259 gdcmCodec.h File Reference . . . 1804
11.260 gdcmCodec.h . . . 1805
11.261 gdcmCoder.h File Reference . . . 1805

11.262 gdcmCoder.h . . . 1806
11.263 gdcmConstCharWrapper.h File Reference . . . 1807
11.264 gdcmConstCharWrapper.h . . . 1807
11.265 gdcmCurve.h File Reference . . . 1808
11.266 gdcmCurve.h . . . 1809
11.267 gdcmDataSetHelper.h File Reference . . . 1810
11.268 gdcmDataSetHelper.h . . . 1810
11.269 gdcmDecoder.h File Reference . . . 1811
11.270 gdcmDecoder.h . . . 1812
11.271 gdcmDeltaEncodingCodec.h File Reference . . . 1813
11.272 gdcmDeltaEncodingCodec.h . . . 1813
11.273 gdcmDICOMDIR.h File Reference . . . 1814
11.274 gdcmDICOMDIR.h . . . 1815
11.275 gdcmDICOMDIRGenerator.h File Reference . . . 1815
11.276 gdcmDICOMDIRGenerator.h . . . 1816
11.277 gdcmDictPrinter.h File Reference . . . 1817
11.278 gdcmDictPrinter.h . . . 1818
11.279 gdcmDirectionCosines.h File Reference . . . 1818
11.280 gdcmDirectionCosines.h . . . 1819
11.281 gdcmDirectoryHelper.h File Reference . . . 1820
11.282 gdcmDirectoryHelper.h . . . 1820
11.283 gdcmDPath.h File Reference . . . 1821
11.284 gdcmDPath.h . . . 1822
11.285 gdcmDumper.h File Reference . . . 1823
11.286 gdcmDumper.h . . . 1824
11.287 gdcmEmptyMaskGenerator.h File Reference . . . 1825
11.288 gdcmEmptyMaskGenerator.h . . . 1825
11.289 gdcmEncapsulatedDocument.h File Reference . . . 1826
11.290 gdcmEncapsulatedDocument.h . . . 1827
11.291 gdcmEquipmentManufacturer.h File Reference . . . 1827
11.292 gdcmEquipmentManufacturer.h . . . 1828
11.293 gdcmFiducials.h File Reference . . . 1829
11.294 gdcmFiducials.h . . . 1829
11.295 gdcmFileAnonymizer.h File Reference . . . 1830
11.296 gdcmFileAnonymizer.h . . . 1831
11.297 gdcmFileChangeTransferSyntax.h File Reference . . . 1831
11.298 gdcmFileChangeTransferSyntax.h . . . 1832
11.299 gdcmFileDecompressLookupTable.h File Reference . . . 1833
11.300 gdcmFileDecompressLookupTable.h . . . 1834

11.301 gdcmFileDerivation.h File Reference . . . 1835
11.302 gdcmFileDerivation.h . . . 1835
11.303 gdcmFileExplicitFilter.h File Reference . . . 1836
11.304 gdcmFileExplicitFilter.h . . . 1837
11.305 gdcmFileStreamer.h File Reference . . . 1838
11.306 gdcmFileStreamer.h . . . 1838
11.307 gdcmIconImage.h File Reference . . . 1839
11.308 gdcmIconImage.h . . . 1840
11.309 gdcmIconImageFilter.h File Reference . . . 1841
11.310 gdcmIconImageFilter.h . . . 1842
11.311 gdcmIconImageGenerator.h File Reference . . . 1843
11.312 gdcmIconImageGenerator.h . . . 1844
11.313 gdcmImage.h File Reference . . . 1844
11.314 gdcmImage.h . . . 1846
11.315 gdcmImageApplyLookupTable.h File Reference . . . 1847
11.316 gdcmImageApplyLookupTable.h . . . 1847
11.317 gdcmImageChangePhotometricInterpretation.h File Reference . . . 1848
11.318 gdcmImageChangePhotometricInterpretation.h . . . 1849
11.319 gdcmImageChangePlanarConfiguration.h File Reference . . . 1851
11.320 gdcmImageChangePlanarConfiguration.h . . . 1851
11.321 gdcmImageChangeTransferSyntax.h File Reference . . . 1852
11.322 gdcmImageChangeTransferSyntax.h . . . 1853
11.323 gdcmImageCodec.h File Reference . . . 1854
11.324 gdcmImageCodec.h . . . 1855
11.325 gdcmImageConverter.h File Reference . . . 1857
11.326 gdcmImageConverter.h . . . 1858
11.327 gdcmImageFragmentSplitter.h File Reference . . . 1859
11.328 gdcmImageFragmentSplitter.h . . . 1859
11.329 gdcmImageHelper.h File Reference . . . 1860
11.330 gdcmImageHelper.h . . . 1861
11.331 gdcmImageReader.h File Reference . . . 1862
11.332 gdcmImageReader.h . . . 1863
11.333 gdcmImageRegionReader.h File Reference . . . 1864
11.334 gdcmImageRegionReader.h . . . 1865
11.335 gdcmImageToImageFilter.h File Reference . . . 1866
11.336 gdcmImageToImageFilter.h . . . 1866
11.337 gdcmImageWriter.h File Reference . . . 1867
11.338 gdcmImageWriter.h . . . 1868
11.339 gdcmIPPSorter.h File Reference . . . 1868

11.340 gdcmIPPSorter.h . . . 1869
11.341 gdcmJPEG12Codec.h File Reference . . . 1870
11.342 gdcmJPEG12Codec.h . . . 1871
11.343 gdcmJPEG16Codec.h File Reference . . . 1872
11.344 gdcmJPEG16Codec.h . . . 1872
11.345 gdcmJPEG2000Codec.h File Reference . . . 1873
11.346 gdcmJPEG2000Codec.h . . . 1874
11.347 gdcmJPEG8Codec.h File Reference . . . 1875
11.348 gdcmJPEG8Codec.h . . . 1875
11.349 gdcmJPEGCodec.h File Reference . . . 1876
11.350 gdcmJPEGCodec.h . . . 1877
11.351 gdcmJPEGLSCodec.h File Reference . . . 1879
11.352 gdcmJPEGLSCodec.h . . . 1879
11.353 gdcmJSON.h File Reference . . . 1880
11.354 gdcmJSON.h . . . 1881
11.355 gdcmKAKADUCodec.h File Reference . . . 1882
11.356 gdcmKAKADUCodec.h . . . 1883
11.357 gdcmLookupTable.h File Reference . . . 1883
11.358 gdcmLookupTable.h . . . 1884
11.359 gdcmMeshPrimitive.h File Reference . . . 1886
11.360 gdcmMeshPrimitive.h . . . 1887
11.361 gdcmOrientation.h File Reference . . . 1889
11.362 gdcmOrientation.h . . . 1889
11.363 gdcmOverlay.h File Reference . . . 1890
11.364 gdcmOverlay.h . . . 1891
11.365 gdcmPDFCodec.h File Reference . . . 1893
11.366 gdcmPDFCodec.h . . . 1893
11.367 gdcmPersonName.h File Reference . . . 1894
11.368 gdcmPersonName.h . . . 1895
11.369 gdcmPGXCodec.h File Reference . . . 1896
11.370 gdcmPGXCodec.h . . . 1896
11.371 gdcmPhotometricInterpretation.h File Reference . . . 1897
11.372 gdcmPhotometricInterpretation.h . . . 1898
11.373 gdcmPixelFormat.h File Reference . . . 1899
11.374 gdcmPixelFormat.h . . . 1901
11.375 gdcmPixmap.h File Reference . . . 1903
11.376 gdcmPixmap.h . . . 1904
11.377 gdcmPixmapReader.h File Reference . . . 1905
11.378 gdcmPixmapReader.h . . . 1907

11.379 gdcmPixmapToPixmapFilter.h File Reference . . . 1908
11.380 gdcmPixmapToPixmapFilter.h . . . 1908
11.381 gdcmPixmapWriter.h File Reference . . . 1909
11.382 gdcmPixmapWriter.h . . . 1910
11.383 gdcmPNMCodec.h File Reference . . . 1911
11.384 gdcmPNMCodec.h . . . 1912
11.385 gdcmPrinter.h File Reference . . . 1912
11.386 gdcmPrinter.h . . . 1914
11.387 gdcmPVRGCodec.h File Reference . . . 1915
11.388 gdcmPVRGCodec.h . . . 1916
11.389 gdcmRAWCodec.h File Reference . . . 1916
11.390 gdcmRAWCodec.h . . . 1917
11.391 gdcmRescaler.h File Reference . . . 1918
11.392 gdcmRescaler.h . . . 1918
11.393 gdcmRLECodec.h File Reference . . . 1920
11.394 gdcmRLECodec.h . . . 1920
11.395 gdcmScanner.h File Reference . . . 1921
11.396 gdcmScanner.h . . . 1922
11.397 gdcmScanner2.h File Reference . . . 1924
11.398 gdcmScanner2.h . . . 1925
11.399 gdcmSegment.h File Reference . . . 1927
11.400 gdcmSegment.h . . . 1929
11.401 gdcmSegmentedPaletteColorLookupTable.h File Reference . . . 1931
11.402 gdcmSegmentedPaletteColorLookupTable.h . . . 1931
11.403 gdcmSegmentHelper.h File Reference . . . 1932
11.404 gdcmSegmentHelper.h . . . 1933
11.405 gdcmSegmentReader.h File Reference . . . 1934
11.406 gdcmSegmentReader.h . . . 1936
11.407 gdcmSegmentWriter.h File Reference . . . 1936
11.408 gdcmSegmentWriter.h . . . 1938
11.409 gdcmSerieHelper.h File Reference . . . 1938
11.410 gdcmSerieHelper.h . . . 1940
11.411 gdcmSimpleSubjectWatcher.h File Reference . . . 1941
11.412 gdcmSimpleSubjectWatcher.h . . . 1942
11.413 gdcmSorter.h File Reference . . . 1943
11.414 gdcmSorter.h . . . 1945
11.415 gdcmSpacing.h File Reference . . . 1946
11.416 gdcmSpacing.h . . . 1946
11.417 gdcmSpectroscopy.h File Reference . . . 1947

11.418 gdcmSpectroscopy.h . . . 1948
11.419 gdcmSplitMosaicFilter.h File Reference . . . 1948
11.420 gdcmSplitMosaicFilter.h . . . 1949
11.421 gdcmStreamImageReader.h File Reference . . . 1950
11.422 gdcmStreamImageReader.h . . . 1951
11.423 gdcmStreamImageWriter.h File Reference . . . 1952
11.424 gdcmStreamImageWriter.h . . . 1953
11.425 gdcmStrictScanner.h File Reference . . . 1954
11.426 gdcmStrictScanner.h . . . 1955
11.427 gdcmStrictScanner2.h File Reference . . . 1956
11.428 gdcmStrictScanner2.h . . . 1957
11.429 gdcmStringFilter.h File Reference . . . 1959
11.430 gdcmStringFilter.h . . . 1960
11.431 gdcmSurface.h File Reference . . . 1961
11.432 gdcmSurface.h . . . 1962
11.433 gdcmSurfaceHelper.h File Reference . . . 1965
11.434 gdcmSurfaceHelper.h . . . 1966
11.435 gdcmSurfaceReader.h File Reference . . . 1968
11.436 gdcmSurfaceReader.h . . . 1969
11.437 gdcmSurfaceWriter.h File Reference . . . 1970
11.438 gdcmSurfaceWriter.h . . . 1971
11.439 gdcmTagPath.h File Reference . . . 1971
11.440 gdcmTagPath.h . . . 1972
11.441 gdcmUIDGenerator.h File Reference . . . 1973
11.442 gdcmUIDGenerator.h . . . 1974
11.443 gdcmUUIDGenerator.h File Reference . . . 1975
11.444 gdcmUUIDGenerator.h . . . 1975
11.445 gdcmValidate.h File Reference . . . 1976
11.446 gdcmValidate.h . . . 1977
11.447 gdcmWaveform.h File Reference . . . 1977
11.448 gdcmWaveform.h . . . 1978
11.449 gdcmXMLPrinter.h File Reference . . . 1978
11.450 gdcmXMLPrinter.h . . . 1979
11.451 gdcmAAbortPDU.h File Reference . . . 1981
11.452 gdcmAAbortPDU.h . . . 1982
11.453 gdcmAAssociateACPDU.h File Reference . . . 1982
11.454 gdcmAAssociateACPDU.h . . . 1983
11.455 gdcmAAssociateRJPDU.h File Reference . . . 1985
11.456 gdcmAAssociateRJPDU.h . . . 1985

11.457 gdcmAAssociateRQPDU.h File Reference 1986
11.458 gdcmAAssociateRQPDU.h 1987
11.459 gdcmAbstractSyntax.h File Reference 1989
11.460 gdcmAbstractSyntax.h 1990
11.461 gdcmApplicationContext.h File Reference 1991
11.462 gdcmApplicationContext.h 1992
11.463 gdcmAReleaseRPPDU.h File Reference 1992
11.464 gdcmAReleaseRPPDU.h 1993
11.465 gdcmAReleaseRQPDU.h File Reference 1994
11.466 gdcmAReleaseRQPDU.h 1995
11.467 gdcmARTIMTimer.h File Reference 1995
11.468 gdcmARTIMTimer.h 1996
11.469 gdcmAsynchronousOperationsWindowSub.h File Reference 1997
11.470 gdcmAsynchronousOperationsWindowSub.h 1997
11.471 gdcmBaseCompositeMessage.h File Reference 1998
11.472 gdcmBaseCompositeMessage.h 1999
11.473 gdcmBaseNormalizedMessage.h File Reference 2000
11.474 gdcmBaseNormalizedMessage.h 2001
11.475 gdcmBasePDU.h File Reference 2001
11.476 gdcmBasePDU.h 2002
11.477 gdcmBaseQuery.h File Reference 2003
11.478 gdcmBaseQuery.h 2004
11.479 gdcmBaseRootQuery.h File Reference 2005
11.480 gdcmBaseRootQuery.h 2006
11.481 gdcmCEchoMessages.h File Reference 2007
11.482 gdcmCEchoMessages.h 2008
11.483 gdcmCFindMessages.h File Reference 2008
11.484 gdcmCFindMessages.h 2009
11.485 gdcmCMoveMessages.h File Reference 2010
11.486 gdcmCMoveMessages.h 2011
11.487 gdcmCommandDataSet.h File Reference 2012
11.488 gdcmCommandDataSet.h 2012
11.489 gdcmCompositeMessageFactory.h File Reference 2013
11.490 gdcmCompositeMessageFactory.h 2014
11.491 gdcmCompositeNetworkFunctions.h File Reference 2015
11.492 gdcmCompositeNetworkFunctions.h 2015
11.493 gdcmCStoreMessages.h File Reference 2016
11.494 gdcmCStoreMessages.h 2017
11.495 gdcmDIMSE.h File Reference 2018

11.496 gdcmDIMSE.h . . . 2018
11.497 gdcmFindPatientRootQuery.h File Reference . . . 2020
11.498 gdcmFindPatientRootQuery.h . . . 2021
11.499 gdcmFindStudyRootQuery.h File Reference . . . 2022
11.500 gdcmFindStudyRootQuery.h . . . 2022
11.501 gdcmImplementationClassUIDSub.h File Reference . . . 2023
11.502 gdcmImplementationClassUIDSub.h . . . 2024
11.503 gdcmImplementationUIDSub.h File Reference . . . 2025
11.504 gdcmImplementationUIDSub.h . . . 2025
11.505 gdcmImplementationVersionNameSub.h File Reference . . . 2026
11.506 gdcmImplementationVersionNameSub.h . . . 2027
11.507 gdcmMaximumLengthSub.h File Reference . . . 2028
11.508 gdcmMaximumLengthSub.h . . . 2029
11.509 gdcmModalityPerformedProcedureStepCreateQuery.h File Reference . . . 2030
11.510 gdcmModalityPerformedProcedureStepCreateQuery.h . . . 2030
11.511 gdcmModalityPerformedProcedureStepSetQuery.h File Reference . . . 2031
11.512 gdcmModalityPerformedProcedureStepSetQuery.h . . . 2032
11.513 gdcmMovePatientRootQuery.h File Reference . . . 2032
11.514 gdcmMovePatientRootQuery.h . . . 2033
11.515 gdcmMoveStudyRootQuery.h File Reference . . . 2034
11.516 gdcmMoveStudyRootQuery.h . . . 2034
11.517 gdcmNActionMessages.h File Reference . . . 2035
11.518 gdcmNActionMessages.h . . . 2036
11.519 gdcmNCreateMessages.h File Reference . . . 2036
11.520 gdcmNCreateMessages.h . . . 2037
11.521 gdcmNDeleteMessages.h File Reference . . . 2038
11.522 gdcmNDeleteMessages.h . . . 2038
11.523 gdcmNetworkEvents.h File Reference . . . 2039
11.524 gdcmNetworkEvents.h . . . 2040
11.525 gdcmNetworkStateID.h File Reference . . . 2041
11.526 gdcmNetworkStateID.h . . . 2042
11.527 gdcmNEventReportMessages.h File Reference . . . 2043
11.528 gdcmNEventReportMessages.h . . . 2044
11.529 gdcmNGetMessages.h File Reference . . . 2044
11.530 gdcmNGetMessages.h . . . 2045
11.531 gdcmNormalizedMessageFactory.h File Reference . . . 2045
11.532 gdcmNormalizedMessageFactory.h . . . 2046
11.533 gdcmNormalizedNetworkFunctions.h File Reference . . . 2047
11.534 gdcmNormalizedNetworkFunctions.h . . . 2048

11.535 gdcmNSetMessages.h File Reference . . . 2049
11.536 gdcmNSetMessages.h . . . 2049
11.537 gdcmPDataTFPDU.h File Reference . . . 2050
11.538 gdcmPDataTFPDU.h . . . 2051
11.539 gdcmPDUFactory.h File Reference . . . 2052
11.540 gdcmPDUFactory.h . . . 2052
11.541 gdcmPresentationContext.h File Reference . . . 2053
11.542 gdcmPresentationContext.h . . . 2054
11.543 gdcmPresentationContextAC.h File Reference . . . 2055
11.544 gdcmPresentationContextAC.h . . . 2057
11.545 gdcmPresentationContextGenerator.h File Reference . . . 2057
11.546 gdcmPresentationContextGenerator.h . . . 2058
11.547 gdcmPresentationContextRQ.h File Reference . . . 2059
11.548 gdcmPresentationContextRQ.h . . . 2060
11.549 gdcmPresentationDataValue.h File Reference . . . 2061
11.550 gdcmPresentationDataValue.h . . . 2062
11.551 gdcmQueryBase.h File Reference . . . 2063
11.552 gdcmQueryBase.h . . . 2065
11.553 gdcmQueryFactory.h File Reference . . . 2066
11.554 gdcmQueryFactory.h . . . 2067
11.555 gdcmQueryImage.h File Reference . . . 2067
11.556 gdcmQueryImage.h . . . 2068
11.557 gdcmQueryPatient.h File Reference . . . 2069
11.558 gdcmQueryPatient.h . . . 2070
11.559 gdcmQuerySeries.h File Reference . . . 2071
11.560 gdcmQuerySeries.h . . . 2071
11.561 gdcmQueryStudy.h File Reference . . . 2072
11.562 gdcmQueryStudy.h . . . 2073
11.563 gdcmRoleSelectionSub.h File Reference . . . 2074
11.564 gdcmRoleSelectionSub.h . . . 2074
11.565 gdcmServiceClassApplicationInformation.h File Reference . . . 2075
11.566 gdcmServiceClassApplicationInformation.h . . . 2076
11.567 gdcmServiceClassUser.h File Reference . . . 2077
11.568 gdcmServiceClassUser.h . . . 2078
11.569 gdcmSOPClassExtendedNegociationSub.h File Reference . . . 2079
11.570 gdcmSOPClassExtendedNegociationSub.h . . . 2080
11.571 gdcmTransferSyntaxSub.h File Reference . . . 2080
11.572 gdcmTransferSyntaxSub.h . . . 2082
11.573 gdcmULAction.h File Reference . . . 2082

11.574 gdcmULAction.h . . . 2083
11.575 gdcmULActionAA.h File Reference . . . 2084
11.576 gdcmULActionAA.h . . . 2085
11.577 gdcmULActionAE.h File Reference . . . 2086
11.578 gdcmULActionAE.h . . . 2087
11.579 gdcmULActionAR.h File Reference . . . 2088
11.580 gdcmULActionAR.h . . . 2089
11.581 gdcmULActionDT.h File Reference . . . 2091
11.582 gdcmULActionDT.h . . . 2091
11.583 gdcmULBasicCallback.h File Reference . . . 2092
11.584 gdcmULBasicCallback.h . . . 2093
11.585 gdcmULConnection.h File Reference . . . 2093
11.586 gdcmULConnection.h . . . 2094
11.587 gdcmULConnectionCallback.h File Reference . . . 2096
11.588 gdcmULConnectionCallback.h . . . 2097
11.589 gdcmULConnectionInfo.h File Reference . . . 2097
11.590 gdcmULConnectionInfo.h . . . 2099
11.591 gdcmULConnectionManager.h File Reference . . . 2099
11.592 gdcmULConnectionManager.h . . . 2100
11.593 gdcmULEvent.h File Reference . . . 2102
11.594 gdcmULEvent.h . . . 2103
11.595 gdcmULTransitionTable.h File Reference . . . 2104
11.596 gdcmULTransitionTable.h . . . 2105
11.597 gdcmULWritingCallback.h File Reference . . . 2107
11.598 gdcmULWritingCallback.h . . . 2107
11.599 gdcmUserInformation.h File Reference . . . 2108
11.600 gdcmUserInformation.h . . . 2109
11.601 gdcmWLMFindQuery.h File Reference . . . 2110
11.602 gdcmWLMFindQuery.h . . . 2111
11.603 vtkGDCMImageReader.h File Reference . . . 2111
11.603.1 Macro Definition Documentation . . . 2113
11.603.1.1 VTK_CMYK . . . 2113
11.603.1.2 VTK_INVERSE_LUMINANCE . . . 2113
11.603.1.3 VTK_LOOKUP_TABLE . . . 2113
11.603.1.4 VTK_YBR . . . 2113
11.604 vtkGDCMImageReader.h . . . 2113
11.605 vtkGDCMImageReader2.h File Reference . . . 2117
11.605.1 Macro Definition Documentation . . . 2118
11.605.1.1 VTK_CMYK . . . 2118

11.605.1.2 VTK_INVERSE_LUMINANCE . . . 2118
11.605.1.3 VTK_LOOKUP_TABLE . . . 2119
11.605.1.4 VTK_YBR . . . 2119
11.606 vtkGDCMImageReader2.h . . . 2119
11.607 vtkGDCMImageWriter.h File Reference . . . 2122
11.608 vtkGDCMImageWriter.h . . . 2123
11.609 vtkGDCMMedicalImageProperties.h File Reference . . . 2126
11.610 vtkGDCMMedicalImageProperties.h . . . 2126
11.611 vtkGDCMPolyDataReader.h File Reference . . . 2131
11.612 vtkGDCMPolyDataReader.h . . . 2132
11.613 vtkGDCMPolyDataWriter.h File Reference . . . 2133
11.614 vtkGDCMPolyDataWriter.h . . . 2133
11.615 vtkGDCMTesting.h File Reference . . . 2135
11.616 vtkGDCMTesting.h . . . 2135
11.617 vtkGDCMThreadedImageReader.h File Reference . . . 2136
11.618 vtkGDCMThreadedImageReader.h . . . 2136
11.619 vtkGDCMThreadedImageReader2.h File Reference . . . 2138
11.620 vtkGDCMThreadedImageReader2.h . . . 2138
11.621 vtkImageColorViewer.h File Reference . . . 2140
11.622 vtkImageColorViewer.h . . . 2141
11.623 vtkImageMapToColors16.h File Reference . . . 2144
11.624 vtkImageMapToColors16.h . . . 2145
11.625 vtkImageMapToWindowLevelColors2.h File Reference . . . 2147
11.626 vtkImageMapToWindowLevelColors2.h . . . 2147
11.627 vtkImagePlanarComponentsToComponents.h File Reference . . . 2149
11.628 vtkImagePlanarComponentsToComponents.h . . . 2149
11.629 vtkImageRGBToYBR.h File Reference . . . 2150
11.630 vtkImageRGBToYBR.h . . . 2151
11.631 vtkImageYBRToRGB.h File Reference . . . 2152
11.632 vtkImageYBRToRGB.h . . . 2152
11.633 vtkLookupTable16.h File Reference . . . 2153
11.634 vtkLookupTable16.h . . . 2154
11.635 vtkRTStructSetProperties.h File Reference . . . 2155
11.636 vtkRTStructSetProperties.h . . . 2155
11.637 gdcmPythonFilter.h File Reference . . . 2157
11.638 gdcmPythonFilter.h . . . 2158

12 Example Documentation **2159**

12.1 TestByteSwap.cxx . . . 2159

12.2 PatchFile.cxx . . . 2161
12.3 SimplePrint.cs . . . 2162
12.4 TestReader.cxx . . . 2163
12.5 TestReader.py . . . 2164
12.6 DecompressJPEGFile.cs . . . 2165
12.7 ManipulateFile.cs . . . 2166
12.8 ClinicalTrialIdentificationWorkflow.cs . . . 2167
12.9 GenerateDICOMDIR.cs . . . 2169
12.10 GenFakeImage.cxx . . . 2170
12.11 ReformatFile.cs . . . 2171
12.12 DecompressImage.cs . . . 2172
12.13 StandardizeFiles.cs . . . 2174
12.14 ScanDirectory.cs . . . 2175
12.15 BasicAnonymizer.cs . . . 2176
12.16 BasicImageAnonymizer.cs . . . 2177
12.17 CompressLossyJPEG.cs . . . 2178
12.18 DecompressImageMultiframe.cs . . . 2179
12.19 DumpCSA.cs . . . 2181
12.20 ExtractEncapsulatedFile.cs . . . 2182
12.21 ExtractImageRegion.cs . . . 2183
12.22 ExtractImageRegionWithLUT.cs . . . 2184
12.23 ExtractOneFrame.cs . . . 2185
12.24 FileAnonymize.cs . . . 2186
12.25 FileChangeTS.cs . . . 2187
12.26 FileChangeTSLossy.cs . . . 2189
12.27 FileStreaming.cs . . . 2191
12.28 GetArray.cs . . . 2192
12.29 MpegVideoInfo.cs . . . 2193
12.30 NewSequence.cs . . . 2197
12.31 RescaleImage.cs . . . 2198
12.32 SendFileSCU.cs . . . 2199
12.33 SimplePrintPatientName.cs . . . 2199
12.34 SortImage2.cs . . . 2200
12.35 CStoreQtProgress.cxx . . . 2200
12.36 ChangePrivateTags.cxx . . . 2202
12.37 ChangeSequenceUltrasound.cxx . . . 2203
12.38 CheckBigEndianBug.cxx . . . 2204
12.39 ClinicalTrialAnnotate.cxx . . . 2206
12.40 CompressImage.cxx . . . 2207

12.41 ConvertToQImage.cxx . . . 2208
12.42 CreateARGBImage.cxx . . . 2209
12.43 CreateCMYKImage.cxx . . . 2210
12.44 CreateJPIPDataSet.cxx . . . 2211
12.45 DeriveSeries.cxx . . . 2212
12.46 DiffFile.cxx . . . 2213
12.47 DiscriminateVolume.cxx . . . 2214
12.48 DumpADAC.cxx . . . 2217
12.49 DumpExamCard.cxx . . . 2221
12.50 DumpGEMSMovieGroup.cxx . . . 2229
12.51 DumpImageHeaderInfo.cxx . . . 2235
12.52 DumpPhilipsECHO.cxx . . . 2237
12.53 DumpSiemensBase64.cxx . . . 2242
12.54 DumpToSQLITE3.cxx . . . 2243
12.55 DumpToshibaDTI.cxx . . . 2244
12.56 DumpToshibaDTI2.cxx . . . 2246
12.57 DumpVisusChange.cxx . . . 2247
12.58 DuplicatePCDE.cxx . . . 2249
12.59 ELSCINT1WaveToText.cxx . . . 2251
12.60 EmptyMask.cxx . . . 2253
12.61 EncapsulateFileInRawData.cxx . . . 2253
12.62 ExtractEncryptedContent.cxx . . . 2254
12.63 ExtractIconFromFile.cxx . . . 2255
12.64 Extracting_All_Resolution.cxx . . . 2256
12.65 Fake_Image_Using_Stream_Image_Writer.cxx . . . 2261
12.66 FixBrokenJ2K.cxx . . . 2263
12.67 FixJAIBugJPEGLS.cxx . . . 2265
12.68 FixOrientation.cxx . . . 2267
12.69 GenAllVR.cxx . . . 2269
12.70 GenFakeIdentifyFile.cxx . . . 2270
12.71 GenLongSeqs.cxx . . . 2273
12.72 GenSeqs.cxx . . . 2274
12.73 GenerateStandardSOPClasses.cxx . . . 2275
12.74 GetJPEGSamplePrecision.cxx . . . 2276
12.75 GetSequenceUltrasound.cxx . . . 2277
12.76 GetSubSequenceData.cxx . . . 2279
12.77 HelloVizWorld.cxx . . . 2281
12.78 HelloWorld.cxx . . . 2282
12.79 LargeVRDSExplicit.cxx . . . 2283

12.80 MakeTemplate.cxx . . . 2285
12.81 MergeTwoFiles.cxx . . . 2286
12.82 MrProtocol.cxx . . . 2287
12.83 PrintLUT.cxx . . . 2293
12.84 PublicDict.cxx . . . 2294
12.85 QIDO-RS.cxx . . . 2295
12.86 ReadAndDumpDICOMDIR.cxx . . . 2295
12.87 ReadAndDumpDICOMDIR2.cxx . . . 2297
12.88 ReadAndPrintAttributes.cxx . . . 2301
12.89 ReadExplicitLengthSQIVR.cxx . . . 2303
12.90 ReadGEMSSDO.cxx . . . 2303
12.91 ReadMultiTimesException.cxx . . . 2305
12.92 ReadUTF8QtDir.cxx . . . 2306
12.93 SimpleScanner.cxx . . . 2307
12.94 SortImage.cxx . . . 2309
12.95 StreamImageReaderTest.cxx . . . 2310
12.96 TemplateEmptyImage.cxx . . . 2313
12.97 TraverseModules.cxx . . . 2315
12.98 VolumeSorter.cxx . . . 2316
12.99 csa2img.cxx . . . 2318
12.100 iU22tomultisc.cxx . . . 2320
12.101 pmsct_rgb1.cxx . . . 2321
12.102 rle2img.cxx . . . 2324
12.103 uid_unique.cxx . . . 2327
12.104 DecompressImage.java . . . 2327
12.105 DecompressPixmap.java . . . 2328
12.106 ExtractImageRegion.java . . . 2329
12.107 FileAnonymize.java . . . 2330
12.108 HelloSimple.java . . . 2331
12.109 ReadFiles.java . . . 2331
12.110 ScanDirectory.java . . . 2332
12.111 SimplePrint.java . . . 2336
12.112 AddPrivateAttribute.py . . . 2337
12.113 ConvertMPL.py . . . 2337
12.114 ConvertNumpy.py . . . 2338
12.115 ConvertPIL.py . . . 2339
12.116 CreateRAWStorage.py . . . 2340
12.117 DecompressImage.py . . . 2342
12.118 DumbAnonymizer.py . . . 2343

12.119 ExtractImageRegion.py . . . 2344
12.120 FindAllPatientName.py . . . 2345
12.121 FixCommaBug.py . . . 2346
12.122 GetPortionCSAHeader.py . . . 2347
12.123 HelloWorld.py . . . 2348
12.124 ManipulateFile.py . . . 2348
12.125 ManipulateSequence.py . . . 2350
12.126 MergeFile.py . . . 2350
12.127 NewSequence.py . . . 2351
12.128 PhilipsPrivateRescaleInterceptSlope.py . . . 2352
12.129 PlaySound.py . . . 2353
12.130 PrivateDict.py . . . 2354
12.131 ReWriteSCAsMR.py . . . 2354
12.132 ReadAndDumpDICOMDIR.py . . . 2355
12.133 RemovePrivateTags.py . . . 2357
12.134 ScanDirectory.py . . . 2358
12.135 SortImage.py . . . 2359
12.136 WriteBuffer.py . . . 2359
12.137 HelloActiviz.cs . . . 2360
12.138 HelloActiviz2.cs . . . 2361
12.139 HelloActiviz3.cs . . . 2362
12.140 HelloActiviz4.cs . . . 2363
12.141 HelloActiviz5.cs . . . 2364
12.142 HelloVTKWorld.cs . . . 2365
12.143 HelloVTKWorld2.cs . . . 2365
12.144 MetaImageMD5Activiz.cs . . . 2366
12.145 RefCounting.cs . . . 2367
12.146 Compute3DSpacing.cxx . . . 2368
12.147 Convert16BitsTo8Bits.cxx . . . 2369
12.148 ConvertMultiFrameToSingleFrame.cxx . . . 2370
12.149 ConvertRGBToLuminance.cxx . . . 2371
12.150 ConvertSingleBitTo8Bits.cxx . . . 2372
12.151 CreateFakePET.cxx . . . 2373
12.152 CreateFakeRTDOSE.cxx . . . 2374
12.153 GenerateRTSTRUCT.cxx . . . 2376
12.154 MagnifyFile.cxx . . . 2378
12.155 gdcmorthoplanes.cxx . . . 2379
12.156 gdcmreslice.cxx . . . 2384
12.157 gdcmrtionplan.cxx . . . 2386

12.158 gdcmrtplan.cxx 2390
12.159 gdcmscene.cxx 2393
12.160 gdcmtexture.cxx 2395
12.161 gdcmvolume.cxx 2396
12.162 offscreenimage.cxx 2398
12.163 reslicesphere.cxx 2399
12.164 rtstructapp.cxx 2405
12.165 threadgdcm.cxx 2407
12.166 AWTMedical3.java 2410
12.167 HelloVTKWorld.java 2413
12.168 MIPViewer.java 2414
12.169 MPRViewer.java 2416
12.170 MPRViewer2.java 2418
12.171 ReadSeriesIntoVTK.java 2422
12.172 CastConvertPhilips.py 2423
12.173 headsq2dcm.py 2425

Index **2427**

Chapter 1

GDCM Documentation

This is the developpers documentation.

A PDF version of this doxygen documentation can be found here:

```
http://gdcm.sourceforge.net/3.0/gdcm-3.0.13.pdf
```

A tarball version of this HTML doxygen documentation can be found here:

```
http://gdcm.sourceforge.net/3.0/gdcm-3.0.13-doc.tar.gz
```

Author

Mathieu Malaterre

Chapter 2

Todo List

Class gdcm::CSAHeader

MrEvaProtocol in 29,1020 contains ^M that would be nice to get rid of on UNIX system...

Class gdcm::network::ApplicationContext

Looks like Application Context can only be 64 bytes at max (see Figure 9-1 / PS 3.8 - 2009)

Class gdcm::Overlay

Is there actually any way to recognize an overlay ? On images with multiple overlay I do not see any way to differentiate them (other than the group tag).

Class gdcm::SequenceOfFragments

I do not enforce that Sequence of Fragments ends with a SQ end del

Class gdcm::TransferSyntax

: The implementation is completely retarded -> see gdcm::UIDs for a replacement We need: IsSupported We need preprocess of raw/xml file We need GetFullName()

Member gdcm::UIDGenerator::IsValid (const char *uid)

: Move that in DataStructureAndEncoding (see FileMetaInformation::CheckFileMetaInformation)

Chapter 3

Deprecated List

Member gdcm::CompositeNetworkFunctions::ConstructQuery (ERootType inRootType, EQueryLevel in↩ QueryLevel, const KeyValuePairArrayType &keys, EQueryType queryType=eFind)

Member gdcm::FileSet::AddFile (File const &)

. Does nothing

Member gdcm::TransferSyntax::GetSwapCode () const

Return the SwapCode associated with the Transfer Syntax. Be careful with the special GE private syntax the DataSet is written in little endian but the Pixel Data is in Big Endian.

Chapter 4

Bug List

Class gdcm::DICOMDIRGenerator

: There is a current limitation of not handling Referenced SOP Class UID / Referenced SOP Instance UID simply because the Scanner does not allow us See PS 3.11 / Table D.3-2 STD-GEN Additional DICOMDIR Keys

Member gdcm::FileStreamer::StartGroupDataElement (const PrivateTag &pt, size_t maxsizede=0, uint8_t startoffset=0)

maxsizede should be a value lower than the actual total size of the buffer to be copied

Class gdcm::IPPSorter

There are currently a couple of bugs in this implementation:

Chapter 5

Namespace Index

5.1 Namespace List

Here is a list of all namespaces with brief descriptions:

gdcm 43
gdcm::network 77
gdcm::SegmentHelper 84
gdcm::terminal
Class for Terminal 84

Chapter 6

Hierarchical Index

6.1 Class Hierarchy

This inheritance list is sorted roughly, but not completely, alphabetically:

gdcm::network::AbstractSyntax . . . 104
gdcm::network::ApplicationContext . . . 121
gdcm::ApplicationEntity . . . 123
gdcm::network::ARTIMTimer . . . 131
gdcm::ASN1 . . . 132
gdcm::network::AsynchronousOperationsWindowSub . . . 134
gdcm::Attribute< Group, Element, TVR, TVM > . . . 136
gdcm::Attribute< Group, Element, TVR, VM::VM1 > . . . 146
gdcm::Attribute< Group, Element, TVR, VM::VM1_n > . . . 155
gdcm::Attribute< Group, Element, TVR, VM::VM1_3 > . . . 153
gdcm::Attribute< Group, Element, TVR, VM::VM1_8 > . . . 154
gdcm::Attribute< Group, Element, TVR, VM::VM2_n > . . . 163
gdcm::Attribute< Group, Element, TVR, VM::VM2_2n > . . . 162
gdcm::Attribute< Group, Element, TVR, VM::VM3_n > . . . 166
gdcm::Attribute< Group, Element, TVR, VM::VM3_3n > . . . 165
gdcm::Base64 . . . 170
gdcm::network::BaseCompositeMessage . . . 173
gdcm::network::CEchoRQ . . . 239
gdcm::network::CEchoRSP . . . 241
gdcm::network::CFindCancelRQ . . . 242
gdcm::network::CFindRQ . . . 244
gdcm::network::CFindRSP . . . 245
gdcm::network::CMoveCancelRq . . . 253
gdcm::network::CMoveRQ . . . 254
gdcm::network::CMoveRSP . . . 255
gdcm::network::CStoreRQ . . . 310
gdcm::network::CStoreRSP . . . 312
gdcm::network::BaseNormalizedMessage . . . 175
gdcm::network::NActionRQ . . . 741
gdcm::network::NActionRSP . . . 742

gdcm::network::NCreateRQ . . . 744
gdcm::network::NCreateRSP . . . 745
gdcm::network::NDeleteRQ . . . 747
gdcm::network::NDeleteRSP . . . 748
gdcm::network::NEventReportRQ . . . 753
gdcm::network::NEventReportRSP . . . 754
gdcm::network::NGetRQ . . . 756
gdcm::network::NGetRSP . . . 757
gdcm::network::NSetRQ . . . 764
gdcm::network::NSetRSP . . . 766
gdcm::network::BasePDU . . . 177
gdcm::network::AAbortPDU . . . 87
gdcm::network::AAssociateACPDU . . . 90
gdcm::network::AAssociateRJPDU . . . 94
gdcm::network::AAssociateRQPDU . . . 97
gdcm::network::AReleaseRPPDU . . . 126
gdcm::network::AReleaseRQPDU . . . 128
gdcm::network::PDataTFPDU . . . 803
std::basic_string< Char >
std::string
gdcm::String< TDelimiter, TMaxLength, TPadChar > . . . 1092
gdcm::SegmentHelper::BasicCodedEntry . . . 190
gdcm::BitmapToBitmapFilter . . . 211
gdcm::PixmapToPixmapFilter . . . 848
gdcm::ImageToImageFilter . . . 597
gdcm::ImageApplyLookupTable . . . 547
gdcm::ImageChangePhotometricInterpretation . . . 550
gdcm::ImageChangePlanarConfiguration . . . 554
gdcm::ImageChangeTransferSyntax . . . 558
gdcm::ImageFragmentSplitter . . . 579
gdcm::ByteBuffer . . . 219
gdcm::ByteSwap< T > . . . 221
gdcm::ByteSwapFilter . . . 223
gdcm::network::CFind . . . 242
gdcm::Coder . . . 258
gdcm::Codec . . . 257
gdcm::AudioCodec . . . 168
gdcm::ImageCodec . . . 563
gdcm::DeltaEncodingCodec . . . 363
gdcm::JPEG2000Codec . . . 638
gdcm::JPEGCodec . . . 649
gdcm::JPEG12Codec . . . 631
gdcm::JPEG16Codec . . . 635
gdcm::JPEG8Codec . . . 645
gdcm::JPEGLSCodec . . . 658
gdcm::KAKADUCodec . . . 667
gdcm::PGXCodec . . . 821
gdcm::PNMCodec . . . 856
gdcm::PVRGCodec . . . 899
gdcm::RAWCodec . . . 919
gdcm::RLECodec . . . 940
gdcm::PDFCodec . . . 812
gdcm::CodeString . . . 260

gdcm::network::CompositeMessageFactory . . . 271
gdcm::CompositeNetworkFunctions . . . 272
gdcm::ConstCharWrapper . . . 278
gdcm::CryptoFactory . . . 282
  gdcm::CAPICryptoFactory . . . 234
  gdcm::OpenSSLCryptoFactory . . . 771
  gdcm::OpenSSLP7CryptoFactory . . . 776
gdcm::CryptographicMessageSyntax . . . 285
  gdcm::CAPICryptographicMessageSyntax . . . 235
  gdcm::OpenSSLCryptographicMessageSyntax . . . 773
  gdcm::OpenSSLP7CryptographicMessageSyntax . . . 778
gdcm::CSAElement . . . 289
gdcm::CSAHeader . . . 297
gdcm::CSAHeaderDict . . . 302
gdcm::CSAHeaderDictEntry . . . 306
gdcm::DataElement . . . 319
  gdcm::CP246ExplicitDataElement . . . 279
  gdcm::ExplicitDataElement . . . 455
  gdcm::ExplicitImplicitDataElement . . . 458
  gdcm::Fragment . . . 522
    gdcm::BasicOffsetTable . . . 193
  gdcm::ImplicitDataElement . . . 608
  gdcm::Item . . . 626
  gdcm::UNExplicitDataElement . . . 1302
  gdcm::UNExplicitImplicitDataElement . . . 1305
  gdcm::VR16ExplicitDataElement . . . 1342
gdcm::DataSet . . . 338
  gdcm::CommandDataSet . . . 268
  gdcm::FileMetaInformation . . . 485
gdcm::DataSetHelper . . . 355
gdcm::Decoder . . . 356
  gdcm::Codec . . . 257
gdcm::DefinedTerms . . . 358
gdcm::Defs . . . 358
gdcm::DICOMDIR . . . 365
gdcm::DICOMDIRGenerator . . . 366
gdcm::Dict . . . 371
gdcm::DictConverter . . . 375
gdcm::DictEntry . . . 380
gdcm::Dicts . . . 388
gdcm::network::DIMSE . . . 392
gdcm::DirectionCosines . . . 394
gdcm::Directory . . . 397
gdcm::DirectoryHelper . . . 402
gdcm::DPath . . . 404
gdcm::DummyValueGenerator . . . 407
gdcm::Element< TVR, TVM > . . . 409
gdcm::Element< TVR, VM::VM1_n > . . . 417
  gdcm::Element< TVR, VM::VM1_2 > . . . 416
  gdcm::Element< TVR, VM::VM2_n > . . . 425
    gdcm::Element< TVR, VM::VM2_2n > . . . 423
  gdcm::Element< TVR, VM::VM3_4 > . . . 428

gdcm::Element< TVR, VM::VM3_n > . . . 430
gdcm::Element< TVR, VM::VM3_3n > . . . 426
gdcm::Element< VR::AS, VM::VM5 > . . . 432
gdcm::Element< VR::OB, VM::VM1_n > . . . 409
gdcm::Element< VR::OB, VM::VM1 > . . . 433
gdcm::Element< VR::OW, VM::VM1_n > . . . 409
gdcm::Element< VR::OW, VM::VM1 > . . . 434
gdcm::ElementDisableCombinations< TVR, TVM > . . . 436
gdcm::ElementDisableCombinations< VR::OB, VM::VM1_n > . . . 437
gdcm::ElementDisableCombinations< VR::OW, VM::VM1_n > . . . 437
gdcm::EmptyMaskGenerator . . . 437
gdcm::EncapsulatedDocument . . . 440
gdcm::EncodingImplementation< T > . . . 441
gdcm::EncodingImplementation< VR::VRASCII > . . . 442
gdcm::EncodingImplementation< VR::VRBINARY > . . . 443
gdcm::EnumeratedValues . . . 446
gdcm::EquipmentManufacturer . . . 446
gdcm::Event . . . 448
gdcm::AnyEvent . . . 120
gdcm::AbortEvent . . . 103
gdcm::AnonymizeEvent . . . 107
gdcm::DataEvent . . . 334
gdcm::DataSetEvent . . . 351
gdcm::EndEvent . . . 445
gdcm::ExitEvent . . . 454
gdcm::FileNameEvent . . . 498
gdcm::InitializeEvent . . . 611
gdcm::IterationEvent . . . 630
gdcm::ModifiedEvent . . . 721
gdcm::ProgressEvent . . . 895
gdcm::StartEvent . . . 1059
gdcm::UserEvent . . . 1311
gdcm::NoEvent . . . 759
std::exception
gdcm::CSAHeaderDictException . . . 310
gdcm::DataElementException . . . 334
gdcm::Exception . . . 452
gdcm::ParseException . . . 795
gdcm::Fiducials . . . 461
gdcm::FileDerivation . . . 477
gdcm::FileExplicitFilter . . . 481
gdcm::Filename . . . 494
gdcm::FilenameGenerator . . . 502
gdcm::FileSet . . . 506
gdcm::Global . . . 525
gdcm::GroupDict . . . 529
gdcm::IconImageFilter . . . 532
gdcm::IconImageGenerator . . . 535
gdcm::ignore_char . . . 539
gdcm::ImageConverter . . . 577
gdcm::ImageHelper . . . 582
gdcm::network::ImplementationClassUIDSub . . . 604
gdcm::network::ImplementationUIDSub . . . 605

gdcm::network::ImplementationVersionNameSub . . . 606
gdcm::IOD . . . 612
gdcm::IODEntry . . . 614
gdcm::IODs . . . 617
gdcm::JSON . . . 664
gdcm::Scanner2::Itstr . . . 682
gdcm::Scanner::Itstr . . . 683
gdcm::StrictScanner2::Itstr . . . 683
gdcm::StrictScanner::Itstr . . . 684
gdcm::Macro . . . 684
gdcm::Macros . . . 687
gdcm::network::MaximumLengthSub . . . 689
gdcm::MD5 . . . 691
gdcm::MediaStorage . . . 692
gdcm::Module . . . 722
gdcm::ModuleEntry . . . 725
gdcm::NestedModuleEntries . . . 750
gdcm::Modules . . . 729
gdcm::MrProtocol . . . 738
gdcm::network::NormalizedMessageFactory . . . 760
gdcm::NormalizedNetworkFunctions . . . 761
gdcm::Object . . . 767
gdcm::BaseQuery . . . 179
gdcm::BaseRootQuery . . . 185
gdcm::FindPatientRootQuery . . . 516
gdcm::FindStudyRootQuery . . . 519
gdcm::MovePatientRootQuery . . . 732
gdcm::MoveStudyRootQuery . . . 735
gdcm::WLMFindQuery . . . 1464
gdcm::ModalityPerformedProcedureStepCreateQuery . . . 715
gdcm::ModalityPerformedProcedureStepSetQuery . . . 718
gdcm::Bitmap . . . 196
gdcm::Pixmap . . . 838
gdcm::Image . . . 540
gdcm::Curve . . . 313
gdcm::File . . . 461
gdcm::FileWithName . . . 514
gdcm::LookupTable . . . 674
gdcm::SegmentedPaletteColorLookupTable . . . 980
gdcm::MeshPrimitive . . . 709
gdcm::Overlay . . . 784
gdcm::Segment . . . 970
gdcm::Subject . . . 1103
gdcm::Anonymizer . . . 111
gdcm::Cleaner . . . 247
gdcm::Command . . . 265
gdcm::MemberCommand< T > . . . 703
gdcm::SimpleMemberCommand< T > . . . 1027
gdcm::FileAnonymizer . . . 466
gdcm::FileChangeTransferSyntax . . . 470
gdcm::FileDecompressLookupTable . . . 474
gdcm::FileStreamer . . . 508
gdcm::Scanner . . . 949

gdcm::Scanner2 . . . 958
gdcm::ServiceClassUser . . . 1017
gdcm::StrictScanner . . . 1072
gdcm::StrictScanner2 . . . 1081
gdcm::network::ULConnectionManager . . . 1288
gdcm::Surface . . . 1107
gdcm::Value . . . 1318
gdcm::ByteValue . . . 225
gdcm::SequenceOfFragments . . . 991
gdcm::SequenceOfItems . . . 999
gdcm::Orientation . . . 781
gdcm::Parser . . . 797
gdcm::Patient . . . 802
gdcm::PDBElement . . . 806
gdcm::PDBHeader . . . 809
gdcm::network::PDUFactory . . . 815
gdcm::PersonName . . . 818
gdcm::PhotometricInterpretation . . . 824
gdcm::PixelFormat . . . 828
gdcm::Preamble . . . 859
gdcm::PresentationContext . . . 864
gdcm::network::PresentationContextAC . . . 868
gdcm::PresentationContextGenerator . . . 870
gdcm::network::PresentationContextRQ . . . 874
gdcm::network::PresentationDataValue . . . 878
gdcm::Printer . . . 882
gdcm::DictPrinter . . . 385
gdcm::Dumper . . . 408
gdcm::PrivateDict . . . 887
gdcm::PythonFilter . . . 902
gdcm::QueryBase . . . 904
gdcm::QueryImage . . . 909
gdcm::QueryPatient . . . 911
gdcm::QuerySeries . . . 914
gdcm::QueryStudy . . . 916
gdcm::QueryFactory . . . 907
gdcm::Reader . . . 923
gdcm::PixmapReader . . . 844
gdcm::ImageReader . . . 589
gdcm::ImageRegionReader . . . 593
gdcm::SegmentReader . . . 983
gdcm::SurfaceReader . . . 1125
gdcm::RealWorldValueMappingContent . . . 931
gdcm::Region . . . 932
gdcm::BoxRegion . . . 214
gdcm::Rescaler . . . 935
gdcm::network::RoleSelectionSub . . . 947
gdcm::SerieHelper . . . 1008
gdcm::Series . . . 1014
gdcm::network::ServiceClassApplicationInformation . . . 1015
gdcm::SHA1 . . . 1025
gdcm::SimpleSubjectWatcher . . . 1032
gdcm::MrProtocol::Slice . . . 1036

gdcm::MrProtocol::SliceArray . . . 1037
gdcm::SmartPointer< ObjectType > . . . 1038
gdcm::SmartPointer< gdcm::Bitmap > . . . 1038
gdcm::SmartPointer< gdcm::File > . . . 1038
gdcm::SmartPointer< gdcm::Image > . . . 1038
gdcm::SmartPointer< gdcm::MemberCommand > . . . 1038
gdcm::SmartPointer< gdcm::MeshPrimitive > . . . 1038
gdcm::SmartPointer< gdcm::Pixmap > . . . 1038
gdcm::SmartPointer< gdcm::SimpleMemberCommand > . . . 1038
gdcm::SmartPointer< gdcm::Subject > . . . 1038
gdcm::SmartPointer< LookupTable > . . . 1038
gdcm::SmartPointer< Segment > . . . 1038
gdcm::SmartPointer< Surface > . . . 1038
gdcm::SmartPointer< Value > . . . 1038
gdcm::network::SOPClassExtendedNegociationSub . . . 1042
gdcm::SOPClassUIDToIOD . . . 1044
gdcm::Sorter . . . 1046
gdcm::IPPSorter . . . 621
gdcm::Spacing . . . 1051
gdcm::Spectroscopy . . . 1054
gdcm::SplitMosaicFilter . . . 1055
gdcm::static_assert_test< x > . . . 1060
gdcm::STATIC_ASSERTION_FAILURE< x > . . . 1060
gdcm::STATIC_ASSERTION_FAILURE< true > . . . 1060
gdcm::StreamImageReader . . . 1061
gdcm::StreamImageWriter . . . 1065
String<'\', 64 >
gdcm::LO . . . 670
gdcm::StringFilter . . . 1098
gdcm::Study . . . 1102
gdcm::SurfaceHelper . . . 1122
gdcm::SwapCode . . . 1132
gdcm::SwapperDoOp . . . 1134
gdcm::SwapperNoOp . . . 1135
gdcm::System . . . 1136
gdcm::Table . . . 1144
gdcm::TableEntry . . . 1147
gdcm::TableReader . . . 1148
gdcm::XMLDictReader . . . 1474
gdcm::XMLPrivateDictReader . . . 1481
gdcm::network::TableRow . . . 1152
gdcm::Tag . . . 1153
gdcm::PrivateTag . . . 890
gdcm::TagPath . . . 1164
gdcm::Testing . . . 1167
gdcm::Trace . . . 1175
gdcm::TransferSyntax . . . 1180
gdcm::network::TransferSyntaxSub . . . 1187
gdcm::network::Transition . . . 1189
gdcm::Type . . . 1191
gdcm::UI . . . 1194
gdcm::UIDGenerator . . . 1195
gdcm::UIDs . . . 1197

gdcm::network::ULAction . . . 1235
gdcm::network::ULActionAA1 . . . 1238
gdcm::network::ULActionAA2 . . . 1239
gdcm::network::ULActionAA3 . . . 1240
gdcm::network::ULActionAA4 . . . 1242
gdcm::network::ULActionAA5 . . . 1243
gdcm::network::ULActionAA6 . . . 1244
gdcm::network::ULActionAA7 . . . 1246
gdcm::network::ULActionAA8 . . . 1247
gdcm::network::ULActionAE1 . . . 1248
gdcm::network::ULActionAE2 . . . 1250
gdcm::network::ULActionAE3 . . . 1251
gdcm::network::ULActionAE4 . . . 1252
gdcm::network::ULActionAE5 . . . 1254
gdcm::network::ULActionAE6 . . . 1255
gdcm::network::ULActionAE7 . . . 1256
gdcm::network::ULActionAE8 . . . 1258
gdcm::network::ULActionAR1 . . . 1259
gdcm::network::ULActionAR10 . . . 1260
gdcm::network::ULActionAR2 . . . 1262
gdcm::network::ULActionAR3 . . . 1263
gdcm::network::ULActionAR4 . . . 1264
gdcm::network::ULActionAR5 . . . 1266
gdcm::network::ULActionAR6 . . . 1267
gdcm::network::ULActionAR7 . . . 1268
gdcm::network::ULActionAR8 . . . 1270
gdcm::network::ULActionAR9 . . . 1271
gdcm::network::ULActionDT1 . . . 1272
gdcm::network::ULActionDT2 . . . 1274
gdcm::network::ULConnection . . . 1277
gdcm::network::ULConnectionCallback . . . 1283
gdcm::network::ULBasicCallback . . . 1275
gdcm::network::ULWritingCallback . . . 1300
gdcm::network::ULConnectionInfo . . . 1286
gdcm::network::ULEvent . . . 1296
gdcm::network::ULTransitionTable . . . 1298
gdcm::Unpacker12Bits . . . 1307
gdcm::Usage . . . 1308
gdcm::network::UserInformation . . . 1312
gdcm::UUIDGenerator . . . 1314
gdcm::Validate . . . 1315
gdcm::ValueIO< TDE, TSwap, TType > . . . 1321
gdcm::MrProtocol::Vector3 . . . 1322
gdcm::Version . . . 1322
gdcm::VL . . . 1325
gdcm::VM . . . 1329
gdcm::VMToLength< T > . . . 1334
gdcm::VR . . . 1334
gdcm::VRToEncoding< T > . . . 1345
gdcm::VRToType< T > . . . 1345
gdcm::VRToType< TagToType< Group, Element >::VRType > . . . 1345
gdcm::VRToType< TVR > . . . 1345
gdcm::VRVLSize< T > . . . 1345
gdcm::VRVLSize< 0 > . . . 1346

gdcm::VRVLSize< 1 > 1346
vtkImageAlgorithm
vtkImagePlanarComponentsToComponents 1441
vtkImageMapToColors
vtkImageMapToWindowLevelColors2 1437
vtkImageWriter
vtkGDCMImageWriter 1377
vtkLookupTable
vtkLookupTable16 1448
vtkMedicalImageProperties
vtkGDCMMedicalImageProperties 1386
vtkMedicalImageReader2
vtkGDCMImageReader 1347
vtkGDCMThreadedImageReader 1403
vtkGDCMImageReader2 1362
vtkObject
vtkGDCMTesting 1399
vtkImageColorViewer 1416
vtkRTStructSetProperties 1452
vtkPolyDataAlgorithm
vtkGDCMPolyDataReader 1389
vtkPolyDataWriter
vtkGDCMPolyDataWriter 1394
vtkThreadedImageAlgorithm
vtkGDCMThreadedImageReader2 1407
vtkImageMapToColors16 1431
vtkImageRGBToYBR 1443
vtkImageYBRToRGB 1446
gdcm::Waveform 1463
gdcm::Writer 1467
gdcm::PixmapWriter 851
gdcm::ImageWriter 600
gdcm::SegmentWriter 987
gdcm::SurfaceWriter 1129
gdcm::XMLPrinter 1477

Chapter 7

Class Index

7.1 Class List

Here are the classes, structs, unions and interfaces with brief descriptions:

gdcm::network::AAbortPDU
AAbortPDU . . . 87
gdcm::network::AAssociateACPDU
AAssociateACPDU . . . 90
gdcm::network::AAssociateRJPDU
AAssociateRJPDU . . . 94
gdcm::network::AAssociateRQPDU
AAssociateRQPDU . . . 97
gdcm::AbortEvent . . . 103
gdcm::network::AbstractSyntax
AbstractSyntax . . . 104
gdcm::AnonymizeEvent
AnonymizeEvent . . . 107
gdcm::Anonymizer
Anonymizer . . . 111
gdcm::AnyEvent . . . 120
gdcm::network::ApplicationContext
ApplicationContext . . . 121
gdcm::ApplicationEntity
ApplicationEntity . . . 123
gdcm::network::AReleaseRPPDU
AReleaseRPPDU . . . 126
gdcm::network::AReleaseRQPDU
AReleaseRQPDU . . . 128
gdcm::network::ARTIMTimer
ARTIMTimer . . . 131
gdcm::ASN1
Class for ASN1 . . . 132
gdcm::network::AsynchronousOperationsWindowSub
AsynchronousOperationsWindowSub . . . 134

gdcm::Attribute< Group, Element, TVR, TVM >
Attribute class This class use template metaprograming tricks to let the user know when the template instantiation does not match the public dictionary . . . 136
gdcm::Attribute< Group, Element, TVR, VM::VM1 > . . . 146
gdcm::Attribute< Group, Element, TVR, VM::VM1_3 > . . . 153
gdcm::Attribute< Group, Element, TVR, VM::VM1_8 > . . . 154
gdcm::Attribute< Group, Element, TVR, VM::VM1_n > . . . 155
gdcm::Attribute< Group, Element, TVR, VM::VM2_2n > . . . 162
gdcm::Attribute< Group, Element, TVR, VM::VM2_n > . . . 163
gdcm::Attribute< Group, Element, TVR, VM::VM3_3n > . . . 165
gdcm::Attribute< Group, Element, TVR, VM::VM3_n > . . . 166
gdcm::AudioCodec
AudioCodec . . . 168
gdcm::Base64
Class for Base64 . . . 170
gdcm::network::BaseCompositeMessage
BaseCompositeMessage . . . 173
gdcm::network::BaseNormalizedMessage
BaseNormalizedMessage . . . 175
gdcm::network::BasePDU
BasePDU . . . 177
gdcm::BaseQuery
BaseQuery . . . 179
gdcm::BaseRootQuery
BaseRootQuery . . . 185
gdcm::SegmentHelper::BasicCodedEntry
This structure defines a basic coded entry with all of its attributes . . . 190
gdcm::BasicOffsetTable
Class to represent a BasicOffsetTable . . . 193
gdcm::Bitmap
Bitmap class . . . 196
gdcm::BitmapToBitmapFilter
BitmapToBitmapFilter class . . . 211
gdcm::BoxRegion
Class for manipulation box region . . . 214
gdcm::ByteBuffer
ByteBuffer . . . 219
gdcm::ByteSwap< T >
ByteSwap . . . 221
gdcm::ByteSwapFilter
ByteSwapFilter . . . 223
gdcm::ByteValue
Class to represent binary value (array of bytes) . . . 225
gdcm::CAPICryptoFactory . . . 234
gdcm::CAPICryptographicMessageSyntax . . . 235
gdcm::network::CEchoRQ
CEchoRQ . . . 239
gdcm::network::CEchoRSP
CEchoRSP this file defines the messages for the cecho action . . . 241
gdcm::network::CFind . . . 242
gdcm::network::CFindCancelRQ
CFindCancelRQ this file defines the messages for the cfind action . . . 242
gdcm::network::CFindRQ
CFindRQ . . . 244

gdcm::network::CFindRSP
CFindRSP this file defines the messages for the cfind action . . . 245
gdcm::Cleaner
Cleaner . . . 247
gdcm::network::CMoveCancelRq . . . 253
gdcm::network::CMoveRQ
CMoveRQ . . . 254
gdcm::network::CMoveRSP
CMoveRSP this file defines the messages for the cmove action . . . 255
gdcm::Codec
Codec class . . . 257
gdcm::Coder
Coder . . . 258
gdcm::CodeString
CodeString . . . 260
gdcm::Command
Command superclass for callback/observer methods . . . 265
gdcm::CommandDataSet
Class to represent a Command DataSet . . . 268
gdcm::network::CompositeMessageFactory
CompositeMessageFactory . . . 271
gdcm::CompositeNetworkFunctions
Composite Network Functions . . . 272
gdcm::ConstCharWrapper
Do not use me . . . 278
gdcm::CP246ExplicitDataElement
Class to read/write a DataElement as CP246Explicit Data Element . . . 279
gdcm::CryptoFactory
Class to do handle the crypto factory . . . 282
gdcm::CryptographicMessageSyntax . . . 285
gdcm::CSAElement
Class to represent a CSA Element . . . 289
gdcm::CSAHeader
Class for CSAHeader . . . 297
gdcm::CSAHeaderDict
Class to represent a map of CSAHeaderDictEntry . . . 302
gdcm::CSAHeaderDictEntry
Class to represent an Entry in the Dict . . . 306
gdcm::CSAHeaderDictException . . . 310
gdcm::network::CStoreRQ
CStoreRQ . . . 310
gdcm::network::CStoreRSP
CStoreRSP this file defines the messages for the cecho action . . . 312
gdcm::Curve
Curve class to handle element 50xx,3000 Curve Data . . . 313
gdcm::DataElement
Class to represent a Data Element either Implicit or Explicit . . . 319
gdcm::DataElementException . . . 334
gdcm::DataEvent
DataEvent . . . 334
gdcm::DataSet
Class to represent a Data Set (which contains Data Elements) . . . 338
gdcm::DataSetEvent
DataSetEvent . . . 351

gdcm::DataSetHelper
DataSetHelper (internal class, not intended for user level) . . . 355
gdcm::Decoder
Decoder . . . 356
gdcm::DefinedTerms
Defined Terms are used when the specified explicit Values may be extended by implementors to include additional new Values. These new Values shall be specified in the Conformance Statement (see PS 3.2) and shall not have the same meaning as currently defined Values in this standard. A Data Element with Defined Terms that does not contain a Value equivalent to one of the Values currently specified in this standard shall not be considered to have an invalid value. Note: Interpretation Type ID (4008,0210) is an example of a Data Element having Defined Terms. It is defined to have a Value that may be one of the set of standard Values; REPORT or AMENDMENT (see PS 3.3). Because this Data Element has Defined Terms other Interpretation Type IDs may be defined by the implementor . . . 358
gdcm::Defs
FIXME I do not like the name 'Defs' . . . 358
gdcm::DeltaEncodingCodec
DeltaEncodingCodec compression used by some private vendor . . . 363
gdcm::DICOMDIR
DICOMDIR class . . . 365
gdcm::DICOMDIRGenerator
DICOMDIRGenerator class . . . 366
gdcm::Dict
Class to represent a map of DictEntry . . . 371
gdcm::DictConverter
Class to convert a .dic file into something else: . . . 375
gdcm::DictEntry
Class to represent an Entry in the Dict . . . 380
gdcm::DictPrinter
DictPrinter class . . . 385
gdcm::Dicts
Class to manipulate the sum of knowledge (all the dict user load) . . . 388
gdcm::network::DIMSE
DIMSE . . . 392
gdcm::DirectionCosines
Class to handle DirectionCosines . . . 394
gdcm::Directory
Class for manipulation directories . . . 397
gdcm::DirectoryHelper
DirectoryHelper . . . 402
gdcm::DPath
Class to handle a DICOM path While supp 118 did introduced a notion of XPath for XML Native model this convention is too XML-centric. Instead prefer DCMTK style notation `https://groups.↩google.com/g/comp.protocols.dicom/c/IyIH0IOBMPA` . . . 404
gdcm::DummyValueGenerator
Class for generating dummy value . . . 407
gdcm::Dumper
Codec class . . . 408
gdcm::Element< TVR, TVM >
Element class . . . 409
gdcm::Element< TVR, VM::VM1_2 > . . . 416
gdcm::Element< TVR, VM::VM1_n > . . . 417
gdcm::Element< TVR, VM::VM2_2n > . . . 423
gdcm::Element< TVR, VM::VM2_n > . . . 425

gdcm::Element< TVR, VM::VM3_3n > . . . 426
gdcm::Element< TVR, VM::VM3_4 > . . . 428
gdcm::Element< TVR, VM::VM3_n > . . . 430
gdcm::Element< VR::AS, VM::VM5 > . . . 432
gdcm::Element< VR::OB, VM::VM1 > . . . 433
gdcm::Element< VR::OW, VM::VM1 > . . . 434
gdcm::ElementDisableCombinations< TVR, TVM >
A class which is used to produce compile errors for an invalid combination of template parameters . 436
gdcm::ElementDisableCombinations< VR::OB, VM::VM1_n > . . . 437
gdcm::ElementDisableCombinations< VR::OW, VM::VM1_n > . . . 437
gdcm::EmptyMaskGenerator
EmptyMaskGenerator Main class to generate a Empty Mask Series from an input Series. This class takes an input folder and generates a series of DICOM files in the specified output directory. This class handles multiples DICOM Series within the same input directory . . . 437
gdcm::EncapsulatedDocument
EncapsulatedDocument . . . 440
gdcm::EncodingImplementation< T >
EncodingImplementation . . . 441
gdcm::EncodingImplementation< VR::VRASCII > . . . 442
gdcm::EncodingImplementation< VR::VRBINARY > . . . 443
gdcm::EndEvent . . . 445
gdcm::EnumeratedValues
Element. A Data Element with Enumerated Values that does not have a Value equivalent to one of the Values specified in this standard has an invalid value within the scope of a specific Information Object/SOP Class definition. Note: . . . 446
gdcm::EquipmentManufacturer . . . 446
gdcm::Event
Superclass for callback/observer methods . . . 448
gdcm::Exception
Exception . . . 452
gdcm::ExitEvent . . . 454
gdcm::ExplicitDataElement
Class to read/write a DataElement as Explicit Data Element . . . 455
gdcm::ExplicitImplicitDataElement
Class to read/write a DataElement as ExplicitImplicit Data Element . . . 458
gdcm::Fiducials
Fiducials . . . 461
gdcm::File
DICOM File . . . 461
gdcm::FileAnonymizer
FileAnonymizer . . . 466
gdcm::FileChangeTransferSyntax
FileChangeTransferSyntax . . . 470
gdcm::FileDecompressLookupTable
FileDecompressLookupTable class . . . 474
gdcm::FileDerivation
FileDerivation class . . . 477
gdcm::FileExplicitFilter
FileExplicitFilter class . . . 481
gdcm::FileMetaInformation
Class to represent a File Meta Information . . . 485
gdcm::Filename
Class to manipulate file name's . . . 494

gdcm::FileNameEvent
FileNameEvent . . . 498
gdcm::FilenameGenerator
FilenameGenerator . . . 502
gdcm::FileSet . . . 506
gdcm::FileStreamer
FileStreamer . . . 508
gdcm::FileWithName
FileWithName . . . 514
gdcm::FindPatientRootQuery
PatientRootQuery . . . 516
gdcm::FindStudyRootQuery
FindStudyRootQuery . . . 519
gdcm::Fragment
Class to represent a Fragment . . . 522
gdcm::Global
Global . . . 525
gdcm::GroupDict
Class to represent the mapping from group number to its abbreviation and name . . . 529
gdcm::IconImageFilter
IconImageFilter . . . 532
gdcm::IconImageGenerator
IconImageGenerator . . . 535
gdcm::ignore_char . . . 539
gdcm::Image
Image . . . 540
gdcm::ImageApplyLookupTable
ImageApplyLookupTable class . . . 547
gdcm::ImageChangePhotometricInterpretation
ImageChangePhotometricInterpretation class . . . 550
gdcm::ImageChangePlanarConfiguration
ImageChangePlanarConfiguration class . . . 554
gdcm::ImageChangeTransferSyntax
ImageChangeTransferSyntax class . . . 558
gdcm::ImageCodec
ImageCodec . . . 563
gdcm::ImageConverter
Image Converter . . . 577
gdcm::ImageFragmentSplitter
ImageFragmentSplitter class . . . 579
gdcm::ImageHelper
ImageHelper (internal class, not intended for user level) . . . 582
gdcm::ImageReader
ImageReader . . . 589
gdcm::ImageRegionReader
ImageRegionReader . . . 593
gdcm::ImageToImageFilter
ImageToImageFilter class . . . 597
gdcm::ImageWriter
ImageWriter . . . 600
gdcm::network::ImplementationClassUIDSub
ImplementationClassUIDSub . . . 604
gdcm::network::ImplementationUIDSub
ImplementationUIDSub . . . 605

gdcm::network::ImplementationVersionNameSub
ImplementationVersionNameSub . . . 606
gdcm::ImplicitDataElement
Class to represent an *Implicit VR* Data Element . . . 608
gdcm::InitializeEvent . . . 611
gdcm::IOD
Class for representing a IOD . . . 612
gdcm::IODEntry
Class for representing a IODEntry . . . 614
gdcm::IODs
Class for representing a IODs . . . 617
gdcm::IPPSorter
IPPSorter . . . 621
gdcm::Item
Class to represent an Item . . . 626
gdcm::IterationEvent . . . 630
gdcm::JPEG12Codec
Class to do JPEG 12bits (lossy & lossless) . . . 631
gdcm::JPEG16Codec
Class to do JPEG 16bits (lossless) . . . 635
gdcm::JPEG2000Codec
Class to do JPEG 2000 . . . 638
gdcm::JPEG8Codec
Class to do JPEG 8bits (lossy & lossless) . . . 645
gdcm::JPEGCodec
JPEG codec . . . 649
gdcm::JPEGLSCodec
JPEG-LS . . . 658
gdcm::JSON . . . 664
gdcm::KAKADUCodec
KAKADUCodec . . . 667
gdcm::LO
LO . . . 670
gdcm::LookupTable
LookupTable class . . . 674
gdcm::Scanner2::ltstr . . . 682
gdcm::Scanner::ltstr . . . 683
gdcm::StrictScanner2::ltstr . . . 683
gdcm::StrictScanner::ltstr . . . 684
gdcm::Macro
Class for representing a Macro . . . 684
gdcm::Macros
Class for representing a Modules . . . 687
gdcm::network::MaximumLengthSub
MaximumLengthSub . . . 689
gdcm::MD5
Class for MD5 . . . 691
gdcm::MediaStorage
MediaStorage . . . 692
gdcm::MemberCommand< T >
Command subclass that calls a pointer to a member function . . . 703
gdcm::MeshPrimitive
This class defines surface mesh primitives . . . 709

gdcm::ModalityPerformedProcedureStepCreateQuery
ModalityPerformedProcedureStepCreateQuery . . . 715
gdcm::ModalityPerformedProcedureStepSetQuery
ModalityPerformedProcedureStepSetQuery . . . 718
gdcm::ModifiedEvent . . . 721
gdcm::Module
Class for representing a Module . . . 722
gdcm::ModuleEntry
Class for representing a ModuleEntry . . . 725
gdcm::Modules
Class for representing a Modules . . . 729
gdcm::MovePatientRootQuery
MovePatientRootQuery . . . 732
gdcm::MoveStudyRootQuery
MoveStudyRootQuery . . . 735
gdcm::MrProtocol
Class for MrProtocol . . . 738
gdcm::network::NActionRQ
NActionRQ . . . 741
gdcm::network::NActionRSP
NActionRSP this file defines the messages for the NAction action . . . 742
gdcm::network::NCreateRQ
NCreateRQ . . . 744
gdcm::network::NCreateRSP
NCreateRSP this file defines the messages for the ncreate action . . . 745
gdcm::network::NDeleteRQ
NDeleteRQ . . . 747
gdcm::network::NDeleteRSP
NDeleteRSP this file defines the messages for the ndelete action . . . 748
gdcm::NestedModuleEntries
Class for representing a NestedModuleEntries . . . 750
gdcm::network::NEventReportRQ
NEventReportRQ . . . 753
gdcm::network::NEventReportRSP
NEventReportRSP this file defines the messages for the neventreport action . . . 754
gdcm::network::NGetRQ
NGetRQ . . . 756
gdcm::network::NGetRSP
NGetRSP this file defines the messages for the nget action . . . 757
gdcm::NoEvent . . . 759
gdcm::network::NormalizedMessageFactory . . . 760
gdcm::NormalizedNetworkFunctions
Normalized Network Functions . . . 761
gdcm::network::NSetRQ
NSetRQ . . . 764
gdcm::network::NSetRSP
NSetRSP this file defines the messages for the nset action . . . 766
gdcm::Object
Object . . . 767
gdcm::OpenSSLCryptoFactory . . . 771
gdcm::OpenSSLCryptographicMessageSyntax . . . 773
gdcm::OpenSSLP7CryptoFactory . . . 776
gdcm::OpenSSLP7CryptographicMessageSyntax . . . 778

gdcm::Orientation
Class to handle Orientation . . . 781
gdcm::Overlay
Overlay class . . . 784
gdcm::ParseException
ParseException Standard exception handling object . . . 795
gdcm::Parser
Parser ala XML_Parser from expat (SAX) . . . 797
gdcm::Patient
See PS 3.3 - 2007 DICOM MODEL OF THE REAL-WORLD, p 54 . . . 802
gdcm::network::PDataTFPDU
PDataTFPDU . . . 803
gdcm::PDBElement
Class to represent a PDB Element . . . 806
gdcm::PDBHeader
Class for PDBHeader . . . 809
gdcm::PDFCodec
PDFCodec class . . . 812
gdcm::network::PDUFactory
PDUFactory basically, given an initial byte, construct the . . . 815
gdcm::PersonName
PersonName class . . . 818
gdcm::PGXCodec
Class to do PGX . . . 821
gdcm::PhotometricInterpretation
Class to represent an PhotometricInterpretation . . . 824
gdcm::PixelFormat
PixelFormat . . . 828
gdcm::Pixmap
Pixmap class . . . 838
gdcm::PixmapReader
PixmapReader . . . 844
gdcm::PixmapToPixmapFilter
PixmapToPixmapFilter class . . . 848
gdcm::PixmapWriter
PixmapWriter . . . 851
gdcm::PNMCodec
Class to do PNM . . . 856
gdcm::Preamble
DICOM Preamble (Part 10) . . . 859
gdcm::PresentationContext
PresentationContext . . . 864
gdcm::network::PresentationContextAC
PresentationContextAC . . . 868
gdcm::PresentationContextGenerator
PresentationContextGenerator . . . 870
gdcm::network::PresentationContextRQ
PresentationContextRQ . . . 874
gdcm::network::PresentationDataValue
PresentationDataValue . . . 878
gdcm::Printer
Printer class . . . 882
gdcm::PrivateDict
Private Dict . . . 887

gdcm::PrivateTag
Class to represent a Private DICOM Data Element (Attribute) Tag (Group, Element, Owner) 890
gdcm::ProgressEvent
ProgressEvent 895
gdcm::PVRGCodec
PVRGCodec 899
gdcm::PythonFilter
PythonFilter PythonFilter is the class that make gdcm2.x looks more like gdcm1 and transform the binary blob contained in a DataElement into a string, typically this is a nice feature to have for wrapped language 902
gdcm::QueryBase
QueryBase 904
gdcm::QueryFactory
QueryFactory.h 907
gdcm::QueryImage
QueryImage 909
gdcm::QueryPatient
QueryPatient 911
gdcm::QuerySeries
QuerySeries 914
gdcm::QueryStudy
QueryStudy.h 916
gdcm::RAWCodec
RAWCodec class 919
gdcm::Reader
Reader ala DOM (Document Object Model) 923
gdcm::RealWorldValueMappingContent 931
gdcm::Region
Class for manipulation region 932
gdcm::Rescaler
Rescale class 935
gdcm::RLECodec
Class to do RLE 940
gdcm::network::RoleSelectionSub
RoleSelectionSub 947
gdcm::Scanner
Scanner 949
gdcm::Scanner2
Scanner2 958
gdcm::Segment
This class defines a segment 970
gdcm::SegmentedPaletteColorLookupTable
SegmentedPaletteColorLookupTable class 980
gdcm::SegmentReader
This class defines a segment reader 983
gdcm::SegmentWriter
This class defines a segment writer 987
gdcm::SequenceOfFragments
Class to represent a Sequence Of Fragments 991
gdcm::SequenceOfItems
Class to represent a Sequence Of Items 999
gdcm::SerieHelper
SerieHelper DO NOT USE this class, it is only a temporary solution for ITK migration from GDCM 1.x to GDCM 2.x It will disappear soon, you've been warned 1008

gdcm::Series
Series . . . 1014
gdcm::network::ServiceClassApplicationInformation . . . 1015
gdcm::ServiceClassUser
ServiceClassUser . . . 1017
gdcm::SHA1
Class for SHA1 . . . 1025
gdcm::SimpleMemberCommand< T >
Command subclass that calls a pointer to a member function . . . 1027
gdcm::SimpleSubjectWatcher
SimpleSubjectWatcher . . . 1032
gdcm::MrProtocol::Slice . . . 1036
gdcm::MrProtocol::SliceArray . . . 1037
gdcm::SmartPointer< ObjectType >
Class for Smart Pointer . . . 1038
gdcm::network::SOPClassExtendedNegociationSub
SOPClassExtendedNegociationSub . . . 1042
gdcm::SOPClassUIDToIOD
Class convert a class SOP Class UID into IOD . . . 1044
gdcm::Sorter
Sorter . . . 1046
gdcm::Spacing
Class for Spacing . . . 1051
gdcm::Spectroscopy
Spectroscopy class . . . 1054
gdcm::SplitMosaicFilter
SplitMosaicFilter class . . . 1055
gdcm::StartEvent . . . 1059
gdcm::static_assert_test< x > . . . 1060
gdcm::STATIC_ASSERTION_FAILURE< x > . . . 1060
gdcm::STATIC_ASSERTION_FAILURE< true > . . . 1060
gdcm::StreamImageReader
StreamImageReader . . . 1061
gdcm::StreamImageWriter
StreamImageReader . . . 1065
gdcm::StrictScanner
StrictScanner . . . 1072
gdcm::StrictScanner2
StrictScanner2 . . . 1081
gdcm::String< TDelimiter, TMaxLength, TPadChar >
String . . . 1092
gdcm::StringFilter
StringFilter . . . 1098
gdcm::Study
Study . . . 1102
gdcm::Subject
Subject . . . 1103
gdcm::Surface
This class defines a SURFACE IE . . . 1107
gdcm::SurfaceHelper
SurfaceHelper . . . 1122
gdcm::SurfaceReader
This class defines a SURFACE IE reader . . . 1125

gdcm::SurfaceWriter
This class defines a SURFACE IE writer . . . 1129
gdcm::SwapCode
SwapCode representation . . . 1132
gdcm::SwapperDoOp . . . 1134
gdcm::SwapperNoOp . . . 1135
gdcm::System
Class to do system operation . . . 1136
gdcm::Table
Table . . . 1144
gdcm::TableEntry
TableEntry . . . 1147
gdcm::TableReader
Class for representing a TableReader . . . 1148
gdcm::network::TableRow . . . 1152
gdcm::Tag
Class to represent a DICOM Data Element (Attribute) Tag (Group, Element) . . . 1153
gdcm::TagPath
Class to handle a path of tag . . . 1164
gdcm::Testing
Class for testing . . . 1167
gdcm::Trace
Trace . . . 1175
gdcm::TransferSyntax
Class to manipulate Transfer Syntax . . . 1180
gdcm::network::TransferSyntaxSub
TransferSyntaxSub . . . 1187
gdcm::network::Transition . . . 1189
gdcm::Type
Type . . . 1191
gdcm::UI . . . 1194
gdcm::UIDGenerator
Class for generating unique UID . . . 1195
gdcm::UIDs
All known uids . . . 1197
gdcm::network::ULAction
ULAction . . . 1235
gdcm::network::ULActionAA1 . . . 1238
gdcm::network::ULActionAA2 . . . 1239
gdcm::network::ULActionAA3 . . . 1240
gdcm::network::ULActionAA4 . . . 1242
gdcm::network::ULActionAA5 . . . 1243
gdcm::network::ULActionAA6 . . . 1244
gdcm::network::ULActionAA7 . . . 1246
gdcm::network::ULActionAA8 . . . 1247
gdcm::network::ULActionAE1 . . . 1248
gdcm::network::ULActionAE2 . . . 1250
gdcm::network::ULActionAE3 . . . 1251
gdcm::network::ULActionAE4 . . . 1252
gdcm::network::ULActionAE5 . . . 1254
gdcm::network::ULActionAE6 . . . 1255
gdcm::network::ULActionAE7 . . . 1256
gdcm::network::ULActionAE8 . . . 1258
gdcm::network::ULActionAR1 . . . 1259

gdcm::network::ULActionAR10 . . . 1260
gdcm::network::ULActionAR2 . . . 1262
gdcm::network::ULActionAR3 . . . 1263
gdcm::network::ULActionAR4 . . . 1264
gdcm::network::ULActionAR5 . . . 1266
gdcm::network::ULActionAR6 . . . 1267
gdcm::network::ULActionAR7 . . . 1268
gdcm::network::ULActionAR8 . . . 1270
gdcm::network::ULActionAR9 . . . 1271
gdcm::network::ULActionDT1 . . . 1272
gdcm::network::ULActionDT2 . . . 1274
gdcm::network::ULBasicCallback
ULBasicCallback . . . 1275
gdcm::network::ULConnection
ULConnection . . . 1277
gdcm::network::ULConnectionCallback . . . 1283
gdcm::network::ULConnectionInfo
ULConnectionInfo . . . 1286
gdcm::network::ULConnectionManager
ULConnectionManager . . . 1288
gdcm::network::ULEvent
ULEvent . . . 1296
gdcm::network::ULTransitionTable
ULTransitionTable The transition table of all the ULEvents, new ULActions, and ULStates . . . 1298
gdcm::network::ULWritingCallback . . . 1300
gdcm::UNExplicitDataElement
Class to read/write a DataElement as UNExplicit Data Element . . . 1302
gdcm::UNExplicitImplicitDataElement
Class to read/write a DataElement as ExplicitImplicit Data Element . . . 1305
gdcm::Unpacker12Bits
Pack/Unpack 12 bits pixel into 16bits . . . 1307
gdcm::Usage
Usage . . . 1308
gdcm::UserEvent . . . 1311
gdcm::network::UserInformation
UserInformation . . . 1312
gdcm::UUIDGenerator
Class for generating unique UUID . . . 1314
gdcm::Validate
Validate class . . . 1315
gdcm::Value
Class to represent the value of a Data Element . . . 1318
gdcm::ValueIO< TDE, TSwap, TType >
Class to dispatch template calls . . . 1321
gdcm::MrProtocol::Vector3 . . . 1322
gdcm::Version
Major/minor and build version . . . 1322
gdcm::VL
Value Length . . . 1325
gdcm::VM
Value Multiplicity Looking at the DICOMV3 dict only there is very few cases: 1 2 3 4 5 6 8 16 24 1-2 1-3 1-8 1-32 1-99 1-n 2-2n 2-n 3-3n 3-n . . . 1329
gdcm::VMToLength< T > . . . 1334

gdcm::VR
VR class . . . 1334
gdcm::VR16ExplicitDataElement
Class to read/write a DataElement as Explicit Data Element . . . 1342
gdcm::VRToEncoding< T > . . . 1345
gdcm::VRToType< T > . . . 1345
gdcm::VRVLSize< T > . . . 1345
gdcm::VRVLSize< 0 > . . . 1346
gdcm::VRVLSize< 1 > . . . 1346
vtkGDCMImageReader . . . 1347
vtkGDCMImageReader2 . . . 1362
vtkGDCMImageWriter . . . 1377
vtkGDCMMedicalImageProperties . . . 1386
vtkGDCMPolyDataReader . . . 1389
vtkGDCMPolyDataWriter . . . 1394
vtkGDCMTesting . . . 1399
vtkGDCMThreadedImageReader . . . 1403
vtkGDCMThreadedImageReader2 . . . 1407
vtkImageColorViewer . . . 1416
vtkImageMapToColors16 . . . 1431
vtkImageMapToWindowLevelColors2 . . . 1437
vtkImagePlanarComponentsToComponents . . . 1441
vtkImageRGBToYBR . . . 1443
vtkImageYBRToRGB . . . 1446
vtkLookupTable16 . . . 1448
vtkRTStructSetProperties . . . 1452
gdcm::Waveform
Waveform class . . . 1463
gdcm::WLMFindQuery
PatientRootQuery . . . 1464
gdcm::Writer
Writer ala DOM (Document Object Model) . . . 1467
gdcm::XMLDictReader
Class for representing a XMLDictReader . . . 1474
gdcm::XMLPrinter . . . 1477
gdcm::XMLPrivateDictReader
Class for representing a XMLPrivateDictReader . . . 1481

Chapter 8

File Index

8.1 File List

Here is a list of all files with brief descriptions:

gdcmASN1.h . . . 1485
gdcmBase64.h . . . 1487
gdcmBoxRegion.h . . . 1488
gdcmByteSwap.h . . . 1489
gdcmCAPICryptoFactory.h . . . 1491
gdcmCAPICryptographicMessageSyntax.h . . . 1492
gdcmCommand.h . . . 1494
gdcmCryptoFactory.h . . . 1497
gdcmCryptographicMessageSyntax.h . . . 1499
gdcmDataEvent.h . . . 1501
gdcmDeflateStream.h . . . 1503
gdcmDirectory.h . . . 1503
gdcmDummyValueGenerator.h . . . 1506
gdcmEvent.h . . . 1507
gdcmException.h . . . 1510
gdcmFilename.h . . . 1512
gdcmFileNameEvent.h . . . 1513
gdcmFilenameGenerator.h . . . 1515
gdcmLegacyMacro.h . . . 1516
gdcmMD5.h . . . 1519
gdcmObject.h . . . 1520
gdcmOpenSSLCryptoFactory.h . . . 1523
gdcmOpenSSLCryptographicMessageSyntax.h . . . 1524
gdcmOpenSSLP7CryptoFactory.h . . . 1526
gdcmOpenSSLP7CryptographicMessageSyntax.h . . . 1527
gdcmProgressEvent.h . . . 1529
gdcmRegion.h . . . 1531
gdcmSHA1.h . . . 1533
gdcmSmartPointer.h . . . 1535
gdcmStaticAssert.h . . . 1537
gdcmString.h . . . 1539

gdcmSubject.h . . . 1542
gdcmSwapCode.h . . . 1543
gdcmSwapper.h . . . 1545
gdcmSystem.h . . . 1548
gdcmTerminal.h . . . 1550
gdcmTestDriver.h . . . 1552
gdcmTesting.h . . . 1553
gdcmTrace.h . . . 1554
gdcmTypes.h . . . 1560
gdcmUnpacker12Bits.h . . . 1562
gdcmVersion.h . . . 1563
gdcmWin32.h . . . 1564
gdcmCSAHeaderDict.h . . . 1566
gdcmCSAHeaderDictEntry.h . . . 1569
gdcmDict.h . . . 1572
gdcmDictConverter.h . . . 1577
gdcmDictEntry.h . . . 1579
gdcmDicts.h . . . 1582
gdcmGlobal.h . . . 1584
gdcmGroupDict.h . . . 1586
gdcmSOPClassUIDToIOD.h . . . 1588
gdcmUIDs.h . . . 1589
gdcmAttribute.h . . . 1603
gdcmBasicOffsetTable.h . . . 1617
gdcmByteBuffer.h . . . 1620
gdcmByteSwapFilter.h . . . 1622
gdcmByteValue.h . . . 1623
gdcmCodeString.h . . . 1628
gdcmCP246ExplicitDataElement.h . . . 1630
gdcmCSAElement.h . . . 1631
gdcmCSAHeader.h . . . 1634
gdcmDataElement.h . . . 1637
gdcmDataSet.h . . . 1640
gdcmDataSetEvent.h . . . 1645
gdcmElement.h . . . 1647
gdcmExplicitDataElement.h . . . 1659
gdcmExplicitImplicitDataElement.h . . . 1661
gdcmFile.h . . . 1662
gdcmFileMetaInformation.h . . . 1664
gdcmFileSet.h . . . 1667
gdcmFragment.h . . . 1669
gdcmImplicitDataElement.h . . . 1674
gdcmItem.h . . . 1675
gdcmLO.h . . . 1681
gdcmMediaStorage.h . . . 1682
gdcmMrProtocol.h . . . 1686
gdcmParseException.h . . . 1688
gdcmParser.h . . . 1690
gdcmPDBElement.h . . . 1693
gdcmPDBHeader.h . . . 1695
gdcmPreamble.h . . . 1696
gdcmPrivateTag.h . . . 1699
gdcmReader.h . . . 1701
gdcmSequenceOfFragments.h . . . 1703

gdcmSequenceOfItems.h . . . 1708
gdcmTag.h . . . 1712
gdcmTagToVR.h . . . 1717
gdcmTransferSyntax.h . . . 1718
gdcmUNExplicitDataElement.h . . . 1720
gdcmUNExplicitImplicitDataElement.h . . . 1722
gdcmValue.h . . . 1723
gdcmValueIO.h . . . 1725
gdcmVL.h . . . 1726
gdcmVM.h . . . 1729
gdcmVR.h . . . 1732
gdcmVR16ExplicitDataElement.h . . . 1739
gdcmWriter.h . . . 1741
gdcmDefinedTerms.h . . . 1743
gdcmDefs.h . . . 1744
gdcmEnumeratedValues.h . . . 1747
gdcmIOD.h . . . 1748
gdcmIODEntry.h . . . 1750
gdcmIODs.h . . . 1753
gdcmMacro.h . . . 1755
gdcmMacroEntry.h . . . 1758
gdcmMacros.h . . . 1761
gdcmModule.h . . . 1763
gdcmModuleEntry.h . . . 1766
gdcmModules.h . . . 1769
gdcmNestedModuleEntries.h . . . 1771
gdcmPatient.h . . . 1773
gdcmSeries.h . . . 1774
gdcmStudy.h . . . 1776
gdcmTable.h . . . 1777
gdcmTableEntry.h . . . 1779
gdcmTableReader.h . . . 1781
gdcmType.h . . . 1783
gdcmUsage.h . . . 1785
gdcmXMLDictReader.h . . . 1788
gdcmXMLPrivateDictReader.h . . . 1789
gdcmAnonymizeEvent.h . . . 1790
gdcmAnonymizer.h . . . 1792
gdcmApplicationEntity.h . . . 1794
gdcmAudioCodec.h . . . 1796
gdcmBitmap.h . . . 1797
gdcmBitmapToBitmapFilter.h . . . 1801
gdcmCleaner.h . . . 1802
gdcmCodec.h . . . 1804
gdcmCoder.h . . . 1805
gdcmConstCharWrapper.h . . . 1807
gdcmCurve.h . . . 1808
gdcmDataSetHelper.h . . . 1810
gdcmDecoder.h . . . 1811
gdcmDeltaEncodingCodec.h . . . 1813
gdcmDICOMDIR.h . . . 1814
gdcmDICOMDIRGenerator.h . . . 1815
gdcmDictPrinter.h . . . 1817
gdcmDirectionCosines.h . . . 1818

gdcmDirectoryHelper.h . . . 1820
gdcmDPath.h . . . 1821
gdcmDumper.h . . . 1823
gdcmEmptyMaskGenerator.h . . . 1825
gdcmEncapsulatedDocument.h . . . 1826
gdcmEquipmentManufacturer.h . . . 1827
gdcmFiducials.h . . . 1829
gdcmFileAnonymizer.h . . . 1830
gdcmFileChangeTransferSyntax.h . . . 1831
gdcmFileDecompressLookupTable.h . . . 1833
gdcmFileDerivation.h . . . 1835
gdcmFileExplicitFilter.h . . . 1836
gdcmFileStreamer.h . . . 1838
gdcmIconImage.h . . . 1839
gdcmIconImageFilter.h . . . 1841
gdcmIconImageGenerator.h . . . 1843
gdcmImage.h . . . 1844
gdcmImageApplyLookupTable.h . . . 1847
gdcmImageChangePhotometricInterpretation.h . . . 1848
gdcmImageChangePlanarConfiguration.h . . . 1851
gdcmImageChangeTransferSyntax.h . . . 1852
gdcmImageCodec.h . . . 1854
gdcmImageConverter.h . . . 1857
gdcmImageFragmentSplitter.h . . . 1859
gdcmImageHelper.h . . . 1860
gdcmImageReader.h . . . 1862
gdcmImageRegionReader.h . . . 1864
gdcmImageToImageFilter.h . . . 1866
gdcmImageWriter.h . . . 1867
gdcmIPPSorter.h . . . 1868
gdcmJPEG12Codec.h . . . 1870
gdcmJPEG16Codec.h . . . 1872
gdcmJPEG2000Codec.h . . . 1873
gdcmJPEG8Codec.h . . . 1875
gdcmJPEGCodec.h . . . 1876
gdcmJPEGLSCodec.h . . . 1879
gdcmJSON.h . . . 1880
gdcmKAKADUCodec.h . . . 1882
gdcmLookupTable.h . . . 1883
gdcmMeshPrimitive.h . . . 1886
gdcmOrientation.h . . . 1889
gdcmOverlay.h . . . 1890
gdcmPDFCodec.h . . . 1893
gdcmPersonName.h . . . 1894
gdcmPGXCodec.h . . . 1896
gdcmPhotometricInterpretation.h . . . 1897
gdcmPixelFormat.h . . . 1899
gdcmPixmap.h . . . 1903
gdcmPixmapReader.h . . . 1905
gdcmPixmapToPixmapFilter.h . . . 1908
gdcmPixmapWriter.h . . . 1909
gdcmPNMCodec.h . . . 1911
gdcmPrinter.h . . . 1912
gdcmPVRGCodec.h . . . 1915

gdcmRAWCodec.h . . . 1916
gdcmRescaler.h . . . 1918
gdcmRLECodec.h . . . 1920
gdcmScanner.h . . . 1921
gdcmScanner2.h . . . 1924
gdcmSegment.h . . . 1927
gdcmSegmentedPaletteColorLookupTable.h . . . 1931
gdcmSegmentHelper.h . . . 1932
gdcmSegmentReader.h . . . 1934
gdcmSegmentWriter.h . . . 1936
gdcmSerieHelper.h . . . 1938
gdcmSimpleSubjectWatcher.h . . . 1941
gdcmSorter.h . . . 1943
gdcmSpacing.h . . . 1946
gdcmSpectroscopy.h . . . 1947
gdcmSplitMosaicFilter.h . . . 1948
gdcmStreamImageReader.h . . . 1950
gdcmStreamImageWriter.h . . . 1952
gdcmStrictScanner.h . . . 1954
gdcmStrictScanner2.h . . . 1956
gdcmStringFilter.h . . . 1959
gdcmSurface.h . . . 1961
gdcmSurfaceHelper.h . . . 1965
gdcmSurfaceReader.h . . . 1968
gdcmSurfaceWriter.h . . . 1970
gdcmTagPath.h . . . 1971
gdcmUIDGenerator.h . . . 1973
gdcmUUIDGenerator.h . . . 1975
gdcmValidate.h . . . 1976
gdcmWaveform.h . . . 1977
gdcmXMLPrinter.h . . . 1978
gdcmAAbortPDU.h . . . 1981
gdcmAAssociateACPDU.h . . . 1982
gdcmAAssociateRJPDU.h . . . 1985
gdcmAAssociateRQPDU.h . . . 1986
gdcmAbstractSyntax.h . . . 1989
gdcmApplicationContext.h . . . 1991
gdcmAReleaseRPPDU.h . . . 1992
gdcmAReleaseRQPDU.h . . . 1994
gdcmARTIMTimer.h . . . 1995
gdcmAsynchronousOperationsWindowSub.h . . . 1997
gdcmBaseCompositeMessage.h . . . 1998
gdcmBaseNormalizedMessage.h . . . 2000
gdcmBasePDU.h . . . 2001
gdcmBaseQuery.h . . . 2003
gdcmBaseRootQuery.h . . . 2005
gdcmCEchoMessages.h . . . 2007
gdcmCFindMessages.h . . . 2008
gdcmCMoveMessages.h . . . 2010
gdcmCommandDataSet.h . . . 2012
gdcmCompositeMessageFactory.h . . . 2013
gdcmCompositeNetworkFunctions.h . . . 2015
gdcmCStoreMessages.h . . . 2016
gdcmDIMSE.h . . . 2018

gdcmFindPatientRootQuery.h . . . 2020
gdcmFindStudyRootQuery.h . . . 2022
gdcmImplementationClassUIDSub.h . . . 2023
gdcmImplementationUIDSub.h . . . 2025
gdcmImplementationVersionNameSub.h . . . 2026
gdcmMaximumLengthSub.h . . . 2028
gdcmModalityPerformedProcedureStepCreateQuery.h . . . 2030
gdcmModalityPerformedProcedureStepSetQuery.h . . . 2031
gdcmMovePatientRootQuery.h . . . 2032
gdcmMoveStudyRootQuery.h . . . 2034
gdcmNActionMessages.h . . . 2035
gdcmNCreateMessages.h . . . 2036
gdcmNDeleteMessages.h . . . 2038
gdcmNetworkEvents.h . . . 2039
gdcmNetworkStateID.h . . . 2041
gdcmNEventReportMessages.h . . . 2043
gdcmNGetMessages.h . . . 2044
gdcmNormalizedMessageFactory.h . . . 2045
gdcmNormalizedNetworkFunctions.h . . . 2047
gdcmNSetMessages.h . . . 2049
gdcmPDataTFPDU.h . . . 2050
gdcmPDUFactory.h . . . 2052
gdcmPresentationContext.h . . . 2053
gdcmPresentationContextAC.h . . . 2055
gdcmPresentationContextGenerator.h . . . 2057
gdcmPresentationContextRQ.h . . . 2059
gdcmPresentationDataValue.h . . . 2061
gdcmQueryBase.h . . . 2063
gdcmQueryFactory.h . . . 2066
gdcmQueryImage.h . . . 2067
gdcmQueryPatient.h . . . 2069
gdcmQuerySeries.h . . . 2071
gdcmQueryStudy.h . . . 2072
gdcmRoleSelectionSub.h . . . 2074
gdcmServiceClassApplicationInformation.h . . . 2075
gdcmServiceClassUser.h . . . 2077
gdcmSOPClassExtendedNegociationSub.h . . . 2079
gdcmTransferSyntaxSub.h . . . 2080
gdcmULAction.h . . . 2082
gdcmULActionAA.h . . . 2084
gdcmULActionAE.h . . . 2086
gdcmULActionAR.h . . . 2088
gdcmULActionDT.h . . . 2091
gdcmULBasicCallback.h . . . 2092
gdcmULConnection.h . . . 2093
gdcmULConnectionCallback.h . . . 2096
gdcmULConnectionInfo.h . . . 2097
gdcmULConnectionManager.h . . . 2099
gdcmULEvent.h . . . 2102
gdcmULTransitionTable.h . . . 2104
gdcmULWritingCallback.h . . . 2107
gdcmUserInformation.h . . . 2108
gdcmWLMFindQuery.h . . . 2110
vtkGDCMImageReader.h . . . 2111

vtkGDCMImageReader2.h . . . 2117
vtkGDCMImageWriter.h . . . 2122
vtkGDCMMedicalImageProperties.h . . . 2126
vtkGDCMPolyDataReader.h . . . 2131
vtkGDCMPolyDataWriter.h . . . 2133
vtkGDCMTesting.h . . . 2135
vtkGDCMThreadedImageReader.h . . . 2136
vtkGDCMThreadedImageReader2.h . . . 2138
vtkImageColorViewer.h . . . 2140
vtkImageMapToColors16.h . . . 2144
vtkImageMapToWindowLevelColors2.h . . . 2147
vtkImagePlanarComponentsToComponents.h . . . 2149
vtkImageRGBToYBR.h . . . 2150
vtkImageYBRToRGB.h . . . 2152
vtkLookupTable16.h . . . 2153
vtkRTStructSetProperties.h . . . 2155
gdcmPythonFilter.h . . . 2157

Chapter 9

Namespace Documentation

9.1 gdcm Namespace Reference

Namespaces

- namespace network
- namespace SegmentHelper
- namespace terminal

  Class for Terminal.

Classes

- class AbortEvent
- class AnonymizeEvent

  AnonymizeEvent.
- class Anonymizer

  Anonymizer.
- class AnyEvent
- class ApplicationEntity

  ApplicationEntity.
- class ASN1

  Class for ASN1.
- class Attribute

  Attribute class This class use template metaprograming tricks to let the user know when the template instantiation does not match the public dictionary.
- class Attribute< Group, Element, TVR, VM::VM1 >
- class Attribute< Group, Element, TVR, VM::VM1_3 >
- class Attribute< Group, Element, TVR, VM::VM1_8 >
- class Attribute< Group, Element, TVR, VM::VM1_n >
- class Attribute< Group, Element, TVR, VM::VM2_2n >
- class Attribute< Group, Element, TVR, VM::VM2_n >
- class Attribute< Group, Element, TVR, VM::VM3_3n >

- class Attribute< Group, Element, TVR, VM::VM3_n >
- class AudioCodec

  AudioCodec.
- class Base64

  Class for Base64.
- class BaseQuery

  BaseQuery.
- class BaseRootQuery

  BaseRootQuery.
- class BasicOffsetTable

  Class to represent a BasicOffsetTable.
- class Bitmap

  Bitmap class.
- class BitmapToBitmapFilter

  BitmapToBitmapFilter class.
- class BoxRegion

  Class for manipulation box region.
- class ByteBuffer

  ByteBuffer.
- class ByteSwap

  ByteSwap.
- class ByteSwapFilter

  ByteSwapFilter.
- class ByteValue

  Class to represent binary value (array of bytes)
- class CAPICryptoFactory
- class CAPICryptographicMessageSyntax
- class Cleaner

  Cleaner.
- class Codec

  Codec class.
- class Coder

  Coder.
- class CodeString

  CodeString.
- class Command

  Command superclass for callback/observer methods.
- class CommandDataSet

  Class to represent a Command DataSet.
- class CompositeNetworkFunctions

  Composite Network Functions.
- class ConstCharWrapper

  Do not use me.
- class CP246ExplicitDataElement

  Class to read/write a DataElement as CP246Explicit Data Element.
- class CryptoFactory

  Class to do handle the crypto factory.

- class CryptographicMessageSyntax
- class CSAElement

  Class to represent a CSA Element.
- class CSAHeader

  Class for CSAHeader.
- class CSAHeaderDict

  Class to represent a map of CSAHeaderDictEntry.
- class CSAHeaderDictEntry

  Class to represent an Entry in the Dict.
- class CSAHeaderDictException
- class Curve

  Curve class to handle element 50xx,3000 Curve Data.
- class DataElement

  Class to represent a Data Element either Implicit or Explicit.
- class DataElementException
- class DataEvent

  DataEvent.
- class DataSet

  Class to represent a Data Set (which contains Data Elements)
- class DataSetEvent

  DataSetEvent.
- class DataSetHelper

  DataSetHelper (internal class, not intended for user level)
- class Decoder

  Decoder.
- class DefinedTerms

  Defined Terms are used when the specified explicit Values may be extended by implementors to include additional new Values. These new Values shall be specified in the Conformance Statement (see PS 3.2) and shall not have the same meaning as currently defined Values in this standard. A Data Element with Defined Terms that does not contain a Value equivalent to one of the Values currently specified in this standard shall not be considered to have an invalid value. Note: Interpretation Type ID (4008,0210) is an example of a Data Element having Defined Terms. It is defined to have a Value that may be one of the set of standard Values; REPORT or AMENDMENT (see PS 3.3). Because this Data Element has Defined Terms other Interpretation Type IDs may be defined by the implementor.
- class Defs

  FIXME I do not like the name 'Defs'.
- class DeltaEncodingCodec

  DeltaEncodingCodec compression used by some private vendor.
- class DICOMDIR

  DICOMDIR class.
- class DICOMDIRGenerator

  DICOMDIRGenerator class.
- class Dict

  Class to represent a map of DictEntry.
- class DictConverter

  Class to convert a .dic file into something else:
- class DictEntry

  Class to represent an Entry in the Dict.
- class DictPrinter

DictPrinter class.

- class Dicts

  Class to manipulate the sum of knowledge (all the dict user load)
- class DirectionCosines

  class to handle DirectionCosines
- class Directory

  Class for manipulation directories.
- class DirectoryHelper

  DirectoryHelper.
- class DPath

  class to handle a DICOM path While supp 118 did introduced a notion of XPath for XML Native model this convention is too XML-centric. Instead prefer DCMTK style notation `https://groups.google.com/g/comp.protocols.dicom/c/IyIH0IOBMPA`
- class DummyValueGenerator

  Class for generating dummy value.
- class Dumper

  Codec class.
- class Element

  Element class.
- class Element< TVR, VM::VM1_2 >
- class Element< TVR, VM::VM1_n >
- class Element< TVR, VM::VM2_2n >
- class Element< TVR, VM::VM2_n >
- class Element< TVR, VM::VM3_3n >
- class Element< TVR, VM::VM3_4 >
- class Element< TVR, VM::VM3_n >
- class Element< VR::AS, VM::VM5 >
- class Element< VR::OB, VM::VM1 >
- class Element< VR::OW, VM::VM1 >
- class ElementDisableCombinations

  A class which is used to produce compile errors for an invalid combination of template parameters.
- class ElementDisableCombinations< VR::OB, VM::VM1_n >
- class ElementDisableCombinations< VR::OW, VM::VM1_n >
- class EmptyMaskGenerator

  EmptyMaskGenerator Main class to generate a Empty Mask Series from an input Series. This class takes an input folder and generates a series of DICOM files in the specified output directory. This class handles multiples DICOM Series within the same input directory.
- class EncapsulatedDocument

  EncapsulatedDocument.
- class EncodingImplementation

  EncodingImplementation.
- class EncodingImplementation< VR::VRASCII >
- class EncodingImplementation< VR::VRBINARY >
- class EndEvent
- class EnumeratedValues

  Element. A Data Element with Enumerated Values that does not have a Value equivalent to one of the Values specified in this standard has an invalid value within the scope of a specific Information Object/SOP Class definition. Note:
- class EquipmentManufacturer
- class Event

superclass for callback/observer methods

- class Exception

  Exception.
- class ExitEvent
- class ExplicitDataElement

  Class to read/write a DataElement as Explicit Data Element.
- class ExplicitImplicitDataElement

  Class to read/write a DataElement as ExplicitImplicit Data Element.
- class Fiducials

  Fiducials.
- class File

  a DICOM File
- class FileAnonymizer

  FileAnonymizer.
- class FileChangeTransferSyntax

  FileChangeTransferSyntax.
- class FileDecompressLookupTable

  FileDecompressLookupTable class.
- class FileDerivation

  FileDerivation class.
- class FileExplicitFilter

  FileExplicitFilter class.
- class FileMetaInformation

  Class to represent a File Meta Information.
- class Filename

  Class to manipulate file name's.
- class FileNameEvent

  FileNameEvent.
- class FilenameGenerator

  FilenameGenerator.
- class FileSet
- class FileStreamer

  FileStreamer.
- class FileWithName

  FileWithName.
- class FindPatientRootQuery

  PatientRootQuery.
- class FindStudyRootQuery

  FindStudyRootQuery.
- class Fragment

  Class to represent a Fragment.
- class Global

  Global.
- class GroupDict

  Class to represent the mapping from group number to its abbreviation and name.
- class IconImageFilter

  IconImageFilter.

- class IconImageGenerator

  IconImageGenerator.
- struct ignore_char
- class Image

  Image.
- class ImageApplyLookupTable

  ImageApplyLookupTable class.
- class ImageChangePhotometricInterpretation

  ImageChangePhotometricInterpretation class.
- class ImageChangePlanarConfiguration

  ImageChangePlanarConfiguration class.
- class ImageChangeTransferSyntax

  ImageChangeTransferSyntax class.
- class ImageCodec

  ImageCodec.
- class ImageConverter

  Image Converter.
- class ImageFragmentSplitter

  ImageFragmentSplitter class.
- class ImageHelper

  ImageHelper (internal class, not intended for user level)
- class ImageReader

  ImageReader.
- class ImageRegionReader

  ImageRegionReader.
- class ImageToImageFilter

  ImageToImageFilter class.
- class ImageWriter

  ImageWriter.
- class ImplicitDataElement

  Class to represent an Implicit VR Data Element.
- class InitializeEvent
- class IOD

  Class for representing a IOD.
- class IODEntry

  Class for representing a IODEntry.
- class IODs

  Class for representing a IODs.
- class IPPSorter

  IPPSorter.
- class Item

  Class to represent an Item.
- class IterationEvent
- class JPEG12Codec

  Class to do JPEG 12bits (lossy & lossless)
- class JPEG16Codec

  Class to do JPEG 16bits (lossless)

- class JPEG2000Codec

  Class to do JPEG 2000.
- class JPEG8Codec

  Class to do JPEG 8bits (lossy & lossless)
- class JPEGCodec

  JPEG codec.
- class JPEGLSCodec

  JPEG-LS.
- class JSON
- class KAKADUCodec

  KAKADUCodec.
- class LO

  LO.
- class LookupTable

  LookupTable class.
- class Macro

  Class for representing a Macro.
- class Macros

  Class for representing a Modules.
- class MD5

  Class for MD5.
- class MediaStorage

  MediaStorage.
- class MemberCommand

  Command subclass that calls a pointer to a member function.
- class MeshPrimitive

  This class defines surface mesh primitives.
- class ModalityPerformedProcedureStepCreateQuery

  ModalityPerformedProcedureStepCreateQuery.
- class ModalityPerformedProcedureStepSetQuery

  ModalityPerformedProcedureStepSetQuery.
- class ModifiedEvent
- class Module

  Class for representing a Module.
- class ModuleEntry

  Class for representing a ModuleEntry.
- class Modules

  Class for representing a Modules.
- class MovePatientRootQuery

  MovePatientRootQuery.
- class MoveStudyRootQuery

  MoveStudyRootQuery.
- class MrProtocol

  Class for MrProtocol.
- class NestedModuleEntries

  Class for representing a NestedModuleEntries.
- class NoEvent

- class NormalizedNetworkFunctions

  Normalized Network Functions.
- class Object

  Object.
- class OpenSSLCryptoFactory
- class OpenSSLCryptographicMessageSyntax
- class OpenSSLP7CryptoFactory
- class OpenSSLP7CryptographicMessageSyntax
- class Orientation

  class to handle Orientation
- class Overlay

  Overlay class.
- class ParseException

  ParseException Standard exception handling object.
- class Parser

  Parser ala XML_Parser from expat (SAX)
- class Patient

  See PS 3.3 - 2007 DICOM MODEL OF THE REAL-WORLD, p 54.
- class PDBElement

  Class to represent a PDB Element.
- class PDBHeader

  Class for PDBHeader.
- class PDFCodec

  PDFCodec class.
- class PersonName

  PersonName class.
- class PGXCodec

  Class to do PGX.
- class PhotometricInterpretation

  Class to represent an PhotometricInterpretation.
- class PixelFormat

  PixelFormat.
- class Pixmap

  Pixmap class.
- class PixmapReader

  PixmapReader.
- class PixmapToPixmapFilter

  PixmapToPixmapFilter class.
- class PixmapWriter

  PixmapWriter.
- class PNMCodec

  Class to do PNM.
- class Preamble

  DICOM Preamble (Part 10)
- class PresentationContext

  PresentationContext.
- class PresentationContextGenerator

PresentationContextGenerator.

- class Printer

  Printer class.
- class PrivateDict

  Private *Dict*.
- class PrivateTag

  Class to represent a Private DICOM Data *Element* (*Attribute*) *Tag* (Group, *Element*, Owner)
- class ProgressEvent

  ProgressEvent.
- class PVRGCodec

  PVRGCodec.
- class PythonFilter

  PythonFilter *PythonFilter* is the class that make gdcm2.x looks more like gdcm1 and transform the binary blob contained in a *DataElement* into a string, typically this is a nice feature to have for wrapped language.
- class QueryBase

  QueryBase.
- class QueryFactory

  QueryFactory.h.
- class QueryImage

  QueryImage.
- class QueryPatient

  QueryPatient.
- class QuerySeries

  QuerySeries.
- class QueryStudy

  QueryStudy.h.
- class RAWCodec

  RAWCodec class.
- class Reader

  Reader ala DOM (Document *Object* Model)
- struct RealWorldValueMappingContent
- class Region

  Class for manipulation region.
- class Rescaler

  Rescale class.
- class RLECodec

  Class to do RLE.
- class Scanner

  Scanner.
- class Scanner2

  Scanner2.
- class Segment

  This class defines a segment.
- class SegmentedPaletteColorLookupTable

  SegmentedPaletteColorLookupTable class.
- class SegmentReader

  This class defines a segment reader.

- class SegmentWriter

  This class defines a segment writer.
- class SequenceOfFragments

  Class to represent a Sequence Of Fragments.
- class SequenceOfItems

  Class to represent a Sequence Of Items.
- class SerieHelper

  SerieHelper DO NOT USE this class, it is only a temporary solution for ITK migration from GDCM 1.x to GDCM 2.x It will disappear soon, you've been warned.
- class Series

  Series.
- class ServiceClassUser

  ServiceClassUser.
- class SHA1

  Class for SHA1.
- class SimpleMemberCommand

  Command subclass that calls a pointer to a member function.
- class SimpleSubjectWatcher

  SimpleSubjectWatcher.
- class SmartPointer

  Class for Smart Pointer.
- class SOPClassUIDToIOD

  Class convert a class SOP Class UID into IOD.
- class Sorter

  Sorter.
- class Spacing

  Class for Spacing.
- class Spectroscopy

  Spectroscopy class.
- class SplitMosaicFilter

  SplitMosaicFilter class.
- class StartEvent
- struct static_assert_test
- struct STATIC_ASSERTION_FAILURE
- struct STATIC_ASSERTION_FAILURE< true >
- class StreamImageReader

  StreamImageReader.
- class StreamImageWriter

  StreamImageReader.
- class StrictScanner

  StrictScanner.
- class StrictScanner2

  StrictScanner2.
- class String

  String.
- class StringFilter

  StringFilter.

- class Study

  Study.
- class Subject

  Subject.
- class Surface

  This class defines a SURFACE IE.
- class SurfaceHelper

  SurfaceHelper.
- class SurfaceReader

  This class defines a SURFACE IE reader.
- class SurfaceWriter

  This class defines a SURFACE IE writer.
- class SwapCode

  SwapCode representation.
- class SwapperDoOp
- class SwapperNoOp
- class System

  Class to do system operation.
- class Table

  Table.
- class TableEntry

  TableEntry.
- class TableReader

  Class for representing a TableReader.
- class Tag

  Class to represent a DICOM Data Element (Attribute) Tag (Group, Element).
- class TagPath

  class to handle a path of tag.
- class Testing

  class for testing
- class Trace

  Trace.
- class TransferSyntax

  Class to manipulate Transfer Syntax.
- class Type

  Type.
- struct UI
- class UIDGenerator

  Class for generating unique UID.
- class UIDs

  all known uids
- class UNExplicitDataElement

  Class to read/write a DataElement as UNExplicit Data Element.
- class UNExplicitImplicitDataElement

  Class to read/write a DataElement as ExplicitImplicit Data Element.
- class Unpacker12Bits

  Pack/Unpack 12 bits pixel into 16bits.

- class Usage

  Usage.
- class UserEvent
- class UUIDGenerator

  Class for generating unique UUID.
- class Validate

  Validate class.
- class Value

  Class to represent the value of a Data Element.
- class ValueIO

  Class to dispatch template calls.
- class Version

  major/minor and build version
- class VL

  Value Length.
- class VM

  Value Multiplicity Looking at the DICOMV3 dict only there is very few cases: 1 2 3 4 5 6 8 16 24 1-2 1-3 1-8 1-32 1-99 1-n 2-2n 2-n 3-3n 3-n.
- struct VMToLength
- class VR

  VR class.
- class VR16ExplicitDataElement

  Class to read/write a DataElement as Explicit Data Element.
- struct VRToEncoding
- struct VRToType
- class VRVLSize
- class VRVLSize< 0 >
- class VRVLSize< 1 >
- class Waveform

  Waveform class.
- class WLMFindQuery

  PatientRootQuery.
- class Writer

  Writer ala DOM (Document Object Model)
- class XMLDictReader

  Class for representing a XMLDictReader.
- class XMLPrinter
- class XMLPrivateDictReader

  Class for representing a XMLPrivateDictReader.

Typedefs

- typedef String<'\\', 16 > AEComp
- typedef String<'\\', 64 > ASComp
- typedef bool(* BOOL_FUNCTION_PFILE_PFILE_POINTER) (File *, File *)
- typedef String<'\\', 16 > CSComp
- typedef String<'\\', 64 > DAComp

- typedef String<'\\', 64 > DTComp
- typedef std::vector< SmartPointer< FileWithName > > FileList
- typedef Bitmap IconImage
- typedef String<'\\', 64 > LOComp
- typedef String<'\\', 64 > LTComp
- typedef ModuleEntry MacroEntry
- typedef NestedModuleEntries NestedMacroEntries
- typedef String<'\\', 64 > PNComp
- typedef String<'\\', 64 > SHComp
- typedef String<'\\', 64 > STComp
- typedef String<'\\', 16 > TMComp
- typedef String<'\\', 4294967294 > UCComp
- typedef String<'\\', 64, 0 > UIComp
- typedef String<'\\', 4294967294 > URComp
- typedef String<'\\', 64 > UTComp

Enumerations

- enum CompOperators {
  GDCM_EQUAL = 0 ,
  GDCM_DIFFERENT ,
  GDCM_GREATER ,
  GDCM_GREATEROREQUAL ,
  GDCM_LESS ,
  GDCM_LESSOREQUAL }
- enum ECharSet {
  eLatin1 = 0 ,
  eLatin2 ,
  eLatin3 ,
  eLatin4 ,
  eCyrillic ,
  eArabic ,
  eGreek ,
  eHebrew ,
  eLatin5 ,
  eJapanese ,
  eThai ,
  eJapaneseKanjiMultibyte ,
  eJapaneseSupplementaryKanjiMultibyte ,
  eKoreanHangulHanjaMultibyte ,
  eUTF8 ,
  eGB18030 }
- enum ENQueryType {
  eCreateMMPS = 0 ,
  eSetMMPS }
- enum EQueryLevel {
  ePatient = 0 ,
  eStudy = 1 ,
  eSeries = 2 ,
  eImage = 3 }

- enum EQueryType {
  eFind = 0 ,
  eMove ,
  eWLMFind }
- enum ERootType {
  ePatientRootType ,
  eStudyRootType }
- enum LodModeType {
  LD_ALL = 0x00000000 ,
  LD_NOSEQ = 0x00000001 ,
  LD_NOSHADOW = 0x00000002 ,
  LD_NOSHADOWSEQ = 0x00000004 }

Functions

- static int add1 (char *buf, int n)
- ignore_char const backslash ('\\')
- template<typename T >
  static T Clamp (int v)
- static void clean (char *mant)
- static int doround (char *buf, unsigned int n)
- VR::VRType GetVRFromTag (Tag const &tag)
- bool operator!= (const CodeString &ref, const CodeString &cs)
- bool operator!= (const DataElement &lhs, const DataElement &rhs)
- std::ostream & operator<< (std::ostream &_os, const GroupDict &_val)
- std::ostream & operator<< (std::ostream &_os, const IOD &_val)
- std::ostream & operator<< (std::ostream &_os, const IODEntry &_val)
- std::ostream & operator<< (std::ostream &_os, const IODs &_val)
- std::ostream & operator<< (std::ostream &_os, const Macro &_val)
- std::ostream & operator<< (std::ostream &_os, const Macros &_val)
- std::ostream & operator<< (std::ostream &_os, const MediaStorage &ms)
- std::ostream & operator<< (std::ostream &_os, const Module &_val)
- std::ostream & operator<< (std::ostream &_os, const ModuleEntry &_val)
- std::ostream & operator<< (std::ostream &_os, const Modules &_val)
- std::ostream & operator<< (std::ostream &_os, const NestedModuleEntries &_val)
- std::ostream & operator<< (std::ostream &_os, const Tag &_val)
- std::ostream & operator<< (std::ostream &_os, const TransferSyntax &ts)
- std::ostream & operator<< (std::ostream &_os, const Type &val)
- std::ostream & operator<< (std::ostream &_os, const UI &_val)
- std::ostream & operator<< (std::ostream &_os, const UIDs &uid)
- std::ostream & operator<< (std::ostream &_os, const Usage &val)
- std::ostream & operator<< (std::ostream &_os, const VM &_val)
- std::ostream & operator<< (std::ostream &_os, const VR &val)
- std::ostream & operator<< (std::ostream &os, const BasicOffsetTable &val)
- std::ostream & operator<< (std::ostream &os, const CodeString &str)
- std::ostream & operator<< (std::ostream &os, const CommandDataSet &val)
- std::ostream & operator<< (std::ostream &os, const CSAElement &val)
- std::ostream & operator<< (std::ostream &os, const CSAHeader &d)
- std::ostream & operator<< (std::ostream &os, const CSAHeaderDict &val)
- std::ostream & operator<< (std::ostream &os, const CSAHeaderDictEntry &val)

- std::ostream & operator<< (std::ostream &os, const DataElement &val)
- std::ostream & operator<< (std::ostream &os, const DataSet &val)
- std::ostream & operator<< (std::ostream &os, const Dict &val)
- std::ostream & operator<< (std::ostream &os, const DictEntry &val)
- std::ostream & operator<< (std::ostream &os, const Dicts &d)
- std::ostream & operator<< (std::ostream &os, const Directory &d)
- std::ostream & operator<< (std::ostream &os, const DPath &val)
- std::ostream & operator<< (std::ostream &os, const File &val)
- std::ostream & operator<< (std::ostream &os, const FileMetaInformation &val)
- std::ostream & operator<< (std::ostream &os, const FileSet &f)
- std::ostream & operator<< (std::ostream &os, const Fragment &val)
- std::ostream & operator<< (std::ostream &os, const Global &g)
- std::ostream & operator<< (std::ostream &os, const Item &val)
- std::ostream & operator<< (std::ostream &os, const MrProtocol &d)
- std::ostream & operator<< (std::ostream &os, const Object &obj)
- std::ostream & operator<< (std::ostream &os, const Orientation &o)
- std::ostream & operator<< (std::ostream &os, const PDBElement &val)
- std::ostream & operator<< (std::ostream &os, const PDBHeader &d)
- std::ostream & operator<< (std::ostream &os, const PhotometricInterpretation &val)
- std::ostream & operator<< (std::ostream &os, const PixelFormat &pf)
- std::ostream & operator<< (std::ostream &os, const Preamble &val)
- std::ostream & operator<< (std::ostream &os, const PrivateDict &val)
- std::ostream & operator<< (std::ostream &os, const PrivateTag &val)
- std::ostream & operator<< (std::ostream &os, const Region &r)
- std::ostream & operator<< (std::ostream &os, const Scanner &s)
- std::ostream & operator<< (std::ostream &os, const Scanner2 &s)
- std::ostream & operator<< (std::ostream &os, const Sorter &s)
- std::ostream & operator<< (std::ostream &os, const StrictScanner &s)
- std::ostream & operator<< (std::ostream &os, const StrictScanner2 &s)
- std::ostream & operator<< (std::ostream &os, const SwapCode &sc)
- std::ostream & operator<< (std::ostream &os, const Version &v)
- std::ostream & operator<< (std::ostream &os, const VL &val)
- std::ostream & operator<< (std::ostream &os, Event &e)

  Generic inserter operator for Event and its subclasses.
- bool operator== (const CodeString &ref, const CodeString &cs)
- std::istream & operator>> (std::istream &_is, Tag &_val)
- std::istream & operator>> (std::istream &in, ignore_char const &ic)
- template<char TDelimiter, unsigned int TMaxLength, char TPadChar>
  std::istream & operator>> (std::istream &is, String< TDelimiter, TMaxLength, TPadChar > &ms)
- template<typename T >
  static int Round (T x)
- static int roundat (char *buf, unsigned int i, int iexp)
- TYPETOENCODING (SQ, VRBINARY, unsigned char) TYPETOENCODING(UN
- template<typename Float >
  static void x16printf (char *buf, int size, Float f)

Variables

- static Global GlobalInstance
- VRBINARY

9.1.1 Detailed Description

This header defines the classes for the AA Actions, Association Abort Related Actions (Table 9-9 of ps 3.8-2009).

Since each class is essentially a placeholder for a function pointer, I'm breaking with having each class have its own file for the sake of brevity of the number of files.

This header defines the classes for the AE Actions, Association Establishment Related Actions (Table 9-6 of ps 3.↩ 8-2009).

Since each class is essentially a placeholder for a function pointer, I'm breaking with having each class have its own file for the sake of brevity of the number of files.

This header defines the classes for the AR Actions, Association Release Related Actions (Table 9-8 of ps 3.8-2009).

Since each class is essentially a placeholder for a function pointer, I'm breaking with having each class have its own file for the sake of brevity of the number of files.

This header defines the classes for the DT Actions, Data Transfer Related Actions (Table 9-8 of ps 3.8-2009).

Since each class is essentially a placeholder for a function pointer, I'm breaking with having each class have its own file for the sake of brevity of the number of files.

9.1.2 Typedef Documentation

9.1.2.1 AEComp

```
typedef String<'\\',16> gdcm::AEComp
```

9.1.2.2 ASComp

```
typedef String<'\\',64> gdcm::ASComp
```

9.1.2.3 BOOL_FUNCTION_PFILE_PFILE_POINTER

```
typedef bool(* gdcm::BOOL_FUNCTION_PFILE_PFILE_POINTER) (File *, File *)
```

9.1.2.4 CSComp

```
typedef String<'\\',16> gdcm::CSComp
```

9.1.2.5 DAComp

```
typedef String<'\\',64> gdcm::DAComp
```

9.1.2.6 DTComp

```
typedef String<'\\',64> gdcm::DTComp
```

9.1.2.7 FileList

```
typedef std::vector< SmartPointer<FileWithName> > gdcm::FileList
```

9.1.2.8 IconImage

```
typedef Bitmap gdcm::IconImage
```

9.1.2.9 LOComp

```
typedef String<'\\',64> gdcm::LOComp
```

9.1.2.10 LTComp

```
typedef String<'\\',64> gdcm::LTComp
```

9.1.2.11 MacroEntry

```
typedef ModuleEntry gdcm::MacroEntry
```

9.1.2.12 NestedMacroEntries

```
typedef NestedModuleEntries gdcm::NestedMacroEntries
```

9.1.2.13 PNComp

```
typedef String<'\\',64> gdcm::PNComp
```

9.1.2.14 SHComp

```
typedef String<'\\',64> gdcm::SHComp
```

9.1.2.15 STComp

```
typedef String<'\\',64> gdcm::STComp
```

9.1.2.16 TMComp

```
typedef String<'\\',16> gdcm::TMComp
```

9.1.2.17 UCComp

```
typedef String<'\\',4294967294> gdcm::UCComp
```

9.1.2.18 UIComp

```
typedef String<'\\',64,0> gdcm::UIComp
```

9.1.2.19 URComp

```
typedef String<'\\',4294967294> gdcm::URComp
```

9.1.2.20 UTComp

```
typedef String<'\\',64> gdcm::UTComp
```

9.1.3 Enumeration Type Documentation

9.1.3.1 CompOperators

```
enum gdcm::CompOperators
```

Enumerator

GDCM_EQUAL	
GDCM_DIFFERENT	
GDCM_GREATER	
GDCM_GREATEROREQUAL	
GDCM_LESS	
GDCM_LESSOREQUAL	

9.1.3.2 ECharSet

```
enum gdcm::ECharSet
```

The character sets enumerated in PS 3.3 2009 Annex C, section C.12.1.1.2 The resulting character set is stored in 0008,0005 The conversion to the data element is performed by the QueryFactory itself

Enumerator

eLatin1	
eLatin2	
eLatin3	
eLatin4	
eCyrillic	
eArabic	
eGreek	
eHebrew	
eLatin5	
eJapanese	
eThai	
eJapaneseKanjiMultibyte	
eJapaneseSupplementaryKanjiMultibyte	
eKoreanHangulHanjaMultibyte	
eUTF8	
eGB18030	

9.1.3.3 ENQueryType

`enum gdcm::ENQueryType`

Enumerator

eCreateMMPS	
eSetMMPS	

9.1.3.4 EQueryLevel

`enum gdcm::EQueryLevel`

Enumerator

ePatient	
eStudy	
eSeries	
eImage	

9.1.3.5 EQueryType

enum gdcm::EQueryType

Enumerator

eFind	
eMove	
eWLMFind	

9.1.3.6 ERootType

enum gdcm::ERootType

Enumerator

ePatientRootType	
eStudyRootType	

9.1.3.7 LodModeType

enum gdcm::LodModeType

Enumerator

LD_ALL	
LD_NOSEQ	
LD_NOSHADOW	
LD_NOSHADOWSEQ	

9.1.4 Function Documentation

9.1.4.1 add1()

```
static int gdcm::add1 (
            char * buf,
            int n )  [static]
```

References add1().

Referenced by add1(), and doround().

9.1.4.2 backslash()

```
ignore_char const gdcm::backslash (
             '\\'  )
```

Referenced by gdcm::EncodingImplementation< VR::VRASCII >::ReadComputeLength().

9.1.4.3 Clamp()

```
template<typename T >
static T gdcm::Clamp (
             int v )  [inline], [static]
```

9.1.4.4 clean()

```
static void gdcm::clean (
             char * mant )  [inline], [static]
```

Referenced by x16printf().

9.1.4.5 doround()

```
static int gdcm::doround (
             char * buf,
             unsigned int n )  [static]
```

References add1().

Referenced by roundat().

9.1.4.6 GetVRFromTag()

```
VR::VRType gdcm::GetVRFromTag (
            Tag const & tag )
```

9.1.4.7 operator"!=() [1/2]

```
bool gdcm::operator!= (
            const CodeString & ref,
            const CodeString & cs )  [inline]
```

9.1.4.8 operator"!=() [2/2]

```
bool gdcm::operator!= (
            const DataElement & lhs,
            const DataElement & rhs )  [inline]
```

9.1.4.9 operator<<() [1/59]

```
std::ostream & gdcm::operator<< (
            std::ostream & _os,
            const GroupDict & _val )  [inline]
```

9.1.4.10 operator<<() [2/59]

```
std::ostream & gdcm::operator<< (
            std::ostream & _os,
            const IOD & _val )  [inline]
```

9.1.4.11 operator<<() [3/59]

```
std::ostream & gdcm::operator<< (
            std::ostream & _os,
            const IODEntry & _val )  [inline]
```

9.1.4.12 operator<<() [4/59]

```
std::ostream & gdcm::operator<< (
            std::ostream & _os,
            const IODs & _val )  [inline]
```

9.1.4.13 operator<<() [5/59]

```
std::ostream & gdcm::operator<< (
            std::ostream & _os,
            const Macro & _val )  [inline]
```

9.1.4.14 operator<<() [6/59]

```
std::ostream & gdcm::operator<< (
            std::ostream & _os,
            const Macros & _val )  [inline]
```

9.1.4.15 operator<<() [7/59]

```
std::ostream & gdcm::operator<< (
            std::ostream & _os,
            const MediaStorage & ms )  [inline]
```

9.1.4.16 operator<<() [8/59]

```
std::ostream & gdcm::operator<< (
            std::ostream & _os,
            const Module & _val )  [inline]
```

9.1.4.17 operator<<() [9/59]

```
std::ostream & gdcm::operator<< (
            std::ostream & _os,
            const ModuleEntry & _val )  [inline]
```

9.1.4.18 operator<<() [10/59]

```
std::ostream & gdcm::operator<< (
            std::ostream & _os,
            const Modules & _val )  [inline]
```

9.1.4.19 operator<<() [11/59]

```
std::ostream & gdcm::operator<< (
            std::ostream & _os,
            const NestedModuleEntries & _val )  [inline]
```

9.1.4.20 operator<<() [12/59]

```
std::ostream & gdcm::operator<< (
            std::ostream & _os,
            const Tag & _val )  [inline]
```

9.1.4.21 operator<<() [13/59]

```
std::ostream & gdcm::operator<< (
            std::ostream & _os,
            const TransferSyntax & ts )  [inline]
```

9.1.4.22 operator<<() [14/59]

```
std::ostream & gdcm::operator<< (
            std::ostream & _os,
            const Type & val )  [inline]
```

9.1.4.23 operator<<() [15/59]

```
std::ostream & gdcm::operator<< (
            std::ostream & _os,
            const UI & _val )  [inline]
```

9.1.4.24 operator<<() [16/59]

```
std::ostream & gdcm::operator<< (
            std::ostream & _os,
            const UIDs & uid )  [inline]
```

References gdcm::UIDs::GetName(), and gdcm::UIDs::GetString().

9.1.4.25 operator<<() [17/59]

```
std::ostream & gdcm::operator<< (
            std::ostream & _os,
            const Usage & val )  [inline]
```

9.1.4.26 operator<<() [18/59]

```
std::ostream & gdcm::operator<< (
            std::ostream & _os,
            const VM & _val )  [inline]
```

9.1.4.27 operator<<() [19/59]

```
std::ostream & gdcm::operator<< (
            std::ostream & _os,
            const VR & val )  [inline]
```

9.1.4.28 operator<<() [20/59]

```
std::ostream & gdcm::operator<< (
            std::ostream & os,
            const BasicOffsetTable & val )  [inline]
```

9.1.4.29 operator<<() [21/59]

```
std::ostream & gdcm::operator<< (
            std::ostream & os,
            const CodeString & str )  [inline]
```

9.1.4.30 operator<<() [22/59]

```
std::ostream & gdcm::operator<< (
            std::ostream & os,
            const CommandDataSet & val )  [inline]
```

9.1.4.31 operator<<() [23/59]

```
std::ostream & gdcm::operator<< (
            std::ostream & os,
            const CSAElement & val )  [inline]
```

9.1.4.32 operator<<() [24/59]

```
std::ostream & gdcm::operator<< (
            std::ostream & os,
            const CSAHeader & d )  [inline]
```

9.1.4.33 operator<<() [25/59]

```
std::ostream & gdcm::operator<< (
            std::ostream & os,
            const CSAHeaderDict & val )  [inline]
```

9.1.4.34 operator<<() [26/59]

```
std::ostream & gdcm::operator<< (
            std::ostream & os,
            const CSAHeaderDictEntry & val )  [inline]
```

9.1.4.35 operator<<() [27/59]

```
std::ostream & gdcm::operator<< (
            std::ostream & os,
            const DataElement & val )  [inline]
```

9.1.4.36 operator<<() [28/59]

```
std::ostream & gdcm::operator<< (
            std::ostream & os,
            const DataSet & val )  [inline]
```

9.1.4.37 operator<<() [29/59]

```
std::ostream & gdcm::operator<< (
            std::ostream & os,
            const Dict & val )  [inline]
```

9.1.4.38 operator<<() [30/59]

```
std::ostream & gdcm::operator<< (
            std::ostream & os,
            const DictEntry & val )  [inline]
```

9.1.4.39 operator<<() [31/59]

```
std::ostream & gdcm::operator<< (
            std::ostream & os,
            const Dicts & d )  [inline]
```

9.1.4.40 operator<<() [32/59]

```
std::ostream & gdcm::operator<< (
            std::ostream & os,
            const Directory & d )  [inline]
```

9.1.4.41 operator<<() [33/59]

```
std::ostream & gdcm::operator<< (
            std::ostream & os,
            const DPath & val )  [inline]
```

9.1.4.42 operator<<() [34/59]

```
std::ostream & gdcm::operator<< (
            std::ostream & os,
            const File & val )  [inline]
```

9.1.4.43 operator<<() [35/59]

```
std::ostream & gdcm::operator<< (
            std::ostream & os,
            const FileMetaInformation & val )  [inline]
```

9.1.4.44 operator<<() [36/59]

```
std::ostream & gdcm::operator<< (
            std::ostream & os,
            const FileSet & f )  [inline]
```

9.1.4.45 operator<<() [37/59]

```
std::ostream & gdcm::operator<< (
            std::ostream & os,
            const Fragment & val )  [inline]
```

9.1.4.46 operator<<() [38/59]

```
std::ostream & gdcm::operator<< (
            std::ostream & os,
            const Global & g )  [inline]
```

9.1.4.47 operator<<() [39/59]

```
std::ostream & gdcm::operator<< (
            std::ostream & os,
            const Item & val )  [inline]
```

9.1.4.48 operator<<() [40/59]

```
std::ostream & gdcm::operator<< (
            std::ostream & os,
            const MrProtocol & d )  [inline]
```

9.1.4.49 operator<<() [41/59]

```
std::ostream & gdcm::operator<< (
            std::ostream & os,
            const Object & obj )  [inline]
```

9.1.4.50 operator<<() [42/59]

```
std::ostream & gdcm::operator<< (
            std::ostream & os,
            const Orientation & o )  [inline]
```

9.1.4.51 operator<<() [43/59]

```
std::ostream & gdcm::operator<< (
            std::ostream & os,
            const PDBElement & val )  [inline]
```

9.1.4.52 operator<<() [44/59]

```
std::ostream & gdcm::operator<< (
            std::ostream & os,
            const PDBHeader & d )  [inline]
```

9.1.4.53 operator<<() [45/59]

```
std::ostream & gdcm::operator<< (
            std::ostream & os,
            const PhotometricInterpretation & val )  [inline]
```

9.1.4.54 operator<<() [46/59]

```
std::ostream & gdcm::operator<< (
            std::ostream & os,
            const PixelFormat & pf )  [inline]
```

9.1.4.55 operator<<() [47/59]

```
std::ostream & gdcm::operator<< (
            std::ostream & os,
            const Preamble & val )  [inline]
```

9.1.4.56 operator<<() [48/59]

```
std::ostream & gdcm::operator<< (
            std::ostream & os,
            const PrivateDict & val )  [inline]
```

9.1.4.57 operator<<() [49/59]

```
std::ostream & gdcm::operator<< (
            std::ostream & os,
            const PrivateTag & val )  [inline]
```

9.1.4.58 operator<<() [50/59]

```
std::ostream & gdcm::operator<< (
            std::ostream & os,
            const Region & r )  [inline]
```

References gdcm::Region::Print().

9.1.4.59 operator<<() [51/59]

```
std::ostream & gdcm::operator<< (
            std::ostream & os,
            const Scanner & s )  [inline]
```

9.1.4.60 operator<<() [52/59]

```
std::ostream & gdcm::operator<< (
            std::ostream & os,
            const Scanner2 & s )  [inline]
```

9.1.4.61 operator<<() [53/59]

```
std::ostream & gdcm::operator<< (
            std::ostream & os,
            const Sorter & s )  [inline]
```

9.1.4.62 operator<<() [54/59]

```
std::ostream & gdcm::operator<< (
            std::ostream & os,
            const StrictScanner & s )  [inline]
```

9.1.4.63 operator<<() [55/59]

```
std::ostream & gdcm::operator<< (
            std::ostream & os,
            const StrictScanner2 & s )  [inline]
```

9.1.4.64 operator<<() [56/59]

```
std::ostream & gdcm::operator<< (
            std::ostream & os,
            const SwapCode & sc )  [inline]
```

9.1.4.65 operator<<() [57/59]

```
std::ostream & gdcm::operator<< (
            std::ostream & os,
            const Version & v )  [inline]
```

9.1.4.66 operator<<() [58/59]

```
std::ostream & gdcm::operator<< (
            std::ostream & os,
            const VL & val )  [inline]
```

9.1.4.67 operator<<() [59/59]

```
std::ostream & gdcm::operator<< (
            std::ostream & os,
            Event & e )  [inline]
```

Generic inserter operator for Event and its subclasses.

References gdcm::Event::Print().

9.1.4.68 operator==()

```
bool gdcm::operator== (
            const CodeString & ref,
            const CodeString & cs )  [inline]
```

Examples

DumpPhilipsECHO.cxx.

9.1.4.69 operator>>() [1/3]

```
std::istream & gdcm::operator>> (
            std::istream & _is,
            Tag & _val )  [inline]
```

9.1.4.70 operator>>() [2/3]

```
std::istream & gdcm::operator>> (
            std::istream & in,
            ignore_char const & ic )  [inline]
```

References gdcm::ignore_char::m_char.

9.1.4.71 operator>>() [3/3]

```
template<char TDelimiter, unsigned int TMaxLength, char TPadChar>
std::istream & gdcm::operator>> (
            std::istream & is,
            String< TDelimiter, TMaxLength, TPadChar > & ms )  [inline]
```

9.1.4.72 Round()

```
template<typename T >
static int gdcm::Round (
            T x )  [inline], [static]
```

Referenced by gdcm::ImageChangePhotometricInterpretation::RGB2YBR(), and gdcm::ImageChangePhotometricInterpretation::YBR2RG

9.1.4.73 roundat()

```
static int gdcm::roundat (
            char * buf,
            unsigned int i,
            int iexp )  [static]
```

References doround().

Referenced by x16printf().

9.1.4.74 TYPETOENCODING()

```
gdcm::TYPETOENCODING (
            SQ ,
            VRBINARY ,
            unsigned char  )
```

9.1.4.75 x16printf()

```
template<typename Float >
static void gdcm::x16printf (
            char * buf,
            int size,
            Float f )  [static]
```

References clean(), and roundat().

Referenced by gdcm::EncodingImplementation< VR::VRASCII >::Write().

9.1.5 Variable Documentation

9.1.5.1 GlobalInstance

```
Global gdcm::GlobalInstance  [static]
```

9.1.5.2 VRBINARY

```
gdcm::VRBINARY
```

Referenced by gdcm::Element< TVR, VM::VM1_n >::Set(), and gdcm::Element< TVR, VM::VM1_n >::SetNoSwap().

9.2 gdcm::network Namespace Reference

Classes

- class AAbortPDU
  AAbortPDU.
- class AAssociateACPDU
  AAssociateACPDU.
- class AAssociateRJPDU
  AAssociateRJPDU.
- class AAssociateRQPDU
  AAssociateRQPDU.
- class AbstractSyntax
  AbstractSyntax.
- class ApplicationContext
  ApplicationContext.

- class AReleaseRPPDU

  AReleaseRPPDU.
- class AReleaseRQPDU

  AReleaseRQPDU.
- class ARTIMTimer

  ARTIMTimer.
- class AsynchronousOperationsWindowSub

  AsynchronousOperationsWindowSub.
- class BaseCompositeMessage

  BaseCompositeMessage.
- class BaseNormalizedMessage

  BaseNormalizedMessage.
- class BasePDU

  BasePDU.
- class CEchoRQ

  CEchoRQ.
- class CEchoRSP

  CEchoRSP this file defines the messages for the cecho action.
- class CFind
- class CFindCancelRQ

  CFindCancelRQ this file defines the messages for the cfind action.
- class CFindRQ

  CFindRQ.
- class CFindRSP

  CFindRSP this file defines the messages for the cfind action.
- class CMoveCancelRq
- class CMoveRQ

  CMoveRQ.
- class CMoveRSP

  CMoveRSP this file defines the messages for the cmove action.
- class CompositeMessageFactory

  CompositeMessageFactory.
- class CStoreRQ

  CStoreRQ.
- class CStoreRSP

  CStoreRSP this file defines the messages for the cecho action.
- class DIMSE

  DIMSE.
- class ImplementationClassUIDSub

  ImplementationClassUIDSub.
- class ImplementationUIDSub

  ImplementationUIDSub.
- class ImplementationVersionNameSub

  ImplementationVersionNameSub.
- class MaximumLengthSub

  MaximumLengthSub.
- class NActionRQ

NActionRQ.

- class NActionRSP

  NActionRSP this file defines the messages for the NAction action.
- class NCreateRQ

  NCreateRQ.
- class NCreateRSP

  NCreateRSP this file defines the messages for the ncreate action.
- class NDeleteRQ

  NDeleteRQ.
- class NDeleteRSP

  NDeleteRSP this file defines the messages for the ndelete action.
- class NEventReportRQ

  NEventReportRQ.
- class NEventReportRSP

  NEventReportRSP this file defines the messages for the neventreport action.
- class NGetRQ

  NGetRQ.
- class NGetRSP

  NGetRSP this file defines the messages for the nget action.
- class NormalizedMessageFactory
- class NSetRQ

  NSetRQ.
- class NSetRSP

  NSetRSP this file defines the messages for the nset action.
- class PDataTFPDU

  PDataTFPDU.
- class PDUFactory

  PDUFactory basically, given an initial byte, construct the.
- class PresentationContextAC

  PresentationContextAC.
- class PresentationContextRQ

  PresentationContextRQ.
- class PresentationDataValue

  PresentationDataValue.
- class RoleSelectionSub

  RoleSelectionSub.
- class ServiceClassApplicationInformation
- class SOPClassExtendedNegociationSub

  SOPClassExtendedNegociationSub.
- class TableRow
- class TransferSyntaxSub

  TransferSyntaxSub.
- struct Transition
- class ULAction

  ULAction.
- class ULActionAA1
- class ULActionAA2

- class ULActionAA3
- class ULActionAA4
- class ULActionAA5
- class ULActionAA6
- class ULActionAA7
- class ULActionAA8
- class ULActionAE1
- class ULActionAE2
- class ULActionAE3
- class ULActionAE4
- class ULActionAE5
- class ULActionAE6
- class ULActionAE7
- class ULActionAE8
- class ULActionAR1
- class ULActionAR10
- class ULActionAR2
- class ULActionAR3
- class ULActionAR4
- class ULActionAR5
- class ULActionAR6
- class ULActionAR7
- class ULActionAR8
- class ULActionAR9
- class ULActionDT1
- class ULActionDT2
- class ULBasicCallback

  ULBasicCallback.
- class ULConnection

  ULConnection.
- class ULConnectionCallback
- class ULConnectionInfo

  ULConnectionInfo.
- class ULConnectionManager

  ULConnectionManager.
- class ULEvent

  ULEvent.
- class ULTransitionTable

  ULTransitionTable The transition table of all the ULEvents, new ULActions, and ULStates.
- class ULWritingCallback
- class UserInformation

  UserInformation.

Enumerations

- enum EEventID {
  eAASSOCIATERequestLocalUser = 0 ,
  eTransportConnConfirmLocal ,
  eASSOCIATE_ACPDUreceived ,
  eASSOCIATE_RJPDUreceived ,
  eTransportConnIndicLocal ,
  eAASSOCIATE_RQPDUreceived ,
  eAASSOCIATEresponseAccept ,
  eAASSOCIATEresponseReject ,
  ePDATArequest ,
  ePDATATFPDU ,
  eARELEASERequest ,
  eARELEASE_RQPDUReceivedOpen ,
  eARELEASE_RPPDUReceived ,
  eARELEASEResponse ,
  eAABORTRequest ,
  eAABORTPDUReceivedOpen ,
  eTransportConnectionClosed ,
  eARTIMTimerExpired ,
  eUnrecognizedPDUReceived ,
  eEventDoesNotExist }
- enum EStateID {
  eStaDoesNotExist = 0 ,
  eSta1Idle = 1 ,
  eSta2Open = 2 ,
  eSta3WaitLocalAssoc = 4 ,
  eSta4LocalAssocDone = 8 ,
  eSta5WaitRemoteAssoc = 16 ,
  eSta6TransferReady = 32 ,
  eSta7WaitRelease = 64 ,
  eSta8WaitLocalRelease = 128 ,
  eSta9ReleaseCollisionRqLocal = 256 ,
  eSta10ReleaseCollisionAc = 512 ,
  eSta11ReleaseCollisionRq = 1024 ,
  eSta12ReleaseCollisionAcLocal = 2048 ,
  eSta13AwaitingClose = 4096 }

Functions

- int GetStateIndex (EStateID inState)

Variables

- const int cMaxEventID = eEventDoesNotExist
- const int cMaxStateID = 13

9.2.1 Enumeration Type Documentation

9.2.1.1 EEventID

enum gdcm::network::EEventID

Enumerator

eAASSOCIATERequestLocalUser	
eTransportConnConfirmLocal	
eASSOCIATE_ACPDUreceived	
eASSOCIATE_RJPDUreceived	
eTransportConnIndicLocal	
eAASSOCIATE_RQPDUreceived	
eAASSOCIATEresponseAccept	
eAASSOCIATEresponseReject	
ePDATArequest	
ePDATATFPDU	
eARELEASERequest	
eARELEASE_RQPDUReceivedOpen	
eARELEASE_RPPDUReceived	
eARELEASEResponse	
eAABORTRequest	
eAABORTPDUReceivedOpen	
eTransportConnectionClosed	
eARTIMTimerExpired	
eUnrecognizedPDUReceived	
eEventDoesNotExist	

9.2.1.2 EStateID

enum gdcm::network::EStateID

Each network connection will be in a particular state at any given time. Those states have IDs as described in the standard ps3.8-2009, roughly 1-13. This enumeration lists those states. The actual ULState class will contain more information about transitions to other states.

name and date: 16 sept 2010 mmr

Enumerator

eStaDoesNotExist	
eSta1Idle	
eSta2Open	
eSta3WaitLocalAssoc	
eSta4LocalAssocDone	
eSta5WaitRemoteAssoc	

Enumerator

eSta6TransferReady	
eSta7WaitRelease	
eSta8WaitLocalRelease	
eSta9ReleaseCollisionRqLocal	
eSta10ReleaseCollisionAc	
eSta11ReleaseCollisionRq	
eSta12ReleaseCollisionAcLocal	
eSta13AwaitingClose	

9.2.2 Function Documentation

9.2.2.1 GetStateIndex()

```
int gdcm::network::GetStateIndex (
            EStateID inState )  [inline]
```

References eSta10ReleaseCollisionAc, eSta11ReleaseCollisionRq, eSta12ReleaseCollisionAcLocal, eSta13AwaitingClose, eSta1Idle, eSta2Open, eSta3WaitLocalAssoc, eSta4LocalAssocDone, eSta5WaitRemoteAssoc, eSta6TransferReady, eSta7WaitRelease, eSta8WaitLocalRelease, eSta9ReleaseCollisionRqLocal, and eStaDoesNotExist.

9.2.3 Variable Documentation

9.2.3.1 cMaxEventID

```
const int gdcm::network::cMaxEventID = eEventDoesNotExist
```

9.2.3.2 cMaxStateID

```
const int gdcm::network::cMaxStateID = 13
```

Referenced by gdcm::network::TableRow::TableRow(), and gdcm::network::TableRow::~TableRow().

9.3 gdcm::SegmentHelper Namespace Reference

Classes

- struct BasicCodedEntry

  This structure defines a basic coded entry with all of its attributes.

9.4 gdcm::terminal Namespace Reference

Class for Terminal.

Enumerations

- enum Attribute {
  reset = 0 ,
  bright = 1 ,
  dim = 2 ,
  underline = 3 ,
  blink = 5 ,
  reverse = 7 ,
  hidden = 8 }
- enum Color {
  black = 0 ,
  red ,
  green ,
  yellow ,
  blue ,
  magenta ,
  cyan ,
  white }
- enum Mode {
  CONSOLE = 0 ,
  VT100 }

Functions

- GDCM_EXPORT std::string setattribute (Attribute att)
- GDCM_EXPORT std::string setbgcolor (Color c)
- GDCM_EXPORT std::string setfgcolor (Color c)
- GDCM_EXPORT void setmode (Mode m)

9.4.1 Detailed Description

Class for Terminal.

Allow one to print in color in a shell

- support VT100 compatible shell
- win32 console

9.4.2 Enumeration Type Documentation

9.4.2.1 Attribute

enum gdcm::terminal::Attribute

Enumerator

reset	
bright	
dim	
underline	
blink	
reverse	
hidden	

9.4.2.2 Color

enum gdcm::terminal::Color

Enumerator

black	
red	
green	
yellow	
blue	
magenta	
cyan	
white	

9.4.2.3 Mode

enum gdcm::terminal::Mode

Enumerator

CONSOLE	
VT100	

9.4.3 Function Documentation

9.4.3.1 setattribute()

```
GDCM_EXPORT std::string gdcm::terminal::setattribute (
            Attribute att )
```

9.4.3.2 setbgcolor()

```
GDCM_EXPORT std::string gdcm::terminal::setbgcolor (
            Color c )
```

9.4.3.3 setfgcolor()

```
GDCM_EXPORT std::string gdcm::terminal::setfgcolor (
            Color c )
```

9.4.3.4 setmode()

```
GDCM_EXPORT void gdcm::terminal::setmode (
            Mode m )
```

Chapter 10

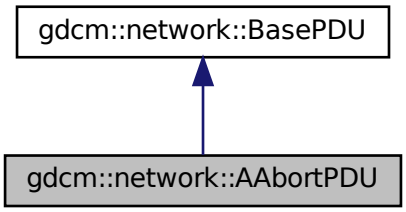
Class Documentation

10.1 gdcm::network::AAbortPDU Class Reference

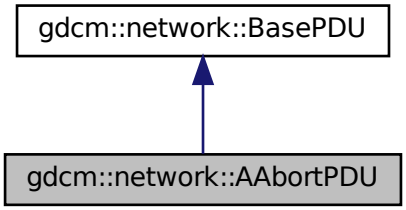
AAbortPDU.

`#include <gdcmAAbortPDU.h>`

Inheritance diagram for gdcm::network::AAbortPDU:



Collaboration diagram for gdcm::network::AAbortPDU:



Public Member Functions

- AAbortPDU ()
- bool IsLastFragment () const override
- void Print (std::ostream &os) const override
- std::istream & Read (std::istream &is) override
- void SetReason (const uint8_t r)
- void SetSource (const uint8_t s)
- size_t Size () const override
- const std::ostream & Write (std::ostream &os) const override

10.1.1 Detailed Description

AAbortPDU.

Table 9-26 A-ABORT PDU FIELDS

10.1.2 Constructor & Destructor Documentation

10.1.2.1 AAbortPDU()

```
gdcm::network::AAbortPDU::AAbortPDU ( )
```

10.1.3 Member Function Documentation

10.1.3.1 IsLastFragment()

```
bool gdcm::network::AAbortPDU::IsLastFragment ( ) const  [inline], [override], [virtual]
```

Implements gdcm::network::BasePDU.

10.1.3.2 Print()

```
void gdcm::network::AAbortPDU::Print (
            std::ostream & os ) const  [override], [virtual]
```

Implements gdcm::network::BasePDU.

10.1.3.3 Read()

```
std::istream & gdcm::network::AAbortPDU::Read (
            std::istream & is )  [override], [virtual]
```

Implements gdcm::network::BasePDU.

10.1.3.4 SetReason()

```
void gdcm::network::AAbortPDU::SetReason (
            const uint8_t r )
```

10.1.3.5 SetSource()

```
void gdcm::network::AAbortPDU::SetSource (
            const uint8_t s )
```

10.1.3.6 Size()

```
size_t gdcm::network::AAbortPDU::Size ( ) const  [override], [virtual]
```

Implements gdcm::network::BasePDU.

10.1.3.7 Write()

```
const std::ostream & gdcm::network::AAbortPDU::Write (
            std::ostream & os ) const  [override], [virtual]
```

Implements gdcm::network::BasePDU.

The documentation for this class was generated from the following file:

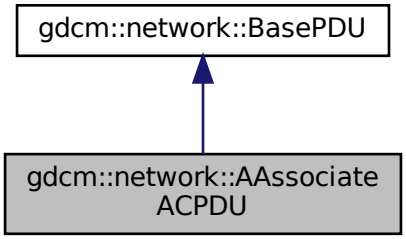
- gdcmAAbortPDU.h

10.2 gdcm::network::AAssociateACPDU Class Reference

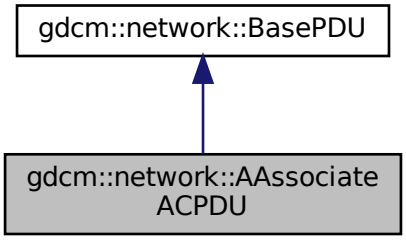
AAssociateACPDU.

`#include <gdcmAAssociateACPDU.h>`

Inheritance diagram for gdcm::network::AAssociateACPDU:



Collaboration diagram for gdcm::network::AAssociateACPDU:



Public Types

- typedef std::vector< PresentationContextAC >::size_type SizeType

Public Member Functions

- AAssociateACPDU ()
- void AddPresentationContextAC (PresentationContextAC const &pcac)
- SizeType GetNumberOfPresentationContextAC () const
- const PresentationContextAC & GetPresentationContextAC (SizeType i)
- const UserInformation & GetUserInformation () const
- void InitFromRQ (AAssociateRQPDU const &rqpdu)
- bool IsLastFragment () const override
- void Print (std::ostream &os) const override
- std::istream & Read (std::istream &is) override
- SizeType Size () const override
- const std::ostream & Write (std::ostream &os) const override

Protected Member Functions

- void SetCalledAETitle (const char calledaetitle[16])
- void SetCallingAETitle (const char callingaetitle[16])

Friends

- class AAssociateRQPDU

10.2.1 Detailed Description

AAssociateACPDU.

Table 9-17 ASSOCIATE-AC PDU fields

10.2.2 Member Typedef Documentation

10.2.2.1 SizeType

```
typedef std::vector<PresentationContextAC>::size_type gdcm::network::AAssociateACPDU::SizeType
```

10.2.3 Constructor & Destructor Documentation

10.2.3.1 AAssociateACPDU()

```
gdcm::network::AAssociateACPDU::AAssociateACPDU ( )
```

10.2.4 Member Function Documentation

10.2.4.1 AddPresentationContextAC()

```
void gdcm::network::AAssociateACPDU::AddPresentationContextAC (
            PresentationContextAC const & pcac )
```

10.2.4.2 GetNumberOfPresentationContextAC()

```
SizeType gdcm::network::AAssociateACPDU::GetNumberOfPresentationContextAC ( ) const  [inline]
```

10.2.4.3 GetPresentationContextAC()

```
const PresentationContextAC & gdcm::network::AAssociateACPDU::GetPresentationContextAC (
            SizeType i )  [inline]
```

10.2.4.4 GetUserInformation()

```
const UserInformation & gdcm::network::AAssociateACPDU::GetUserInformation ( ) const  [inline]
```

10.2.4.5 InitFromRQ()

```
void gdcm::network::AAssociateACPDU::InitFromRQ (
            AAssociateRQPDU const & rqpdu )
```

10.2.4.6 IsLastFragment()

```
bool gdcm::network::AAssociateACPDU::IsLastFragment ( ) const  [inline], [override], [virtual]
```

Implements gdcm::network::BasePDU.

10.2.4.7 Print()

```
void gdcm::network::AAssociateACPDU::Print (
            std::ostream & os ) const  [override], [virtual]
```

Implements gdcm::network::BasePDU.

10.2.4.8 Read()

```
std::istream & gdcm::network::AAssociateACPDU::Read (
            std::istream & is )  [override], [virtual]
```

Implements gdcm::network::BasePDU.

10.2.4.9 SetCalledAETitle()

```
void gdcm::network::AAssociateACPDU::SetCalledAETitle (
            const char calledaetitle[16] )  [protected]
```

10.2.4.10 SetCallingAETitle()

```
void gdcm::network::AAssociateACPDU::SetCallingAETitle (
            const char callingaetitle[16] )  [protected]
```

10.2.4.11 Size()

```
SizeType gdcm::network::AAssociateACPDU::Size ( ) const  [override], [virtual]
```

Implements gdcm::network::BasePDU.

10.2.4.12 Write()

```
const std::ostream & gdcm::network::AAssociateACPDU::Write (
            std::ostream & os ) const  [override], [virtual]
```

Implements gdcm::network::BasePDU.

10.2.5 Friends And Related Function Documentation

10.2.5.1 AAssociateRQPDU

```
friend class AAssociateRQPDU  [friend]
```

The documentation for this class was generated from the following file:

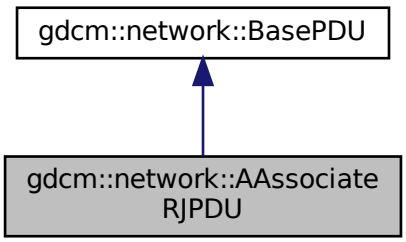
- gdcmAAssociateACPDU.h

10.3 gdcm::network::AAssociateRJPDU Class Reference

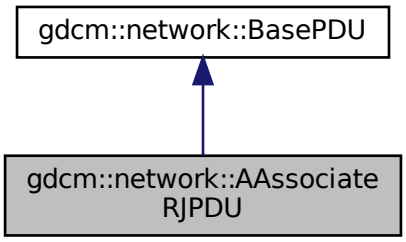
AAssociateRJPDU.

```
#include <gdcmAAssociateRJPDU.h>
```

Inheritance diagram for gdcm::network::AAssociateRJPDU:



Collaboration diagram for gdcm::network::AAssociateRJPDU:



Public Member Functions

- AAssociateRJPDU ()
- bool IsLastFragment () const override
- void Print (std::ostream &os) const override
- std::istream & Read (std::istream &is) override
- size_t Size () const override
- const std::ostream & Write (std::ostream &os) const override

10.3.1 Detailed Description

AAssociateRJPDU.

Table 9-21 ASSOCIATE-RJ PDU FIELDS

10.3.2 Constructor & Destructor Documentation

10.3.2.1 AAssociateRJPDU()

```
gdcm::network::AAssociateRJPDU::AAssociateRJPDU ( )
```

10.3.3 Member Function Documentation

10.3.3.1 IsLastFragment()

```
bool gdcm::network::AAssociateRJPDU::IsLastFragment ( ) const  [inline], [override], [virtual]
```

Implements gdcm::network::BasePDU.

10.3.3.2 Print()

```
void gdcm::network::AAssociateRJPDU::Print (
            std::ostream & os ) const  [override], [virtual]
```

Implements gdcm::network::BasePDU.

10.3.3.3 Read()

```
std::istream & gdcm::network::AAssociateRJPDU::Read (
            std::istream & is )  [override], [virtual]
```

Implements gdcm::network::BasePDU.

10.3.3.4 Size()

```
size_t gdcm::network::AAssociateRJPDU::Size ( ) const  [override], [virtual]
```

Implements gdcm::network::BasePDU.

10.3.3.5 Write()

```
const std::ostream & gdcm::network::AAssociateRJPDU::Write (
            std::ostream & os ) const  [override], [virtual]
```

Implements gdcm::network::BasePDU.

The documentation for this class was generated from the following file:

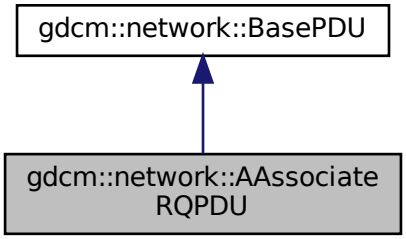
- gdcmAAssociateRJPDU.h

10.4 gdcm::network::AAssociateRQPDU Class Reference

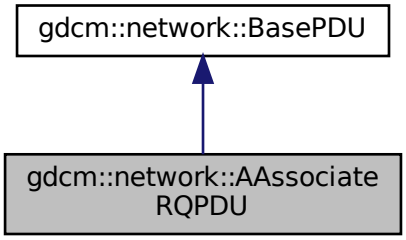
AAssociateRQPDU.

`#include <gdcmAAssociateRQPDU.h>`

Inheritance diagram for gdcm::network::AAssociateRQPDU:



Collaboration diagram for gdcm::network::AAssociateRQPDU:



Public Types

- typedef std::vector< PresentationContextRQ > PresentationContextArrayType
- typedef std::vector< PresentationContextRQ >::size_type SizeType

Public Member Functions

- AAssociateRQPDU ()
- AAssociateRQPDU (const AAssociateRQPDU &pdu)
- void AddPresentationContext (PresentationContextRQ const &pc)
- std::string GetCalledAETitle () const
- std::string GetCallingAETitle () const
- SizeType GetNumberOfPresentationContext () const
- PresentationContextRQ const & GetPresentationContext (SizeType i) const
- const PresentationContextRQ * GetPresentationContextByAbstractSyntax (AbstractSyntax const &absyn) const
- const PresentationContextRQ * GetPresentationContextByID (uint8_t i) const
- PresentationContextArrayType const & GetPresentationContexts ()
- const UserInformation & GetUserInformation () const
- bool IsLastFragment () const override
- void Print (std::ostream &os) const override
- std::istream & Read (std::istream &is) override
- void SetCalledAETitle (const char calledaetitle[16])

  Set the Called AE Title.
- void SetCallingAETitle (const char callingaetitle[16])

  Set the Calling AE Title.
- void SetUserInformation (UserInformation const &ui)
- size_t Size () const override
- const std::ostream & Write (std::ostream &os) const override

Static Public Member Functions

- static bool IsAETitleValid (const char title[16])

  Check whether or not the.

Protected Member Functions

- std::string GetReserved43_74 () const

Friends

- class AAssociateACPDU

10.4.1 Detailed Description

AAssociateRQPDU.

Table 9-11 ASSOCIATE-RQ PDU fields

10.4.2 Member Typedef Documentation

10.4.2.1 PresentationContextArrayType

```
typedef std::vector<PresentationContextRQ> gdcm::network::AAssociateRQPDU::PresentationContextArrayType
```

10.4.2.2 SizeType

```
typedef std::vector<PresentationContextRQ>::size_type gdcm::network::AAssociateRQPDU::SizeType
```

10.4.3 Constructor & Destructor Documentation

10.4.3.1 AAssociateRQPDU() [1/2]

```
gdcm::network::AAssociateRQPDU::AAssociateRQPDU ( )
```

10.4.3.2 AAssociateRQPDU() [2/2]

```
gdcm::network::AAssociateRQPDU::AAssociateRQPDU (
            const AAssociateRQPDU & pdu )  [inline]
```

10.4.4 Member Function Documentation

10.4.4.1 AddPresentationContext()

```
void gdcm::network::AAssociateRQPDU::AddPresentationContext (
            PresentationContextRQ const & pc )
```

10.4.4.2 GetCalledAETitle()

```
std::string gdcm::network::AAssociateRQPDU::GetCalledAETitle ( ) const  [inline]
```

10.4.4.3 GetCallingAETitle()

```
std::string gdcm::network::AAssociateRQPDU::GetCallingAETitle ( ) const  [inline]
```

10.4.4.4 GetNumberOfPresentationContext()

```
SizeType gdcm::network::AAssociateRQPDU::GetNumberOfPresentationContext ( ) const  [inline]
```

10.4.4.5 GetPresentationContext()

```
PresentationContextRQ const & gdcm::network::AAssociateRQPDU::GetPresentationContext (
            SizeType i ) const  [inline]
```

10.4.4.6 GetPresentationContextByAbstractSyntax()

```
const PresentationContextRQ * gdcm::network::AAssociateRQPDU::GetPresentationContextByAbstract↩
Syntax (
            AbstractSyntax const & absyn ) const
```

10.4.4.7 GetPresentationContextByID()

```
const PresentationContextRQ * gdcm::network::AAssociateRQPDU::GetPresentationContextByID (
            uint8_t i ) const
```

10.4.4.8 GetPresentationContexts()

```
PresentationContextArrayType const & gdcm::network::AAssociateRQPDU::GetPresentationContexts ( )
[inline]
```

10.4.4.9 GetReserved43_74()

```
std::string gdcm::network::AAssociateRQPDU::GetReserved43_74 ( ) const  [protected]
```

10.4.4.10 GetUserInformation()

```
const UserInformation & gdcm::network::AAssociateRQPDU::GetUserInformation ( ) const  [inline]
```

10.4.4.11 IsAETitleValid()

```
static bool gdcm::network::AAssociateRQPDU::IsAETitleValid (
            const char title[16] )  [static]
```

Check whether or not the.

Parameters

title	is a valid AE title

10.4.4.12 IsLastFragment()

```
bool gdcm::network::AAssociateRQPDU::IsLastFragment ( ) const  [inline], [override], [virtual]
```

Implements gdcm::network::BasePDU.

10.4.4.13 Print()

```
void gdcm::network::AAssociateRQPDU::Print (
            std::ostream & os ) const  [override], [virtual]
```

This function will initialize an AAssociateACPDU from the fields in the AAssociateRQPDU structure

Implements gdcm::network::BasePDU.

10.4.4.14 Read()

```
std::istream & gdcm::network::AAssociateRQPDU::Read (
            std::istream & is )  [override], [virtual]
```

Implements gdcm::network::BasePDU.

10.4.4.15 SetCalledAETitle()

```
void gdcm::network::AAssociateRQPDU::SetCalledAETitle (
            const char calledaetitle[16] )
```

Set the Called AE Title.

10.4.4.16 SetCallingAETitle()

```
void gdcm::network::AAssociateRQPDU::SetCallingAETitle (
            const char callingaetitle[16] )
```

Set the Calling AE Title.

10.4.4.17 SetUserInformation()

```
void gdcm::network::AAssociateRQPDU::SetUserInformation (
            UserInformation const & ui )
```

10.4.4.18 Size()

```
size_t gdcm::network::AAssociateRQPDU::Size ( ) const  [override], [virtual]
```

Implements gdcm::network::BasePDU.

10.4.4.19 Write()

```
const std::ostream & gdcm::network::AAssociateRQPDU::Write (
            std::ostream & os ) const  [override], [virtual]
```

Implements gdcm::network::BasePDU.

10.4.5 Friends And Related Function Documentation

10.4.5.1 AAssociateACPDU

```
friend class AAssociateACPDU  [friend]
```

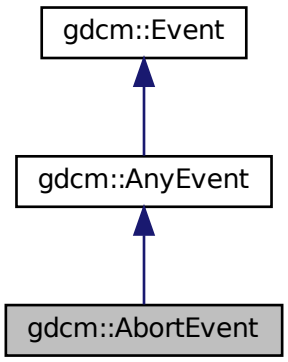
The documentation for this class was generated from the following file:

- gdcmAAssociateRQPDU.h

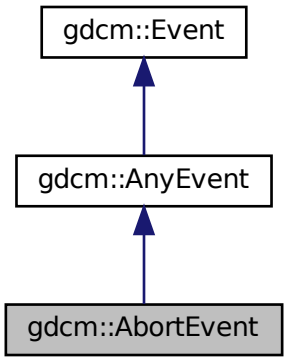
10.5 gdcm::AbortEvent Class Reference

```
#include <gdcmEvent.h>
```

Inheritance diagram for gdcm::AbortEvent:



Collaboration diagram for gdcm::AbortEvent:



Additional Inherited Members

The documentation for this class was generated from the following file:

- gdcmEvent.h

10.6 gdcm::network::AbstractSyntax Class Reference

AbstractSyntax.

`#include <gdcmAbstractSyntax.h>`

Public Member Functions

- AbstractSyntax ()
- DataElement GetAsDataElement () const
- const char * GetName () const
- bool operator== (const AbstractSyntax &as) const
- void Print (std::ostream &os) const
- std::istream & Read (std::istream &is)
- void SetName (const char *name)
- void SetNameFromUID (UIDs::TSName tsname)
- size_t Size () const
- const std::ostream & Write (std::ostream &os) const

10.6.1 Detailed Description

AbstractSyntax.

Table 9-14 ABSTRACT SYNTAX SUB-ITEM FIELDS

10.6.2 Constructor & Destructor Documentation

10.6.2.1 AbstractSyntax()

```
gdcm::network::AbstractSyntax::AbstractSyntax ( )
```

10.6.3 Member Function Documentation

10.6.3.1 GetAsDataElement()

```
DataElement gdcm::network::AbstractSyntax::GetAsDataElement ( ) const
```

10.6.3.2 GetName()

```
const char * gdcm::network::AbstractSyntax::GetName ( ) const  [inline]
```

10.6.3.3 operator==()

```
bool gdcm::network::AbstractSyntax::operator== (
            const AbstractSyntax & as ) const  [inline]
```

10.6.3.4 Print()

```
void gdcm::network::AbstractSyntax::Print (
            std::ostream & os ) const
```

10.6.3.5 Read()

```
std::istream & gdcm::network::AbstractSyntax::Read (
            std::istream & is )
```

10.6.3.6 SetName()

```
void gdcm::network::AbstractSyntax::SetName (
            const char * name )  [inline]
```

10.6.3.7 SetNameFromUID()

```
void gdcm::network::AbstractSyntax::SetNameFromUID (
            UIDs::TSName tsname )
```

10.6.3.8 Size()

```
size_t gdcm::network::AbstractSyntax::Size ( ) const
```

10.6.3.9 Write()

```
const std::ostream & gdcm::network::AbstractSyntax::Write (
            std::ostream & os ) const
```

The documentation for this class was generated from the following file:

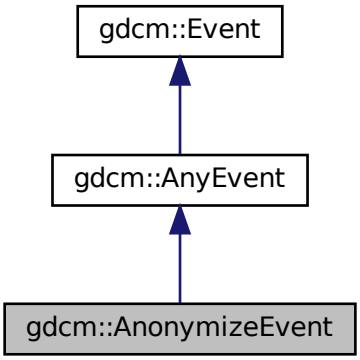
- gdcmAbstractSyntax.h

10.7 gdcm::AnonymizeEvent Class Reference

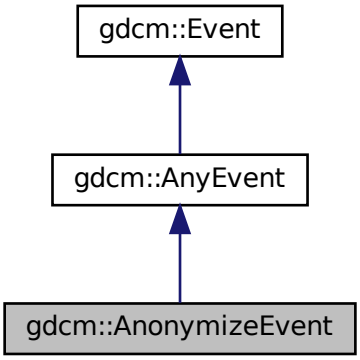
AnonymizeEvent.

`#include <gdcmAnonymizeEvent.h>`

Inheritance diagram for gdcm::AnonymizeEvent:



Collaboration diagram for gdcm::AnonymizeEvent:



Public Types

- typedef AnonymizeEvent Self
- typedef AnyEvent Superclass

Public Member Functions

- AnonymizeEvent (const Self &s)
- AnonymizeEvent (Tag const &tag=0)
- ~AnonymizeEvent () override=default
- bool CheckEvent (const ::gdcm::Event *e) const override
- const char * GetEventName () const override
- Tag const & GetTag () const
- ::gdcm::Event * MakeObject () const override
- void operator= (const Self &)=delete
- void SetTag (const Tag &t)

10.7.1 Detailed Description

AnonymizeEvent.

Special type of event triggered during the Anonymization process

See also

Anonymizer

Examples

BasicAnonymizer.cs, and ClinicalTrialIdentificationWorkflow.cs.

10.7.2 Member Typedef Documentation

10.7.2.1 Self

```
typedef AnonymizeEvent gdcm::AnonymizeEvent::Self
```

10.7.2.2 Superclass

```
typedef AnyEvent gdcm::AnonymizeEvent::Superclass
```

10.7.3 Constructor & Destructor Documentation

10.7.3.1 AnonymizeEvent() [1/2]

```
gdcm::AnonymizeEvent::AnonymizeEvent (
            Tag const & tag = 0 )  [inline]
```

10.7.3.2 ~AnonymizeEvent()

```
gdcm::AnonymizeEvent::~AnonymizeEvent ( )  [override], [default]
```

10.7.3.3 AnonymizeEvent() [2/2]

```
gdcm::AnonymizeEvent::AnonymizeEvent (
            const Self & s )  [inline]
```

10.7.4 Member Function Documentation

10.7.4.1 CheckEvent()

```
bool gdcm::AnonymizeEvent::CheckEvent (
            const ::gdcm::Event * e ) const  [inline], [override]
```

10.7.4.2 GetEventName()

```
const char * gdcm::AnonymizeEvent::GetEventName ( ) const  [inline], [override], [virtual]
```

Return the StringName associated with the event.

Implements gdcm::Event.

10.7.4.3 GetTag()

```
Tag const & gdcm::AnonymizeEvent::GetTag ( ) const  [inline]
```

Examples

BasicAnonymizer.cs, and ClinicalTrialIdentificationWorkflow.cs.

10.7.4.4 MakeObject()

```
::gdcm::Event * gdcm::AnonymizeEvent::MakeObject ( ) const  [inline], [override], [virtual]
```

Create an Event of this type This method work as a Factory for creating events of each particular type.

Implements gdcm::Event.

10.7.4.5 operator=()

```
void gdcm::AnonymizeEvent::operator= (
            const Self &  )  [delete]
```

10.7.4.6 SetTag()

```
void gdcm::AnonymizeEvent::SetTag (
            const Tag & t )  [inline]
```

The documentation for this class was generated from the following file:

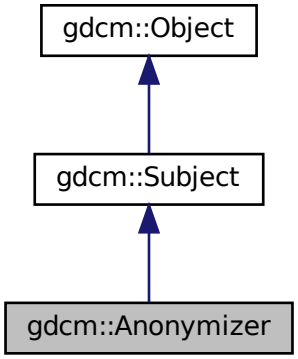
- gdcmAnonymizeEvent.h

10.8 gdcm::Anonymizer Class Reference

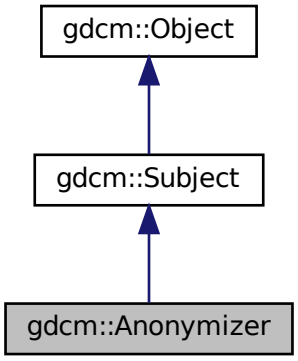
Anonymizer.

`#include <gdcmAnonymizer.h>`

Inheritance diagram for gdcm::Anonymizer:



Collaboration diagram for gdcm::Anonymizer:



Public Member Functions

- Anonymizer ()
- ~Anonymizer () override
- bool BasicApplicationLevelConfidentialityProfile (bool deidentify=true)
- bool Clear (PrivateTag const &pt)
- bool Clear (Tag const &t)

  Identical to 'Empty' except no action is done when tag is not present.
- bool Empty (PrivateTag const &pt)
- bool Empty (Tag const &t)

  Make Tag t empty (if not found tag will be created)
- const CryptographicMessageSyntax * GetCryptographicMessageSyntax () const
- File & GetFile ()
- bool Remove (PrivateTag const &pt)
- bool Remove (Tag const &t)

  remove a tag (even a SQ can be removed)
- bool RemoveGroupLength ()

  Main function that loop over all elements and remove group length.
- bool RemovePrivateTags ()

  Main function that loop over all elements and remove private tags.
- bool RemoveRetired ()

  Main function that loop over all elements and remove retired element.
- bool Replace (PrivateTag const &t, const char *value)
- bool Replace (PrivateTag const &t, const char *value, VL const &vl)
- bool Replace (Tag const &t, const char *value)
- bool Replace (Tag const &t, const char *value, VL const &vl)
- void SetCryptographicMessageSyntax (CryptographicMessageSyntax *cms)

  Set/Get CMS key that will be used to encrypt the dataset within BasicApplicationLevelConfidentialityProfile.
- void SetFile (const File &f)

  Set/Get File.

Static Public Member Functions

- static void ClearInternalUIDs ()
- static std::vector< Tag > GetBasicApplicationLevelConfidentialityProfileAttributes ()

  Return the list of Tag that will be considered when anonymizing a DICOM file.
- static SmartPointer< Anonymizer > New ()

  for wrapped language: instantiate a reference counted object

Protected Member Functions

- bool BALCPProtect (DataSet &ds, Tag const &tag, const IOD &iod)
- bool CanEmptyTag (Tag const &tag, const IOD &iod) const
- void RecurseDataSet (DataSet &ds)

10.8.1 Detailed Description

Anonymizer.

This class is a multi purpose anonymizer. It can work in 2 mode:

- Full (irreversible) anonymizer (aka dumb mode)
- reversible de-identifier/re-identifier (aka smart mode). This implements the Basic Application Level Confidentiality Profile, DICOM PS 3.15-2009

1. dumb mode This is a dumb anonymizer implementation. All it allows user is simple operation such as:

Tag based functions:

- complete removal of DICOM attribute (Remove)
- make a tag empty, ie make it's length 0 (Empty)
- replace with another string-based value (Replace)

DataSet based functions:

- Remove all group length attribute from a DICOM dataset (Group Length element are deprecated, DICOM 2008)
- Remove all private attributes
- Remove all retired attributes

All function calls actually execute the user specified request. Previous implementation were calling a general Anonymize function but traversing a std::set is O(n) operation, while a simple user specified request is O(log(n)) operation. So 'm' user interaction is O(m∗log(n)) which is $<$ O(n) complexity.

1. smart mode this mode implements the Basic Application Level Confidentiality Profile (DICOM PS 3.15-2008) In this case, it is extremely important to use the same Anonymizer class when anonymizing a FileSet. Once the Anonymizer is destroyed its memory of known (already processed) UIDs will be lost. which will make the anonymizer behaves incorrectly for attributes such as Series UID Study UID where user want some consistency. When attribute is Type 1 / Type 1C, a dummy generator will take in the existing value and produce a dummy value (a sha1 representation). sha1 algorithm is considered to be cryptographically strong (compared to md5sum) so that we meet the following two conditions:
   - Produce the same dummy value for the same input value
   - do not provide an easy way to retrieve the original value from the sha1 generated value

This class implement the Subject/Observer pattern trigger the following event:

- AnonymizeEvent
- IterationEvent
- StartEvent
- EndEvent

See also

CryptographicMessageSyntax

Examples

BasicAnonymizer.cs, ClinicalTrialAnnotate.cxx, ClinicalTrialIdentificationWorkflow.cs, CreateJPIPDataSet.cxx, EncapsulateFileInRawData.cxx, ManipulateFile.cs, and MpegVideoInfo.cs.

10.8.2 Constructor & Destructor Documentation

10.8.2.1 Anonymizer()

```
gdcm::Anonymizer::Anonymizer ( )  [inline]
```

10.8.2.2 ~Anonymizer()

```
gdcm::Anonymizer::~Anonymizer ( )  [override]
```

10.8.3 Member Function Documentation

10.8.3.1 BALCPProtect()

```
bool gdcm::Anonymizer::BALCPProtect (
            DataSet & ds,
            Tag const & tag,
            const IOD & iod )  [protected]
```

10.8.3.2 BasicApplicationLevelConfidentialityProfile()

```
bool gdcm::Anonymizer::BasicApplicationLevelConfidentialityProfile (
            bool deidentify = true )
```

PS 3.15 / E.1.1 De-Identifier An Application may claim conformance to the Basic Application Level Confidentiality Profile as a deidentifier if it protects all Attributes that might be used by unauthorized entities to identify the patient. NOT THREAD SAFE

Examples

BasicAnonymizer.cs.

10.8.3.3 CanEmptyTag()

```
bool gdcm::Anonymizer::CanEmptyTag (
            Tag const & tag,
            const IOD & iod ) const  [protected]
```

10.8.3.4 Clear() [1/2]

```
bool gdcm::Anonymizer::Clear (
            PrivateTag const & pt )
```

10.8.3.5 Clear() [2/2]

```
bool gdcm::Anonymizer::Clear (
            Tag const & t )
```

Identical to 'Empty' except no action is done when tag is not present.

10.8.3.6 ClearInternalUIDs()

```
static void gdcm::Anonymizer::ClearInternalUIDs ( )  [static]
```

Clear the internal mapping of real UIDs to generated UIDs

Warning

the mapping is definitely lost

10.8.3.7 Empty() [1/2]

```
bool gdcm::Anonymizer::Empty (
            PrivateTag const & pt )
```

Make PrivateTag pt empty (if not found tag will be created) Pay special attention that this code must be done before any call to Empty/Remove of the associated Private Creator, but before any call to Replace.

10.8.3.8 Empty() [2/2]

```
bool gdcm::Anonymizer::Empty (
            Tag const & t )
```

Make Tag t empty (if not found tag will be created)

Examples

CreateJPIPDataSet.cxx.

10.8.3.9 GetBasicApplicationLevelConfidentialityProfileAttributes()

```
static std::vector< Tag > gdcm::Anonymizer::GetBasicApplicationLevelConfidentialityProfile↩
Attributes ( )  [static]
```

Return the list of Tag that will be considered when anonymizing a DICOM file.

Examples

GenFakeIdentifyFile.cxx, and TraverseModules.cxx.

10.8.3.10 GetCryptographicMessageSyntax()

```
const CryptographicMessageSyntax * gdcm::Anonymizer::GetCryptographicMessageSyntax ( ) const
```

10.8.3.11 GetFile()

```
File & gdcm::Anonymizer::GetFile ( )  [inline]
```

Examples

BasicAnonymizer.cs, and ManipulateFile.cs.

10.8.3.12 New()

```
static SmartPointer< Anonymizer > gdcm::Anonymizer::New ( )  [inline], [static]
```

for wrapped language: instantiate a reference counted object

Examples

BasicAnonymizer.cs, and ClinicalTrialIdentificationWorkflow.cs.

10.8.3.13 RecurseDataSet()

```
void gdcm::Anonymizer::RecurseDataSet (
            DataSet & ds )  [protected]
```

10.8.3.14 Remove() [1/2]

```
bool gdcm::Anonymizer::Remove (
            PrivateTag const & pt )
```

remove a private tag (even a SQ can be removed) Pay special attention that this code must be done before any call to Empty/Remove of the associated Private Creator, but before any call to Replace. When the private reservation becomes empty, no check is done to automatically remove the private creator

10.8.3.15 Remove() [2/2]

```
bool gdcm::Anonymizer::Remove (
            Tag const & t )
```

remove a tag (even a SQ can be removed)

10.8.3.16 RemoveGroupLength()

```
bool gdcm::Anonymizer::RemoveGroupLength ( )
```

Main function that loop over all elements and remove group length.

Examples

ClinicalTrialAnnotate.cxx, and ManipulateFile.cs.

10.8.3.17 RemovePrivateTags()

```
bool gdcm::Anonymizer::RemovePrivateTags ( )
```

Main function that loop over all elements and remove private tags.

Examples

ClinicalTrialAnnotate.cxx, and ManipulateFile.cs.

10.8.3.18 RemoveRetired()

```
bool gdcm::Anonymizer::RemoveRetired ( )
```

Main function that loop over all elements and remove retired element.

10.8.3.19 Replace() [1/4]

```
bool gdcm::Anonymizer::Replace (
            PrivateTag const & t,
            const char * value )
```

10.8.3.20 Replace() [2/4]

```
bool gdcm::Anonymizer::Replace (
            PrivateTag const & t,
            const char * value,
            VL const & vl )
```

10.8.3.21 Replace() [3/4]

```
bool gdcm::Anonymizer::Replace (
            Tag const & t,
            const char * value )
```

Replace tag with another value, if tag is not found it will be created: WARNING: this function can only execute if tag is a VRASCII

Examples

ClinicalTrialAnnotate.cxx, CreateJPIPDataSet.cxx, EncapsulateFileInRawData.cxx, ManipulateFile.cs, and MpegVideoInfo.cs.

10.8.3.22 Replace() [4/4]

```
bool gdcm::Anonymizer::Replace (
            Tag const & t,
            const char * value,
            VL const & vl )
```

when the value contains \0, it is a good idea to specify the length. This function is required when dealing with VRBINARY tag

10.8.3.23 SetCryptographicMessageSyntax()

```
void gdcm::Anonymizer::SetCryptographicMessageSyntax (
            CryptographicMessageSyntax * cms )
```

Set/Get CMS key that will be used to encrypt the dataset within BasicApplicationLevelConfidentialityProfile.

Examples

BasicAnonymizer.cs, and ClinicalTrialIdentificationWorkflow.cs.

10.8.3.24 SetFile()

```
void gdcm::Anonymizer::SetFile (
            const File & f )  [inline]
```

Set/Get File.

Examples

BasicAnonymizer.cs, ClinicalTrialAnnotate.cxx, CreateJPIPDataSet.cxx, EncapsulateFileInRawData.cxx, ManipulateFile.cs, and MpegVideoInfo.cs.

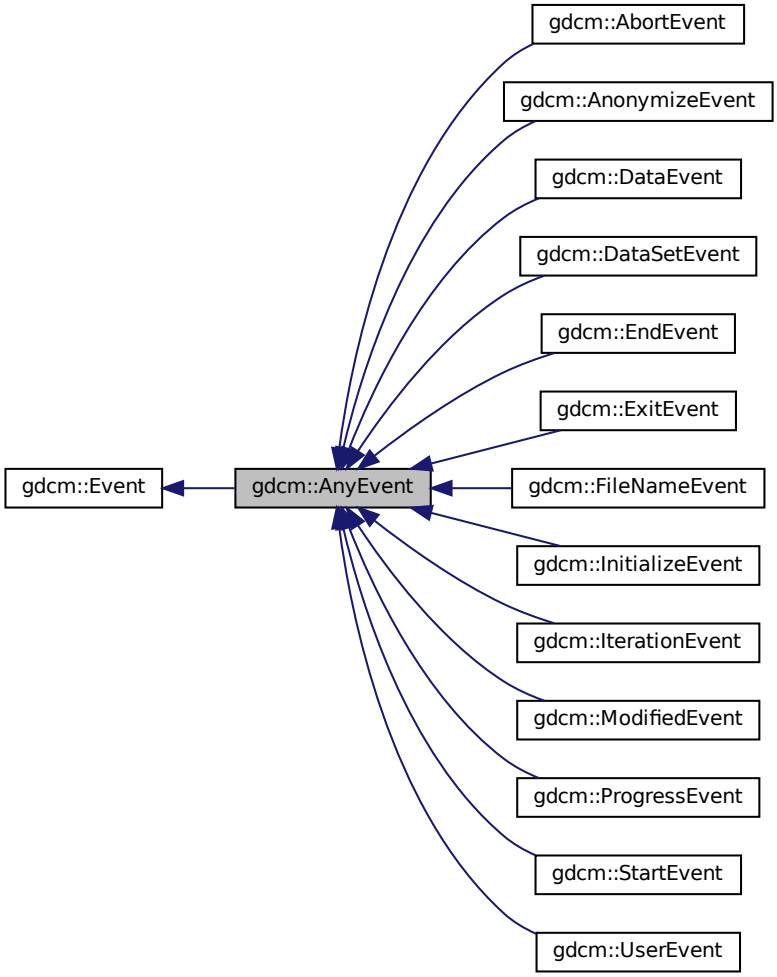
The documentation for this class was generated from the following file:

- gdcmAnonymizer.h

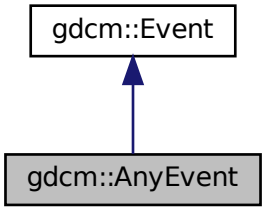
10.9 gdcm::AnyEvent Class Reference

`#include <gdcmEvent.h>`

Inheritance diagram for gdcm::AnyEvent:



Collaboration diagram for gdcm::AnyEvent:



Additional Inherited Members

The documentation for this class was generated from the following file:

- gdcmEvent.h

10.10 gdcm::network::ApplicationContext Class Reference

ApplicationContext.

`#include <gdcmApplicationContext.h>`

Public Member Functions

- ApplicationContext ()
- const char * GetName () const
- void Print (std::ostream &os) const
- std::istream & Read (std::istream &is)
- void SetName (const char *name)
- size_t Size () const
- const std::ostream & Write (std::ostream &os) const

10.10.1 Detailed Description

ApplicationContext.

Table 9-12 APPLICATION CONTEXT ITEM FIELDS

Todo Looks like Application Context can only be 64 bytes at max (see Figure 9-1 / PS 3.8 - 2009)

10.10.2 Constructor & Destructor Documentation

10.10.2.1 ApplicationContext()

```
gdcm::network::ApplicationContext::ApplicationContext ( )
```

10.10.3 Member Function Documentation

10.10.3.1 GetName()

```
const char * gdcm::network::ApplicationContext::GetName ( ) const  [inline]
```

10.10.3.2 Print()

```
void gdcm::network::ApplicationContext::Print (
            std::ostream & os ) const
```

10.10.3.3 Read()

```
std::istream & gdcm::network::ApplicationContext::Read (
            std::istream & is )
```

10.10.3.4 SetName()

```
void gdcm::network::ApplicationContext::SetName (
            const char * name )  [inline]
```

10.10.3.5 Size()

```
size_t gdcm::network::ApplicationContext::Size ( ) const
```

10.10.3.6 Write()

```
const std::ostream & gdcm::network::ApplicationContext::Write (
            std::ostream & os ) const
```

The documentation for this class was generated from the following file:

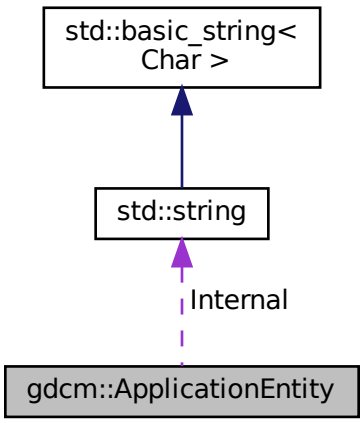
- gdcmApplicationContext.h

10.11 gdcm::ApplicationEntity Class Reference

ApplicationEntity.

```
#include <gdcmApplicationEntity.h>
```

Collaboration diagram for gdcm::ApplicationEntity:



Public Member Functions

- bool IsValid () const
- void Print (std::ostream &os) const
- void SetBlob (const std::vector< char > &v)
- void Squeeze ()

Public Attributes

- std::string Internal

Static Public Attributes

- static const unsigned int MaxLength = 16
- static const unsigned int MaxNumberOfComponents = 1
- static const char Padding = ' '
- static const char Separator = ' '

10.11.1 Detailed Description

ApplicationEntity.

- AE Application Entity
- A string of characters that identifies an Application Entity with leading and trailing spaces (20H) being non-significant. A value consisting solely of spaces shall not be used.
- Default Character Repertoire excluding character code 5CH (the BACKSLASH \ in ISO-IR 6), and control characters LF, FF, CR and ESC.
- 16 bytes maximum

10.11.2 Member Function Documentation

10.11.2.1 IsValid()

```
bool gdcm::ApplicationEntity::IsValid ( ) const  [inline]
```

10.11.2.2 Print()

```
void gdcm::ApplicationEntity::Print (
            std::ostream & os ) const  [inline]
```

10.11.2.3 SetBlob()

```
void gdcm::ApplicationEntity::SetBlob (
            const std::vector< char > & v )  [inline]
```

10.11.2.4 Squeeze()

```
void gdcm::ApplicationEntity::Squeeze ( )  [inline]
```

10.11.3 Member Data Documentation

10.11.3.1 Internal

```
std::string gdcm::ApplicationEntity::Internal
```

10.11.3.2 MaxLength

```
const unsigned int gdcm::ApplicationEntity::MaxLength = 16  [static]
```

10.11.3.3 MaxNumberOfComponents

```
const unsigned int gdcm::ApplicationEntity::MaxNumberOfComponents = 1  [static]
```

10.11.3.4 Padding

```
const char gdcm::ApplicationEntity::Padding = ' '  [static]
```

10.11.3.5 Separator

```
const char gdcm::ApplicationEntity::Separator = ' '  [static]
```

The documentation for this class was generated from the following file:

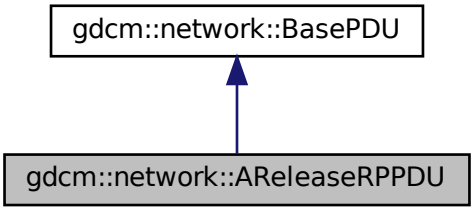
- gdcmApplicationEntity.h

10.12 gdcm::network::AReleaseRPPDU Class Reference

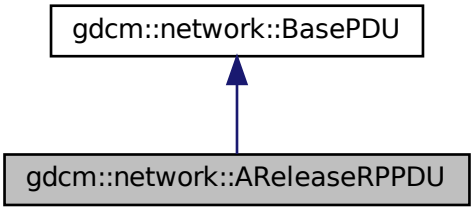
AReleaseRPPDU.

```
#include <gdcmAReleaseRPPDU.h>
```

Inheritance diagram for gdcm::network::AReleaseRPPDU:



Collaboration diagram for gdcm::network::AReleaseRPPDU:



Public Member Functions

- AReleaseRPPDU ()
- bool IsLastFragment () const override
- void Print (std::ostream &os) const override
- std::istream & Read (std::istream &is) override
- size_t Size () const override
- const std::ostream & Write (std::ostream &os) const override

10.12.1 Detailed Description

AReleaseRPPDU.

Table 9-25 A-RELEASE-RP PDU fields

10.12.2 Constructor & Destructor Documentation

10.12.2.1 AReleaseRPPDU()

```
gdcm::network::AReleaseRPPDU::AReleaseRPPDU ( )
```

10.12.3 Member Function Documentation

10.12.3.1 IsLastFragment()

```
bool gdcm::network::AReleaseRPPDU::IsLastFragment ( ) const  [inline], [override], [virtual]
```

Implements gdcm::network::BasePDU.

10.12.3.2 Print()

```
void gdcm::network::AReleaseRPPDU::Print (
            std::ostream & os ) const  [override], [virtual]
```

Implements gdcm::network::BasePDU.

10.12.3.3 Read()

```
std::istream & gdcm::network::AReleaseRPPDU::Read (
            std::istream & is )  [override], [virtual]
```

Implements gdcm::network::BasePDU.

10.12.3.4 Size()

```
size_t gdcm::network::AReleaseRPPDU::Size ( ) const  [override], [virtual]
```

Implements gdcm::network::BasePDU.

10.12.3.5 Write()

```
const std::ostream & gdcm::network::AReleaseRPPDU::Write (
            std::ostream & os ) const  [override], [virtual]
```

Implements gdcm::network::BasePDU.

The documentation for this class was generated from the following file:

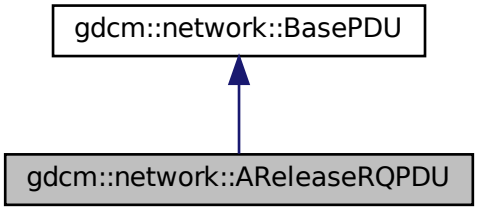
- gdcmAReleaseRPPDU.h

10.13 gdcm::network::AReleaseRQPDU Class Reference

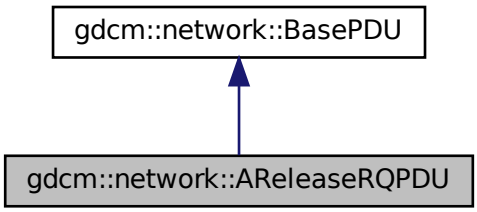
AReleaseRQPDU.

```
#include <gdcmAReleaseRQPDU.h>
```

Inheritance diagram for gdcm::network::AReleaseRQPDU:



Collaboration diagram for gdcm::network::AReleaseRQPDU:



Public Member Functions

- AReleaseRQPDU ()
- bool IsLastFragment () const override
- void Print (std::ostream &os) const override
- std::istream & Read (std::istream &is) override
- size_t Size () const override
- const std::ostream & Write (std::ostream &os) const override

10.13.1 Detailed Description

AReleaseRQPDU.

Table 9-24 A-RELEASE-RQ PDU FIELDS

10.13.2 Constructor & Destructor Documentation

10.13.2.1 AReleaseRQPDU()

```
gdcm::network::AReleaseRQPDU::AReleaseRQPDU ( )
```

10.13.3 Member Function Documentation

10.13.3.1 IsLastFragment()

```
bool gdcm::network::AReleaseRQPDU::IsLastFragment ( ) const  [inline], [override], [virtual]
```

Implements gdcm::network::BasePDU.

10.13.3.2 Print()

```
void gdcm::network::AReleaseRQPDU::Print (
            std::ostream & os ) const  [override], [virtual]
```

Implements gdcm::network::BasePDU.

10.13.3.3 Read()

```
std::istream & gdcm::network::AReleaseRQPDU::Read (
            std::istream & is )  [override], [virtual]
```

Implements gdcm::network::BasePDU.

10.13.3.4 Size()

```
size_t gdcm::network::AReleaseRQPDU::Size ( ) const  [override], [virtual]
```

Implements gdcm::network::BasePDU.

10.13.3.5 Write()

```
const std::ostream & gdcm::network::AReleaseRQPDU::Write (
            std::ostream & os ) const  [override], [virtual]
```

Implements gdcm::network::BasePDU.

The documentation for this class was generated from the following file:

- gdcmAReleaseRQPDU.h

10.14 gdcm::network::ARTIMTimer Class Reference

ARTIMTimer.

`#include <gdcmARTIMTimer.h>`

Public Member Functions

- ARTIMTimer ()
- double GetElapsedTime () const
- bool GetHasExpired () const
- double GetTimeout () const
- void SetTimeout (double inTimeout)
- void Start ()
- void Stop ()

10.14.1 Detailed Description

ARTIMTimer.

This file contains the code for the ARTIM timer.

Basically, the ARTIM timer will just get the wall time when it's started, and then can be queried for the current time, and then can be stopped (ie, the start time reset).

Because we're trying to do this without threading, we should be able to 'start' the ARTIM timer by this mechanism, and then when waiting for a particular response, tight loop that with sleep calls and determinations of when the ARTIM timer has reached its peak. As such, this isn't a strict 'timer' in the traditional sense of the word, but more of a time keeper.

There can be only one ARTIM timer per connection.

10.14.2 Constructor & Destructor Documentation

10.14.2.1 ARTIMTimer()

`gdcm::network::ARTIMTimer::ARTIMTimer ( )`

10.14.3 Member Function Documentation

10.14.3.1 GetElapsedTime()

```
double gdcm::network::ARTIMTimer::GetElapsedTime ( ) const
```

10.14.3.2 GetHasExpired()

```
bool gdcm::network::ARTIMTimer::GetHasExpired ( ) const
```

10.14.3.3 GetTimeout()

```
double gdcm::network::ARTIMTimer::GetTimeout ( ) const
```

10.14.3.4 SetTimeout()

```
void gdcm::network::ARTIMTimer::SetTimeout (
            double inTimeout )
```

10.14.3.5 Start()

```
void gdcm::network::ARTIMTimer::Start ( )
```

10.14.3.6 Stop()

```
void gdcm::network::ARTIMTimer::Stop ( )
```

The documentation for this class was generated from the following file:

- gdcmARTIMTimer.h

10.15 gdcm::ASN1 Class Reference

Class for ASN1.

```
#include <gdcmASN1.h>
```

Public Member Functions

- ASN1 ()
- ASN1 (const ASN1 &)=delete
- ~ASN1 ()
- void operator= (const ASN1 &)=delete

Static Public Member Functions

- static bool ParseDump (const char *array, size_t length)
- static bool ParseDumpFile (const char *filename)

Protected Member Functions

- int TestPBKDF2 ()

10.15.1 Detailed Description

Class for ASN1.

10.15.2 Constructor & Destructor Documentation

10.15.2.1 ASN1() [1/2]

```
gdcm::ASN1::ASN1 ( )
```

10.15.2.2 ~ASN1()

```
gdcm::ASN1::~ASN1 ( )
```

10.15.2.3 ASN1() [2/2]

```
gdcm::ASN1::ASN1 (
            const ASN1 &  )  [delete]
```

10.15.3 Member Function Documentation

10.15.3.1 operator=()

```
void gdcm::ASN1::operator= (
            const ASN1 &  )  [delete]
```

10.15.3.2 ParseDump()

```
static bool gdcm::ASN1::ParseDump (
            const char * array,
            size_t length )  [static]
```

10.15.3.3 ParseDumpFile()

```
static bool gdcm::ASN1::ParseDumpFile (
            const char * filename )  [static]
```

10.15.3.4 TestPBKDF2()

```
int gdcm::ASN1::TestPBKDF2 ( )  [protected]
```

The documentation for this class was generated from the following file:

- gdcmASN1.h

10.16 gdcm::network::AsynchronousOperationsWindowSub Class Reference

AsynchronousOperationsWindowSub.

```
#include <gdcmAsynchronousOperationsWindowSub.h>
```

Public Member Functions

- AsynchronousOperationsWindowSub ()
- void Print (std::ostream &os) const
- std::istream & Read (std::istream &is)
- size_t Size () const
- const std::ostream & Write (std::ostream &os) const

10.16.1 Detailed Description

AsynchronousOperationsWindowSub.

PS 3.7 Table D.3-7 ASYNCHRONOUS OPERATIONS WINDOW SUB-ITEM FIELDS (A-ASSOCIATE-RQ)

10.16.2 Constructor & Destructor Documentation

10.16.2.1 AsynchronousOperationsWindowSub()

```
gdcm::network::AsynchronousOperationsWindowSub::AsynchronousOperationsWindowSub ( )
```

10.16.3 Member Function Documentation

10.16.3.1 Print()

```
void gdcm::network::AsynchronousOperationsWindowSub::Print (
            std::ostream & os ) const
```

10.16.3.2 Read()

```
std::istream & gdcm::network::AsynchronousOperationsWindowSub::Read (
            std::istream & is )
```

10.16.3.3 Size()

```
size_t gdcm::network::AsynchronousOperationsWindowSub::Size ( ) const
```

10.16.3.4 Write()

```
const std::ostream & gdcm::network::AsynchronousOperationsWindowSub::Write (
            std::ostream & os ) const
```

The documentation for this class was generated from the following file:

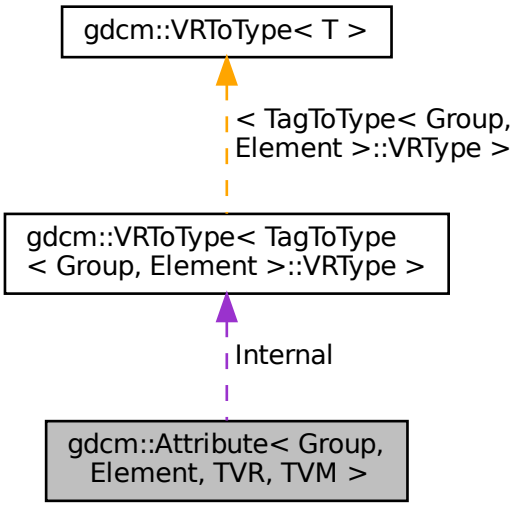
- gdcmAsynchronousOperationsWindowSub.h

10.17 gdcm::Attribute< Group, Element, TVR, TVM > Class Template Reference

Attribute class This class use template metaprograming tricks to let the user know when the template instantiation does not match the public dictionary.

```
#include <gdcmAttribute.h>
```

Collaboration diagram for gdcm::Attribute< Group, Element, TVR, TVM >:



Public Types

- enum { VMType = VMToLength<TVM>::Length }
- typedef VRToType< TVR >::Type ArrayType

Public Member Functions

- GDCM_STATIC_ASSERT (((((VR::VRType) TVR &VR::VR_VM1) &&((VM::VMType) TVM==VM::VM1))||!((VR::VRType) TVR &VR::VR_VM1)))
- GDCM_STATIC_ASSERT (((VM::VMType) TVM &(VM::VMType)(TagToType< Group, Element >::VMType)))
- GDCM_STATIC_ASSERT (((VR::VRType) TVR &(VR::VRType)(TagToType< Group, Element >::VRType)))
- DataElement GetAsDataElement () const
- unsigned int GetNumberOfValues () const
- ArrayType & GetValue (unsigned int idx=0)
- ArrayType const & GetValue (unsigned int idx=0) const
- const ArrayType * GetValues () const
- bool operator!= (const Attribute &att) const
- bool operator< (const Attribute &att) const
- bool operator== (const Attribute &att) const
- ArrayType & operator[ ] (unsigned int idx)
- ArrayType const & operator[ ] (unsigned int idx) const
- void Print (std::ostream &os) const
- void Set (DataSet const &ds)
- void SetFromDataElement (DataElement const &de)
- void SetFromDataSet (DataSet const &ds)
- void SetValue (ArrayType v, unsigned int idx=0)
- void SetValues (const ArrayType *array, unsigned int numel=VMType)

Static Public Member Functions

- static VM GetDictVM ()
- static VR GetDictVR ()
- static Tag GetTag ()
- static VM GetVM ()
- static VR GetVR ()

Public Attributes

- ArrayType Internal [VMToLength< TVM >::Length]

Protected Member Functions

- void SetByteValue (const ByteValue *bv)
- void SetByteValueNoSwap (const ByteValue *bv)

10.17.1 Detailed Description

template<uint16_t Group, uint16_t Element, long long TVR = TagToType<Group, Element>::VRType, int TVM = TagToType<Group, Element>::VMType>
class gdcm::Attribute< Group, Element, TVR, TVM >

Attribute class This class use template metaprograming tricks to let the user know when the template instantiation does not match the public dictionary.

Typical example that compile is: Attribute<0x0008,0x9007> a = {"ORIGINAL","PRIMARY","T1","NONE"};

Examples that will NOT compile are:

Attribute<0x0018,0x1182, VR::IS, VM::VM1> fd1 = {}; // not enough parameters Attribute<0x0018,0x1182, VR::IS, VM::VM2> fd2 = {0,1,2}; // too many initializers Attribute<0x0018,0x1182, VR::IS, VM::VM3> fd3 = {0,1,2}; // VM3 is not valid Attribute<0x0018,0x1182, VR::UL, VM::VM2> fd3 = {0,1}; // UL is not valid VR

Examples

CreateFakeRTDOSE.cxx, CreateJPIPDataSet.cxx, DeriveSeries.cxx, Extracting_All_Resolution.cxx, Fake_Image_Using_Stream_In FixOrientation.cxx, GenFakeIdentifyFile.cxx, GetSequenceUltrasound.cxx, HelloWorld.cxx, LargeVRDSExplicit.cxx, PatchFile.cxx, ReadAndDumpDICOMDIR2.cxx, ReadAndPrintAttributes.cxx, SortImage.cxx, StreamImageReaderTest.cxx, VolumeSorter.cxx, gdcmrtionplan.cxx, gdcmrtplan.cxx, pmsct_rgb1.cxx, and rle2img.cxx.

10.17.2 Member Typedef Documentation

10.17.2.1 ArrayType

```
template<uint16_t Group, uint16_t Element, long long TVR = TagToType<Group, Element>::VRType, int
TVM = TagToType<Group, Element>::VMType>
typedef VRToType<TVR>::Type gdcm::Attribute< Group, Element, TVR, TVM >::ArrayType
```

10.17.3 Member Enumeration Documentation

10.17.3.1 anonymous enum

```
template<uint16_t Group, uint16_t Element, long long TVR = TagToType<Group, Element>::VRType, int
TVM = TagToType<Group, Element>::VMType>
anonymous enum
```

Enumerator

VMType	

10.17.4 Member Function Documentation

10.17.4.1 GDCM_STATIC_ASSERT() [1/3]

```
template<uint16_t Group, uint16_t Element, long long TVR = TagToType<Group, Element>::VRType, int
TVM = TagToType<Group, Element>::VMType>
gdcm::Attribute< Group, Element, TVR, TVM >::GDCM_STATIC_ASSERT (
             ((((VR::VRType) TVR &VR::VR_VM1) &&((VM::VMType) TVM==VM::VM1))||!((VR::VRType) TVR
&VR::VR_VM1))  )
```

10.17.4.2 GDCM_STATIC_ASSERT() [2/3]

```
template<uint16_t Group, uint16_t Element, long long TVR = TagToType<Group, Element>::VRType, int
TVM = TagToType<Group, Element>::VMType>
gdcm::Attribute< Group, Element, TVR, TVM >::GDCM_STATIC_ASSERT (
             ((VM::VMType) TVM &(VM::VMType)(TagToType< Group, Element >::VMType))  )
```

10.17.4.3 GDCM_STATIC_ASSERT() [3/3]

```
template<uint16_t Group, uint16_t Element, long long TVR = TagToType<Group, Element>::VRType, int
TVM = TagToType<Group, Element>::VMType>
gdcm::Attribute< Group, Element, TVR, TVM >::GDCM_STATIC_ASSERT (
             ((VR::VRType) TVR &(VR::VRType)(TagToType< Group, Element >::VRType))  )
```

10.17.4.4 GetAsDataElement()

```
template<uint16_t Group, uint16_t Element, long long TVR = TagToType<Group, Element>::VRType, int
TVM = TagToType<Group, Element>::VMType>
DataElement gdcm::Attribute< Group, Element, TVR, TVM >::GetAsDataElement ( ) const  [inline]
```

Examples

CreateFakeRTDOSE.cxx, CreateJPIPDataSet.cxx, Extracting_All_Resolution.cxx, Fake_Image_Using_Stream_Image_Writer.cxx, FixOrientation.cxx, GenFakeIdentifyFile.cxx, HelloWorld.cxx, LargeVRDSExplicit.cxx, PatchFile.cxx, and StreamImageReaderTest.cxx.

References gdcm::DataElement::GetVR(), gdcm::DataElement::SetByteValue(), and gdcm::DataElement::SetVR().

10.17.4.5 GetDictVM()

```
template<uint16_t Group, uint16_t Element, long long TVR = TagToType<Group, Element>::VRType, int
TVM = TagToType<Group, Element>::VMType>
static VM gdcm::Attribute< Group, Element, TVR, TVM >::GetDictVM ( )  [inline], [static]
```

10.17.4.6 GetDictVR()

```
template<uint16_t Group, uint16_t Element, long long TVR = TagToType<Group, Element>::VRType, int
TVM = TagToType<Group, Element>::VMType>
static VR gdcm::Attribute< Group, Element, TVR, TVM >::GetDictVR ( )  [inline], [static]
```

10.17.4.7 GetNumberOfValues()

```
template<uint16_t Group, uint16_t Element, long long TVR = TagToType<Group, Element>::VRType, int
TVM = TagToType<Group, Element>::VMType>
unsigned int gdcm::Attribute< Group, Element, TVR, TVM >::GetNumberOfValues ( ) const  [inline]
```

Examples

LargeVRDSExplicit.cxx.

Referenced by gdcm::Attribute< Group, Element, TVR, TVM >::operator<(), and gdcm::Attribute< Group, Element, TVR, VM::VM1 >::o

10.17.4.8 GetTag()

```
template<uint16_t Group, uint16_t Element, long long TVR = TagToType<Group, Element>::VRType, int
TVM = TagToType<Group, Element>::VMType>
static Tag gdcm::Attribute< Group, Element, TVR, TVM >::GetTag ( )  [inline], [static]
```

Examples

PatchFile.cxx, ReadAndPrintAttributes.cxx, gdcmrtionplan.cxx, and gdcmrtplan.cxx.

10.17.4.9 GetValue() [1/2]

```
template<uint16_t Group, uint16_t Element, long long TVR = TagToType<Group, Element>::VRType, int
TVM = TagToType<Group, Element>::VMType>
ArrayType & gdcm::Attribute< Group, Element, TVR, TVM >::GetValue (
            unsigned int idx = 0 )  [inline]
```

Examples

DeriveSeries.cxx, FixOrientation.cxx, GetSequenceUltrasound.cxx, PatchFile.cxx, ReadAndDumpDICOMDIR2.cxx, ReadAndPrintAttributes.cxx, gdcmrtionplan.cxx, gdcmrtplan.cxx, pmsct_rgb1.cxx, and rle2img.cxx.

10.17.4.10 GetValue() [2/2]

```
template<uint16_t Group, uint16_t Element, long long TVR = TagToType<Group, Element>::VRType, int
TVM = TagToType<Group, Element>::VMType>
ArrayType const & gdcm::Attribute< Group, Element, TVR, TVM >::GetValue (
            unsigned int idx = 0 ) const  [inline]
```

10.17.4.11 GetValues()

```
template<uint16_t Group, uint16_t Element, long long TVR = TagToType<Group, Element>::VRType, int
TVM = TagToType<Group, Element>::VMType>
const ArrayType * gdcm::Attribute< Group, Element, TVR, TVM >::GetValues ( ) const  [inline]
```

Examples

FixOrientation.cxx, LargeVRDSExplicit.cxx, gdcmrtionplan.cxx, and gdcmrtplan.cxx.

Referenced by gdcm::Attribute< Group, Element, TVR, TVM >::operator!=(), gdcm::Attribute< Group, Element, TVR, VM::VM1 >::opera
gdcm::Attribute< Group, Element, TVR, TVM >::operator<(), gdcm::Attribute< Group, Element, TVR, VM::VM1 >::operator<(),
gdcm::Attribute< Group, Element, TVR, TVM >::operator==(), and gdcm::Attribute< Group, Element, TVR, VM::VM1 >::operator==().

10.17.4.12 GetVM()

```
template<uint16_t Group, uint16_t Element, long long TVR = TagToType<Group, Element>::VRType, int
TVM = TagToType<Group, Element>::VMType>
static VM gdcm::Attribute< Group, Element, TVR, TVM >::GetVM ( )  [inline], [static]
```

10.17.4.13 GetVR()

```
template<uint16_t Group, uint16_t Element, long long TVR = TagToType<Group, Element>::VRType, int
TVM = TagToType<Group, Element>::VMType>
static VR gdcm::Attribute< Group, Element, TVR, TVM >::GetVR ( )  [inline], [static]
```

10.17.4.14 operator"!=()

```
template<uint16_t Group, uint16_t Element, long long TVR = TagToType<Group, Element>::VRType, int
TVM = TagToType<Group, Element>::VMType>
bool gdcm::Attribute< Group, Element, TVR, TVM >::operator!= (
            const Attribute< Group, Element, TVR, TVM > & att ) const  [inline]
```

References gdcm::Attribute< Group, Element, TVR, TVM >::GetValues().

10.17.4.15 operator<()

```
template<uint16_t Group, uint16_t Element, long long TVR = TagToType<Group, Element>::VRType, int
TVM = TagToType<Group, Element>::VMType>
bool gdcm::Attribute< Group, Element, TVR, TVM >::operator< (
            const Attribute< Group, Element, TVR, TVM > & att ) const  [inline]
```

References gdcm::Attribute< Group, Element, TVR, TVM >::GetNumberOfValues(), and gdcm::Attribute< Group, Element, TVR, TVM >

10.17.4.16 operator==()

```
template<uint16_t Group, uint16_t Element, long long TVR = TagToType<Group, Element>::VRType, int
TVM = TagToType<Group, Element>::VMType>
bool gdcm::Attribute< Group, Element, TVR, TVM >::operator== (
            const Attribute< Group, Element, TVR, TVM > & att ) const  [inline]
```

References gdcm::Attribute< Group, Element, TVR, TVM >::GetValues().

10.17.4.17 operator[]() [1/2]

```
template<uint16_t Group, uint16_t Element, long long TVR = TagToType<Group, Element>::VRType, int
TVM = TagToType<Group, Element>::VMType>
ArrayType & gdcm::Attribute< Group, Element, TVR, TVM >::operator[] (
            unsigned int idx )  [inline]
```

10.17.4.18 operator[]() [2/2]

```
template<uint16_t Group, uint16_t Element, long long TVR = TagToType<Group, Element>::VRType, int
TVM = TagToType<Group, Element>::VMType>
ArrayType const & gdcm::Attribute< Group, Element, TVR, TVM >::operator[] (
            unsigned int idx ) const  [inline]
```

10.17.4.19 Print()

```
template<uint16_t Group, uint16_t Element, long long TVR = TagToType<Group, Element>::VRType, int
TVM = TagToType<Group, Element>::VMType>
void gdcm::Attribute< Group, Element, TVR, TVM >::Print (
            std::ostream & os ) const  [inline]
```

10.17.4.20 Set()

```
template<uint16_t Group, uint16_t Element, long long TVR = TagToType<Group, Element>::VRType, int
TVM = TagToType<Group, Element>::VMType>
void gdcm::Attribute< Group, Element, TVR, TVM >::Set (
            DataSet const & ds )  [inline]
```

Examples

LargeVRDSExplicit.cxx, SortImage.cxx, and VolumeSorter.cxx.

References gdcm::DataSet::GetDataElement().

10.17.4.21 SetByteValue()

```
template<uint16_t Group, uint16_t Element, long long TVR = TagToType<Group, Element>::VRType, int
TVM = TagToType<Group, Element>::VMType>
void gdcm::Attribute< Group, Element, TVR, TVM >::SetByteValue (
            const ByteValue * bv )  [inline], [protected]
```

References gdcm::ByteValue::GetLength(), and gdcm::ByteValue::GetPointer().

10.17.4.22 SetByteValueNoSwap()

```
template<uint16_t Group, uint16_t Element, long long TVR = TagToType<Group, Element>::VRType, int
TVM = TagToType<Group, Element>::VMType>
void gdcm::Attribute< Group, Element, TVR, TVM >::SetByteValueNoSwap (
            const ByteValue * bv )  [inline], [protected]
```

References gdcm::ByteValue::GetLength(), and gdcm::ByteValue::GetPointer().

10.17.4.23 SetFromDataElement()

```
template<uint16_t Group, uint16_t Element, long long TVR = TagToType<Group, Element>::VRType, int
TVM = TagToType<Group, Element>::VMType>
void gdcm::Attribute< Group, Element, TVR, TVM >::SetFromDataElement (
            DataElement const & de )  [inline]
```

Examples

GetSequenceUltrasound.cxx, LargeVRDSExplicit.cxx, PatchFile.cxx, ReadAndDumpDICOMDIR2.cxx, gdcmrtionplan.cxx, and gdcmrtplan.cxx.

References gdcm::DataElement::GetByteValue(), gdcm::DataElement::GetTag(), gdcm::DataElement::GetVR(), and gdcm::DataElement::IsEmpty().

10.17.4.24 SetFromDataSet()

```
template<uint16_t Group, uint16_t Element, long long TVR = TagToType<Group, Element>::VRType, int
TVM = TagToType<Group, Element>::VMType>
void gdcm::Attribute< Group, Element, TVR, TVM >::SetFromDataSet (
            DataSet const & ds )  [inline]
```

Examples

DeriveSeries.cxx, FixOrientation.cxx, ReadAndPrintAttributes.cxx, pmsct_rgb1.cxx, and rle2img.cxx.

References gdcm::DataSet::FindDataElement(), and gdcm::DataSet::GetDataElement().

10.17.4.25 SetValue()

```
template<uint16_t Group, uint16_t Element, long long TVR = TagToType<Group, Element>::VRType, int
TVM = TagToType<Group, Element>::VMType>
void gdcm::Attribute< Group, Element, TVR, TVM >::SetValue (
            ArrayType v,
            unsigned int idx = 0 )  [inline]
```

Examples

CreateFakeRTDOSE.cxx, CreateJPIPDataSet.cxx, FixOrientation.cxx, HelloWorld.cxx, LargeVRDSExplicit.cxx, and PatchFile.cxx.

10.17.4.26 SetValues()

```
template<uint16_t Group, uint16_t Element, long long TVR = TagToType<Group, Element>::VRType, int
TVM = TagToType<Group, Element>::VMType>
void gdcm::Attribute< Group, Element, TVR, TVM >::SetValues (
            const ArrayType * array,
            unsigned int numel = VMType )  [inline]
```

Examples

FixOrientation.cxx, and LargeVRDSExplicit.cxx.

10.17.5 Member Data Documentation

10.17.5.1 Internal

```
template<uint16_t Group, uint16_t Element, long long TVR = TagToType<Group, Element>::VRType, int
TVM = TagToType<Group, Element>::VMType>
ArrayType gdcm::Attribute< Group, Element, TVR, TVM >::Internal[VMToLength< TVM >::Length]
```

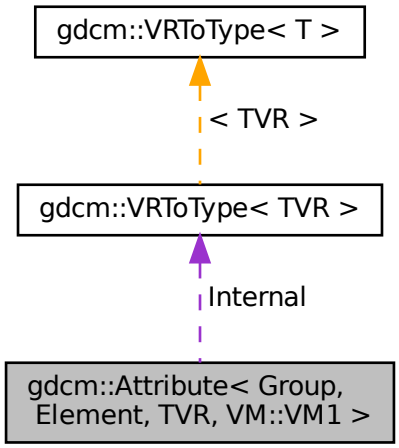
The documentation for this class was generated from the following file:

- gdcmAttribute.h

10.18 gdcm::Attribute< Group, Element, TVR, VM::VM1 > Class Template Reference

`#include <gdcmAttribute.h>`

Collaboration diagram for gdcm::Attribute< Group, Element, TVR, VM::VM1 >:



Public Types

- enum { VMType = VMToLength<VM::VM1>::Length }
- typedef VRToType< TVR >::Type ArrayType

Public Member Functions

- GDCM_STATIC_ASSERT (((((VR::VRType) TVR &VR::VR_VM1) &&((VM::VMType) VM::VM1==VM::VM1))||!((VR::VRType) TVR &VR::VR_VM1)))
- GDCM_STATIC_ASSERT (((VM::VMType) VM::VM1 &(VM::VMType)(TagToType< Group, Element >::VMType)))
- GDCM_STATIC_ASSERT (((VR::VRType) TVR &(VR::VRType)(TagToType< Group, Element >::VRType)))
- GDCM_STATIC_ASSERT (VMToLength< VM::VM1 >::Length==1)
- DataElement GetAsDataElement () const
- unsigned int GetNumberOfValues () const
- ArrayType & GetValue ()
- ArrayType const & GetValue () const
- const ArrayType * GetValues () const
- bool operator!= (const Attribute &att) const
- bool operator< (const Attribute &att) const
- bool operator== (const Attribute &att) const

- void Print (std::ostream &os) const
- void Set (DataSet const &ds)
- void SetFromDataElement (DataElement const &de)
- void SetFromDataSet (DataSet const &ds)
- void SetValue (ArrayType v)

Static Public Member Functions

- static VM GetDictVM ()
- static VR GetDictVR ()
- static Tag GetTag ()
- static VM GetVM ()
- static VR GetVR ()

Public Attributes

- ArrayType Internal

Protected Member Functions

- void SetByteValue (const ByteValue *bv)
- void SetByteValueNoSwap (const ByteValue *bv)

10.18.1 Member Typedef Documentation

10.18.1.1 ArrayType

```
template<uint16_t Group, uint16_t Element, long long TVR>
typedef VRToType<TVR>::Type gdcm::Attribute< Group, Element, TVR, VM::VM1 >::ArrayType
```

10.18.2 Member Enumeration Documentation

10.18.2.1 anonymous enum

```
template<uint16_t Group, uint16_t Element, long long TVR>
anonymous enum
```

Enumerator

VMType	

10.18.3 Member Function Documentation

10.18.3.1 GDCM_STATIC_ASSERT() [1/4]

```
template<uint16_t Group, uint16_t Element, long long TVR>
gdcm::Attribute< Group, Element, TVR, VM::VM1 >::GDCM_STATIC_ASSERT (
             ((((VR::VRType) TVR &VR::VR_VM1) &&((VM::VMType) VM::VM1==VM::VM1))||!((VR::VRType)
TVR &VR::VR_VM1))  )
```

10.18.3.2 GDCM_STATIC_ASSERT() [2/4]

```
template<uint16_t Group, uint16_t Element, long long TVR>
gdcm::Attribute< Group, Element, TVR, VM::VM1 >::GDCM_STATIC_ASSERT (
             ((VM::VMType) VM::VM1 &(VM::VMType)(TagToType< Group, Element >::VMType))  )
```

10.18.3.3 GDCM_STATIC_ASSERT() [3/4]

```
template<uint16_t Group, uint16_t Element, long long TVR>
gdcm::Attribute< Group, Element, TVR, VM::VM1 >::GDCM_STATIC_ASSERT (
             ((VR::VRType) TVR &(VR::VRType)(TagToType< Group, Element >::VRType))  )
```

10.18.3.4 GDCM_STATIC_ASSERT() [4/4]

```
template<uint16_t Group, uint16_t Element, long long TVR>
gdcm::Attribute< Group, Element, TVR, VM::VM1 >::GDCM_STATIC_ASSERT (
             VMToLength< VM::VM1 >::Length  = =1 )
```

10.18.3.5 GetAsDataElement()

```
template<uint16_t Group, uint16_t Element, long long TVR>
DataElement gdcm::Attribute< Group, Element, TVR, VM::VM1 >::GetAsDataElement ( ) const  [inline]
```

References gdcm::DataElement::GetVR(), gdcm::DataElement::SetByteValue(), and gdcm::DataElement::SetVR().

10.18.3.6 GetDictVM()

```
template<uint16_t Group, uint16_t Element, long long TVR>
static VM gdcm::Attribute< Group, Element, TVR, VM::VM1 >::GetDictVM ( )  [inline], [static]
```

10.18.3.7 GetDictVR()

```
template<uint16_t Group, uint16_t Element, long long TVR>
static VR gdcm::Attribute< Group, Element, TVR, VM::VM1 >::GetDictVR ( )  [inline], [static]
```

10.18.3.8 GetNumberOfValues()

```
template<uint16_t Group, uint16_t Element, long long TVR>
unsigned int gdcm::Attribute< Group, Element, TVR, VM::VM1 >::GetNumberOfValues ( ) const  [inline]
```

10.18.3.9 GetTag()

```
template<uint16_t Group, uint16_t Element, long long TVR>
static Tag gdcm::Attribute< Group, Element, TVR, VM::VM1 >::GetTag ( )  [inline], [static]
```

10.18.3.10 GetValue() [1/2]

```
template<uint16_t Group, uint16_t Element, long long TVR>
ArrayType & gdcm::Attribute< Group, Element, TVR, VM::VM1 >::GetValue ( )  [inline]
```

10.18.3.11 GetValue() [2/2]

```
template<uint16_t Group, uint16_t Element, long long TVR>
ArrayType const & gdcm::Attribute< Group, Element, TVR, VM::VM1 >::GetValue ( ) const  [inline]
```

10.18.3.12 GetValues()

```
template<uint16_t Group, uint16_t Element, long long TVR>
const ArrayType * gdcm::Attribute< Group, Element, TVR, VM::VM1 >::GetValues ( ) const  [inline]
```

10.18.3.13 GetVM()

```
template<uint16_t Group, uint16_t Element, long long TVR>
static VM gdcm::Attribute< Group, Element, TVR, VM::VM1 >::GetVM ( )  [inline], [static]
```

10.18.3.14 GetVR()

```
template<uint16_t Group, uint16_t Element, long long TVR>
static VR gdcm::Attribute< Group, Element, TVR, VM::VM1 >::GetVR ( )  [inline], [static]
```

10.18.3.15 operator"!=()

```
template<uint16_t Group, uint16_t Element, long long TVR>
bool gdcm::Attribute< Group, Element, TVR, VM::VM1 >::operator!= (
            const Attribute< Group, Element, TVR, VM::VM1 > & att ) const  [inline]
```

References gdcm::Attribute< Group, Element, TVR, TVM >::GetValues().

10.18.3.16 operator<()

```
template<uint16_t Group, uint16_t Element, long long TVR>
bool gdcm::Attribute< Group, Element, TVR, VM::VM1 >::operator< (
            const Attribute< Group, Element, TVR, VM::VM1 > & att ) const  [inline]
```

References gdcm::Attribute< Group, Element, TVR, TVM >::GetNumberOfValues(), and gdcm::Attribute< Group, Element, TVR, TVM >

10.18.3.17 operator==()

```
template<uint16_t Group, uint16_t Element, long long TVR>
bool gdcm::Attribute< Group, Element, TVR, VM::VM1 >::operator== (
            const Attribute< Group, Element, TVR, VM::VM1 > & att ) const  [inline]
```

References gdcm::Attribute< Group, Element, TVR, TVM >::GetValues().

10.18.3.18 Print()

```
template<uint16_t Group, uint16_t Element, long long TVR>
void gdcm::Attribute< Group, Element, TVR, VM::VM1 >::Print (
            std::ostream & os ) const  [inline]
```

10.18.3.19 Set()

```
template<uint16_t Group, uint16_t Element, long long TVR>
void gdcm::Attribute< Group, Element, TVR, VM::VM1 >::Set (
            DataSet const & ds )  [inline]
```

References gdcm::DataSet::GetDataElement().

10.18.3.20 SetByteValue()

```
template<uint16_t Group, uint16_t Element, long long TVR>
void gdcm::Attribute< Group, Element, TVR, VM::VM1 >::SetByteValue (
            const ByteValue * bv )  [inline], [protected]
```

References gdcm::ByteValue::GetLength(), and gdcm::ByteValue::GetPointer().

10.18.3.21 SetByteValueNoSwap()

```
template<uint16_t Group, uint16_t Element, long long TVR>
void gdcm::Attribute< Group, Element, TVR, VM::VM1 >::SetByteValueNoSwap (
            const ByteValue * bv )  [inline], [protected]
```

References gdcm::ByteValue::GetLength(), and gdcm::ByteValue::GetPointer().

10.18.3.22 SetFromDataElement()

```
template<uint16_t Group, uint16_t Element, long long TVR>
void gdcm::Attribute< Group, Element, TVR, VM::VM1 >::SetFromDataElement (
            DataElement const & de )  [inline]
```

References gdcm::DataElement::GetByteValue(), gdcm::DataElement::GetTag(), gdcm::DataElement::GetVR(), and gdcm::DataElement::IsEmpty().

10.18.3.23 SetFromDataSet()

```
template<uint16_t Group, uint16_t Element, long long TVR>
void gdcm::Attribute< Group, Element, TVR, VM::VM1 >::SetFromDataSet (
            DataSet const & ds )  [inline]
```

References gdcm::DataSet::FindDataElement(), and gdcm::DataSet::GetDataElement().

10.18.3.24 SetValue()

```
template<uint16_t Group, uint16_t Element, long long TVR>
void gdcm::Attribute< Group, Element, TVR, VM::VM1 >::SetValue (
            ArrayType v )  [inline]
```

10.18.4 Member Data Documentation

10.18.4.1 Internal

```
template<uint16_t Group, uint16_t Element, long long TVR>
ArrayType gdcm::Attribute< Group, Element, TVR, VM::VM1 >::Internal
```

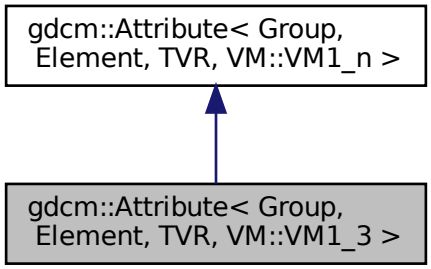
The documentation for this class was generated from the following file:

- gdcmAttribute.h

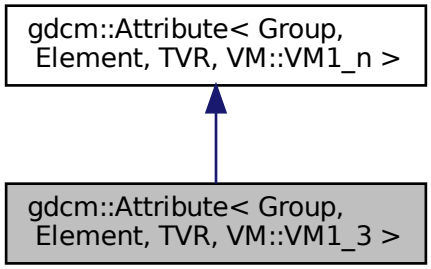
10.19 gdcm::Attribute< Group, Element, TVR, VM::VM1_3 > Class Template Reference

`#include <gdcmAttribute.h>`

Inheritance diagram for gdcm::Attribute< Group, Element, TVR, VM::VM1_3 >:



Collaboration diagram for gdcm::Attribute< Group, Element, TVR, VM::VM1_3 >:



Public Member Functions

- VM GetVM () const

Additional Inherited Members

10.19.1 Member Function Documentation

10.19.1.1 GetVM()

```
template<uint16_t Group, uint16_t Element, long long TVR>
VM gdcm::Attribute< Group, Element, TVR, VM::VM1_3 >::GetVM ( ) const  [inline]
```

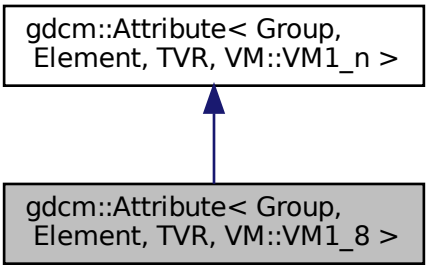
The documentation for this class was generated from the following file:

- gdcmAttribute.h

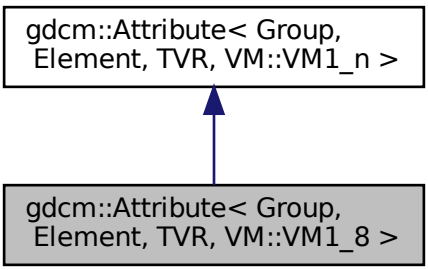
10.20 gdcm::Attribute< Group, Element, TVR, VM::VM1_8 > Class Template Reference

`#include <gdcmAttribute.h>`

Inheritance diagram for gdcm::Attribute< Group, Element, TVR, VM::VM1_8 >:



Collaboration diagram for gdcm::Attribute< Group, Element, TVR, VM::VM1_8 >:



Public Member Functions

- VM GetVM () const

Additional Inherited Members

10.20.1 Member Function Documentation

10.20.1.1 GetVM()

```
template<uint16_t Group, uint16_t Element, long long TVR>
VM gdcm::Attribute< Group, Element, TVR, VM::VM1_8 >::GetVM ( ) const  [inline]
```

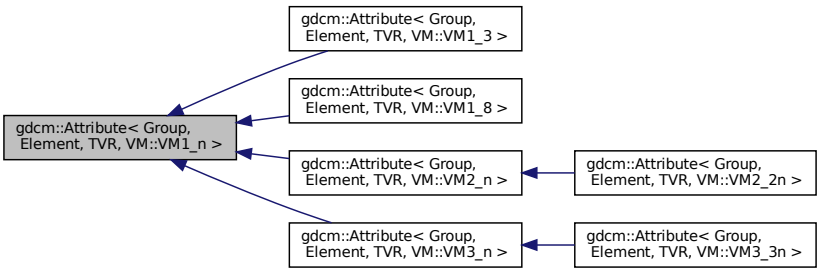
The documentation for this class was generated from the following file:

- gdcmAttribute.h

10.21 gdcm::Attribute< Group, Element, TVR, VM::VM1_n > Class Template Reference

```
#include <gdcmAttribute.h>
```

Inheritance diagram for gdcm::Attribute< Group, Element, TVR, VM::VM1_n >:



Public Types

- typedef VRToType< TVR >::Type ArrayType

Public Member Functions

- Attribute ()
- ~Attribute ()
- GDCM_STATIC_ASSERT (((((VR::VRType) TVR &VR::VR_VM1) &&((VM::VMType) TagToType< Group, Element >::VMType==VM::VM1))||!((VR::VRType) TVR &VR::VR_VM1)))
- GDCM_STATIC_ASSERT (((VR::VRType) TVR &(VR::VRType)(TagToType< Group, Element >::VRType)))
- GDCM_STATIC_ASSERT ((VM::VM1_n &(VM::VMType)(TagToType< Group, Element >::VMType)))
- DataElement GetAsDataElement () const
- unsigned int GetNumberOfValues () const
- ArrayType & GetValue (unsigned int idx=0)
- ArrayType const & GetValue (unsigned int idx=0) const
- const ArrayType * GetValues () const
- ArrayType & operator[ ] (unsigned int idx)
- ArrayType const & operator[ ] (unsigned int idx) const
- void Print (std::ostream &os) const
- void Set (DataSet const &ds)
- void SetFromDataElement (DataElement const &de)
- void SetFromDataSet (DataSet const &ds)
- void SetNumberOfValues (unsigned int numel)
- void SetValue (ArrayType v)
- void SetValue (unsigned int idx, ArrayType v)
- void SetValues (const ArrayType *array, unsigned int numel, bool own=false)

Static Public Member Functions

- static VM GetDictVM ()
- static VR GetDictVR ()
- static Tag GetTag ()
- static VM GetVM ()
- static VR GetVR ()

Protected Member Functions

- void SetByteValue (const ByteValue *bv)

10.21.1 Member Typedef Documentation

10.21.1.1 ArrayType

```
template<uint16_t Group, uint16_t Element, long long TVR>
typedef VRToType<TVR>::Type gdcm::Attribute< Group, Element, TVR, VM::VM1_n >::ArrayType
```

10.21.2 Constructor & Destructor Documentation

10.21.2.1 Attribute()

```
template<uint16_t Group, uint16_t Element, long long TVR>
gdcm::Attribute< Group, Element, TVR, VM::VM1_n >::Attribute ( )  [inline], [explicit]
```

10.21.2.2 ~Attribute()

```
template<uint16_t Group, uint16_t Element, long long TVR>
gdcm::Attribute< Group, Element, TVR, VM::VM1_n >::~Attribute ( )  [inline]
```

10.21.3 Member Function Documentation

10.21.3.1 GDCM_STATIC_ASSERT() [1/3]

```
template<uint16_t Group, uint16_t Element, long long TVR>
gdcm::Attribute< Group, Element, TVR, VM::VM1_n >::GDCM_STATIC_ASSERT (
             ((((VR::VRType) TVR &VR::VR_VM1) &&((VM::VMType) TagToType< Group, Element >::VMType==VM::VM1))||!((V
TVR &VR::VR_VM1))  )
```

10.21.3.2 GDCM_STATIC_ASSERT() [2/3]

```
template<uint16_t Group, uint16_t Element, long long TVR>
gdcm::Attribute< Group, Element, TVR, VM::VM1_n >::GDCM_STATIC_ASSERT (
             ((VR::VRType) TVR &(VR::VRType)(TagToType< Group, Element >::VRType))  )
```

10.21.3.3 GDCM_STATIC_ASSERT() [3/3]

```
template<uint16_t Group, uint16_t Element, long long TVR>
gdcm::Attribute< Group, Element, TVR, VM::VM1_n >::GDCM_STATIC_ASSERT (
             (VM::VM1_n &(VM::VMType)(TagToType< Group, Element >::VMType))  )
```

10.21.3.4 GetAsDataElement()

```
template<uint16_t Group, uint16_t Element, long long TVR>
DataElement gdcm::Attribute< Group, Element, TVR, VM::VM1_n >::GetAsDataElement ( ) const  [inline]
```

References gdcm::DataElement::GetVR(), gdcm::DataElement::SetByteValue(), and gdcm::DataElement::SetVR().

10.21.3.5 GetDictVM()

```
template<uint16_t Group, uint16_t Element, long long TVR>
static VM gdcm::Attribute< Group, Element, TVR, VM::VM1_n >::GetDictVM ( )  [inline], [static]
```

10.21.3.6 GetDictVR()

```
template<uint16_t Group, uint16_t Element, long long TVR>
static VR gdcm::Attribute< Group, Element, TVR, VM::VM1_n >::GetDictVR ( )  [inline], [static]
```

10.21.3.7 GetNumberOfValues()

```
template<uint16_t Group, uint16_t Element, long long TVR>
unsigned int gdcm::Attribute< Group, Element, TVR, VM::VM1_n >::GetNumberOfValues ( ) const  [inline]
```

10.21.3.8 GetTag()

```
template<uint16_t Group, uint16_t Element, long long TVR>
static Tag gdcm::Attribute< Group, Element, TVR, VM::VM1_n >::GetTag ( )  [inline], [static]
```

10.21.3.9 GetValue() [1/2]

```
template<uint16_t Group, uint16_t Element, long long TVR>
ArrayType & gdcm::Attribute< Group, Element, TVR, VM::VM1_n >::GetValue (
            unsigned int idx = 0 )  [inline]
```

10.21.3.10 GetValue() [2/2]

```
template<uint16_t Group, uint16_t Element, long long TVR>
ArrayType const & gdcm::Attribute< Group, Element, TVR, VM::VM1_n >::GetValue (
            unsigned int idx = 0 ) const  [inline]
```

10.21.3.11 GetValues()

```
template<uint16_t Group, uint16_t Element, long long TVR>
const ArrayType * gdcm::Attribute< Group, Element, TVR, VM::VM1_n >::GetValues ( ) const  [inline]
```

10.21.3.12 GetVM()

```
template<uint16_t Group, uint16_t Element, long long TVR>
static VM gdcm::Attribute< Group, Element, TVR, VM::VM1_n >::GetVM ( )  [inline], [static]
```

10.21.3.13 GetVR()

```
template<uint16_t Group, uint16_t Element, long long TVR>
static VR gdcm::Attribute< Group, Element, TVR, VM::VM1_n >::GetVR ( )  [inline], [static]
```

10.21.3.14 operator[]() [1/2]

```
template<uint16_t Group, uint16_t Element, long long TVR>
ArrayType & gdcm::Attribute< Group, Element, TVR, VM::VM1_n >::operator[] (
            unsigned int idx )  [inline]
```

10.21.3.15 operator[]() [2/2]

```
template<uint16_t Group, uint16_t Element, long long TVR>
ArrayType const & gdcm::Attribute< Group, Element, TVR, VM::VM1_n >::operator[] (
            unsigned int idx ) const  [inline]
```

10.21.3.16 Print()

```
template<uint16_t Group, uint16_t Element, long long TVR>
void gdcm::Attribute< Group, Element, TVR, VM::VM1_n >::Print (
            std::ostream & os ) const  [inline]
```

10.21.3.17 Set()

```
template<uint16_t Group, uint16_t Element, long long TVR>
void gdcm::Attribute< Group, Element, TVR, VM::VM1_n >::Set (
            DataSet const & ds )  [inline]
```

References gdcm::DataSet::GetDataElement().

10.21.3.18 SetByteValue()

```
template<uint16_t Group, uint16_t Element, long long TVR>
void gdcm::Attribute< Group, Element, TVR, VM::VM1_n >::SetByteValue (
            const ByteValue * bv )  [inline], [protected]
```

References gdcm::ByteValue::GetLength(), and gdcm::ByteValue::GetPointer().

10.21.3.19 SetFromDataElement()

```
template<uint16_t Group, uint16_t Element, long long TVR>
void gdcm::Attribute< Group, Element, TVR, VM::VM1_n >::SetFromDataElement (
            DataElement const & de )  [inline]
```

References gdcm::DataElement::GetByteValue(), gdcm::DataElement::GetTag(), gdcm::DataElement::GetVR(), and gdcm::DataElement::IsEmpty().

10.21.3.20 SetFromDataSet()

```
template<uint16_t Group, uint16_t Element, long long TVR>
void gdcm::Attribute< Group, Element, TVR, VM::VM1_n >::SetFromDataSet (
            DataSet const & ds )  [inline]
```

References gdcm::DataSet::FindDataElement(), and gdcm::DataSet::GetDataElement().

10.21.3.21 SetNumberOfValues()

```
template<uint16_t Group, uint16_t Element, long long TVR>
void gdcm::Attribute< Group, Element, TVR, VM::VM1_n >::SetNumberOfValues (
            unsigned int numel )  [inline]
```

10.21.3.22 SetValue() [1/2]

```
template<uint16_t Group, uint16_t Element, long long TVR>
void gdcm::Attribute< Group, Element, TVR, VM::VM1_n >::SetValue (
            ArrayType v )  [inline]
```

References SetValue().

Referenced by SetValue().

10.21.3.23 SetValue() [2/2]

```
template<uint16_t Group, uint16_t Element, long long TVR>
void gdcm::Attribute< Group, Element, TVR, VM::VM1_n >::SetValue (
            unsigned int idx,
            ArrayType v )  [inline]
```

10.21.3.24 SetValues()

```
template<uint16_t Group, uint16_t Element, long long TVR>
void gdcm::Attribute< Group, Element, TVR, VM::VM1_n >::SetValues (
            const ArrayType * array,
            unsigned int numel,
            bool own = false )  [inline]
```

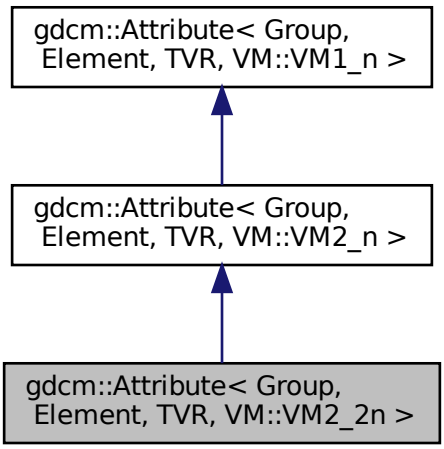
The documentation for this class was generated from the following file:

- gdcmAttribute.h

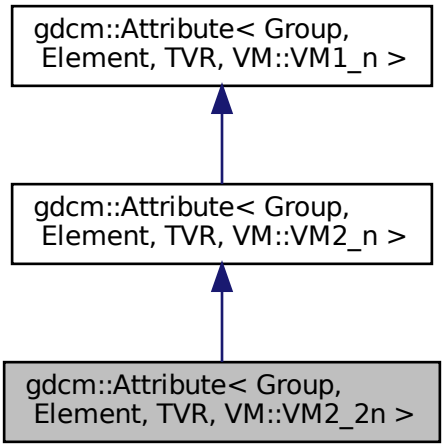
10.22 gdcm::Attribute< Group, Element, TVR, VM::VM2_2n > Class Template Reference

`#include <gdcmAttribute.h>`

Inheritance diagram for gdcm::Attribute< Group, Element, TVR, VM::VM2_2n >:



Collaboration diagram for gdcm::Attribute< Group, Element, TVR, VM::VM2_2n >:



Static Public Member Functions

- static VM GetVM ()

Additional Inherited Members

10.22.1 Member Function Documentation

10.22.1.1 GetVM()

```
template<uint16_t Group, uint16_t Element, long long TVR>
static VM gdcm::Attribute< Group, Element, TVR, VM::VM2_2n >::GetVM ( )  [inline], [static]
```

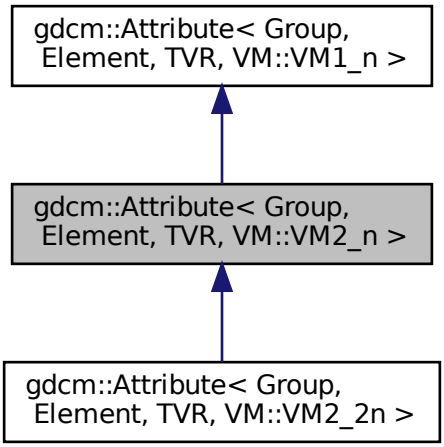
The documentation for this class was generated from the following file:

- gdcmAttribute.h

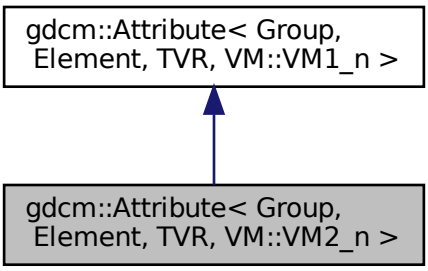
10.23 gdcm::Attribute< Group, Element, TVR, VM::VM2_n > Class Template Reference

```
#include <gdcmAttribute.h>
```

Inheritance diagram for gdcm::Attribute< Group, Element, TVR, VM::VM2_n >:



Collaboration diagram for gdcm::Attribute< Group, Element, TVR, VM::VM2_n >:



Public Member Functions

- VM GetVM () const

Additional Inherited Members

10.23.1 Member Function Documentation

10.23.1.1 GetVM()

```
template<uint16_t Group, uint16_t Element, long long TVR>
VM gdcm::Attribute< Group, Element, TVR, VM::VM2_n >::GetVM ( ) const  [inline]
```

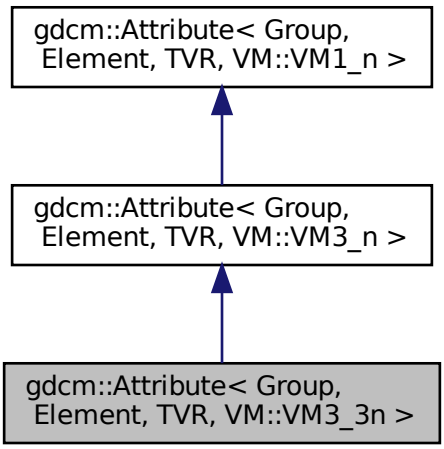
The documentation for this class was generated from the following file:

- gdcmAttribute.h

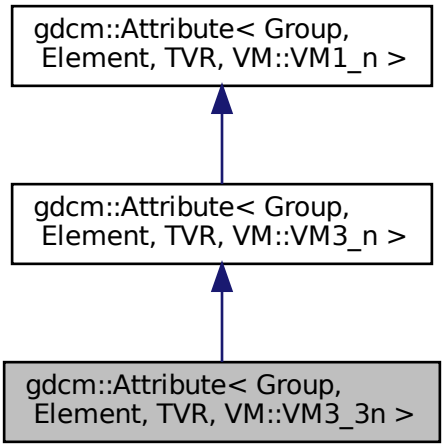
10.24 gdcm::Attribute< Group, Element, TVR, VM::VM3_3n > Class Template Reference

`#include <gdcmAttribute.h>`

Inheritance diagram for gdcm::Attribute< Group, Element, TVR, VM::VM3_3n >:



Collaboration diagram for gdcm::Attribute< Group, Element, TVR, VM::VM3_3n >:



Static Public Member Functions

- static VM GetVM ()

Additional Inherited Members

10.24.1 Member Function Documentation

10.24.1.1 GetVM()

```
template<uint16_t Group, uint16_t Element, long long TVR>
static VM gdcm::Attribute< Group, Element, TVR, VM::VM3_3n >::GetVM ( )  [inline], [static]
```

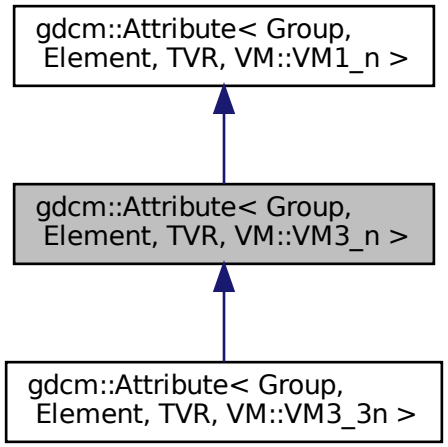
The documentation for this class was generated from the following file:

- gdcmAttribute.h

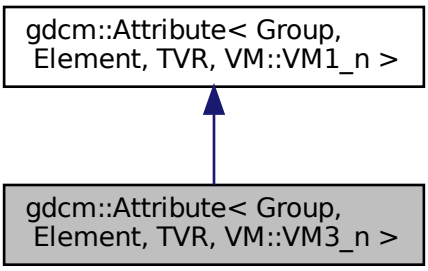
10.25 gdcm::Attribute< Group, Element, TVR, VM::VM3_n > Class Template Reference

```
#include <gdcmAttribute.h>
```

Inheritance diagram for gdcm::Attribute< Group, Element, TVR, VM::VM3_n >:



Collaboration diagram for gdcm::Attribute< Group, Element, TVR, VM::VM3_n >:



Static Public Member Functions

- static VM GetVM ()

Additional Inherited Members

10.25.1 Member Function Documentation

10.25.1.1 GetVM()

```
template<uint16_t Group, uint16_t Element, long long TVR>
static VM gdcm::Attribute< Group, Element, TVR, VM::VM3_n >::GetVM ( )  [inline], [static]
```

The documentation for this class was generated from the following file:

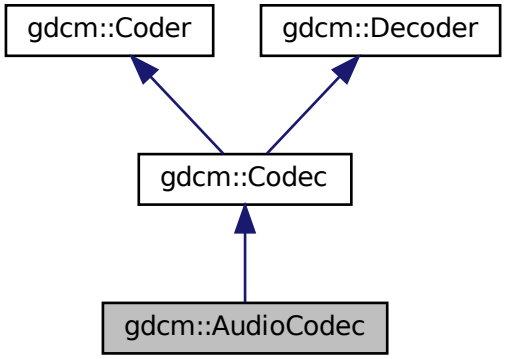
- gdcmAttribute.h

10.26 gdcm::AudioCodec Class Reference

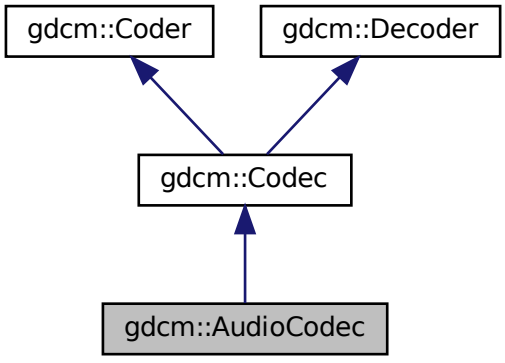
AudioCodec.

`#include <gdcmAudioCodec.h>`

Inheritance diagram for gdcm::AudioCodec:



Collaboration diagram for gdcm::AudioCodec:



Public Member Functions

- AudioCodec ()
- ~AudioCodec () override
- bool CanCode (TransferSyntax const &) const override

  Return whether this coder support this transfer syntax (can code it)
- bool CanDecode (TransferSyntax const &) const override

  Return whether this decoder support this transfer syntax (can decode it)
- bool Decode (DataElement const &is, DataElement &os) override

  Decode.

Additional Inherited Members

10.26.1 Detailed Description

AudioCodec.

10.26.2 Constructor & Destructor Documentation

10.26.2.1 AudioCodec()

```
gdcm::AudioCodec::AudioCodec ( )
```

10.26.2.2 ~AudioCodec()

```
gdcm::AudioCodec::~AudioCodec ( )  [override]
```

10.26.3 Member Function Documentation

10.26.3.1 CanCode()

```
bool gdcm::AudioCodec::CanCode (
            TransferSyntax const &  ) const  [inline], [override], [virtual]
```

Return whether this coder support this transfer syntax (can code it)

Implements gdcm::Coder.

10.26.3.2 CanDecode()

```
bool gdcm::AudioCodec::CanDecode (
            TransferSyntax const &  ) const  [inline], [override], [virtual]
```

Return whether this decoder support this transfer syntax (can decode it)

Implements gdcm::Decoder.

10.26.3.3 Decode()

```
bool gdcm::AudioCodec::Decode (
            DataElement const & ,
            DataElement &  )  [override], [virtual]
```

Decode.

Reimplemented from gdcm::Decoder.

The documentation for this class was generated from the following file:

- gdcmAudioCodec.h

10.27 gdcm::Base64 Class Reference

Class for Base64.

```
#include <gdcmBase64.h>
```

Public Member Functions

- Base64 (const Base64 &)=delete
- void operator= (const Base64 &)=delete

Static Public Member Functions

- static size_t Decode (char *dst, size_t dlen, const char *src, size_t slen)

  Decode a base64-formatted buffer.
- static size_t Encode (char *dst, size_t dlen, const char *src, size_t slen)

  Encode a buffer into base64 format.
- static size_t GetDecodeLength (const char *src, size_t len)
- static size_t GetEncodeLength (const char *src, size_t srclen)

10.27.1 Detailed Description

Class for Base64.

10.27.2 Constructor & Destructor Documentation

10.27.2.1 Base64()

```
gdcm::Base64::Base64 (
            const Base64 &  )  [delete]
```

10.27.3 Member Function Documentation

10.27.3.1 Decode()

```
static size_t gdcm::Base64::Decode (
            char * dst,
            size_t dlen,
            const char * src,
            size_t slen )  [static]
```

Decode a base64-formatted buffer.

Parameters

dst	destination buffer
dlen	size of the buffer
src	source buffer
slen	amount of data to be decoded

Returns

0 if not successful, size of decoded otherwise

Examples

DumpExamCard.cxx, and DumpSiemensBase64.cxx.

10.27.3.2 Encode()

```
static size_t gdcm::Base64::Encode (
            char * dst,
            size_t dlen,
            const char * src,
            size_t slen )  [static]
```

Encode a buffer into base64 format.

Parameters

dst	destination buffer
dlen	size of the buffer
src	source buffer
slen	amount of data to be encoded

Returns

0 if not successful, size of encoded otherwise

10.27.3.3 GetDecodeLength()

```
static size_t gdcm::Base64::GetDecodeLength (
            const char * src,
            size_t len )  [static]
```

Call this function to obtain the required buffer size

Examples

DumpExamCard.cxx, and DumpSiemensBase64.cxx.

10.27.3.4 GetEncodeLength()

```
static size_t gdcm::Base64::GetEncodeLength (
            const char * src,
            size_t srclen )  [static]
```

Call this function to obtain the required buffer size

10.27.3.5 operator=()

```
void gdcm::Base64::operator= (
            const Base64 &  )  [delete]
```

The documentation for this class was generated from the following file:

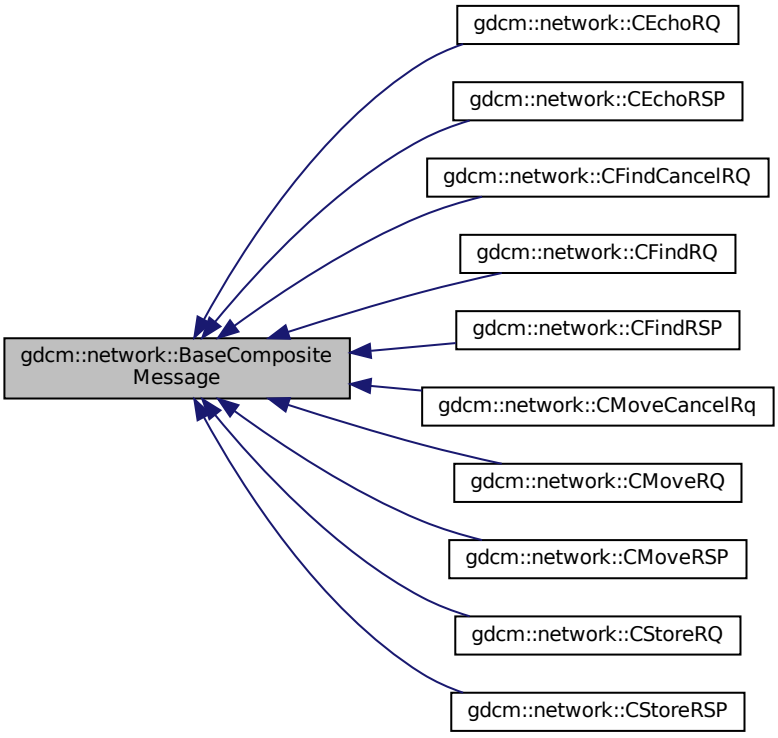
- gdcmBase64.h

10.28 gdcm::network::BaseCompositeMessage Class Reference

BaseCompositeMessage.

`#include <gdcmBaseCompositeMessage.h>`

Inheritance diagram for gdcm::network::BaseCompositeMessage:



Public Member Functions

- virtual ~BaseCompositeMessage ()=default
- virtual std::vector< PresentationDataValue > ConstructPDV (const ULConnection &inConnection, const BaseRootQuery *inRootQuery)=0

10.28.1 Detailed Description

BaseCompositeMessage.

The Composite events described in section 3.7-2009 of the DICOM standard all use their own messages. These messages are constructed using Presentation Data Values, from section 3.8-2009 of the standard, and then fill in appropriate values in their datasets.

So, for the five composites:

- C-ECHO
- C-FIND
- C-MOVE
- C-GET
- C-STORE there are a series of messages. However, all of these messages are obtained as part of a PDataPDU, and all have to be placed there. Therefore, since they all have shared functionality and construction tropes, that will be put into a base class. Further, the base class will be then returned by the factory class, gdcmComposite↩ PDUFactory.

This is an abstract class. It cannot be instantiated on its own.

10.28.2 Constructor & Destructor Documentation

10.28.2.1 ~BaseCompositeMessage()

```
virtual gdcm::network::BaseCompositeMessage::~BaseCompositeMessage ( )  [virtual], [default]
```

10.28.3 Member Function Documentation

10.28.3.1 ConstructPDV()

```
virtual std::vector< PresentationDataValue > gdcm::network::BaseCompositeMessage::ConstructPDV (
            const ULConnection & inConnection,
            const BaseRootQuery * inRootQuery )  [pure virtual]
```

Implemented in gdcm::network::CEchoRQ, gdcm::network::CFindRQ, and gdcm::network::CMoveRQ.

The documentation for this class was generated from the following file:

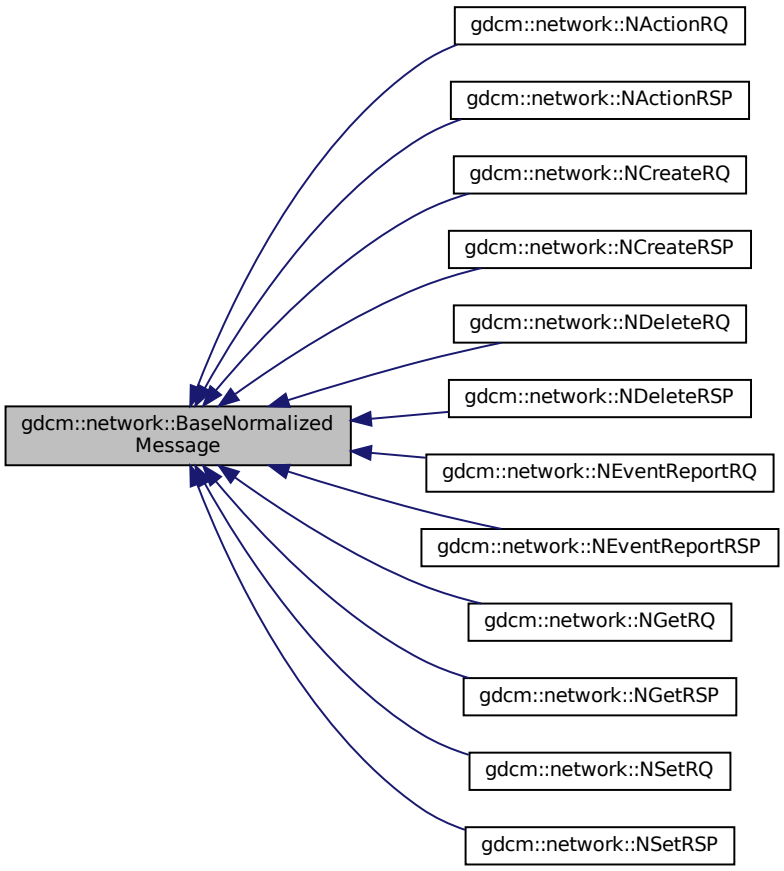
- gdcmBaseCompositeMessage.h

10.29 gdcm::network::BaseNormalizedMessage Class Reference

BaseNormalizedMessage.

`#include <gdcmBaseNormalizedMessage.h>`

Inheritance diagram for gdcm::network::BaseNormalizedMessage:



Public Member Functions

- virtual ~BaseNormalizedMessage ()=default
- virtual std::vector< PresentationDataValue > ConstructPDV (const ULConnection &inConnection, const BaseQuery *inQuery)=0

10.29.1 Detailed Description

BaseNormalizedMessage.

The Normalized events described in section 3.7-2011 of the DICOM standard all use their own messages. These messages are constructed using Presentation Data Values, from section 3.8-2011 of the standard, and then fill in appropriate values in their datasets.

So, for the five normalized:

- N-ACTION
- N-CREATE
- N-DELETE
- N-EVENT
- N-GET
- N-SET there are a series of messages. However, all of these messages are obtained as part of a PData↩ PDU, and all have to be placed there. Therefore, since they all have shared functionality and construction tropes, that will be put into a base class. Further, the base class will be then returned by the factory class, gdcmNormalizedMessageFactory.h.

This is an abstract class. It cannot be instantiated on its own.

10.29.2 Constructor & Destructor Documentation

10.29.2.1 ~BaseNormalizedMessage()

```
virtual gdcm::network::BaseNormalizedMessage::~BaseNormalizedMessage ( )  [virtual], [default]
```

10.29.3 Member Function Documentation

10.29.3.1 ConstructPDV()

```
virtual std::vector< PresentationDataValue > gdcm::network::BaseNormalizedMessage::ConstructPDV (
            const ULConnection & inConnection,
            const BaseQuery * inQuery )  [pure virtual]
```

Implemented in gdcm::network::NActionRQ, gdcm::network::NCreateRQ, gdcm::network::NDeleteRQ, gdcm::network::NEventReportRQ, gdcm::network::NGetRQ, and gdcm::network::NSetRQ.

The documentation for this class was generated from the following file:

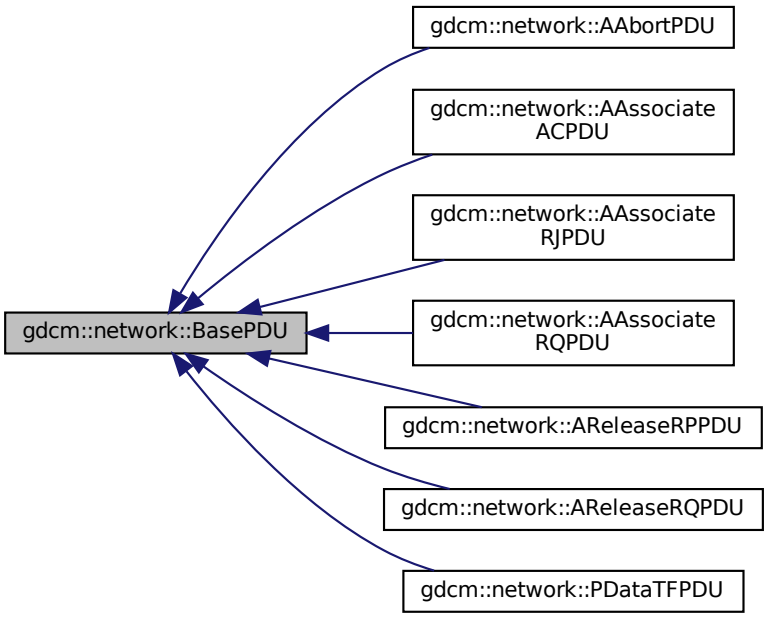
- gdcmBaseNormalizedMessage.h

10.30 gdcm::network::BasePDU Class Reference

BasePDU.

`#include <gdcmBasePDU.h>`

Inheritance diagram for gdcm::network::BasePDU:



Public Member Functions

- virtual ~BasePDU ()=default
- virtual bool IsLastFragment () const =0
- virtual void Print (std::ostream &os) const =0
- virtual std::istream & Read (std::istream &is)=0
- virtual size_t Size () const =0
- virtual const std::ostream & Write (std::ostream &os) const =0

10.30.1 Detailed Description

BasePDU.

base class for PDUs

all PDUs start with the first ten bytes as specified: 01 PDU type 02 reserved 3-6 PDU Length (unsigned) 7-10 variable

on some, 7-10 are split (7-8 as protocol version in Associate-RQ, for instance, while associate-rj splits those four bytes differently).

Also common to all the PDUs is their ability to read and write to a stream.

So, let's just get them all bunched together into one (abstract) class, shall we?

Why? 1) so that the ULEvent can have the PDU stored in it, since the event takes PDUs and not other class structures (other class structures get converted into PDUs) 2) to make reading PDUs in the event loop cleaner

10.30.2 Constructor & Destructor Documentation

10.30.2.1 ~BasePDU()

```
virtual gdcm::network::BasePDU::~BasePDU ( )  [virtual], [default]
```

10.30.3 Member Function Documentation

10.30.3.1 IsLastFragment()

```
virtual bool gdcm::network::BasePDU::IsLastFragment ( ) const  [pure virtual]
```

Implemented in gdcm::network::AAbortPDU, gdcm::network::AAssociateACPDU, gdcm::network::AAssociateRJPDU, gdcm::network::AAssociateRQPDU, gdcm::network::AReleaseRPPDU, gdcm::network::AReleaseRQPDU, and gdcm::network::PDataTFPDU.

10.30.3.2 Print()

```
virtual void gdcm::network::BasePDU::Print (
            std::ostream & os ) const  [pure virtual]
```

Implemented in gdcm::network::AAbortPDU, gdcm::network::AAssociateACPDU, gdcm::network::AAssociateRJPDU, gdcm::network::AAssociateRQPDU, gdcm::network::AReleaseRPPDU, gdcm::network::AReleaseRQPDU, and gdcm::network::PDataTFPDU.

10.30.3.3 Read()

```
virtual std::istream & gdcm::network::BasePDU::Read (
            std::istream & is )  [pure virtual]
```

Implemented in gdcm::network::AAbortPDU, gdcm::network::AAssociateACPDU, gdcm::network::AAssociateRJPDU, gdcm::network::AAssociateRQPDU, gdcm::network::AReleaseRPPDU, gdcm::network::AReleaseRQPDU, and gdcm::network::PDataTFPDU.

10.30.3.4 Size()

```
virtual size_t gdcm::network::BasePDU::Size ( ) const  [pure virtual]
```

Implemented in gdcm::network::AAbortPDU, gdcm::network::AAssociateACPDU, gdcm::network::AAssociateRJPDU, gdcm::network::AAssociateRQPDU, gdcm::network::AReleaseRPPDU, gdcm::network::AReleaseRQPDU, and gdcm::network::PDataTFPDU.

10.30.3.5 Write()

```
virtual const std::ostream & gdcm::network::BasePDU::Write (
            std::ostream & os ) const  [pure virtual]
```

Implemented in gdcm::network::AAbortPDU, gdcm::network::AAssociateACPDU, gdcm::network::AAssociateRJPDU, gdcm::network::AAssociateRQPDU, gdcm::network::AReleaseRPPDU, gdcm::network::AReleaseRQPDU, and gdcm::network::PDataTFPDU.

The documentation for this class was generated from the following file:

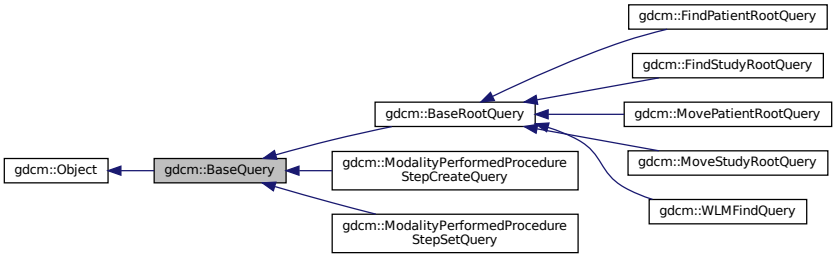
- gdcmBasePDU.h

10.31 gdcm::BaseQuery Class Reference

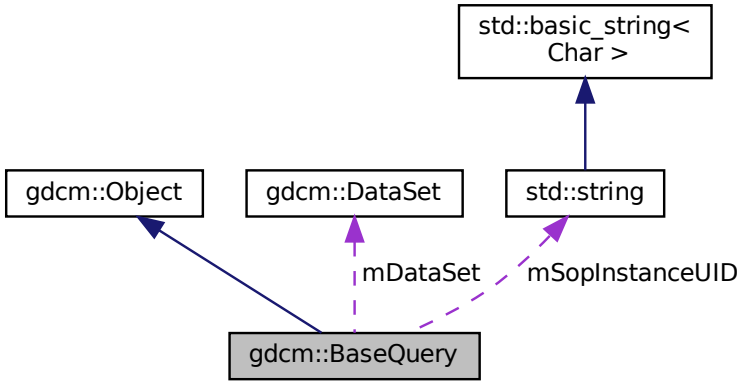
BaseQuery.

```
#include <gdcmBaseQuery.h>
```

Inheritance diagram for gdcm::BaseQuery:



Collaboration diagram for gdcm::BaseQuery:



Public Member Functions

- ~BaseQuery () override
- void AddQueryDataSet (const DataSet &ds)
- virtual UIDs::TSName GetAbstractSyntaxUID () const =0
- DataSet & GetQueryDataSet ()
- DataSet const & GetQueryDataSet () const

  Set/Get the internal representation of the query as a DataSet.
- std::string GetSOPInstanceUID () const
- void Print (std::ostream &os) const override
- void SetSearchParameter (const std::string &inKeyword, const std::string &inValue)
- void SetSearchParameter (const Tag &inTag, const std::string &inValue)
- void SetSOPInstanceUID (const std::string &iSopInstanceUID)
- virtual bool ValidateQuery (bool inStrict=true) const =0
- const std::ostream & WriteHelpFile (std::ostream &os)
- bool WriteQuery (const std::string &inFileName)

Protected Member Functions

- BaseQuery ()
- void SetSearchParameter (const Tag &inTag, const DictEntry &inDictEntry, const std::string &inValue)
- bool ValidDataSet (const DataSet &dataSetToValid, const DataSet &dataSetReference) const

Protected Attributes

- DataSet mDataSet
- std::string mSopInstanceUID

Friends

- class QueryFactory

10.31.1 Detailed Description

BaseQuery.

contains: a baseclass which will produce a dataset for all dimse messages

10.31.2 Constructor & Destructor Documentation

10.31.2.1 BaseQuery()

```
gdcm::BaseQuery::BaseQuery ( )  [protected]
```

10.31.2.2 ~BaseQuery()

```
gdcm::BaseQuery::~BaseQuery ( )  [override]
```

10.31.3 Member Function Documentation

10.31.3.1 AddQueryDataSet()

```
void gdcm::BaseQuery::AddQueryDataSet (
            const DataSet & ds )
```

10.31.3.2 GetAbstractSyntaxUID()

```
virtual UIDs::TSName gdcm::BaseQuery::GetAbstractSyntaxUID ( ) const  [pure virtual]
```

Implemented in gdcm::FindPatientRootQuery, gdcm::FindStudyRootQuery, gdcm::ModalityPerformedProcedureStepCreateQuery, gdcm::ModalityPerformedProcedureStepSetQuery, gdcm::MovePatientRootQuery, gdcm::MoveStudyRootQuery, and gdcm::WLMFindQuery.

10.31.3.3 GetQueryDataSet() [1/2]

```
DataSet & gdcm::BaseQuery::GetQueryDataSet ( )
```

10.31.3.4 GetQueryDataSet() [2/2]

```
DataSet const & gdcm::BaseQuery::GetQueryDataSet ( ) const
```

Set/Get the internal representation of the query as a DataSet.

10.31.3.5 GetSOPInstanceUID()

```
std::string gdcm::BaseQuery::GetSOPInstanceUID ( ) const  [inline]
```

10.31.3.6 Print()

```
void gdcm::BaseQuery::Print (
            std::ostream & os ) const  [override], [virtual]
```

Reimplemented from gdcm::Object.

10.31.3.7 SetSearchParameter() [1/3]

```
void gdcm::BaseQuery::SetSearchParameter (
            const std::string & inKeyword,
            const std::string & inValue )
```

10.31.3.8 SetSearchParameter() [2/3]

```
void gdcm::BaseQuery::SetSearchParameter (
            const Tag & inTag,
            const DictEntry & inDictEntry,
            const std::string & inValue )  [protected]
```

10.31.3.9 SetSearchParameter() [3/3]

```
void gdcm::BaseQuery::SetSearchParameter (
            const Tag & inTag,
            const std::string & inValue )
```

10.31.3.10 SetSOPInstanceUID()

```
void gdcm::BaseQuery::SetSOPInstanceUID (
            const std::string & iSopInstanceUID )  [inline]
```

10.31.3.11 ValidateQuery()

```
virtual bool gdcm::BaseQuery::ValidateQuery (
            bool inStrict = true ) const  [pure virtual]
```

Implemented in gdcm::FindPatientRootQuery, gdcm::FindStudyRootQuery, gdcm::ModalityPerformedProcedureStepCreateQuery, gdcm::ModalityPerformedProcedureStepSetQuery, gdcm::MovePatientRootQuery, gdcm::MoveStudyRootQuery, gdcm::WLMFindQuery, and gdcm::BaseRootQuery.

10.31.3.12 ValidDataSet()

```
bool gdcm::BaseQuery::ValidDataSet (
            const DataSet & dataSetToValid,
            const DataSet & dataSetReference ) const  [protected]
```

10.31.3.13 WriteHelpFile()

```
const std::ostream & gdcm::BaseQuery::WriteHelpFile (
            std::ostream & os )
```

10.31.3.14 WriteQuery()

```
bool gdcm::BaseQuery::WriteQuery (
            const std::string & inFileName )
```

10.31.4 Friends And Related Function Documentation

10.31.4.1 QueryFactory

```
friend class QueryFactory  [friend]
```

10.31.5 Member Data Documentation

10.31.5.1 mDataSet

```
DataSet gdcm::BaseQuery::mDataSet  [protected]
```

10.31.5.2 mSopInstanceUID

```
std::string gdcm::BaseQuery::mSopInstanceUID  [protected]
```

The documentation for this class was generated from the following file:

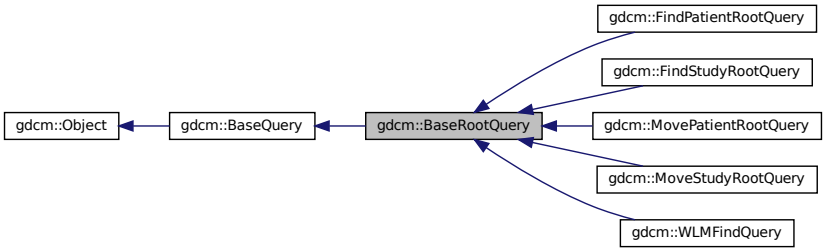
- gdcmBaseQuery.h

10.32 gdcm::BaseRootQuery Class Reference

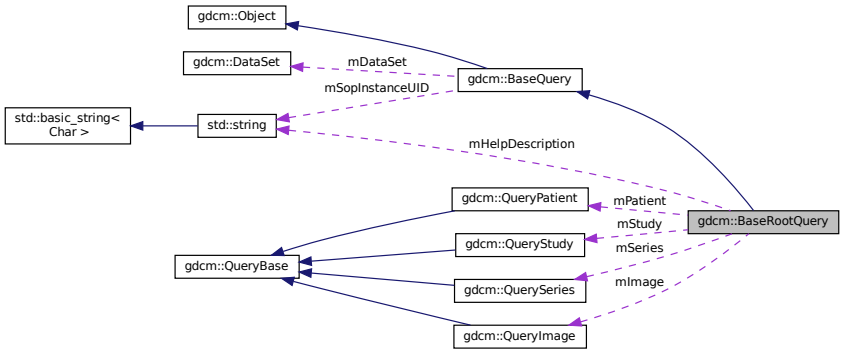
BaseRootQuery.

```
#include <gdcmBaseRootQuery.h>
```

Inheritance diagram for gdcm::BaseRootQuery:



Collaboration diagram for gdcm::BaseRootQuery:



Public Member Functions

- ~BaseRootQuery () override
- EQueryLevel GetQueryLevelFromQueryRoot (ERootType roottype)
- virtual std::vector< Tag > GetTagListByLevel (const EQueryLevel &inQueryLevel)=0
- virtual void InitializeDataSet (const EQueryLevel &inQueryLevel)=0
- bool ValidateQuery (bool inStrict=true) const override=0

Static Public Member Functions

- static QueryBase * Construct (ERootType inRootType, EQueryLevel qlevel)
- static int GetQueryLevelFromString (const char *str)
- static const char * GetQueryLevelString (EQueryLevel ql)

Protected Member Functions

- BaseRootQuery ()

Protected Attributes

- std::string mHelpDescription
- QueryImage mImage
- QueryPatient mPatient
- ERootType mRootType
- QuerySeries mSeries
- QueryStudy mStudy

Friends

- class QueryFactory

10.32.1 Detailed Description

BaseRootQuery.

contains: a baseclass which will produce a dataset for c-find and c-move with patient/study root

This class contains the functionality used in patient c-find and c-move queries. PatientRootQuery and StudyRootQuery derive from this class.

Namely: 1) list all tags associated with a particular query type 2) produce a query dataset via tag association

Eventually, it can be used to validate a particular dataset type.

The dataset held by this object (or, really, one of its derivates) should be passed to a c-find or c-move query.

10.32.2 Constructor & Destructor Documentation

10.32.2.1 BaseRootQuery()

```
gdcm::BaseRootQuery::BaseRootQuery ( )  [protected]
```

10.32.2.2 ~BaseRootQuery()

```
gdcm::BaseRootQuery::~BaseRootQuery ( )  [override]
```

10.32.3 Member Function Documentation

10.32.3.1 Construct()

```
static QueryBase * gdcm::BaseRootQuery::Construct (
            ERootType inRootType,
            EQueryLevel qlevel )  [static]
```

10.32.3.2 GetQueryLevelFromQueryRoot()

```
EQueryLevel gdcm::BaseRootQuery::GetQueryLevelFromQueryRoot (
            ERootType roottype )
```

10.32.3.3 GetQueryLevelFromString()

```
static int gdcm::BaseRootQuery::GetQueryLevelFromString (
            const char * str )  [static]
```

10.32.3.4 GetQueryLevelString()

```
static const char * gdcm::BaseRootQuery::GetQueryLevelString (
            EQueryLevel ql )  [static]
```

10.32.3.5 GetTagListByLevel()

```
virtual std::vector< Tag > gdcm::BaseRootQuery::GetTagListByLevel (
            const EQueryLevel & inQueryLevel )  [pure virtual]
```

this function will return all tags at a given query level, so that they maybe selected for searching. The boolean forFind is true if the query is a find query, or false for a move query.

Implemented in gdcm::FindPatientRootQuery, gdcm::FindStudyRootQuery, gdcm::MovePatientRootQuery, gdcm::MoveStudyRootQuery, and gdcm::WLMFindQuery.

10.32.3.6 InitializeDataSet()

```
virtual void gdcm::BaseRootQuery::InitializeDataSet (
            const EQueryLevel & inQueryLevel )  [pure virtual]
```

this function sets tag 8,52 to the appropriate value based on query level also fills in the right unique tags, as per the standard's requirements should allow for connection with dcmtk

Implemented in gdcm::FindPatientRootQuery, gdcm::FindStudyRootQuery, gdcm::MovePatientRootQuery, gdcm::MoveStudyRootQuery, and gdcm::WLMFindQuery.

10.32.3.7 ValidateQuery()

```
bool gdcm::BaseRootQuery::ValidateQuery (
            bool inStrict = true ) const  [override], [pure virtual]
```

have to be able to ensure that 0x8,0x52 is set (which will be true if InitializeDataSet is called...) that the level is appropriate (ie, not setting PATIENT for a study query that the tags in the query match the right level (either required, unique, optional) by default, this function checks to see if the query is for finding, which is more permissive than for moving. For moving, only the unique tags are allowed. 10 Jan 2011: adding in the 'strict' mode. according to the standard (at least, how I've read it), only tags for a particular level should be allowed in a particular query (ie, just series level tags in a series level query). However, it seems that dcm4chee doesn't share that interpretation. So, if 'inStrict' is false, then tags from the current level and all higher levels are now considered valid. So, if you're doing a non-strict series-level query, tags from the patient and study level can be passed along as well.

Implements gdcm::BaseQuery.

Implemented in gdcm::FindPatientRootQuery, gdcm::FindStudyRootQuery, gdcm::MovePatientRootQuery, gdcm::MoveStudyRootQuery, and gdcm::WLMFindQuery.

10.32.4 Friends And Related Function Documentation

10.32.4.1 QueryFactory

```
friend class QueryFactory  [friend]
```

10.32.5 Member Data Documentation

10.32.5.1 mHelpDescription

```
std::string gdcm::BaseRootQuery::mHelpDescription  [protected]
```

10.32.5.2 mImage

```
QueryImage gdcm::BaseRootQuery::mImage  [protected]
```

10.32.5.3 mPatient

```
QueryPatient gdcm::BaseRootQuery::mPatient  [protected]
```

10.32.5.4 mRootType

```
ERootType gdcm::BaseRootQuery::mRootType  [protected]
```

10.32.5.5 mSeries

```
QuerySeries gdcm::BaseRootQuery::mSeries  [protected]
```

10.32.5.6 mStudy

```
QueryStudy gdcm::BaseRootQuery::mStudy  [protected]
```

The documentation for this class was generated from the following file:

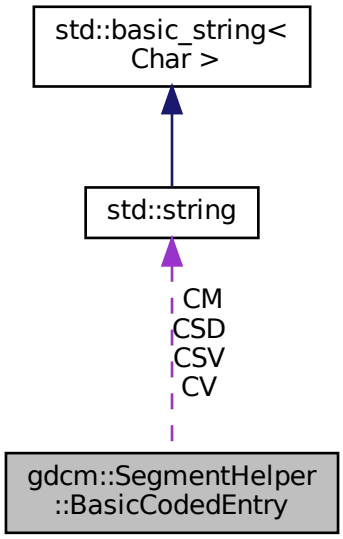
- gdcmBaseRootQuery.h

10.33 gdcm::SegmentHelper::BasicCodedEntry Struct Reference

This structure defines a basic coded entry with all of its attributes.

`#include <gdcmSegmentHelper.h>`

Collaboration diagram for gdcm::SegmentHelper::BasicCodedEntry:



Public Member Functions

- BasicCodedEntry ()
  Constructor.
- BasicCodedEntry (const char *a_CV, const char *a_CSD, const char *a_CM)
  constructor which defines type 1 attributes.
- BasicCodedEntry (const char *a_CV, const char *a_CSD, const char *a_CSV, const char *a_CM)
  constructor which defines attributes.
- bool IsEmpty (const bool checkOptionalAttributes=false) const
  Check if each attributes of the basic coded entry is defined.

Public Attributes

- std::string CM

  Coding Scheme Version attribute.
- std::string CSD

  Code Value attribute.
- std::string CSV

  Coding Scheme Designator attribute.
- std::string CV

10.33.1 Detailed Description

This structure defines a basic coded entry with all of its attributes.

See also

PS 3.3 section 8.8.

10.33.2 Constructor & Destructor Documentation

10.33.2.1 BasicCodedEntry() [1/3]

```
gdcm::SegmentHelper::BasicCodedEntry::BasicCodedEntry ( )  [inline]
```

Constructor.

10.33.2.2 BasicCodedEntry() [2/3]

```
gdcm::SegmentHelper::BasicCodedEntry::BasicCodedEntry (
            const char * a_CV,
            const char * a_CSD,
            const char * a_CM )  [inline]
```

constructor which defines type 1 attributes.

10.33.2.3 BasicCodedEntry() [3/3]

```
gdcm::SegmentHelper::BasicCodedEntry::BasicCodedEntry (
            const char * a_CV,
            const char * a_CSD,
            const char * a_CSV,
            const char * a_CM )  [inline]
```

constructor which defines attributes.

10.33.3 Member Function Documentation

10.33.3.1 IsEmpty()

```
bool gdcm::SegmentHelper::BasicCodedEntry::IsEmpty (
            const bool checkOptionalAttributes = false ) const
```

Check if each attributes of the basic coded entry is defined.

Parameters

checkOptionalAttributes	Check also type 1C attributes.

10.33.4 Member Data Documentation

10.33.4.1 CM

```
std::string gdcm::SegmentHelper::BasicCodedEntry::CM
```

Coding Scheme Version attribute.

10.33.4.2 CSD

```
std::string gdcm::SegmentHelper::BasicCodedEntry::CSD
```

Code Value attribute.

10.33.4.3 CSV

```
std::string gdcm::SegmentHelper::BasicCodedEntry::CSV
```

Coding Scheme Designator attribute.

10.33.4.4 CV

```
std::string gdcm::SegmentHelper::BasicCodedEntry::CV
```

The documentation for this struct was generated from the following file:

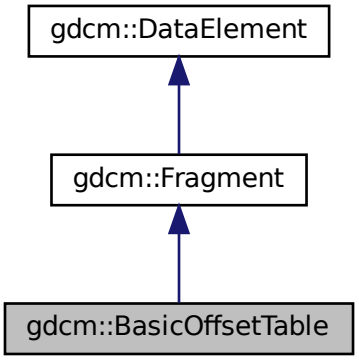
- gdcmSegmentHelper.h

10.34 gdcm::BasicOffsetTable Class Reference

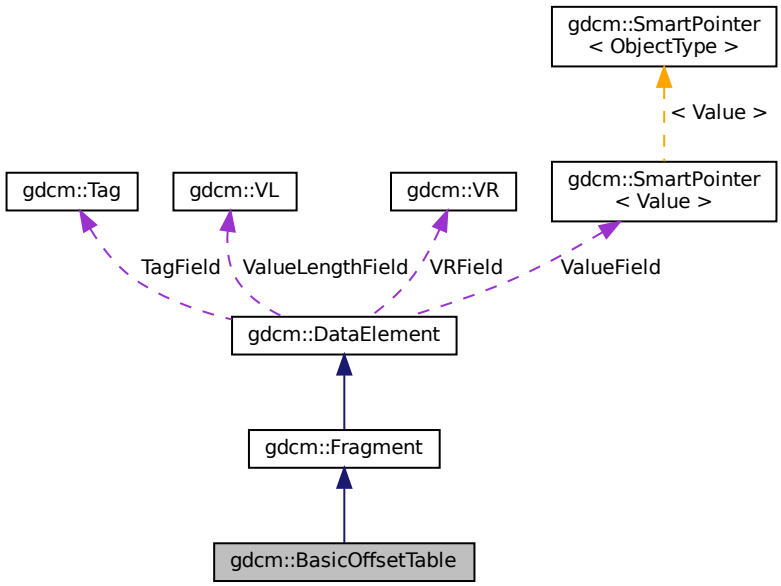
Class to represent a BasicOffsetTable.

```
#include <gdcmBasicOffsetTable.h>
```

Inheritance diagram for gdcm::BasicOffsetTable:



Collaboration diagram for gdcm::BasicOffsetTable:



Public Member Functions

- BasicOffsetTable ()
- template<typename TSwap >
  std::istream & Read (std::istream &is)

Friends

- std::ostream & operator<< (std::ostream &os, const BasicOffsetTable &val)

Additional Inherited Members

10.34.1 Detailed Description

Class to represent a BasicOffsetTable.

10.34.2 Constructor & Destructor Documentation

10.34.2.1 BasicOffsetTable()

```
gdcm::BasicOffsetTable::BasicOffsetTable ( )  [inline]
```

10.34.3 Member Function Documentation

10.34.3.1 Read()

```
template<typename TSwap >
std::istream & gdcm::BasicOffsetTable::Read (
            std::istream & is )  [inline]
```

References gdcmAssertAlwaysMacro, and gdcm::ParseException::SetLastElement().

10.34.4 Friends And Related Function Documentation

10.34.4.1 operator<<

```
std::ostream & operator<< (
            std::ostream & os,
            const BasicOffsetTable & val )  [friend]
```

The documentation for this class was generated from the following file:

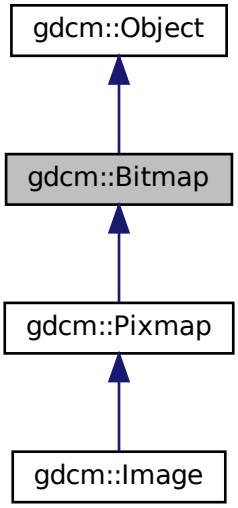
- gdcmBasicOffsetTable.h

10.35 gdcm::Bitmap Class Reference

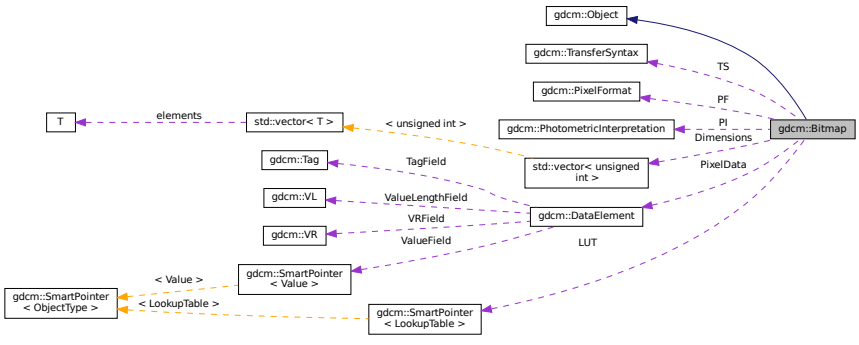
Bitmap class.

`#include <gdcmBitmap.h>`

Inheritance diagram for gdcm::Bitmap:



Collaboration diagram for gdcm::Bitmap:



Public Member Functions

- Bitmap ()
- ~Bitmap () override
- virtual bool AreOverlaysInPixelData () const
- void Clear ()
- bool GetBuffer (char *buffer) const

  Access the raw data.
- unsigned long GetBufferLength () const
- unsigned int GetColumns () const
- DataElement & GetDataElement ()
- const DataElement & GetDataElement () const
- unsigned int GetDimension (unsigned int idx) const
- const unsigned int * GetDimensions () const

  Return the dimension of the pixel data, first dimension (x), then 2nd (y), then 3rd (z)...
- LookupTable & GetLUT ()
- const LookupTable & GetLUT () const
- bool GetNeedByteSwap () const

  INTERNAL do not use.
- unsigned int GetNumberOfDimensions () const

  Return the number of dimension of the pixel data bytes; for example 2 for a 2D matrices of values.
- const PhotometricInterpretation & GetPhotometricInterpretation () const

  return the photometric interpretation
- PixelFormat & GetPixelFormat ()
- const PixelFormat & GetPixelFormat () const

  Get/Set PixelFormat.
- unsigned int GetPlanarConfiguration () const

  return the planar configuration
- unsigned int GetRows () const
- const TransferSyntax & GetTransferSyntax () const
- bool IsEmpty () const
- bool IsLossy () const

  Return whether or not the image was compressed using a lossy compressor or not.
- bool IsTransferSyntaxCompatible (TransferSyntax const &ts) const
- void Print (std::ostream &) const override
- void SetColumns (unsigned int col)
- void SetDataElement (DataElement const &de)
- void SetDimension (unsigned int idx, unsigned int dim)
- void SetDimensions (const unsigned int dims[3])
- void SetLossyFlag (bool f)

  Specifically set that the image was compressed using a lossy compression mechanism.
- void SetLUT (LookupTable const &lut)

  Set/Get LUT.
- void SetNeedByteSwap (bool b)
- void SetNumberOfDimensions (unsigned int dim)
- void SetPhotometricInterpretation (PhotometricInterpretation const &pi)
- void SetPixelFormat (PixelFormat const &pf)
- void SetPlanarConfiguration (unsigned int pc)
- void SetRows (unsigned int rows)
- void SetTransferSyntax (TransferSyntax const &ts)

  Transfer syntax.
- virtual bool UnusedBitsPresentInPixelData () const

Protected Types

- typedef SmartPointer< LookupTable > LUTPtr

Protected Member Functions

- bool ComputeLossyFlag ()
- bool GetBuffer2 (std::ostream &os) const
- bool TryJPEG2000Codec (char *buffer, bool &lossyflag) const
- bool TryJPEG2000Codec2 (std::ostream &os) const
- bool TryJPEGCodec (char *buffer, bool &lossyflag) const
- bool TryJPEGCodec2 (std::ostream &os) const
- bool TryJPEGLSCodec (char *buffer, bool &lossyflag) const
- bool TryKAKADUCodec (char *buffer, bool &lossyflag) const
- bool TryPVRGCodec (char *buffer, bool &lossyflag) const
- bool TryRAWCodec (char *buffer, bool &lossyflag) const
- bool TryRLECodec (char *buffer, bool &lossyflag) const

Protected Attributes

- std::vector< unsigned int > Dimensions
- bool LossyFlag
- LUTPtr LUT
- bool NeedByteSwap
- unsigned int NumberOfDimensions
- PixelFormat PF
- PhotometricInterpretation PI
- DataElement PixelData
- unsigned int PlanarConfiguration
- TransferSyntax TS

Friends

- class ImageChangeTransferSyntax
- class PixmapReader

10.35.1 Detailed Description

Bitmap class.

A bitmap based image. Used as parent for both IconImage and the main Pixel Data Image It does not contains any World Space information (IPP, IOP)

Examples

ExtractIconFromFile.cxx.

10.35.2 Member Typedef Documentation

10.35.2.1 LUTPtr

```
typedef SmartPointer<LookupTable> gdcm::Bitmap::LUTPtr  [protected]
```

10.35.3 Constructor & Destructor Documentation

10.35.3.1 Bitmap()

```
gdcm::Bitmap::Bitmap ( )
```

10.35.3.2 ~Bitmap()

```
gdcm::Bitmap::~Bitmap ( )  [override]
```

10.35.4 Member Function Documentation

10.35.4.1 AreOverlaysInPixelData()

```
virtual bool gdcm::Bitmap::AreOverlaysInPixelData ( ) const  [inline], [virtual]
```

Reimplemented in gdcm::Pixmap.

10.35.4.2 Clear()

```
void gdcm::Bitmap::Clear ( )
```

10.35.4.3 ComputeLossyFlag()

```
bool gdcm::Bitmap::ComputeLossyFlag ( )  [protected]
```

10.35.4.4 GetBuffer()

```
bool gdcm::Bitmap::GetBuffer (
            char * buffer ) const
```

Access the raw data.

Examples

BasicImageAnonymizer.cs, ConvertToQImage.cxx, DecompressImage.cs, DecompressImageMultiframe.cs, DecompressJPEGFile.cs, GetArray.cs, ReadMultiTimesException.cxx, and threadgdcm.cxx.

10.35.4.5 GetBuffer2()

```
bool gdcm::Bitmap::GetBuffer2 (
            std::ostream & os ) const  [protected]
```

10.35.4.6 GetBufferLength()

```
unsigned long gdcm::Bitmap::GetBufferLength ( ) const
```

Return the length of the image after decompression WARNING for palette color: It will NOT take into account the Palette Color thus you need to multiply this length by 3 if computing the size of equivalent RGB image

Examples

BasicImageAnonymizer.cs, ConvertToQImage.cxx, DecompressImage.cs, DecompressImageMultiframe.cs, DecompressJPEGFile.cs, GenFakeImage.cxx, GetArray.cs, GetSubSequenceData.cxx, PatchFile.cxx, ReadMultiTimesException.cx RescaleImage.cs, and threadgdcm.cxx.

10.35.4.7 GetColumns()

```
unsigned int gdcm::Bitmap::GetColumns ( ) const  [inline]
```

10.35.4.8 GetDataElement() [1/2]

```
DataElement & gdcm::Bitmap::GetDataElement ( )  [inline]
```

10.35.4.9 GetDataElement() [2/2]

```
const DataElement & gdcm::Bitmap::GetDataElement ( ) const  [inline]
```

Examples

ExtractIconFromFile.cxx.

10.35.4.10 GetDimension()

```
unsigned int gdcm::Bitmap::GetDimension (
            unsigned int idx ) const
```

Examples

BasicImageAnonymizer.cs, DecompressImage.cs, and GetArray.cs.

10.35.4.11 GetDimensions()

```
const unsigned int * gdcm::Bitmap::GetDimensions ( ) const
```

Return the dimension of the pixel data, first dimension (x), then 2nd (y), then 3rd (z)...

Examples

ConvertToQImage.cxx, ExtractIconFromFile.cxx, FixJAIBugJPEGLS.cxx, HelloVizWorld.cxx, and threadgdcm.cxx.

10.35.4.12 GetLUT() [1/2]

```
LookupTable & gdcm::Bitmap::GetLUT ( )  [inline]
```

10.35.4.13 GetLUT() [2/2]

```
const LookupTable & gdcm::Bitmap::GetLUT ( ) const  [inline]
```

Examples

ExtractIconFromFile.cxx, ExtractImageRegionWithLUT.cs, and PrintLUT.cxx.

10.35.4.14 GetNeedByteSwap()

```
bool gdcm::Bitmap::GetNeedByteSwap ( ) const  [inline]
```

INTERNAL do not use.

10.35.4.15 GetNumberOfDimensions()

```
unsigned int gdcm::Bitmap::GetNumberOfDimensions ( ) const
```

Return the number of dimension of the pixel data bytes; for example 2 for a 2D matrices of values.

Examples

DecompressImage.cs, GetArray.cs, HelloVizWorld.cxx, and threadgdcm.cxx.

10.35.4.16 GetPhotometricInterpretation()

```
const PhotometricInterpretation & gdcm::Bitmap::GetPhotometricInterpretation ( ) const
```

return the photometric interpretation

Examples

ConvertToQImage.cxx, DecompressImage.cs, ExtractIconFromFile.cxx, and HelloVizWorld.cxx.

10.35.4.17 GetPixelFormat() [1/2]

```
PixelFormat & gdcm::Bitmap::GetPixelFormat ( )  [inline]
```

10.35.4.18 GetPixelFormat() [2/2]

```
const PixelFormat & gdcm::Bitmap::GetPixelFormat ( ) const  [inline]
```

Get/Set PixelFormat.

Examples

ConvertToQImage.cxx, DecompressImage.cs, ExtractIconFromFile.cxx, FixJAIBugJPEGLS.cxx, GenFakeImage.cxx, GetArray.cs, GetJPEGSamplePrecision.cxx, RescaleImage.cs, TemplateEmptyImage.cxx, and threadgdcm.cxx.

10.35.4.19 GetPlanarConfiguration()

```
unsigned int gdcm::Bitmap::GetPlanarConfiguration ( ) const
```

return the planar configuration

10.35.4.20 GetRows()

```
unsigned int gdcm::Bitmap::GetRows ( ) const  [inline]
```

10.35.4.21 GetTransferSyntax()

```
const TransferSyntax & gdcm::Bitmap::GetTransferSyntax ( ) const  [inline]
```

Examples

ExtractIconFromFile.cxx.

10.35.4.22 IsEmpty()

```
bool gdcm::Bitmap::IsEmpty ( ) const  [inline]
```

10.35.4.23 IsLossy()

```
bool gdcm::Bitmap::IsLossy ( ) const
```

Return whether or not the image was compressed using a lossy compressor or not.

10.35.4.24 IsTransferSyntaxCompatible()

```
bool gdcm::Bitmap::IsTransferSyntaxCompatible (
            TransferSyntax const & ts ) const
```

10.35.4.25 Print()

```
void gdcm::Bitmap::Print (
            std::ostream &  ) const  [override], [virtual]
```

Reimplemented from gdcm::Object.

Reimplemented in gdcm::Pixmap, and gdcm::Image.

Examples

ExtractIconFromFile.cxx.

10.35.4.26 SetColumns()

```
void gdcm::Bitmap::SetColumns (
            unsigned int col )  [inline]
```

10.35.4.27 SetDataElement()

```
void gdcm::Bitmap::SetDataElement (
            DataElement const & de )  [inline]
```

Examples

BasicImageAnonymizer.cs, CreateARGBImage.cxx, CreateCMYKImage.cxx, DecompressImage.cs, DecompressImageMultiframe.c DecompressJPEGFile.cs, FileChangeTS.cs, FileChangeTSLossy.cs, GenFakeImage.cxx, GetSubSequenceData.cxx, MpegVideoInfo.cs, csa2img.cxx, and iU22tomultisc.cxx.

10.35.4.28 SetDimension()

```
void gdcm::Bitmap::SetDimension (
            unsigned int idx,
            unsigned int dim )
```

Examples

DecompressImageMultiframe.cs, DecompressJPEGFile.cs, FileChangeTS.cs, FileChangeTSLossy.cs, GenFakeImage.cxx, GetSubSequenceData.cxx, MpegVideoInfo.cs, csa2img.cxx, and iU22tomultisc.cxx.

10.35.4.29 SetDimensions()

```
void gdcm::Bitmap::SetDimensions (
            const unsigned int dims[3] )
```

Examples

CreateARGBImage.cxx, CreateCMYKImage.cxx, and DecompressImage.cs.

10.35.4.30 SetLossyFlag()

```
void gdcm::Bitmap::SetLossyFlag (
            bool f )  [inline]
```

Specifically set that the image was compressed using a lossy compression mechanism.

10.35.4.31 SetLUT()

```
void gdcm::Bitmap::SetLUT (
            LookupTable const & lut )  [inline]
```

Set/Get LUT.

10.35.4.32 SetNeedByteSwap()

```
void gdcm::Bitmap::SetNeedByteSwap (
            bool b )  [inline]
```

10.35.4.33 SetNumberOfDimensions()

```
void gdcm::Bitmap::SetNumberOfDimensions (
            unsigned int dim )
```

Examples

CreateARGBImage.cxx, CreateCMYKImage.cxx, DecompressImage.cs, DecompressImageMultiframe.cs, DecompressJPEGFile.cs, FileChangeTS.cs, FileChangeTSLossy.cs, GenFakeImage.cxx, GetSubSequenceData.cxx, MpegVideoInfo.cs, csa2img.cxx, and iU22tomultisc.cxx.

10.35.4.34 SetPhotometricInterpretation()

```
void gdcm::Bitmap::SetPhotometricInterpretation (
            PhotometricInterpretation const & pi )
```

Examples

CreateARGBImage.cxx, CreateCMYKImage.cxx, DecompressImage.cs, DecompressImageMultiframe.cs, DecompressJPEGFile.cs, FileChangeTS.cs, FileChangeTSLossy.cs, GenFakeImage.cxx, GetSubSequenceData.cxx, MpegVideoInfo.cs, csa2img.cxx, and iU22tomultisc.cxx.

10.35.4.35 SetPixelFormat()

```
void gdcm::Bitmap::SetPixelFormat (
            PixelFormat const & pf )  [inline]
```

Examples

CreateARGBImage.cxx, CreateCMYKImage.cxx, DecompressImage.cs, DecompressImageMultiframe.cs, DecompressJPEGFile.cs, MpegVideoInfo.cs, csa2img.cxx, and iU22tomultisc.cxx.

References gdcm::PixelFormat::Validate().

10.35.4.36 SetPlanarConfiguration()

```
void gdcm::Bitmap::SetPlanarConfiguration (
            unsigned int pc )
```

Warning

you need to call SetPixelFormat first (before SetPlanarConfiguration) for consistency checking

10.35.4.37 SetRows()

```
void gdcm::Bitmap::SetRows (
            unsigned int rows )  [inline]
```

10.35.4.38 SetTransferSyntax()

```
void gdcm::Bitmap::SetTransferSyntax (
            TransferSyntax const & ts )  [inline]
```

Transfer syntax.

Examples

BasicImageAnonymizer.cs, CreateARGBImage.cxx, CreateCMYKImage.cxx, DecompressImageMultiframe.cs, DecompressJPEGFile.cs, MergeTwoFiles.cxx, and MpegVideoInfo.cs.

10.35.4.39 TryJPEG2000Codec()

```
bool gdcm::Bitmap::TryJPEG2000Codec (
            char * buffer,
            bool & lossyflag ) const  [protected]
```

10.35.4.40 TryJPEG2000Codec2()

```
bool gdcm::Bitmap::TryJPEG2000Codec2 (
            std::ostream & os ) const  [protected]
```

10.35.4.41 TryJPEGCodec()

```
bool gdcm::Bitmap::TryJPEGCodec (
            char * buffer,
            bool & lossyflag ) const  [protected]
```

10.35.4.42 TryJPEGCodec2()

```
bool gdcm::Bitmap::TryJPEGCodec2 (
            std::ostream & os ) const  [protected]
```

10.35.4.43 TryJPEGLSCodec()

```
bool gdcm::Bitmap::TryJPEGLSCodec (
            char * buffer,
            bool & lossyflag ) const  [protected]
```

10.35.4.44 TryKAKADUCodec()

```
bool gdcm::Bitmap::TryKAKADUCodec (
            char * buffer,
            bool & lossyflag ) const  [protected]
```

10.35.4.45 TryPVRGCodec()

```
bool gdcm::Bitmap::TryPVRGCodec (
            char * buffer,
            bool & lossyflag ) const  [protected]
```

10.35.4.46 TryRAWCodec()

```
bool gdcm::Bitmap::TryRAWCodec (
            char * buffer,
            bool & lossyflag ) const  [protected]
```

10.35.4.47 TryRLECodec()

```
bool gdcm::Bitmap::TryRLECodec (
            char * buffer,
            bool & lossyflag ) const  [protected]
```

10.35.4.48 UnusedBitsPresentInPixelData()

```
virtual bool gdcm::Bitmap::UnusedBitsPresentInPixelData ( ) const  [inline], [virtual]
```

Reimplemented in gdcm::Pixmap.

10.35.5 Friends And Related Function Documentation

10.35.5.1 ImageChangeTransferSyntax

```
friend class ImageChangeTransferSyntax  [friend]
```

10.35.5.2 PixmapReader

```
friend class PixmapReader  [friend]
```

10.35.6 Member Data Documentation

10.35.6.1 Dimensions

```
std::vector<unsigned int> gdcm::Bitmap::Dimensions  [protected]
```

10.35.6.2 LossyFlag

```
bool gdcm::Bitmap::LossyFlag  [protected]
```

10.35.6.3 LUT

```
LUTPtr gdcm::Bitmap::LUT  [protected]
```

10.35.6.4 NeedByteSwap

```
bool gdcm::Bitmap::NeedByteSwap  [protected]
```

10.35.6.5 NumberOfDimensions

```
unsigned int gdcm::Bitmap::NumberOfDimensions  [protected]
```

10.35.6.6 PF

```
PixelFormat gdcm::Bitmap::PF  [protected]
```

10.35.6.7 PI

```
PhotometricInterpretation gdcm::Bitmap::PI  [protected]
```

10.35.6.8 PixelData

```
DataElement gdcm::Bitmap::PixelData  [protected]
```

10.35.6.9 PlanarConfiguration

```
unsigned int gdcm::Bitmap::PlanarConfiguration  [protected]
```

10.35.6.10 TS

```
TransferSyntax gdcm::Bitmap::TS  [protected]
```

The documentation for this class was generated from the following file:

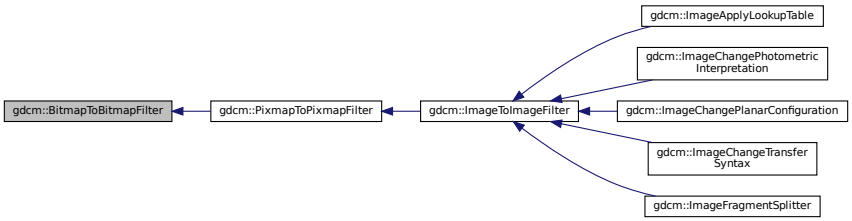
- gdcmBitmap.h

10.36 gdcm::BitmapToBitmapFilter Class Reference

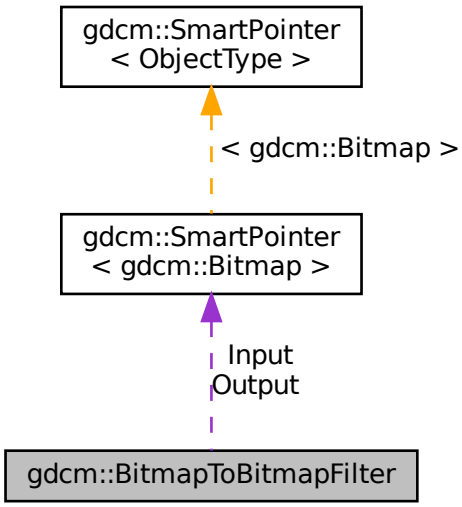
BitmapToBitmapFilter class.

```
#include <gdcmBitmapToBitmapFilter.h>
```

Inheritance diagram for gdcm::BitmapToBitmapFilter:



Collaboration diagram for gdcm::BitmapToBitmapFilter:



Public Member Functions

- BitmapToBitmapFilter ()
- ~BitmapToBitmapFilter ()=default
- const Bitmap & GetOutput () const

  Get Output image.
- const Bitmap & GetOutputAsBitmap () const
- void SetInput (const Bitmap &image)

  Set input image.

Protected Attributes

- SmartPointer< Bitmap > Input
- SmartPointer< Bitmap > Output

10.36.1 Detailed Description

BitmapToBitmapFilter class.

Super class for all filter taking an image and producing an output image

10.36.2 Constructor & Destructor Documentation

10.36.2.1 BitmapToBitmapFilter()

```
gdcm::BitmapToBitmapFilter::BitmapToBitmapFilter ( )
```

10.36.2.2 ~BitmapToBitmapFilter()

```
gdcm::BitmapToBitmapFilter::~BitmapToBitmapFilter ( )  [default]
```

10.36.3 Member Function Documentation

10.36.3.1 GetOutput()

```
const Bitmap & gdcm::BitmapToBitmapFilter::GetOutput ( ) const  [inline]
```

Get Output image.

10.36.3.2 GetOutputAsBitmap()

```
const Bitmap & gdcm::BitmapToBitmapFilter::GetOutputAsBitmap ( ) const
```

10.36.3.3 SetInput()

```
void gdcm::BitmapToBitmapFilter::SetInput (
            const Bitmap & image )
```

Set input image.

Examples

BasicImageAnonymizer.cs, CompressImage.cxx, CompressLossyJPEG.cs, and StandardizeFiles.cs.

10.36.4 Member Data Documentation

10.36.4.1 Input

```
SmartPointer<Bitmap> gdcm::BitmapToBitmapFilter::Input  [protected]
```

10.36.4.2 Output

```
SmartPointer<Bitmap> gdcm::BitmapToBitmapFilter::Output  [protected]
```

The documentation for this class was generated from the following file:

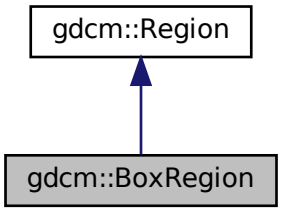
- gdcmBitmapToBitmapFilter.h

10.37 gdcm::BoxRegion Class Reference

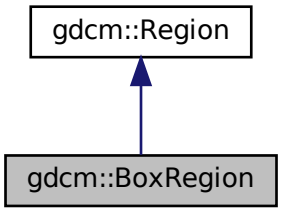
Class for manipulation box region.

`#include <gdcmBoxRegion.h>`

Inheritance diagram for gdcm::BoxRegion:



Collaboration diagram for gdcm::BoxRegion:



Public Member Functions

- BoxRegion ()
- BoxRegion (const BoxRegion &)

  copy/cstor and al.
- ~BoxRegion () override
- size_t Area () const override

  compute the area
- Region * Clone () const override
- BoxRegion ComputeBoundingBox () override

  Return the Axis-Aligned minimum bounding box for all regions.
- bool Empty () const override

  return whether this domain is empty:
- unsigned int GetXMax () const
- unsigned int GetXMin () const

  Get domain.
- unsigned int GetYMax () const
- unsigned int GetYMin () const
- unsigned int GetZMax () const
- unsigned int GetZMin () const
- bool IsValid () const override

  return whether this is valid domain
- void operator= (const BoxRegion &)
- void Print (std::ostream &os=std::cout) const override

  Print.
- void SetDomain (unsigned int xmin, unsigned int xmax, unsigned int ymin, unsigned int ymax, unsigned int zmin, unsigned int zmax)

  Set domain.

Static Public Member Functions

- static BoxRegion BoundingBox (BoxRegion const &b1, BoxRegion const &b2)

  Helper class to compute the bounding box of two BoxRegion.

10.37.1 Detailed Description

Class for manipulation box region.

This is a very simple implementation of the Region class. It only support 3D box type region. It assumes the 3D Box does not have a tilt Origin is as (0,0,0)

Examples

ExtractImageRegion.cs, and ExtractImageRegionWithLUT.cs.

10.37.2 Constructor & Destructor Documentation

10.37.2.1 BoxRegion() [1/2]

```
gdcm::BoxRegion::BoxRegion ( )
```

10.37.2.2 ~BoxRegion()

```
gdcm::BoxRegion::~BoxRegion ( )  [override]
```

10.37.2.3 BoxRegion() [2/2]

```
gdcm::BoxRegion::BoxRegion (
            const BoxRegion &  )
```

copy/cstor and al.

10.37.3 Member Function Documentation

10.37.3.1 Area()

```
size_t gdcm::BoxRegion::Area ( ) const  [override], [virtual]
```

compute the area

Implements gdcm::Region.

10.37.3.2 BoundingBox()

```
static BoxRegion gdcm::BoxRegion::BoundingBox (
            BoxRegion const & b1,
            BoxRegion const & b2 )  [static]
```

Helper class to compute the bounding box of two BoxRegion.

10.37.3.3 Clone()

```
Region * gdcm::BoxRegion::Clone ( ) const  [override], [virtual]
```

Implements gdcm::Region.

10.37.3.4 ComputeBoundingBox()

```
BoxRegion gdcm::BoxRegion::ComputeBoundingBox ( )  [override], [virtual]
```

Return the Axis-Aligned minimum bounding box for all regions.

Implements gdcm::Region.

10.37.3.5 Empty()

```
bool gdcm::BoxRegion::Empty ( ) const  [override], [virtual]
```

return whether this domain is empty:

Implements gdcm::Region.

10.37.3.6 GetXMax()

```
unsigned int gdcm::BoxRegion::GetXMax ( ) const
```

10.37.3.7 GetXMin()

```
unsigned int gdcm::BoxRegion::GetXMin ( ) const
```

Get domain.

10.37.3.8 GetYMax()

```
unsigned int gdcm::BoxRegion::GetYMax ( ) const
```

10.37.3.9 GetYMin()

```
unsigned int gdcm::BoxRegion::GetYMin ( ) const
```

10.37.3.10 GetZMax()

```
unsigned int gdcm::BoxRegion::GetZMax ( ) const
```

10.37.3.11 GetZMin()

```
unsigned int gdcm::BoxRegion::GetZMin ( ) const
```

10.37.3.12 IsValid()

```
bool gdcm::BoxRegion::IsValid ( ) const  [override], [virtual]
```

return whether this is valid domain

Implements gdcm::Region.

10.37.3.13 operator=()

```
void gdcm::BoxRegion::operator= (
            const BoxRegion &  )
```

10.37.3.14 Print()

```
void gdcm::BoxRegion::Print (
            std::ostream & os = std::cout ) const  [override], [virtual]
```

Print.

Reimplemented from gdcm::Region.

10.37.3.15 SetDomain()

```
void gdcm::BoxRegion::SetDomain (
            unsigned int xmin,
            unsigned int xmax,
            unsigned int ymin,
            unsigned int ymax,
            unsigned int zmin,
            unsigned int zmax )
```

Set domain.

Examples

ExtractImageRegion.cs, and ExtractImageRegionWithLUT.cs.

The documentation for this class was generated from the following file:

- gdcmBoxRegion.h

10.38 gdcm::ByteBuffer Class Reference

ByteBuffer.

```
#include <gdcmByteBuffer.h>
```

Public Member Functions

- ByteBuffer ()
- char * Get (int len)
- const char * GetStart () const
- void ShiftEnd (int len)
- void UpdatePosition ()

10.38.1 Detailed Description

ByteBuffer.

Detailed description here

Note

looks like a std::streambuf or std::filebuf class with the get and peek pointer

10.38.2 Constructor & Destructor Documentation

10.38.2.1 ByteBuffer()

```
gdcm::ByteBuffer::ByteBuffer ( )  [inline]
```

10.38.3 Member Function Documentation

10.38.3.1 Get()

```
char * gdcm::ByteBuffer::Get (
            int len )  [inline]
```

10.38.3.2 GetStart()

```
const char * gdcm::ByteBuffer::GetStart ( ) const  [inline]
```

10.38.3.3 ShiftEnd()

```
void gdcm::ByteBuffer::ShiftEnd (
            int len )  [inline]
```

10.38.3.4 UpdatePosition()

```
void gdcm::ByteBuffer::UpdatePosition ( )  [inline]
```

The documentation for this class was generated from the following file:

- gdcmByteBuffer.h

10.39 gdcm::ByteSwap< T > Class Template Reference

ByteSwap.

```
#include <gdcmByteSwap.h>
```

Static Public Member Functions

- static void Swap (T &p)
- static void SwapFromSwapCodeIntoSystem (T &p, SwapCode const &sc)
- static void SwapRange (T *p, unsigned int num)
- static void SwapRangeFromSwapCodeIntoSystem (T *p, SwapCode const &sc, std::streamoff num)
- static bool SystemIsBigEndian ()
- static bool SystemIsLittleEndian ()

10.39.1 Detailed Description

template<class T>
class gdcm::ByteSwap< T >

ByteSwap.

Perform machine dependent byte swapping (Little Endian, Big Endian, Bad Little Endian, Bad Big Endian). TODO: bswap_32 / bswap_64 ...

Examples

TestByteSwap.cxx.

10.39.2 Member Function Documentation

10.39.2.1 Swap()

```
template<class T >
static void gdcm::ByteSwap< T >::Swap (
            T & p )  [static]
```

10.39.2.2 SwapFromSwapCodeIntoSystem()

```
template<class T >
static void gdcm::ByteSwap< T >::SwapFromSwapCodeIntoSystem (
            T & p,
            SwapCode const & sc )  [static]
```

Examples

TestByteSwap.cxx.

10.39.2.3 SwapRange()

```
template<class T >
static void gdcm::ByteSwap< T >::SwapRange (
            T * p,
            unsigned int num )  [static]
```

10.39.2.4 SwapRangeFromSwapCodeIntoSystem()

```
template<class T >
static void gdcm::ByteSwap< T >::SwapRangeFromSwapCodeIntoSystem (
            T * p,
            SwapCode const & sc,
            std::streamoff num )  [static]
```

Examples

TestByteSwap.cxx.

10.39.2.5 SystemIsBigEndian()

```
template<class T >
static bool gdcm::ByteSwap< T >::SystemIsBigEndian ( )  [static]
```

Query the machine Endian-ness.

10.39.2.6 SystemIsLittleEndian()

```
template<class T >
static bool gdcm::ByteSwap< T >::SystemIsLittleEndian ( )  [static]
```

The documentation for this class was generated from the following file:

- gdcmByteSwap.h

10.40 gdcm::ByteSwapFilter Class Reference

ByteSwapFilter.

```
#include <gdcmByteSwapFilter.h>
```

Public Member Functions

- ByteSwapFilter (const ByteSwapFilter &)=delete
- ByteSwapFilter (DataSet &ds)
- ~ByteSwapFilter ()
- bool ByteSwap ()
- ByteSwapFilter & operator= (const ByteSwapFilter &)=delete
- void SetByteSwapTag (bool b)

10.40.1 Detailed Description

ByteSwapFilter.

In place byte-swapping of a dataset FIXME: FL status ??

10.40.2 Constructor & Destructor Documentation

10.40.2.1 ByteSwapFilter() [1/2]

```
gdcm::ByteSwapFilter::ByteSwapFilter (
            DataSet & ds )  [inline]
```

10.40.2.2 ~ByteSwapFilter()

```
gdcm::ByteSwapFilter::~ByteSwapFilter ( )
```

10.40.2.3 ByteSwapFilter() [2/2]

```
gdcm::ByteSwapFilter::ByteSwapFilter (
            const ByteSwapFilter &  )  [delete]
```

10.40.3 Member Function Documentation

10.40.3.1 ByteSwap()

```
bool gdcm::ByteSwapFilter::ByteSwap ( )
```

Referenced by gdcm::Item::Read().

10.40.3.2 operator=()

```
ByteSwapFilter & gdcm::ByteSwapFilter::operator= (
            const ByteSwapFilter &  )  [delete]
```

10.40.3.3 SetByteSwapTag()

```
void gdcm::ByteSwapFilter::SetByteSwapTag (
            bool b )  [inline]
```

Referenced by gdcm::Item::Read().

The documentation for this class was generated from the following file:

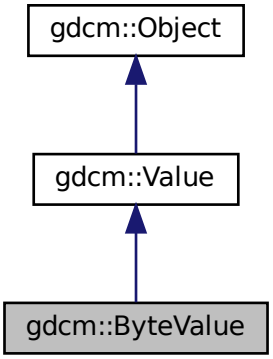
- gdcmByteSwapFilter.h

10.41 gdcm::ByteValue Class Reference

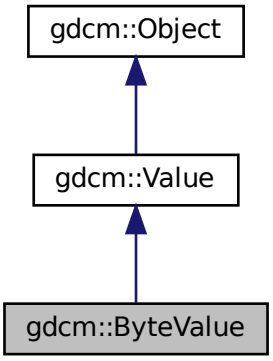
Class to represent binary value (array of bytes)

`#include <gdcmByteValue.h>`

Inheritance diagram for gdcm::ByteValue:



Collaboration diagram for gdcm::ByteValue:



Public Member Functions

- ByteValue (const char *array=nullptr, VL const &vl=0)
- ByteValue (std::vector< char > &v)
- ~ByteValue () override
- void Append (ByteValue const &bv)
- void Clear () override
- VL ComputeLength () const
- void Fill (char c)
- bool GetBuffer (char *buffer, unsigned long length) const
- VL GetLength () const override
- const char * GetPointer () const
- void * GetVoidPointer ()
- const void * GetVoidPointer () const
- bool IsEmpty () const
- bool IsPrintable (VL length) const

  Checks whether a 'ByteValue' is printable or not (in order to avoid corrupting the terminal of invocation when printing) I don't think this function is working since it does not handle UNICODE or character set...
- operator const std::vector< char > & () const
- ByteValue & operator= (const ByteValue &val)
- bool operator== (const ByteValue &val) const
- bool operator== (const Value &val) const override
- void PrintASCII (std::ostream &os, VL maxlength) const
- void PrintASCIIXML (std::ostream &os) const
- void PrintGroupLength (std::ostream &os)
- void PrintHex (std::ostream &os, VL maxlength) const
- void PrintHexXML (std::ostream &os) const
- void PrintPNXML (std::ostream &os) const
- template<typename TSwap >
  std::istream & Read (std::istream &is)
- template<typename TSwap , typename TType >
  std::istream & Read (std::istream &is, bool readvalues=true)
- void SetLength (VL vl) override
- template<typename TSwap , typename TType >
  std::ostream const & Write (std::ostream &os) const
- template<typename TSwap >
  std::ostream const & Write (std::ostream &os) const
- bool WriteBuffer (std::ostream &os) const

Protected Member Functions

- void Print (std::ostream &os) const override
- void SetLengthOnly (VL vl) override

10.41.1 Detailed Description

Class to represent binary value (array of bytes)

Note

Examples

DumpADAC.cxx, DumpExamCard.cxx, DumpGEMSMovieGroup.cxx, DumpImageHeaderInfo.cxx, DumpPhilipsECHO.cxx, DumpSiemensBase64.cxx, DumpToshibaDTI.cxx, DumpToshibaDTI2.cxx, DuplicatePCDE.cxx, ELSCINT1WaveToText.cxx, ExtractEncapsulatedFile.cs, ExtractEncryptedContent.cxx, ExtractIconFromFile.cxx, FixBrokenJ2K.cxx, FixJAIBugJPEGLS.cxx, GetSubSequenceData.cxx, MrProtocol.cxx, PatchFile.cxx, ReadExplicitLengthSQIVR.cxx, pmsct_rgb1.cxx, and rle2img.cxx.

10.41.2 Constructor & Destructor Documentation

10.41.2.1 ByteValue() [1/2]

```
gdcm::ByteValue::ByteValue (
            const char * array = nullptr,
            VL const & vl = 0 )  [inline]
```

References gdcmDebugMacro.

10.41.2.2 ByteValue() [2/2]

```
gdcm::ByteValue::ByteValue (
            std::vector< char > & v )  [inline]
```

Warning

casting to uint32_t

10.41.2.3 ~ByteValue()

```
gdcm::ByteValue::~ByteValue ( )  [inline], [override]
```

10.41.3 Member Function Documentation

10.41.3.1 Append()

```
void gdcm::ByteValue::Append (
            ByteValue const & bv )
```

10.41.3.2 Clear()

```
void gdcm::ByteValue::Clear ( )  [inline], [override], [virtual]
```

Implements gdcm::Value.

10.41.3.3 ComputeLength()

```
VL gdcm::ByteValue::ComputeLength ( ) const  [inline]
```

Referenced by gdcm::Fragment::Write().

10.41.3.4 Fill()

```
void gdcm::ByteValue::Fill (
            char c )  [inline]
```

Examples

DuplicatePCDE.cxx.

10.41.3.5 GetBuffer()

```
bool gdcm::ByteValue::GetBuffer (
            char * buffer,
            unsigned long length ) const
```

Examples

ExtractEncapsulatedFile.cs, and FixJAIBugJPEGLS.cxx.

10.41.3.6 GetLength()

```
VL gdcm::ByteValue::GetLength ( ) const  [inline], [override], [virtual]
```

Implements gdcm::Value.

Examples

DumpADAC.cxx, DumpExamCard.cxx, DumpGEMSMovieGroup.cxx, DumpImageHeaderInfo.cxx, DumpPhilipsECHO.cxx, DumpSiemensBase64.cxx, DumpToshibaDTI.cxx, DumpToshibaDTI2.cxx, ELSCINT1WaveToText.cxx, ExtractEncapsulatedFile.cs, ExtractEncryptedContent.cxx, ExtractIconFromFile.cxx, FixBrokenJ2K.cxx, FixJAIBugJPEGLS.cxx, GetSubSequenceData.cxx, MrProtocol.cxx, PatchFile.cxx, ReadExplicitLengthSQIVR.cxx, ReadGEMSSDO.cxx, pmsct_rgb1.cxx, and rle2img.cxx.

Referenced by gdcm::SequenceOfFragments::ReadValue(), gdcm::Element< TVR, TVM >::Set(), gdcm::Element< TVR, VM::VM1_n >:
gdcm::Attribute< Group, Element, TVR, TVM >::SetByteValue(), gdcm::Attribute< Group, Element, TVR, VM::VM1 >::SetByteValue(),
gdcm::Attribute< Group, Element, TVR, VM::VM1_n >::SetByteValue(), gdcm::Attribute< Group, Element, TVR, TVM >::SetByteValueN
gdcm::Attribute< Group, Element, TVR, VM::VM1 >::SetByteValueNoSwap(), gdcm::Element< TVR, TVM >::SetNoSwap(),
gdcm::Element< TVR, VM::VM1_n >::SetNoSwap(), and gdcm::Fragment::Write().

10.41.3.7 GetPointer()

```
const char * gdcm::ByteValue::GetPointer ( ) const  [inline]
```

Examples

DumpADAC.cxx, DumpExamCard.cxx, DumpGEMSMovieGroup.cxx, DumpImageHeaderInfo.cxx, DumpPhilipsECHO.cxx, DumpSiemensBase64.cxx, DumpToshibaDTI.cxx, DumpToshibaDTI2.cxx, ELSCINT1WaveToText.cxx, ExtractEncryptedContent.cxx
ExtractIconFromFile.cxx, GetSubSequenceData.cxx, MrProtocol.cxx, ReadExplicitLengthSQIVR.cxx, ReadGEMSSDO.cxx, pmsct_rgb1.cxx, and rle2img.cxx.

Referenced by gdcm::SequenceOfFragments::ReadValue(), gdcm::Element< TVR, TVM >::Set(), gdcm::Element< TVR, VM::VM1_n >:
gdcm::Attribute< Group, Element, TVR, TVM >::SetByteValue(), gdcm::Attribute< Group, Element, TVR, VM::VM1 >::SetByteValue(),
gdcm::Attribute< Group, Element, TVR, VM::VM1_n >::SetByteValue(), gdcm::Attribute< Group, Element, TVR, TVM >::SetByteValueN
gdcm::Attribute< Group, Element, TVR, VM::VM1 >::SetByteValueNoSwap(), gdcm::Element< TVR, TVM >::SetNoSwap(),
and gdcm::Element< TVR, VM::VM1_n >::SetNoSwap().

10.41.3.8 GetVoidPointer() [1/2]

```
void * gdcm::ByteValue::GetVoidPointer ( )  [inline]
```

10.41.3.9 GetVoidPointer() [2/2]

```
const void * gdcm::ByteValue::GetVoidPointer ( ) const  [inline]
```

Examples

FixBrokenJ2K.cxx.

Referenced by gdcm::Element< TVR, VM::VM1_n >::Set().

10.41.3.10 IsEmpty()

```
bool gdcm::ByteValue::IsEmpty ( ) const  [inline]
```

10.41.3.11 IsPrintable()

```
bool gdcm::ByteValue::IsPrintable (
            VL length ) const  [inline]
```

Checks whether a 'ByteValue' is printable or not (in order to avoid corrupting the terminal of invocation when printing) I don't think this function is working since it does not handle UNICODE or character set...

10.41.3.12 operator const std::vector< char > &()

```
gdcm::ByteValue::operator const std::vector< char > & ( ) const  [inline]
```

10.41.3.13 operator=()

```
ByteValue & gdcm::ByteValue::operator= (
            const ByteValue & val )  [inline]
```

10.41.3.14 operator==() [1/2]

```
bool gdcm::ByteValue::operator== (
            const ByteValue & val ) const  [inline]
```

10.41.3.15 operator==() [2/2]

```
bool gdcm::ByteValue::operator== (
            const Value & val ) const  [inline], [override], [virtual]
```

Implements gdcm::Value.

10.41.3.16 Print()

```
void gdcm::ByteValue::Print (
            std::ostream & os ) const  [inline], [override], [protected], [virtual]
```

Reimplemented from gdcm::Object.

10.41.3.17 PrintASCII()

```
void gdcm::ByteValue::PrintASCII (
            std::ostream & os,
            VL maxlength ) const
```

10.41.3.18 PrintASCIIXML()

```
void gdcm::ByteValue::PrintASCIIXML (
            std::ostream & os ) const
```

10.41.3.19 PrintGroupLength()

```
void gdcm::ByteValue::PrintGroupLength (
            std::ostream & os )  [inline]
```

10.41.3.20 PrintHex()

```
void gdcm::ByteValue::PrintHex (
            std::ostream & os,
            VL maxlength ) const
```

10.41.3.21 PrintHexXML()

```
void gdcm::ByteValue::PrintHexXML (
            std::ostream & os ) const
```

10.41.3.22 PrintPNXML()

```
void gdcm::ByteValue::PrintPNXML (
            std::ostream & os ) const
```

To Print Values in Native DICOM format

10.41.3.23 Read() [1/2]

```
template<typename TSwap >
std::istream & gdcm::ByteValue::Read (
            std::istream & is )  [inline]
```

10.41.3.24 Read() [2/2]

```
template<typename TSwap , typename TType >
std::istream & gdcm::ByteValue::Read (
            std::istream & is,
            bool readvalues = true )  [inline]
```

10.41.3.25 SetLength()

```
void gdcm::ByteValue::SetLength (
            VL vl )  [override], [virtual]
```

Implements gdcm::Value.

10.41.3.26 SetLengthOnly()

```
void gdcm::ByteValue::SetLengthOnly (
            VL vl )  [inline], [override], [protected], [virtual]
```

Reimplemented from gdcm::Value.

10.41.3.27 Write() [1/2]

```
template<typename TSwap , typename TType >
std::ostream const & gdcm::ByteValue::Write (
            std::ostream & os ) const  [inline]
```

Referenced by gdcm::Fragment::Write().

10.41.3.28 Write() [2/2]

```
template<typename TSwap >
std::ostream const & gdcm::ByteValue::Write (
            std::ostream & os ) const  [inline]
```

10.41.3.29 WriteBuffer()

```
bool gdcm::ByteValue::WriteBuffer (
            std::ostream & os ) const  [inline]
```

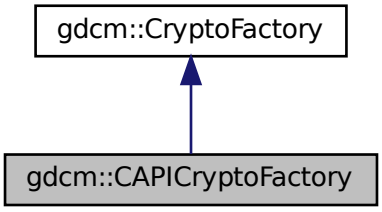
The documentation for this class was generated from the following file:

- gdcmByteValue.h

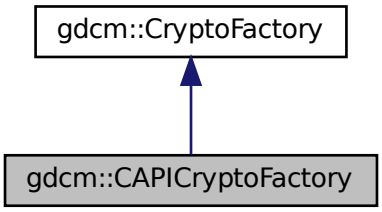
10.42 gdcm::CAPICryptoFactory Class Reference

`#include <gdcmCAPICryptoFactory.h>`

Inheritance diagram for gdcm::CAPICryptoFactory:



Collaboration diagram for gdcm::CAPICryptoFactory:



Public Member Functions

- CAPICryptoFactory (CryptoLib id)
- CryptographicMessageSyntax * CreateCMSProvider ()

Additional Inherited Members

10.42.1 Constructor & Destructor Documentation

10.42.1.1 CAPICryptoFactory()

```
gdcm::CAPICryptoFactory::CAPICryptoFactory (
            CryptoLib id )
```

10.42.2 Member Function Documentation

10.42.2.1 CreateCMSProvider()

```
CryptographicMessageSyntax * gdcm::CAPICryptoFactory::CreateCMSProvider ( )  [virtual]
```

Implements gdcm::CryptoFactory.

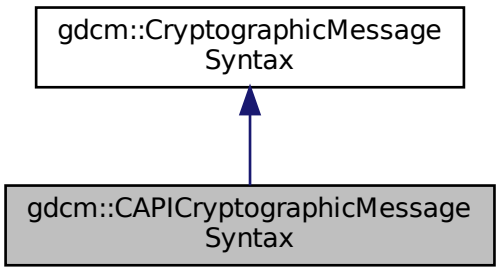
The documentation for this class was generated from the following file:

- gdcmCAPICryptoFactory.h

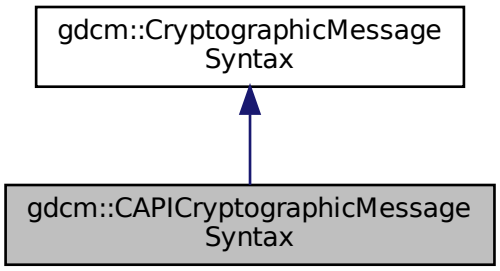
10.43 gdcm::CAPICryptographicMessageSyntax Class Reference

```
#include <gdcmCAPICryptographicMessageSyntax.h>
```

Inheritance diagram for gdcm::CAPICryptographicMessageSyntax:



Collaboration diagram for gdcm::CAPICryptographicMessageSyntax:



Public Member Functions

- CAPICryptographicMessageSyntax ()
- ~CAPICryptographicMessageSyntax ()
- bool Decrypt (char *output, size_t &outlen, const char *array, size_t len) const

  decrypt content from a CMS envelopedData structure
- bool Encrypt (char *output, size_t &outlen, const char *array, size_t len) const

  create a CMS envelopedData structure
- CipherTypes GetCipherType () const
- bool GetInitialized () const
- bool ParseCertificateFile (const char *filename)
- bool ParseKeyFile (const char *filename)
- void SetCipherType (CipherTypes type)
- bool SetPassword (const char *pass, size_t passLen)

Additional Inherited Members

10.43.1 Constructor & Destructor Documentation

10.43.1.1 CAPICryptographicMessageSyntax()

```
gdcm::CAPICryptographicMessageSyntax::CAPICryptographicMessageSyntax ( )
```

10.43.1.2 ~CAPICryptographicMessageSyntax()

```
gdcm::CAPICryptographicMessageSyntax::~CAPICryptographicMessageSyntax ( )
```

10.43.2 Member Function Documentation

10.43.2.1 Decrypt()

```
bool gdcm::CAPICryptographicMessageSyntax::Decrypt (
            char * output,
            size_t & outlen,
            const char * array,
            size_t len ) const  [virtual]
```

decrypt content from a CMS envelopedData structure

Implements gdcm::CryptographicMessageSyntax.

10.43.2.2 Encrypt()

```
bool gdcm::CAPICryptographicMessageSyntax::Encrypt (
            char * output,
            size_t & outlen,
            const char * array,
            size_t len ) const  [virtual]
```

create a CMS envelopedData structure

Implements gdcm::CryptographicMessageSyntax.

10.43.2.3 GetCipherType()

```
CipherTypes gdcm::CAPICryptographicMessageSyntax::GetCipherType ( ) const  [virtual]
```

Implements gdcm::CryptographicMessageSyntax.

10.43.2.4 GetInitialized()

```
bool gdcm::CAPICryptographicMessageSyntax::GetInitialized ( ) const  [inline]
```

10.43.2.5 ParseCertificateFile()

```
bool gdcm::CAPICryptographicMessageSyntax::ParseCertificateFile (
            const char * filename )  [virtual]
```

Implements gdcm::CryptographicMessageSyntax.

10.43.2.6 ParseKeyFile()

```
bool gdcm::CAPICryptographicMessageSyntax::ParseKeyFile (
            const char * filename )  [virtual]
```

Implements gdcm::CryptographicMessageSyntax.

10.43.2.7 SetCipherType()

```
void gdcm::CAPICryptographicMessageSyntax::SetCipherType (
            CipherTypes type )  [virtual]
```

Implements gdcm::CryptographicMessageSyntax.

10.43.2.8 SetPassword()

```
bool gdcm::CAPICryptographicMessageSyntax::SetPassword (
            const char * pass,
            size_t passLen )  [virtual]
```

Implements gdcm::CryptographicMessageSyntax.

The documentation for this class was generated from the following file:

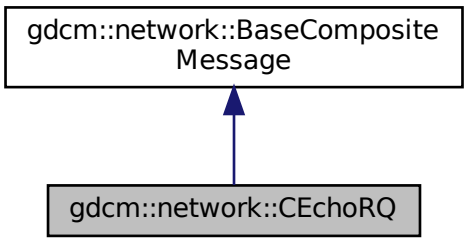
- gdcmCAPICryptographicMessageSyntax.h

10.44 gdcm::network::CEchoRQ Class Reference

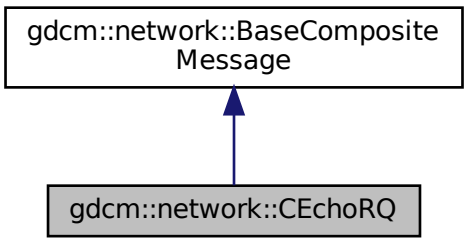
CEchoRQ.

`#include <gdcmCEchoMessages.h>`

Inheritance diagram for gdcm::network::CEchoRQ:



Collaboration diagram for gdcm::network::CEchoRQ:



Public Member Functions

- std::vector< PresentationDataValue > ConstructPDV (const ULConnection &inConnection, const BaseRootQuery *inRootQuery) override

Public Attributes

- UIComp AffectedSOPClassUID
- uint16_t MessageID

10.44.1 Detailed Description

CEchoRQ.

this file defines the messages for the cecho action

10.44.2 Member Function Documentation

10.44.2.1 ConstructPDV()

```
std::vector< PresentationDataValue > gdcm::network::CEchoRQ::ConstructPDV (
            const ULConnection & inConnection,
            const BaseRootQuery * inRootQuery )  [override], [virtual]
```

Implements gdcm::network::BaseCompositeMessage.

10.44.3 Member Data Documentation

10.44.3.1 AffectedSOPClassUID

```
UIComp gdcm::network::CEchoRQ::AffectedSOPClassUID
```

10.44.3.2 MessageID

```
uint16_t gdcm::network::CEchoRQ::MessageID
```

The documentation for this class was generated from the following files:

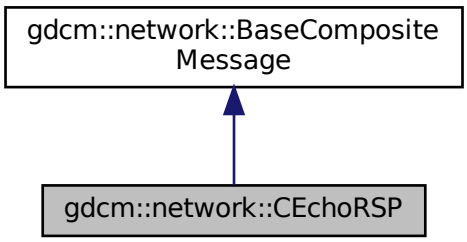
- gdcmCEchoMessages.h
- gdcmDIMSE.h

10.45 gdcm::network::CEchoRSP Class Reference

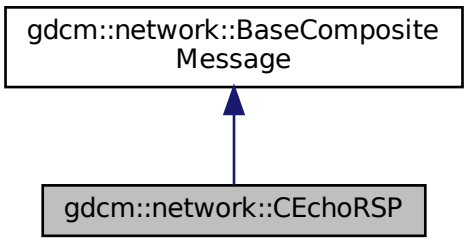
CEchoRSP this file defines the messages for the cecho action.

```
#include <gdcmCEchoMessages.h>
```

Inheritance diagram for gdcm::network::CEchoRSP:



Collaboration diagram for gdcm::network::CEchoRSP:



Public Member Functions

- std::vector< PresentationDataValue > ConstructPDVByDataSet (const DataSet *inDataSet)

10.45.1 Detailed Description

CEchoRSP this file defines the messages for the cecho action.

10.45.2 Member Function Documentation

10.45.2.1 ConstructPDVByDataSet()

```
std::vector< PresentationDataValue > gdcm::network::CEchoRSP::ConstructPDVByDataSet (
            const DataSet * inDataSet )
```

The documentation for this class was generated from the following file:

- gdcmCEchoMessages.h

10.46 gdcm::network::CFind Class Reference

```
#include <gdcmDIMSE.h>
```

10.46.1 Detailed Description

PS 3.4 - 2009 Table B.2-1 C-STORE STATUS

The documentation for this class was generated from the following file:

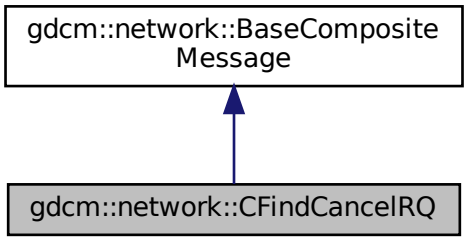
- gdcmDIMSE.h

10.47 gdcm::network::CFindCancelRQ Class Reference

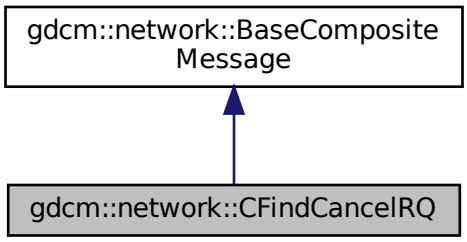
CFindCancelRQ this file defines the messages for the cfind action.

```
#include <gdcmCFindMessages.h>
```

Inheritance diagram for gdcm::network::CFindCancelRQ:



Collaboration diagram for gdcm::network::CFindCancelRQ:



Public Member Functions

- std::vector< PresentationDataValue > ConstructPDVByDataSet (const DataSet *inDataSet)

10.47.1 Detailed Description

CFindCancelRQ this file defines the messages for the cfind action.

10.47.2 Member Function Documentation

10.47.2.1 ConstructPDVByDataSet()

```
std::vector< PresentationDataValue > gdcm::network::CFindCancelRQ::ConstructPDVByDataSet (
            const DataSet * inDataSet )
```

The documentation for this class was generated from the following file:

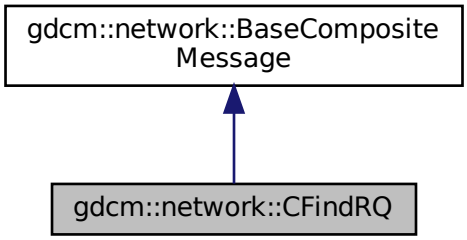
- gdcmCFindMessages.h

10.48 gdcm::network::CFindRQ Class Reference

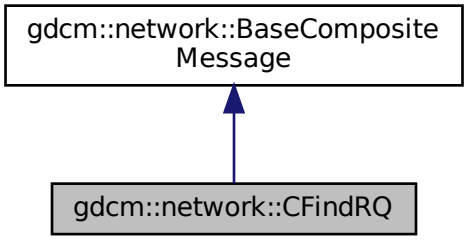
CFindRQ.

`#include <gdcmCFindMessages.h>`

Inheritance diagram for gdcm::network::CFindRQ:



Collaboration diagram for gdcm::network::CFindRQ:



Public Member Functions

- std::vector< PresentationDataValue > ConstructPDV (const ULConnection &inConnection, const BaseRootQuery *inRootQuery) override

10.48.1 Detailed Description

CFindRQ.

this file defines the messages for the cfind action

10.48.2 Member Function Documentation

10.48.2.1 ConstructPDV()

```
std::vector< PresentationDataValue > gdcm::network::CFindRQ::ConstructPDV (
            const ULConnection & inConnection,
            const BaseRootQuery * inRootQuery )  [override], [virtual]
```

Implements gdcm::network::BaseCompositeMessage.

The documentation for this class was generated from the following file:

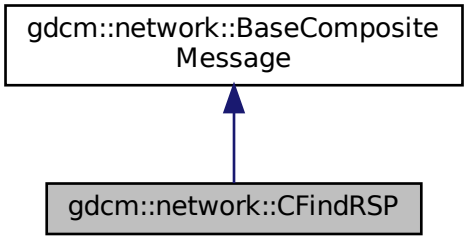
- gdcmCFindMessages.h

10.49 gdcm::network::CFindRSP Class Reference

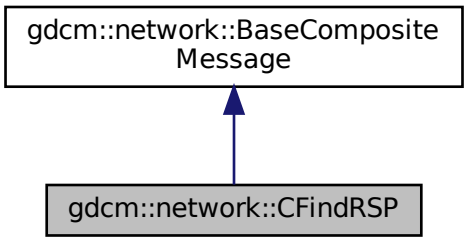
CFindRSP this file defines the messages for the cfind action.

```
#include <gdcmCFindMessages.h>
```

Inheritance diagram for gdcm::network::CFindRSP:



Collaboration diagram for gdcm::network::CFindRSP:



Public Member Functions

- std::vector< PresentationDataValue > ConstructPDVByDataSet (const DataSet *inDataSet)

10.49.1 Detailed Description

CFindRSP this file defines the messages for the cfind action.

10.49.2 Member Function Documentation

10.49.2.1 ConstructPDVByDataSet()

```
std::vector< PresentationDataValue > gdcm::network::CFindRSP::ConstructPDVByDataSet (
            const DataSet * inDataSet )
```

The documentation for this class was generated from the following file:

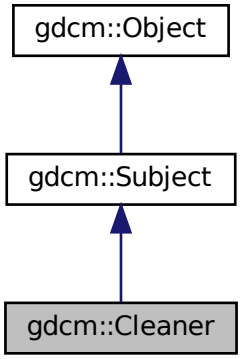
- gdcmCFindMessages.h

10.50 gdcm::Cleaner Class Reference

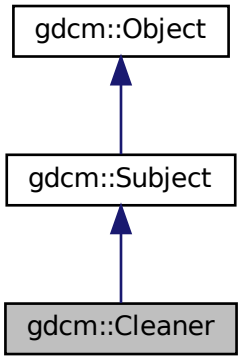
Cleaner.

`#include <gdcmCleaner.h>`

Inheritance diagram for gdcm::Cleaner:



Collaboration diagram for gdcm::Cleaner:



Public Member Functions

- Cleaner ()
- ∼Cleaner ()
- bool Clean ()

  main loop
- bool Empty (DPath const &dpath)
- bool Empty (PrivateTag const &pt)
- bool Empty (Tag const &t)
- bool Empty (VR const &vr)
- File & GetFile ()
- bool Preserve (DPath const &dpath)
- bool Remove (DPath const &dpath)
- bool Remove (PrivateTag const &pt)
- bool Remove (Tag const &t)
- bool Remove (VR const &vr)
- void RemoveAllGroupLength (bool remove)

  Should I remove all group length (deprecated). Default: true.
- void RemoveAllIllegal (bool remove)

  Should I remove all illegal attribute. Default: true.
- void RemoveAllMissingPrivateCreator (bool remove)
- bool RemoveMissingPrivateCreator (Tag const &t)
- bool Scrub (DPath const &dpath)
- bool Scrub (PrivateTag const &pt)
- bool Scrub (Tag const &t)

  Clean digital tash (typically SIEMENS CSA header):
- bool Scrub (VR const &vr)
- void SetFile (const File &f)

  Set/Get File.

Static Public Member Functions

- static SmartPointer< Cleaner > New ()

  for wrapped language: instantiate a reference counted object

Additional Inherited Members

10.50.1 Detailed Description

Cleaner.

This class implement the Subject/Observer pattern trigger the following event:

- AnonymizeEvent
- IterationEvent
- StartEvent
- EndEvent

10.50.2 Constructor & Destructor Documentation

10.50.2.1 Cleaner()

```
gdcm::Cleaner::Cleaner ( )
```

10.50.2.2 ~Cleaner()

```
gdcm::Cleaner::~Cleaner ( )
```

10.50.3 Member Function Documentation

10.50.3.1 Clean()

```
bool gdcm::Cleaner::Clean ( )
```

main loop

10.50.3.2 Empty() [1/4]

```
bool gdcm::Cleaner::Empty (
            DPath const & dpath )
```

10.50.3.3 Empty() [2/4]

```
bool gdcm::Cleaner::Empty (
            PrivateTag const & pt )
```

10.50.3.4 Empty() [3/4]

```
bool gdcm::Cleaner::Empty (
            Tag const & t )
```

10.50.3.5 Empty() [4/4]

```
bool gdcm::Cleaner::Empty (
            VR const & vr )
```

10.50.3.6 GetFile()

```
File & gdcm::Cleaner::GetFile ( )  [inline]
```

10.50.3.7 New()

```
static SmartPointer< Cleaner > gdcm::Cleaner::New ( )  [inline], [static]
```

for wrapped language: instantiate a reference counted object

10.50.3.8 Preserve()

```
bool gdcm::Cleaner::Preserve (
            DPath const & dpath )
```

10.50.3.9 Remove() [1/4]

```
bool gdcm::Cleaner::Remove (
            DPath const & dpath )
```

10.50.3.10 Remove() [2/4]

```
bool gdcm::Cleaner::Remove (
            PrivateTag const & pt )
```

10.50.3.11 Remove() [3/4]

```
bool gdcm::Cleaner::Remove (
            Tag const & t )
```

10.50.3.12 Remove() [4/4]

```
bool gdcm::Cleaner::Remove (
            VR const & vr )
```

10.50.3.13 RemoveAllGroupLength()

```
void gdcm::Cleaner::RemoveAllGroupLength (
            bool remove )
```

Should I remove all group length (deprecated). Default: true.

10.50.3.14 RemoveAllIllegal()

```
void gdcm::Cleaner::RemoveAllIllegal (
            bool remove )
```

Should I remove all illegal attribute. Default: true.

10.50.3.15 RemoveAllMissingPrivateCreator()

```
void gdcm::Cleaner::RemoveAllMissingPrivateCreator (
            bool remove )
```

Should I remove all private tag for which no private creator is found. Default: true

10.50.3.16 RemoveMissingPrivateCreator()

```
bool gdcm::Cleaner::RemoveMissingPrivateCreator (
            Tag const & t )
```

Specify a private tag (odd number) without a private creator (root level only for now):

10.50.3.17 Scrub() [1/4]

```
bool gdcm::Cleaner::Scrub (
            DPath const & dpath )
```

10.50.3.18 Scrub() [2/4]

```
bool gdcm::Cleaner::Scrub (
            PrivateTag const & pt )
```

10.50.3.19 Scrub() [3/4]

```
bool gdcm::Cleaner::Scrub (
            Tag const & t )
```

Clean digital tash (typically SIEMENS CSA header):

10.50.3.20 Scrub() [4/4]

```
bool gdcm::Cleaner::Scrub (
            VR const & vr )
```

10.50.3.21 SetFile()

```
void gdcm::Cleaner::SetFile (
            const File & f )  [inline]
```

Set/Get File.

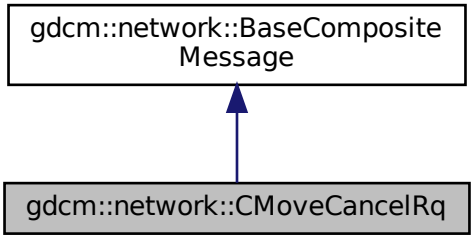
The documentation for this class was generated from the following file:

- gdcmCleaner.h

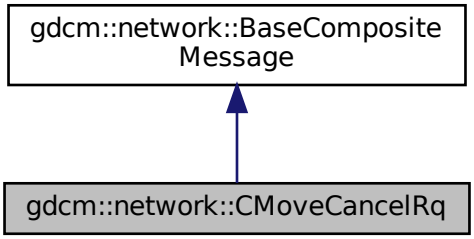
10.51 gdcm::network::CMoveCancelRq Class Reference

`#include <gdcmCMoveMessages.h>`

Inheritance diagram for gdcm::network::CMoveCancelRq:



Collaboration diagram for gdcm::network::CMoveCancelRq:



Public Member Functions

- std::vector< PresentationDataValue > ConstructPDVByDataSet (const DataSet *inDataSet)

10.51.1 Member Function Documentation

10.51.1.1 ConstructPDVByDataSet()

```
std::vector< PresentationDataValue > gdcm::network::CMoveCancelRq::ConstructPDVByDataSet (
            const DataSet * inDataSet )
```

The documentation for this class was generated from the following file:

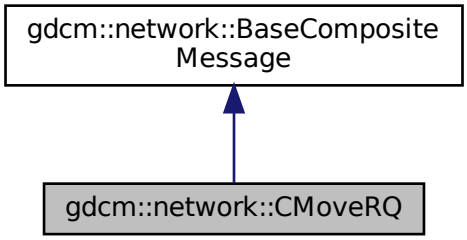
- gdcmCMoveMessages.h

10.52 gdcm::network::CMoveRQ Class Reference

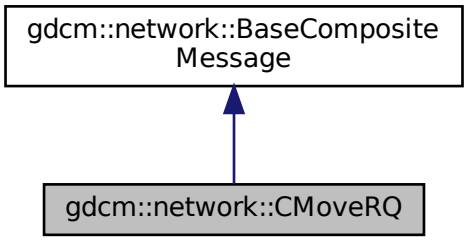
CMoveRQ.

```
#include <gdcmCMoveMessages.h>
```

Inheritance diagram for gdcm::network::CMoveRQ:



Collaboration diagram for gdcm::network::CMoveRQ:



Public Member Functions

- std::vector< PresentationDataValue > ConstructPDV (const ULConnection &inConnection, const BaseRootQuery *inRootQuery) override

10.52.1 Detailed Description

CMoveRQ.

this file defines the messages for the cmove action

10.52.2 Member Function Documentation

10.52.2.1 ConstructPDV()

```
std::vector< PresentationDataValue > gdcm::network::CMoveRQ::ConstructPDV (
            const ULConnection & inConnection,
            const BaseRootQuery * inRootQuery )  [override], [virtual]
```

Implements gdcm::network::BaseCompositeMessage.

The documentation for this class was generated from the following file:

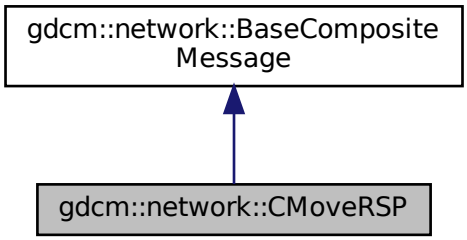
- gdcmCMoveMessages.h

10.53 gdcm::network::CMoveRSP Class Reference

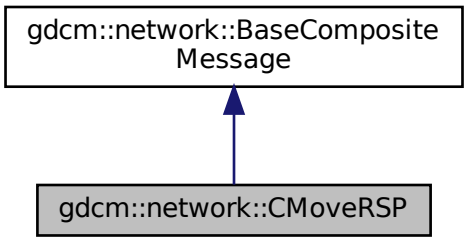
CMoveRSP this file defines the messages for the cmove action.

```
#include <gdcmCMoveMessages.h>
```

Inheritance diagram for gdcm::network::CMoveRSP:



Collaboration diagram for gdcm::network::CMoveRSP:



Public Member Functions

- std::vector< PresentationDataValue > ConstructPDVByDataSet (const DataSet *inDataSet)

10.53.1 Detailed Description

CMoveRSP this file defines the messages for the cmove action.

10.53.2 Member Function Documentation

10.53.2.1 ConstructPDVByDataSet()

```
std::vector< PresentationDataValue > gdcm::network::CMoveRSP::ConstructPDVByDataSet (
            const DataSet * inDataSet )
```

The documentation for this class was generated from the following file:

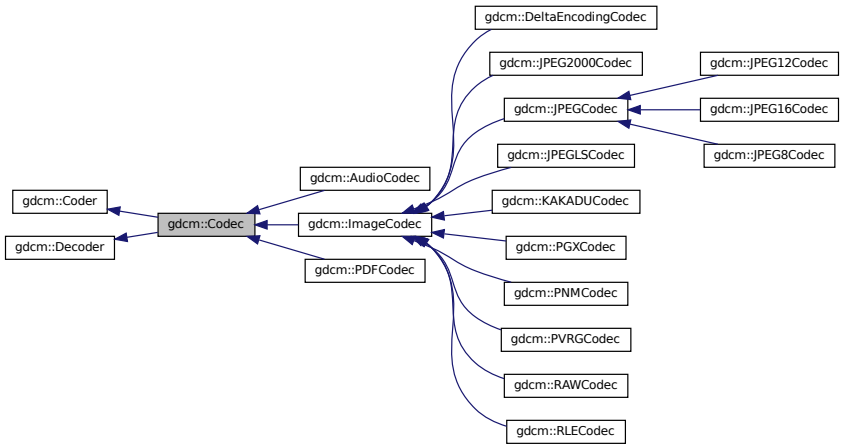
- gdcmCMoveMessages.h

10.54 gdcm::Codec Class Reference

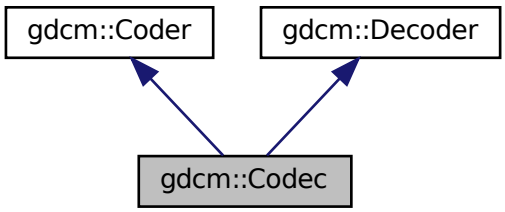
Codec class.

`#include <gdcmCodec.h>`

Inheritance diagram for gdcm::Codec:



Collaboration diagram for gdcm::Codec:



Additional Inherited Members

10.54.1 Detailed Description

Codec class.

The documentation for this class was generated from the following file:

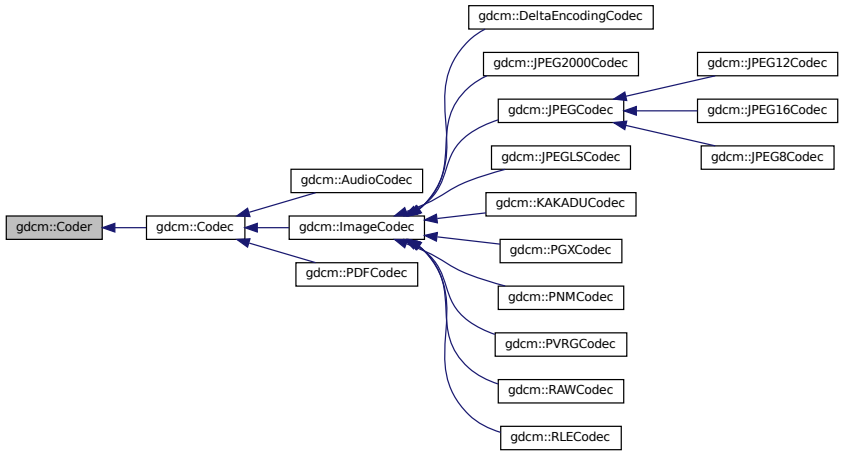
- gdcmCodec.h

10.55 gdcm::Coder Class Reference

Coder.

`#include <gdcmCoder.h>`

Inheritance diagram for gdcm::Coder:



Public Member Functions

- virtual ~Coder ()=default
- virtual bool CanCode (TransferSyntax const &) const =0

  Return whether this coder support this transfer syntax (can code it)
- virtual bool Code (DataElement const &in_, DataElement &out_)

  Code.

Protected Member Functions

- virtual bool InternalCode (const char *bv, unsigned long len, std::ostream &os)

10.55.1 Detailed Description

Coder.

10.55.2 Constructor & Destructor Documentation

10.55.2.1 ~Coder()

```
virtual gdcm::Coder::~Coder ( )  [virtual], [default]
```

10.55.3 Member Function Documentation

10.55.3.1 CanCode()

```
virtual bool gdcm::Coder::CanCode (
            TransferSyntax const &  ) const  [pure virtual]
```

Return whether this coder support this transfer syntax (can code it)

Implemented in gdcm::AudioCodec, gdcm::ImageCodec, gdcm::PDFCodec, gdcm::JPEG2000Codec, gdcm::JPEGCodec, gdcm::JPEGLSCodec, gdcm::KAKADUCodec, gdcm::PGXCodec, gdcm::PNMCodec, gdcm::PVRGCodec, gdcm::RAWCodec, and gdcm::RLECodec.

10.55.3.2 Code()

```
virtual bool gdcm::Coder::Code (
            DataElement const & in_,
            DataElement & out_ )  [inline], [virtual]
```

Code.

Reimplemented in gdcm::JPEG2000Codec, gdcm::JPEGCodec, gdcm::JPEGLSCodec, gdcm::KAKADUCodec, gdcm::PVRGCodec, gdcm::RAWCodec, and gdcm::RLECodec.

10.55.3.3 InternalCode()

```
virtual bool gdcm::Coder::InternalCode (
            const char * bv,
            unsigned long len,
            std::ostream & os )  [inline], [protected], [virtual]
```

Reimplemented in gdcm::JPEG12Codec, gdcm::JPEG16Codec, and gdcm::JPEG8Codec.

The documentation for this class was generated from the following file:

- gdcmCoder.h

10.56 gdcm::CodeString Class Reference

CodeString.

`#include <gdcmCodeString.h>`

Public Types

- typedef InternalClass::const_iterator const_iterator
- typedef InternalClass::const_reference const_reference
- typedef InternalClass::const_reverse_iterator const_reverse_iterator
- typedef InternalClass::difference_type difference_type
- typedef InternalClass::iterator iterator
- typedef InternalClass::pointer pointer
- typedef InternalClass::reference reference
- typedef InternalClass::reverse_iterator reverse_iterator
- typedef InternalClass::size_type size_type
- typedef InternalClass::value_type value_type

Public Member Functions

- CodeString ()

  CodeString constructors.
- CodeString (const InternalClass &s, size_type pos=0, size_type n=InternalClass::npos)
- CodeString (const value_type *s)
- CodeString (const value_type *s, size_type n)
- std::string GetAsString () const

  Return the full code string as std::string.
- bool IsValid () const

  Check if CodeString obj is correct..
- size_type Size () const

  Return the size of the string.

Protected Member Functions

- std::string TrimInternal () const

Friends

- bool operator!= (const CodeString &ref, const CodeString &cs)
- std::ostream & operator<< (std::ostream &os, const CodeString &str)
- bool operator== (const CodeString &ref, const CodeString &cs)

10.56.1 Detailed Description

CodeString.

This is an implementation of DICOM VR: CS The cstor will properly Trim so that operator== is correct.

Note

the cstor of CodeString will Trim the string on the fly so as to remove the extra leading and ending spaces. However it will not perform validation on the fly (CodeString obj can contains invalid char such as lower cases). This design was chosen to be a little tolerant to broken DICOM implementation, and thus allow user to compare lower case CS from there input file without the need to first rewrite them to get rid of invalid character (validation is a different operation from searching, querying).

Warning

when writing out DICOM file it is highly recommended to perform the IsValid() call, at least to check that the length of the string match the definition in the standard.

10.56.2 Member Typedef Documentation

10.56.2.1 const_iterator

```
typedef InternalClass::const_iterator gdcm::CodeString::const_iterator
```

10.56.2.2 const_reference

```
typedef InternalClass::const_reference gdcm::CodeString::const_reference
```

10.56.2.3 const_reverse_iterator

```
typedef InternalClass::const_reverse_iterator gdcm::CodeString::const_reverse_iterator
```

10.56.2.4 difference_type

```
typedef InternalClass::difference_type gdcm::CodeString::difference_type
```

10.56.2.5 iterator

```
typedef InternalClass::iterator gdcm::CodeString::iterator
```

10.56.2.6 pointer

```
typedef InternalClass::pointer gdcm::CodeString::pointer
```

10.56.2.7 reference

```
typedef InternalClass::reference gdcm::CodeString::reference
```

10.56.2.8 reverse_iterator

```
typedef InternalClass::reverse_iterator gdcm::CodeString::reverse_iterator
```

10.56.2.9 size_type

```
typedef InternalClass::size_type gdcm::CodeString::size_type
```

10.56.2.10 value_type

```
typedef InternalClass::value_type gdcm::CodeString::value_type
```

10.56.3 Constructor & Destructor Documentation

10.56.3.1 CodeString() [1/4]

```
gdcm::CodeString::CodeString ( )  [inline]
```

CodeString constructors.

10.56.3.2 CodeString() [2/4]

```
gdcm::CodeString::CodeString (
            const value_type * s )  [inline]
```

10.56.3.3 CodeString() [3/4]

```
gdcm::CodeString::CodeString (
            const value_type * s,
            size_type n )  [inline]
```

10.56.3.4 CodeString() [4/4]

```
gdcm::CodeString::CodeString (
            const InternalClass & s,
            size_type pos = 0,
            size_type n = InternalClass::npos )  [inline]
```

10.56.4 Member Function Documentation

10.56.4.1 GetAsString()

```
std::string gdcm::CodeString::GetAsString ( ) const  [inline]
```

Return the full code string as std::string.

10.56.4.2 IsValid()

```
bool gdcm::CodeString::IsValid ( ) const
```

Check if CodeString obj is correct..

10.56.4.3 Size()

```
size_type gdcm::CodeString::Size ( ) const  [inline]
```

Return the size of the string.

10.56.4.4 TrimInternal()

```
std::string gdcm::CodeString::TrimInternal ( ) const  [inline], [protected]
```

10.56.5 Friends And Related Function Documentation

10.56.5.1 operator"!=

```
bool operator!= (
            const CodeString & ref,
            const CodeString & cs )  [friend]
```

10.56.5.2 operator<<

```
std::ostream & operator<< (
            std::ostream & os,
            const CodeString & str )  [friend]
```

10.56.5.3 operator==

```
bool operator== (
            const CodeString & ref,
            const CodeString & cs )  [friend]
```

The documentation for this class was generated from the following file:

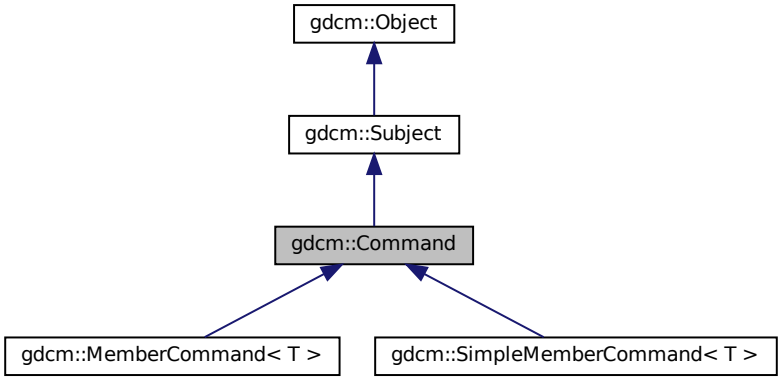
- gdcmCodeString.h

10.57 gdcm::Command Class Reference

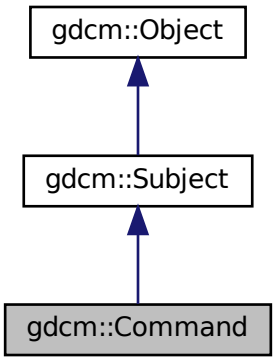
Command superclass for callback/observer methods.

```
#include <gdcmCommand.h>
```

Inheritance diagram for gdcm::Command:



Collaboration diagram for gdcm::Command:



Public Member Functions

- Command (const Command &)=delete
- virtual void Execute (const Subject *caller, const Event &event)=0
- virtual void Execute (Subject *caller, const Event &event)=0

  Abstract method that defines the action to be taken by the command.
- void operator= (const Command &)=delete

Protected Member Functions

- Command ()
- ~Command () override

10.57.1 Detailed Description

Command superclass for callback/observer methods.

See also

Subject

10.57.2 Constructor & Destructor Documentation

10.57.2.1 Command() [1/2]

```
gdcm::Command::Command (
            const Command &  )  [delete]
```

10.57.2.2 Command() [2/2]

```
gdcm::Command::Command ( )  [protected]
```

10.57.2.3 ~Command()

```
gdcm::Command::~Command ( )  [override], [protected]
```

10.57.3 Member Function Documentation

10.57.3.1 Execute() [1/2]

```
virtual void gdcm::Command::Execute (
            const Subject * caller,
            const Event & event )  [pure virtual]
```

Abstract method that defines the action to be taken by the command. This variant is expected to be used when requests comes from a const Object

Implemented in gdcm::SimpleMemberCommand< T >, and gdcm::MemberCommand< T >.

10.57.3.2 Execute() [2/2]

```
virtual void gdcm::Command::Execute (
            Subject * caller,
            const Event & event )  [pure virtual]
```

Abstract method that defines the action to be taken by the command.

Implemented in gdcm::SimpleMemberCommand< T >, and gdcm::MemberCommand< T >.

10.57.3.3 operator=()

```
void gdcm::Command::operator= (
            const Command &  )  [delete]
```

The documentation for this class was generated from the following file:

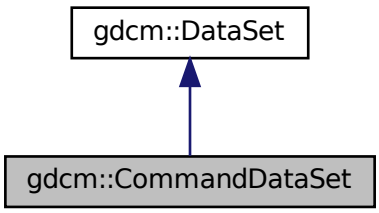
- gdcmCommand.h

10.58 gdcm::CommandDataSet Class Reference

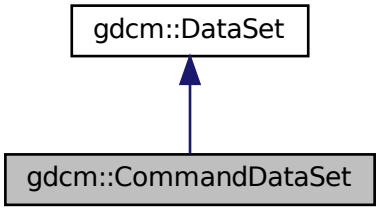
Class to represent a Command DataSet.

```
#include <gdcmCommandDataSet.h>
```

Inheritance diagram for gdcm::CommandDataSet:



Collaboration diagram for gdcm::CommandDataSet:



Public Member Functions

- CommandDataSet ()=default
- ~CommandDataSet ()=default
- void Insert (const DataElement &de)
- std::istream & Read (std::istream &is)

  Read.
- void Replace (const DataElement &de)
- std::ostream & Write (std::ostream &os) const

  Write.

Friends

- std::ostream & operator<< (std::ostream &_os, const CommandDataSet &_val)

Additional Inherited Members

10.58.1 Detailed Description

Class to represent a Command DataSet.

See also

DataSet

10.58.2 Constructor & Destructor Documentation

10.58.2.1 CommandDataSet()

```
gdcm::CommandDataSet::CommandDataSet ( )  [default]
```

10.58.2.2 ~CommandDataSet()

```
gdcm::CommandDataSet::~CommandDataSet ( )  [default]
```

10.58.3 Member Function Documentation

10.58.3.1 Insert()

```
void gdcm::CommandDataSet::Insert (
            const DataElement & de )  [inline]
```

References gdcmErrorMacro, gdcm::Tag::GetGroup(), and gdcm::DataElement::GetTag().

10.58.3.2 Read()

```
std::istream & gdcm::CommandDataSet::Read (
            std::istream & is )
```

Read.

10.58.3.3 Replace()

```
void gdcm::CommandDataSet::Replace (
            const DataElement & de )  [inline]
```

References gdcm::DataElement::GetTag().

10.58.3.4 Write()

```
std::ostream & gdcm::CommandDataSet::Write (
            std::ostream & os ) const
```

Write.

10.58.4 Friends And Related Function Documentation

10.58.4.1 operator<<

```
std::ostream & operator<< (
            std::ostream & _os,
            const CommandDataSet & _val )  [friend]
```

The documentation for this class was generated from the following file:

- gdcmCommandDataSet.h

10.59 gdcm::network::CompositeMessageFactory Class Reference

CompositeMessageFactory.

`#include <gdcmCompositeMessageFactory.h>`

Static Public Member Functions

- static std::vector< PresentationDataValue > ConstructCEchoRQ (const ULConnection &inConnection)
- static std::vector< PresentationDataValue > ConstructCFindRQ (const ULConnection &inConnection, const BaseRootQuery *inRootQuery)
- static std::vector< PresentationDataValue > ConstructCMoveRQ (const ULConnection &inConnection, const BaseRootQuery *inRootQuery)
- static std::vector< PresentationDataValue > ConstructCStoreRQ (const ULConnection &inConnection, const File &file, bool writeDataSet=true)
- static std::vector< PresentationDataValue > ConstructCStoreRSP (const DataSet *inDataSet, const BasePDU *inPC)

10.59.1 Detailed Description

CompositeMessageFactory.

This class constructs PDataPDUs, but that have been specifically constructed for the composite DICOM services (C-Echo, C-Find, C-Get, C-Move, and C-Store). It will also handle parsing the incoming data to determine which of the CompositePDUs the incoming data is, and so therefore allowing the scu to determine what to do with incoming data (if acting as a storescp server, for instance).

10.59.2 Member Function Documentation

10.59.2.1 ConstructCEchoRQ()

```
static std::vector< PresentationDataValue > gdcm::network::CompositeMessageFactory::ConstructCEchoRQ (
            const ULConnection & inConnection )  [static]
```

10.59.2.2 ConstructCFindRQ()

```
static std::vector< PresentationDataValue > gdcm::network::CompositeMessageFactory::ConstructCFindRQ (
            const ULConnection & inConnection,
            const BaseRootQuery * inRootQuery )  [static]
```

10.59.2.3 ConstructCMoveRQ()

```
static std::vector< PresentationDataValue > gdcm::network::CompositeMessageFactory::Construct↩
CMoveRQ (
            const ULConnection & inConnection,
            const BaseRootQuery * inRootQuery )  [static]
```

10.59.2.4 ConstructCStoreRQ()

```
static std::vector< PresentationDataValue > gdcm::network::CompositeMessageFactory::Construct↩
CStoreRQ (
            const ULConnection & inConnection,
            const File & file,
            bool writeDataSet = true )  [static]
```

10.59.2.5 ConstructCStoreRSP()

```
static std::vector< PresentationDataValue > gdcm::network::CompositeMessageFactory::Construct↩
CStoreRSP (
            const DataSet * inDataSet,
            const BasePDU * inPC )  [static]
```

The documentation for this class was generated from the following file:

- gdcmCompositeMessageFactory.h

10.60 gdcm::CompositeNetworkFunctions Class Reference

Composite Network Functions.

```
#include <gdcmCompositeNetworkFunctions.h>
```

Public Types

- typedef std::vector< KeyValuePairType > KeyValuePairArrayType
- typedef std::pair< Tag, std::string > KeyValuePairType

Static Public Member Functions

- static bool CEcho (const char *remote, uint16_t portno, const char *aetitle=nullptr, const char *call=nullptr)
- static bool CFind (const char *remote, uint16_t portno, const BaseRootQuery *query, std::vector< DataSet > &retDataSets, const char *aetitle=nullptr, const char *call=nullptr)
- static bool CMove (const char *remote, uint16_t portno, const BaseRootQuery *query, uint16_t portscp, const char *aetitle=nullptr, const char *call=nullptr, const char *outputdir=nullptr)
- static BaseRootQuery * ConstructQuery (ERootType inRootType, EQueryLevel inQueryLevel, const DataSet &queryds, EQueryType queryType=eFind)
- static BaseRootQuery * ConstructQuery (ERootType inRootType, EQueryLevel inQueryLevel, const KeyValuePairArrayType &keys, EQueryType queryType=eFind)
- static bool CStore (const char *remote, uint16_t portno, const Directory::FilenamesType &filenames, const char *aetitle=nullptr, const char *call=nullptr)

10.60.1 Detailed Description

Composite Network Functions.

These functions provide a generic API to the DICOM functions implemented in GDCM. Advanced users can use this code as a template for building their own versions of these functions (for instance, to provide progress bars or some other way of handling returned query information), but for most users, these functions should be sufficient to interface with a PACS to a local machine. Note that these functions are not contained within a static class or some other class-style interface, because multiple connections can be instantiated in the same program. The DICOM standard is much more function oriented rather than class oriented in this instance, so the design of this API reflects that functional approach. These functions implements the following SCU operations:

- C-ECHO SCU
- C-FIND SCU
- C-STORE SCU
- C-MOVE SCU (+internal C-STORE SCP)

Examples

SendFileSCU.cs.

10.60.2 Member Typedef Documentation

10.60.2.1 KeyValuePairArrayType

```
typedef std::vector< KeyValuePairType > gdcm::CompositeNetworkFunctions::KeyValuePairArrayType
```

10.60.2.2 KeyValuePairType

```
typedef std::pair<Tag, std::string> gdcm::CompositeNetworkFunctions::KeyValuePairType
```

10.60.3 Member Function Documentation

10.60.3.1 CEcho()

```
static bool gdcm::CompositeNetworkFunctions::CEcho (
            const char * remote,
            uint16_t portno,
            const char * aetitle = nullptr,
            const char * call = nullptr )  [static]
```

The most basic network function. Use this function to ensure that the remote server is responding on the given IP and port number as expected.

Parameters

aetitle	when not set will default to 'GDCMSCU'
call	when not set will default to 'ANY-SCP'

Warning

This is an error to set remote to NULL or portno to 0

Returns

true if it worked.

Examples

SendFileSCU.cs.

10.60.3.2 CFind()

```
static bool gdcm::CompositeNetworkFunctions::CFind (
            const char * remote,
            uint16_t portno,
            const BaseRootQuery * query,
            std::vector< DataSet > & retDataSets,
```

```
const char * aetitle = nullptr,
const char * call = nullptr )  [static]
```

This function will use the provided query to determine what files a remote server contains that match the query strings. The return is a vector of datasets that contain tags as reported by the server. If the dataset is empty, then it is possible that an error condition was encountered; in which case, the user should monitor the error and warning streams.

Parameters

aetitle	when not set will default to 'GDCMSCU'
call	when not set will default to 'ANY-SCP'

Warning

This is an error to set remote to NULL or portno to 0

Returns

true if it worked.

10.60.3.3 CMove()

```
static bool gdcm::CompositeNetworkFunctions::CMove (
            const char * remote,
            uint16_t portno,
            const BaseRootQuery * query,
            uint16_t portscp,
            const char * aetitle = nullptr,
            const char * call = nullptr,
            const char * outputdir = nullptr )  [static]
```

This function will use the provided query to get files from a remote server. NOTE that this functionality is essentially equivalent to C-GET in the DICOM standard; however, C-GET has been deprecated, so this function allows for the user to ask a remote server for files matching a query and return them to the local machine. Files will be written to the given output directory. If the operation succeeds, the function returns true. This function is a prime candidate for being overwritten by expert users; if the datasets should remain in memory, for instance, that behavior can be changed by creating a user-level version of this function.

Parameters

aetitle	when not set will default to 'GDCMSCU'
call	when not set will default to 'ANY-SCP' This is an error to set remote to NULL or portno to 0 when
outputdir	is not set default to current dir ('.')

Returns

true if it worked.

10.60.3.4 ConstructQuery() [1/2]

```
static BaseRootQuery * gdcm::CompositeNetworkFunctions::ConstructQuery (
            ERootType inRootType,
            EQueryLevel inQueryLevel,
            const DataSet & queryds,
            EQueryType queryType = eFind )  [static]
```

This function will take a list of strings and tags and fill in a query that can be used for either CFind or CMove (depending on the input boolean

Parameters

inMove).	Note that the caller is responsible for deleting the constructed query. This function is used to build both a move and a find query (true for inMove if it's move, false if it's find)

10.60.3.5 ConstructQuery() [2/2]

```
static BaseRootQuery * gdcm::CompositeNetworkFunctions::ConstructQuery (
            ERootType inRootType,
            EQueryLevel inQueryLevel,
            const KeyValuePairArrayType & keys,
            EQueryType queryType = eFind )  [static]
```

Deprecated

10.60.3.6 CStore()

```
static bool gdcm::CompositeNetworkFunctions::CStore (
            const char * remote,
            uint16_t portno,
            const Directory::FilenamesType & filenames,
            const char * aetitle = nullptr,
            const char * call = nullptr )  [static]
```

This function will place the provided files into the remote server. The function returns true if it worked for all files.

Warning

the server side can refuse an association on a given file

Parameters

aetitle	when not set will default to 'GDCMSCU'
call	when not set will default to 'ANY-SCP'

Warning

This is an error to set remote to NULL or portno to 0

Returns

true if it worked for all files

Examples

SendFileSCU.cs.

The documentation for this class was generated from the following file:

- gdcmCompositeNetworkFunctions.h

10.61 gdcm::ConstCharWrapper Class Reference

Do not use me.

`#include <gdcmConstCharWrapper.h>`

Public Member Functions

- ConstCharWrapper (const char *i=0)
- operator const char * () const

10.61.1 Detailed Description

Do not use me.

10.61.2 Constructor & Destructor Documentation

10.61.2.1 ConstCharWrapper()

```
gdcm::ConstCharWrapper::ConstCharWrapper (
            const char * i = 0 )  [inline]
```

10.61.3 Member Function Documentation

10.61.3.1 operator const char *()

```
gdcm::ConstCharWrapper::operator const char * ( ) const  [inline]
```

The documentation for this class was generated from the following file:

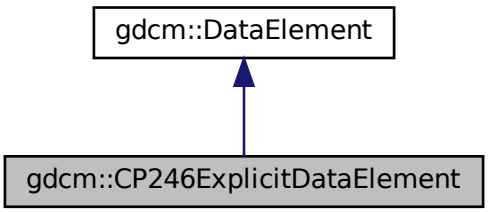
- gdcmConstCharWrapper.h

10.62 gdcm::CP246ExplicitDataElement Class Reference

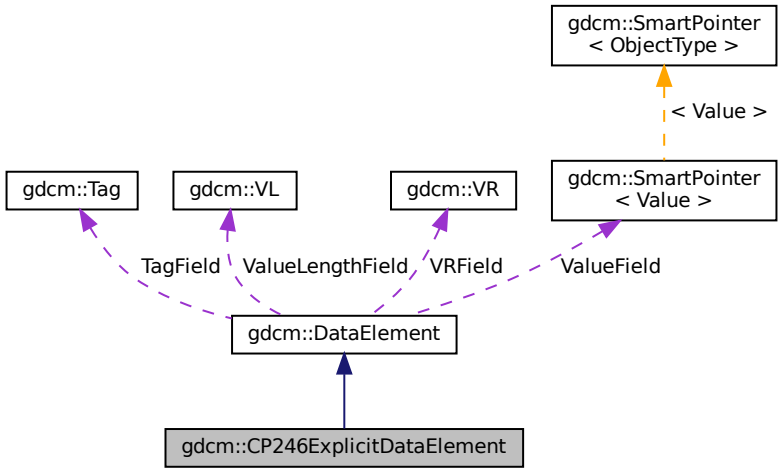
Class to read/write a DataElement as CP246Explicit Data Element.

```
#include <gdcmCP246ExplicitDataElement.h>
```

Inheritance diagram for gdcm::CP246ExplicitDataElement:



Collaboration diagram for gdcm::CP246ExplicitDataElement:



Public Member Functions

- VL GetLength () const
- template<typename TSwap >
  std::istream & Read (std::istream &is)
- template<typename TSwap >
  std::istream & ReadPreValue (std::istream &is)
- template<typename TSwap >
  std::istream & ReadValue (std::istream &is, bool readvalues=true)
- template<typename TSwap >
  std::istream & ReadWithLength (std::istream &is, VL &length)

Additional Inherited Members

10.62.1 Detailed Description

Class to read/write a DataElement as CP246Explicit Data Element.

Note

Some system are producing SQ, declare them as UN, but encode the SQ as 'Explicit' instead of Implicit

10.62.2 Member Function Documentation

10.62.2.1 GetLength()

```
VL gdcm::CP246ExplicitDataElement::GetLength ( ) const
```

10.62.2.2 Read()

```
template<typename TSwap >
std::istream & gdcm::CP246ExplicitDataElement::Read (
             std::istream & is )
```

10.62.2.3 ReadPreValue()

```
template<typename TSwap >
std::istream & gdcm::CP246ExplicitDataElement::ReadPreValue (
             std::istream & is )
```

10.62.2.4 ReadValue()

```
template<typename TSwap >
std::istream & gdcm::CP246ExplicitDataElement::ReadValue (
             std::istream & is,
             bool readvalues = true )
```

10.62.2.5 ReadWithLength()

```
template<typename TSwap >
std::istream & gdcm::CP246ExplicitDataElement::ReadWithLength (
             std::istream & is,
             VL & length )
```

The documentation for this class was generated from the following file:

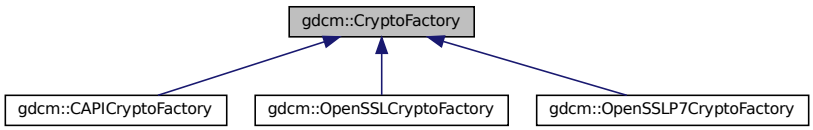
- gdcmCP246ExplicitDataElement.h

10.63 gdcm::CryptoFactory Class Reference

Class to do handle the crypto factory.

`#include <gdcmCryptoFactory.h>`

Inheritance diagram for gdcm::CryptoFactory:



Public Types

- enum CryptoLib {
  DEFAULT = 0 ,
  OPENSSL = 1 ,
  CAPI = 2 ,
  OPENSSLP7 = 3 }

Public Member Functions

- virtual CryptographicMessageSyntax * CreateCMSProvider ()=0

Static Public Member Functions

- static CryptoFactory * GetFactoryInstance (CryptoLib id=DEFAULT)

Protected Member Functions

- CryptoFactory ()=default
- CryptoFactory (CryptoLib id)
- ~CryptoFactory ()=default

10.63.1 Detailed Description

Class to do handle the crypto factory.

GDCM needs to access in a platform independent way the user specified crypto engine. It can be:

- CAPI (windows only)
- OPENSSL (portable)
- OPENSSLP7 (portable) By default the factory will try: CAPI if on windows OPENSSL if possible OPENSSLP7 when older OpenSSL is used.

Examples

BasicAnonymizer.cs, and ClinicalTrialIdentificationWorkflow.cs.

10.63.2 Member Enumeration Documentation

10.63.2.1 CryptoLib

```
enum gdcm::CryptoFactory::CryptoLib
```

Enumerator

DEFAULT	
OPENSSL	
CAPI	
OPENSSLP7	

10.63.3 Constructor & Destructor Documentation

10.63.3.1 CryptoFactory() [1/2]

```
gdcm::CryptoFactory::CryptoFactory (
            CryptoLib id )  [inline], [protected]
```

10.63.3.2 CryptoFactory() [2/2]

```
gdcm::CryptoFactory::CryptoFactory ( )  [protected], [default]
```

10.63.3.3 ~CryptoFactory()

```
gdcm::CryptoFactory::~CryptoFactory ( )  [protected], [default]
```

10.63.4 Member Function Documentation

10.63.4.1 CreateCMSProvider()

```
virtual CryptographicMessageSyntax * gdcm::CryptoFactory::CreateCMSProvider ( )  [pure virtual]
```

Implemented in gdcm::CAPICryptoFactory, gdcm::OpenSSLCryptoFactory, and gdcm::OpenSSLP7CryptoFactory.

Examples

BasicAnonymizer.cs, and ClinicalTrialIdentificationWorkflow.cs.

10.63.4.2 GetFactoryInstance()

```
static CryptoFactory * gdcm::CryptoFactory::GetFactoryInstance (
            CryptoLib id = DEFAULT )  [static]
```

Examples

BasicAnonymizer.cs, and ClinicalTrialIdentificationWorkflow.cs.

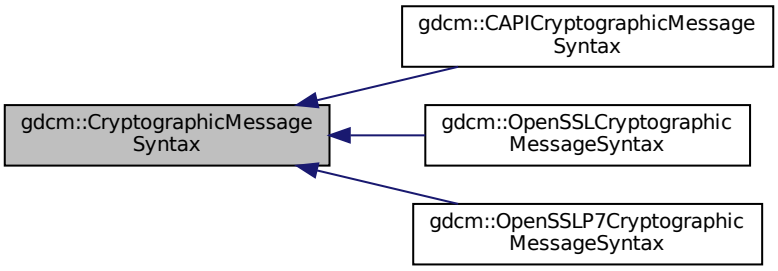
The documentation for this class was generated from the following file:

- gdcmCryptoFactory.h

10.64 gdcm::CryptographicMessageSyntax Class Reference

`#include <gdcmCryptographicMessageSyntax.h>`

Inheritance diagram for gdcm::CryptographicMessageSyntax:



Public Types

- enum CipherTypes {
  DES3_CIPHER ,
  AES128_CIPHER ,
  AES192_CIPHER ,
  AES256_CIPHER }

Public Member Functions

- CryptographicMessageSyntax ()=default
- CryptographicMessageSyntax (const CryptographicMessageSyntax &)=delete
- virtual ~CryptographicMessageSyntax ()=default
- virtual bool Decrypt (char *output, size_t &outlen, const char *array, size_t len) const =0
  decrypt content from a CMS envelopedData structure
- virtual bool Encrypt (char *output, size_t &outlen, const char *array, size_t len) const =0
  create a CMS envelopedData structure
- virtual CipherTypes GetCipherType () const =0
- void operator= (const CryptographicMessageSyntax &)=delete
- virtual bool ParseCertificateFile (const char *filename)=0
- virtual bool ParseKeyFile (const char *filename)=0
- virtual void SetCipherType (CipherTypes type)=0
- virtual bool SetPassword (const char *pass, size_t passLen)=0

10.64.1 Detailed Description

Examples

BasicAnonymizer.cs, and ClinicalTrialIdentificationWorkflow.cs.

10.64.2 Member Enumeration Documentation

10.64.2.1 CipherTypes

```
enum gdcm::CryptographicMessageSyntax::CipherTypes
```

Enumerator

DES3_CIPHER	
AES128_CIPHER	
AES192_CIPHER	
AES256_CIPHER	

10.64.3 Constructor & Destructor Documentation

10.64.3.1 CryptographicMessageSyntax() [1/2]

```
gdcm::CryptographicMessageSyntax::CryptographicMessageSyntax ( )  [default]
```

10.64.3.2 ~CryptographicMessageSyntax()

```
virtual gdcm::CryptographicMessageSyntax::~CryptographicMessageSyntax ( )  [virtual], [default]
```

10.64.3.3 CryptographicMessageSyntax() [2/2]

```
gdcm::CryptographicMessageSyntax::CryptographicMessageSyntax (
            const CryptographicMessageSyntax &  )  [delete]
```

10.64.4 Member Function Documentation

10.64.4.1 Decrypt()

```
virtual bool gdcm::CryptographicMessageSyntax::Decrypt (
            char * output,
            size_t & outlen,
            const char * array,
            size_t len ) const  [pure virtual]
```

decrypt content from a CMS envelopedData structure

Implemented in gdcm::CAPICryptographicMessageSyntax, gdcm::OpenSSLCryptographicMessageSyntax, and gdcm::OpenSSLP7CryptographicMessageSyntax.

10.64.4.2 Encrypt()

```
virtual bool gdcm::CryptographicMessageSyntax::Encrypt (
            char * output,
            size_t & outlen,
            const char * array,
            size_t len ) const  [pure virtual]
```

create a CMS envelopedData structure

Implemented in gdcm::CAPICryptographicMessageSyntax, gdcm::OpenSSLCryptographicMessageSyntax, and gdcm::OpenSSLP7CryptographicMessageSyntax.

10.64.4.3 GetCipherType()

```
virtual CipherTypes gdcm::CryptographicMessageSyntax::GetCipherType ( ) const  [pure virtual]
```

Implemented in gdcm::CAPICryptographicMessageSyntax, gdcm::OpenSSLCryptographicMessageSyntax, and gdcm::OpenSSLP7CryptographicMessageSyntax.

10.64.4.4 operator=()

```
void gdcm::CryptographicMessageSyntax::operator= (
            const CryptographicMessageSyntax &  )  [delete]
```

10.64.4.5 ParseCertificateFile()

```
virtual bool gdcm::CryptographicMessageSyntax::ParseCertificateFile (
            const char * filename )  [pure virtual]
```

Implemented in gdcm::CAPICryptographicMessageSyntax, gdcm::OpenSSLCryptographicMessageSyntax, and gdcm::OpenSSLP7CryptographicMessageSyntax.

Examples

BasicAnonymizer.cs, and ClinicalTrialIdentificationWorkflow.cs.

10.64.4.6 ParseKeyFile()

```
virtual bool gdcm::CryptographicMessageSyntax::ParseKeyFile (
            const char * filename )  [pure virtual]
```

Implemented in gdcm::CAPICryptographicMessageSyntax, gdcm::OpenSSLCryptographicMessageSyntax, and gdcm::OpenSSLP7CryptographicMessageSyntax.

10.64.4.7 SetCipherType()

```
virtual void gdcm::CryptographicMessageSyntax::SetCipherType (
            CipherTypes type )  [pure virtual]
```

Implemented in gdcm::CAPICryptographicMessageSyntax, gdcm::OpenSSLCryptographicMessageSyntax, and gdcm::OpenSSLP7CryptographicMessageSyntax.

10.64.4.8 SetPassword()

```
virtual bool gdcm::CryptographicMessageSyntax::SetPassword (
            const char * pass,
            size_t passLen )  [pure virtual]
```

Implemented in gdcm::OpenSSLP7CryptographicMessageSyntax, gdcm::CAPICryptographicMessageSyntax, and gdcm::OpenSSLCryptographicMessageSyntax.

The documentation for this class was generated from the following file:

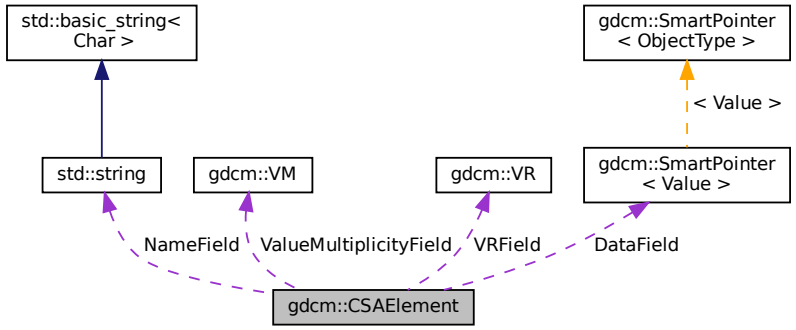
- gdcmCryptographicMessageSyntax.h

10.65 gdcm::CSAElement Class Reference

Class to represent a CSA Element.

`#include <gdcmCSAElement.h>`

Collaboration diagram for gdcm::CSAElement:



Public Member Functions

- CSAElement (const CSAElement &_val)
- CSAElement (unsigned int kf=0)
- const ByteValue * GetByteValue () const
- unsigned int GetKey () const

  Set/Get Key.
- const char * GetName () const

  Set/Get Name.
- unsigned int GetNoOfItems () const

  Set/Get NoOfItems.
- unsigned int GetSyngoDT () const

  Set/Get SyngoDT.
- Value & GetValue ()
- Value const & GetValue () const

  Set/Get Value (bytes array, SQ of items, SQ of fragments):
- const VM & GetVM () const

  Set/Get VM.
- VR const & GetVR () const

  Set/Get VR.
- bool IsEmpty () const

  Check if CSA Element is empty.
- bool operator< (const CSAElement &de) const
- CSAElement & operator= (const CSAElement &de)=default

- bool operator== (const CSAElement &de) const
- void SetByteValue (const char *array, VL length)

  Set.
- void SetKey (unsigned int key)
- void SetName (const char *name)
- void SetNoOfItems (unsigned int items)
- void SetSyngoDT (unsigned int syngodt)
- void SetValue (Value const &vl)
- void SetVM (const VM &vm)
- void SetVR (VR const &vr)

Protected Types

- typedef SmartPointer< Value > DataPtr

Protected Attributes

- DataPtr DataField
- unsigned int KeyField
- std::string NameField
- unsigned int NoOfItemsField
- unsigned int SyngoDTField
- VM ValueMultiplicityField
- VR VRField

Friends

- std::ostream & operator<< (std::ostream &os, const CSAElement &val)

10.65.1 Detailed Description

Class to represent a CSA Element.

See also

CSAHeader

Examples

DumpCSA.cs, DumpSiemensBase64.cxx, MrProtocol.cxx, and csa2img.cxx.

10.65.2 Member Typedef Documentation

10.65.2.1 DataPtr

```
typedef SmartPointer<Value> gdcm::CSAElement::DataPtr  [protected]
```

10.65.3 Constructor & Destructor Documentation

10.65.3.1 CSAElement() [1/2]

```
gdcm::CSAElement::CSAElement (
            unsigned int kf = 0 )  [inline]
```

10.65.3.2 CSAElement() [2/2]

```
gdcm::CSAElement::CSAElement (
            const CSAElement & _val )  [inline]
```

10.65.4 Member Function Documentation

10.65.4.1 GetByteValue()

```
const ByteValue * gdcm::CSAElement::GetByteValue ( ) const  [inline]
```

Return the Value of CSAElement as a ByteValue (if possible)

Warning

: You need to check for NULL return value

Examples

DumpSiemensBase64.cxx, and MrProtocol.cxx.

10.65.4.2 GetKey()

```
unsigned int gdcm::CSAElement::GetKey ( ) const  [inline]
```

Set/Get Key.

Referenced by operator<().

10.65.4.3 GetName()

```
const char * gdcm::CSAElement::GetName ( ) const  [inline]
```

Set/Get Name.

10.65.4.4 GetNoOfItems()

```
unsigned int gdcm::CSAElement::GetNoOfItems ( ) const  [inline]
```

Set/Get NoOfItems.

10.65.4.5 GetSyngoDT()

```
unsigned int gdcm::CSAElement::GetSyngoDT ( ) const  [inline]
```

Set/Get SyngoDT.

10.65.4.6 GetValue() [1/2]

```
Value & gdcm::CSAElement::GetValue ( )  [inline]
```

10.65.4.7 GetValue() [2/2]

```
Value const & gdcm::CSAElement::GetValue ( ) const  [inline]
```

Set/Get Value (bytes array, SQ of items, SQ of fragments):

Examples

csa2img.cxx.

10.65.4.8 GetVM()

```
const VM & gdcm::CSAElement::GetVM ( ) const  [inline]
```

Set/Get VM.

10.65.4.9 GetVR()

```
VR const & gdcm::CSAElement::GetVR ( ) const  [inline]
```

Set/Get VR.

10.65.4.10 IsEmpty()

```
bool gdcm::CSAElement::IsEmpty ( ) const  [inline]
```

Check if CSA Element is empty.

Examples

csa2img.cxx.

10.65.4.11 operator<()

```
bool gdcm::CSAElement::operator< (
            const CSAElement & de ) const  [inline]
```

References GetKey().

10.65.4.12 operator=()

```
CSAElement & gdcm::CSAElement::operator= (
            const CSAElement & de )  [default]
```

10.65.4.13 operator==()

```
bool gdcm::CSAElement::operator== (
            const CSAElement & de ) const  [inline]
```

References KeyField, NameField, SyngoDTField, ValueMultiplicityField, and VRField.

10.65.4.14 SetByteValue()

```
void gdcm::CSAElement::SetByteValue (
            const char * array,
            VL length )  [inline]
```

Set.

10.65.4.15 SetKey()

```
void gdcm::CSAElement::SetKey (
            unsigned int key )  [inline]
```

10.65.4.16 SetName()

```
void gdcm::CSAElement::SetName (
            const char * name )  [inline]
```

10.65.4.17 SetNoOfItems()

```
void gdcm::CSAElement::SetNoOfItems (
            unsigned int items )  [inline]
```

10.65.4.18 SetSyngoDT()

```
void gdcm::CSAElement::SetSyngoDT (
            unsigned int syngodt )  [inline]
```

10.65.4.19 SetValue()

```
void gdcm::CSAElement::SetValue (
            Value const & vl )  [inline]
```

10.65.4.20 SetVM()

```
void gdcm::CSAElement::SetVM (
            const VM & vm )  [inline]
```

10.65.4.21 SetVR()

```
void gdcm::CSAElement::SetVR (
            VR const & vr )  [inline]
```

10.65.5 Friends And Related Function Documentation

10.65.5.1 operator<<

```
std::ostream & operator<< (
            std::ostream & os,
            const CSAElement & val )  [friend]
```

10.65.6 Member Data Documentation

10.65.6.1 DataField

```
DataPtr gdcm::CSAElement::DataField  [protected]
```

10.65.6.2 KeyField

```
unsigned int gdcm::CSAElement::KeyField  [protected]
```

Referenced by operator==().

10.65.6.3 NameField

```
std::string gdcm::CSAElement::NameField  [protected]
```

Referenced by operator==().

10.65.6.4 NoOfItemsField

```
unsigned int gdcm::CSAElement::NoOfItemsField  [protected]
```

10.65.6.5 SyngoDTField

```
unsigned int gdcm::CSAElement::SyngoDTField  [protected]
```

Referenced by operator==().

10.65.6.6 ValueMultiplicityField

```
VM gdcm::CSAElement::ValueMultiplicityField  [protected]
```

Referenced by operator==().

10.65.6.7 VRField

```
VR gdcm::CSAElement::VRField  [protected]
```

Referenced by operator==().

The documentation for this class was generated from the following file:

- gdcmCSAElement.h

10.66 gdcm::CSAHeader Class Reference

Class for CSAHeader.

```
#include <gdcmCSAHeader.h>
```

Public Types

- enum CSAHeaderType {
  UNKNOWN = 0 ,
  SV10 ,
  NOMAGIC ,
  DATASET_FORMAT ,
  INTERFILE ,
  ZEROED_OUT }

  Divers format of CSAHeader as found 'in the wild'.

Public Member Functions

- CSAHeader ()
- ~CSAHeader ()=default
- bool FindCSAElementByName (const char *name)
- const CSAElement & GetCSAElementByName (const char *name)
- const DataSet & GetDataSet () const

  Return the DataSet output (use only if Format == DATASET_FORMAT)
- CSAHeaderType GetFormat () const
- const char * GetInterfile () const

  Return the string output (use only if Format == Interfile)
- bool GetMrProtocol (const DataSet &ds, MrProtocol &mrProtocol)

  Retrieve the ASCII portion stored within the MrProtocol/MrPhoenixProtocol:
- bool LoadFromDataElement (DataElement const &de)

  Decode the CSAHeader from element 'de'.
- void Print (std::ostream &os) const

  Print the CSAHeader (use only if Format == SV10 or NOMAGIC)

Static Public Member Functions

- static const PrivateTag & GetCSADataInfo ()
- static const PrivateTag & GetCSAImageHeaderInfoTag ()
- static const PrivateTag & GetCSASeriesHeaderInfoTag ()

Protected Member Functions

- const CSAElement & GetCSAEEnd () const

Friends

- std::ostream & operator<< (std::ostream &_os, const CSAHeader &d)

10.66.1 Detailed Description

Class for CSAHeader.

SIEMENS store private information in tag (0x0029,0x10,"SIEMENS CSA HEADER") this class is meant for user wishing to access values stored within this private attribute. There are basically two main 'format' for this attribute : SV10/↩ NOMAGIC and DATASET_FORMAT SV10 and NOMAGIC are from a user prospective identical, see CSAHeader.xml for possible name / value stored in this format. DATASET_FORMAT is in fact simply just another DICOM dataset (implicit) with -currently unknown- value. This can be only be printed for now.

Warning

Everything you do with this code is at your own risk, since decoding process was not written from specification documents.

the API of this class might change.

Todo MrEvaProtocol in 29,1020 contains ^M that would be nice to get rid of on UNIX system...

See also

PDBHeader

External references: 5.1.3.2.4.1 MEDCOM History Information and 5.1.4.3 CSA Non-Image Module in `http↩ ://tamsinfo.toshiba.com/docrequest/pdf/E.Soft_v2.0.pdf`

Examples

DumpCSA.cs, DumpSiemensBase64.cxx, MrProtocol.cxx, and csa2img.cxx.

10.66.2 Member Enumeration Documentation

10.66.2.1 CSAHeaderType

`enum gdcm::CSAHeader::CSAHeaderType`

Divers format of CSAHeader as found 'in the wild'.

Enumerator

UNKNOWN	
SV10	
NOMAGIC	
DATASET_FORMAT	
INTERFILE	
ZEROED_OUT	

10.66.3 Constructor & Destructor Documentation

10.66.3.1 CSAHeader()

```
gdcm::CSAHeader::CSAHeader ( )  [inline]
```

10.66.3.2 ~CSAHeader()

```
gdcm::CSAHeader::~CSAHeader ( )  [default]
```

10.66.4 Member Function Documentation

10.66.4.1 FindCSAElementByName()

```
bool gdcm::CSAHeader::FindCSAElementByName (
            const char * name )
```

Return true if the CSA element matching 'name' is found or not

Warning

Case Sensitive

Examples

DumpCSA.cs, DumpSiemensBase64.cxx, MrProtocol.cxx, and csa2img.cxx.

10.66.4.2 GetCSADataInfo()

```
static const PrivateTag & gdcm::CSAHeader::GetCSADataInfo ( )  [static]
```

Return the private tag used by SIEMENS to store the CSA Data Info This is: PrivateTag(0x0029,0x0010,"SIEMENS CSA NON-IMAGE");

10.66.4.3 GetCSAEEnd()

```
const CSAElement & gdcm::CSAHeader::GetCSAEEnd ( ) const  [protected]
```

10.66.4.4 GetCSAElementByName()

```
const CSAElement & gdcm::CSAHeader::GetCSAElementByName (
            const char * name )
```

Return the CSAElement corresponding to name 'name'

Warning

Case Sensitive

Examples

DumpCSA.cs, DumpSiemensBase64.cxx, MrProtocol.cxx, and csa2img.cxx.

10.66.4.5 GetCSAImageHeaderInfoTag()

```
static const PrivateTag & gdcm::CSAHeader::GetCSAImageHeaderInfoTag ( )  [static]
```

Return the private tag used by SIEMENS to store the CSA Image Header This is: PrivateTag(0x0029,0x0010,"SIEMENS CSA HEADER");

Examples

DumpCSA.cs, DumpSiemensBase64.cxx, PublicDict.cxx, and csa2img.cxx.

10.66.4.6 GetCSASeriesHeaderInfoTag()

```
static const PrivateTag & gdcm::CSAHeader::GetCSASeriesHeaderInfoTag ( )  [static]
```

Return the private tag used by SIEMENS to store the CSA Series Header This is: PrivateTag(0x0029,0x0020,"SIEMENS CSA HEADER");

Examples

MrProtocol.cxx.

10.66.4.7 GetDataSet()

```
const DataSet & gdcm::CSAHeader::GetDataSet ( ) const  [inline]
```

Return the DataSet output (use only if Format == DATASET_FORMAT)

10.66.4.8 GetFormat()

```
CSAHeaderType gdcm::CSAHeader::GetFormat ( ) const
```

return the format of the CSAHeader SV10 and NOMAGIC are equivalent.

10.66.4.9 GetInterfile()

```
const char * gdcm::CSAHeader::GetInterfile ( ) const  [inline]
```

Return the string output (use only if Format == Interfile)

10.66.4.10 GetMrProtocol()

```
bool gdcm::CSAHeader::GetMrProtocol (
            const DataSet & ds,
            MrProtocol & mrProtocol )
```

Retrieve the ASCII portion stored within the MrProtocol/MrPhoenixProtocol:

Examples

MrProtocol.cxx.

10.66.4.11 LoadFromDataElement()

```
bool gdcm::CSAHeader::LoadFromDataElement (
            DataElement const & de )
```

Decode the CSAHeader from element 'de'.

Examples

DumpCSA.cs, DumpSiemensBase64.cxx, MrProtocol.cxx, and csa2img.cxx.

10.66.4.12 Print()

```
void gdcm::CSAHeader::Print (
            std::ostream & os ) const
```

Print the CSAHeader (use only if Format == SV10 or NOMAGIC)

Examples

csa2img.cxx.

10.66.5 Friends And Related Function Documentation

10.66.5.1 operator<<

```
std::ostream & operator<< (
            std::ostream & _os,
            const CSAHeader & d )  [friend]
```

The documentation for this class was generated from the following file:

- gdcmCSAHeader.h

10.67 gdcm::CSAHeaderDict Class Reference

Class to represent a map of CSAHeaderDictEntry.

```
#include <gdcmCSAHeaderDict.h>
```

Public Types

- typedef MapCSAHeaderDictEntry::const_iterator ConstIterator
- typedef MapCSAHeaderDictEntry::iterator Iterator
- typedef std::set< CSAHeaderDictEntry > MapCSAHeaderDictEntry

Public Member Functions

- CSAHeaderDict ()
- CSAHeaderDict (const CSAHeaderDict &_val)=delete
- void AddCSAHeaderDictEntry (const CSAHeaderDictEntry &de)
- ConstIterator Begin () const
- ConstIterator End () const
- const CSAHeaderDictEntry & GetCSAHeaderDictEntry (const char *name) const
- bool IsEmpty () const
- CSAHeaderDict & operator= (const CSAHeaderDict &_val)=delete

Protected Member Functions

- void LoadDefault ()

Friends

- class Dicts
- std::ostream & operator<< (std::ostream &_os, const CSAHeaderDict &_val)

10.67.1 Detailed Description

Class to represent a map of CSAHeaderDictEntry.

Examples

MrProtocol.cxx.

10.67.2 Member Typedef Documentation

10.67.2.1 ConstIterator

```
typedef MapCSAHeaderDictEntry::const_iterator gdcm::CSAHeaderDict::ConstIterator
```

10.67.2.2 Iterator

```
typedef MapCSAHeaderDictEntry::iterator gdcm::CSAHeaderDict::Iterator
```

10.67.2.3 MapCSAHeaderDictEntry

```
typedef std::set<CSAHeaderDictEntry> gdcm::CSAHeaderDict::MapCSAHeaderDictEntry
```

10.67.3 Constructor & Destructor Documentation

10.67.3.1 CSAHeaderDict() [1/2]

```
gdcm::CSAHeaderDict::CSAHeaderDict ( )  [inline]
```

10.67.3.2 CSAHeaderDict() [2/2]

```
gdcm::CSAHeaderDict::CSAHeaderDict (
            const CSAHeaderDict & _val )  [delete]
```

10.67.4 Member Function Documentation

10.67.4.1 AddCSAHeaderDictEntry()

```
void gdcm::CSAHeaderDict::AddCSAHeaderDictEntry (
            const CSAHeaderDictEntry & de )  [inline]
```

10.67.4.2 Begin()

```
ConstIterator gdcm::CSAHeaderDict::Begin ( ) const  [inline]
```

10.67.4.3 End()

```
ConstIterator gdcm::CSAHeaderDict::End ( ) const  [inline]
```

10.67.4.4 GetCSAHeaderDictEntry()

```
const CSAHeaderDictEntry & gdcm::CSAHeaderDict::GetCSAHeaderDictEntry (
            const char * name ) const  [inline]
```

Examples

MrProtocol.cxx.

10.67.4.5 IsEmpty()

```
bool gdcm::CSAHeaderDict::IsEmpty ( ) const  [inline]
```

10.67.4.6 LoadDefault()

```
void gdcm::CSAHeaderDict::LoadDefault ( )  [protected]
```

10.67.4.7 operator=()

```
CSAHeaderDict & gdcm::CSAHeaderDict::operator= (
            const CSAHeaderDict & _val )  [delete]
```

10.67.5 Friends And Related Function Documentation

10.67.5.1 Dicts

```
friend class Dicts  [friend]
```

10.67.5.2 operator<<

```
std::ostream & operator<< (
            std::ostream & _os,
            const CSAHeaderDict & _val )  [friend]
```

The documentation for this class was generated from the following file:

- gdcmCSAHeaderDict.h

10.68 gdcm::CSAHeaderDictEntry Class Reference

Class to represent an Entry in the Dict.

```
#include <gdcmCSAHeaderDictEntry.h>
```

Public Member Functions

- CSAHeaderDictEntry (const char *name="", VR const &vr=VR::INVALID, VM const &vm=VM::VM0, const char *desc="")
- const char * GetDescription () const

  Set/Get Description.
- const char * GetName () const

  Set/Get Name.
- const VM & GetVM () const

  Set/Get VM.
- const VR & GetVR () const

  Set/Get VR.
- bool operator< (const CSAHeaderDictEntry &entry) const
- void SetDescription (const char *desc)
- void SetName (const char *name)
- void SetVM (VM const &vm)
- void SetVR (const VR &vr)

Friends

- std::ostream & operator<< (std::ostream &_os, const CSAHeaderDictEntry &_val)

10.68.1 Detailed Description

Class to represent an Entry in the Dict.

Does not really exist within the DICOM definition, just a way to minimize storage and have a mapping from gdcm::Tag to the needed information

Note

bla TODO FIXME: Need a PublicCSAHeaderDictEntry...indeed CSAHeaderDictEntry has a notion of retired which does not exist in PrivateCSAHeaderDictEntry...

See also

gdcm::Dict

Examples

MrProtocol.cxx.

10.68.2 Constructor & Destructor Documentation

10.68.2.1 CSAHeaderDictEntry()

```
gdcm::CSAHeaderDictEntry::CSAHeaderDictEntry (
            const char * name = "",
            VR const & vr = VR::INVALID,
            VM const & vm = VM::VM0,
            const char * desc = "" )  [inline]
```

10.68.3 Member Function Documentation

10.68.3.1 GetDescription()

```
const char * gdcm::CSAHeaderDictEntry::GetDescription ( ) const  [inline]
```

Set/Get Description.

10.68.3.2 GetName()

```
const char * gdcm::CSAHeaderDictEntry::GetName ( ) const  [inline]
```

Set/Get Name.

Referenced by operator<().

10.68.3.3 GetVM()

```
const VM & gdcm::CSAHeaderDictEntry::GetVM ( ) const  [inline]
```

Set/Get VM.

10.68.3.4 GetVR()

```
const VR & gdcm::CSAHeaderDictEntry::GetVR ( ) const  [inline]
```

Set/Get VR.

10.68.3.5 operator<()

```
bool gdcm::CSAHeaderDictEntry::operator< (
            const CSAHeaderDictEntry & entry ) const  [inline]
```

References GetName().

10.68.3.6 SetDescription()

```
void gdcm::CSAHeaderDictEntry::SetDescription (
            const char * desc )  [inline]
```

10.68.3.7 SetName()

```
void gdcm::CSAHeaderDictEntry::SetName (
            const char * name )  [inline]
```

10.68.3.8 SetVM()

```
void gdcm::CSAHeaderDictEntry::SetVM (
            VM const & vm )  [inline]
```

10.68.3.9 SetVR()

```
void gdcm::CSAHeaderDictEntry::SetVR (
            const VR & vr )  [inline]
```

10.68.4 Friends And Related Function Documentation

10.68.4.1 operator<<

```
std::ostream & operator<< (
            std::ostream & _os,
            const CSAHeaderDictEntry & _val )  [friend]
```

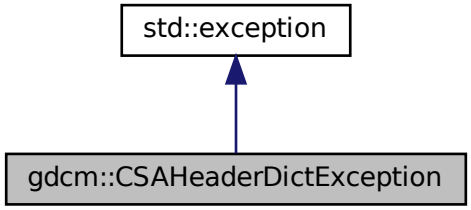
The documentation for this class was generated from the following file:

- gdcmCSAHeaderDictEntry.h

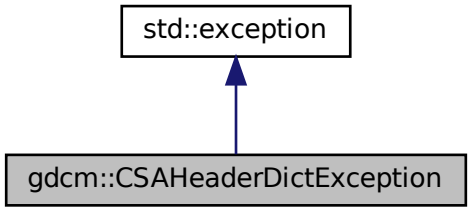
10.69 gdcm::CSAHeaderDictException Class Reference

`#include <gdcmCSAHeaderDict.h>`

Inheritance diagram for gdcm::CSAHeaderDictException:



Collaboration diagram for gdcm::CSAHeaderDictException:



The documentation for this class was generated from the following file:

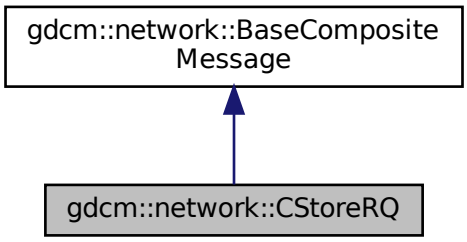
- gdcmCSAHeaderDict.h

10.70 gdcm::network::CStoreRQ Class Reference

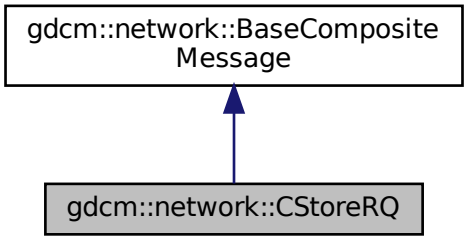
CStoreRQ.

`#include <gdcmCStoreMessages.h>`

Inheritance diagram for gdcm::network::CStoreRQ:



Collaboration diagram for gdcm::network::CStoreRQ:



Public Member Functions

- std::vector< PresentationDataValue > ConstructPDV (const ULConnection &inConnection, const File &file, bool writeDataSet=true)

10.70.1 Detailed Description

CStoreRQ.

this file defines the messages for the cecho action

10.70.2 Member Function Documentation

10.70.2.1 ConstructPDV()

```
std::vector< PresentationDataValue > gdcm::network::CStoreRQ::ConstructPDV (
            const ULConnection & inConnection,
            const File & file,
            bool writeDataSet = true )
```

The documentation for this class was generated from the following file:

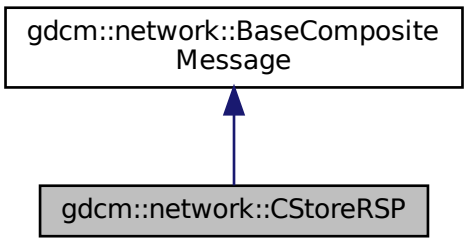
- gdcmCStoreMessages.h

10.71 gdcm::network::CStoreRSP Class Reference

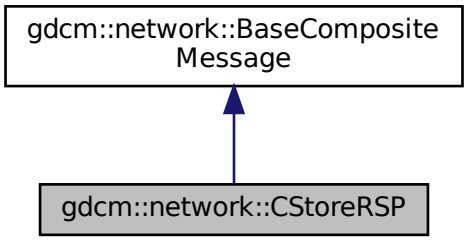
CStoreRSP this file defines the messages for the cecho action.

```
#include <gdcmCStoreMessages.h>
```

Inheritance diagram for gdcm::network::CStoreRSP:



Collaboration diagram for gdcm::network::CStoreRSP:



Public Member Functions

- std::vector< PresentationDataValue > ConstructPDV (const DataSet *inDataSet, const BasePDU *inPC)

10.71.1 Detailed Description

CStoreRSP this file defines the messages for the cecho action.

10.71.2 Member Function Documentation

10.71.2.1 ConstructPDV()

```
std::vector< PresentationDataValue > gdcm::network::CStoreRSP::ConstructPDV (
            const DataSet * inDataSet,
            const BasePDU * inPC )
```

The documentation for this class was generated from the following file:

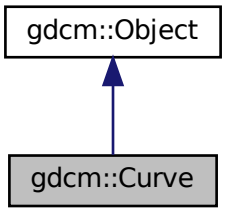
- gdcmCStoreMessages.h

10.72 gdcm::Curve Class Reference

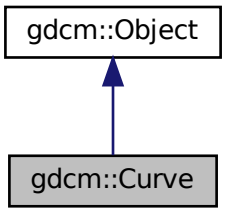
Curve class to handle element 50xx,3000 Curve Data.

```
#include <gdcmCurve.h>
```

Inheritance diagram for gdcm::Curve:



Collaboration diagram for gdcm::Curve:



Public Member Functions

- Curve ()
- Curve (Curve const &ov)
- ~Curve () override
- void Decode (std::istream &is, std::ostream &os)
- void GetAsPoints (float *array) const
- std::vector< unsigned short > const & GetCurveDataDescriptor () const
- unsigned short GetDataValueRepresentation () const
- unsigned short GetDimensions () const
- unsigned short GetGroup () const
- unsigned short GetNumberOfPoints () const
- const char * GetTypeOfData () const
- const char * GetTypeOfDataDescription () const
- bool IsEmpty () const
- void Print (std::ostream &) const override
- void SetCoordinateStartValue (unsigned short v)
- void SetCoordinateStepValue (unsigned short v)
- void SetCurve (const char *array, unsigned int length)
- void SetCurveDataDescriptor (const uint16_t *values, size_t num)
- void SetCurveDescription (const char *curvedescription)
- void SetDataValueRepresentation (unsigned short datavaluerepresentation)
- void SetDimensions (unsigned short dimensions)
- void SetGroup (unsigned short group)
- void SetNumberOfPoints (unsigned short numberofpoints)
- void SetTypeOfData (const char *typeofdata)
- void Update (const DataElement &de)

Static Public Member Functions

- static unsigned int GetNumberOfCurves (DataSet const &ds)

Additional Inherited Members

10.72.1 Detailed Description

Curve class to handle element 50xx,3000 Curve Data.

WARNING: This is deprecated and lastly defined in PS 3.3 - 2004

Examples:

- GE_DLX-8-MONO2-Multiframe-Jpeg_Lossless.dcm
- GE_DLX-8-MONO2-Multiframe.dcm
- gdcmSampleData/Philips_Medical_Images/integris_HV_5000/xa_integris.dcm
- TOSHIBA-CurveData[1-3].dcm

10.72.2 Constructor & Destructor Documentation

10.72.2.1 Curve() [1/2]

```
gdcm::Curve::Curve ( )
```

10.72.2.2 ~Curve()

```
gdcm::Curve::~Curve ( )  [override]
```

10.72.2.3 Curve() [2/2]

```
gdcm::Curve::Curve (
            Curve const & ov )
```

10.72.3 Member Function Documentation

10.72.3.1 Decode()

```
void gdcm::Curve::Decode (
            std::istream & is,
            std::ostream & os )
```

10.72.3.2 GetAsPoints()

```
void gdcm::Curve::GetAsPoints (
            float * array ) const
```

10.72.3.3 GetCurveDataDescriptor()

```
std::vector< unsigned short > const & gdcm::Curve::GetCurveDataDescriptor ( ) const
```

10.72.3.4 GetDataValueRepresentation()

```
unsigned short gdcm::Curve::GetDataValueRepresentation ( ) const
```

10.72.3.5 GetDimensions()

```
unsigned short gdcm::Curve::GetDimensions ( ) const
```

10.72.3.6 GetGroup()

```
unsigned short gdcm::Curve::GetGroup ( ) const
```

10.72.3.7 GetNumberOfCurves()

```
static unsigned int gdcm::Curve::GetNumberOfCurves (
            DataSet const & ds )  [static]
```

10.72.3.8 GetNumberOfPoints()

```
unsigned short gdcm::Curve::GetNumberOfPoints ( ) const
```

10.72.3.9 GetTypeOfData()

```
const char * gdcm::Curve::GetTypeOfData ( ) const
```

10.72.3.10 GetTypeOfDataDescription()

```
const char * gdcm::Curve::GetTypeOfDataDescription ( ) const
```

10.72.3.11 IsEmpty()

```
bool gdcm::Curve::IsEmpty ( ) const
```

10.72.3.12 Print()

```
void gdcm::Curve::Print (
            std::ostream &  ) const  [override], [virtual]
```

Reimplemented from gdcm::Object.

10.72.3.13 SetCoordinateStartValue()

```
void gdcm::Curve::SetCoordinateStartValue (
            unsigned short v )
```

10.72.3.14 SetCoordinateStepValue()

```
void gdcm::Curve::SetCoordinateStepValue (
            unsigned short v )
```

10.72.3.15 SetCurve()

```
void gdcm::Curve::SetCurve (
            const char * array,
            unsigned int length )
```

10.72.3.16 SetCurveDataDescriptor()

```
void gdcm::Curve::SetCurveDataDescriptor (
            const uint16_t * values,
            size_t num )
```

10.72.3.17 SetCurveDescription()

```
void gdcm::Curve::SetCurveDescription (
            const char * curvedescription )
```

10.72.3.18 SetDataValueRepresentation()

```
void gdcm::Curve::SetDataValueRepresentation (
            unsigned short datavaluerepresentation )
```

10.72.3.19 SetDimensions()

```
void gdcm::Curve::SetDimensions (
            unsigned short dimensions )
```

10.72.3.20 SetGroup()

```
void gdcm::Curve::SetGroup (
            unsigned short group )
```

10.72.3.21 SetNumberOfPoints()

```
void gdcm::Curve::SetNumberOfPoints (
            unsigned short numberofpoints )
```

10.72.3.22 SetTypeOfData()

```
void gdcm::Curve::SetTypeOfData (
            const char * typeofdata )
```

10.72.3.23 Update()

```
void gdcm::Curve::Update (
            const DataElement & de )
```

The documentation for this class was generated from the following file:

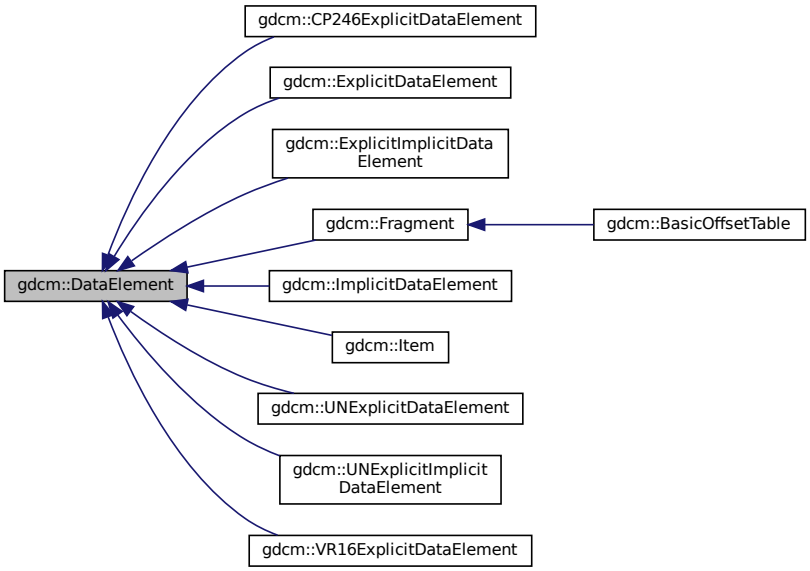
- gdcmCurve.h

10.73 gdcm::DataElement Class Reference

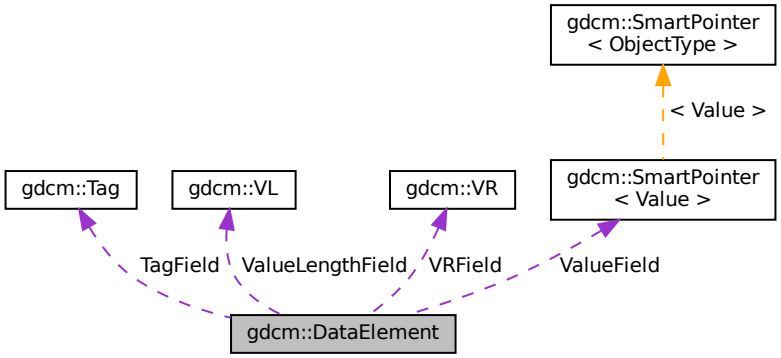
Class to represent a Data Element either Implicit or Explicit.

```
#include <gdcmDataElement.h>
```

Inheritance diagram for gdcm::DataElement:



Collaboration diagram for gdcm::DataElement:



Public Member Functions

- DataElement (const DataElement &_val)
- DataElement (const Tag &t=Tag(0), const VL &vl=0, const VR &vr=VR::INVALID)
- void Clear ()

  Clear Data Element (make Value empty and invalidate Tag & VR)

- void Empty ()

  Make Data Element empty (no Value)
- const ByteValue * GetByteValue () const
- template<typename TDE >
  VL GetLength () const
- SequenceOfFragments * GetSequenceOfFragments ()
- const SequenceOfFragments * GetSequenceOfFragments () const
- Tag & GetTag ()
- const Tag & GetTag () const

  Get Tag.
- Value & GetValue ()
- Value const & GetValue () const

  Set/Get Value (bytes array, SQ of items, SQ of fragments):
- SmartPointer< SequenceOfItems > GetValueAsSQ () const
- VL & GetVL ()
- const VL & GetVL () const

  Get VL.
- VR const & GetVR () const
- bool IsEmpty () const

  Check if Data Element is empty.
- bool IsUndefinedLength () const

  return if Value Length if of undefined length
- bool operator< (const DataElement &de) const
- DataElement & operator= (const DataElement &)=default
- bool operator== (const DataElement &de) const
- template<typename TDE , typename TSwap >
  std::istream & Read (std::istream &is)
- template<typename TDE , typename TSwap >
  std::istream & ReadOrSkip (std::istream &is, std::set< Tag > const &skiptags)
- template<typename TDE , typename TSwap >
  std::istream & ReadPreValue (std::istream &is, std::set< Tag > const &skiptags)
- template<typename TDE , typename TSwap >
  std::istream & ReadValue (std::istream &is, std::set< Tag > const &skiptags)
- template<typename TDE , typename TSwap >
  std::istream & ReadValueWithLength (std::istream &is, VL &length, std::set< Tag > const &skiptags)
- template<typename TDE , typename TSwap >
  std::istream & ReadWithLength (std::istream &is, VL &length)
- void SetByteValue (const char *array, VL length)
- void SetTag (const Tag &t)
- void SetValue (Value const &vl)
- void SetVL (const VL &vl)
- void SetVLToUndefined ()
- void SetVR (VR const &vr)
- template<typename TDE , typename TSwap >
  const std::ostream & Write (std::ostream &os) const

Protected Types

- typedef SmartPointer< Value > ValuePtr

Protected Member Functions

- void SetValueFieldLength (VL vl, bool readvalues)

Protected Attributes

- Tag TagField
- ValuePtr ValueField
- VL ValueLengthField
- VR VRField

Friends

- std::ostream & operator<< (std::ostream &_os, const DataElement &_val)

10.73.1 Detailed Description

Class to represent a Data Element either Implicit or Explicit.

DATA ELEMENT: A unit of information as defined by a single entry in the data dictionary. An encoded Information Object Definition (IOD) Attribute that is composed of, at a minimum, three fields: a Data Element Tag, a Value Length, and a Value Field. For some specific Transfer Syntaxes, a Data Element also contains a VR Field where the Value Representation of that Data Element is specified explicitly.

Design:

- A DataElement in GDCM always store VL (Value Length) on a 32 bits integer even when VL is 16 bits
- A DataElement always store the VR even for Implicit TS, in which case VR is defaulted to VR::INVALID
- For Item start/end (See 0xfffe tags), Value is NULL

See also

ExplicitDataElement ImplicitDataElement

Examples

BasicImageAnonymizer.cs, ChangePrivateTags.cxx, ChangeSequenceUltrasound.cxx, CreateARGBImage.cxx, CreateCMYKImage.cxx, DecompressImage.cs, DecompressImageMultiframe.cs, DecompressJPEGFile.cs, DiffFile.cxx, DumpADAC.cxx, DumpCSA.cs, DumpExamCard.cxx, DumpGEMSMovieGroup.cxx, DumpImageHeaderInfo.cxx, DumpPhilipsECHO.cxx, DumpSiemensBase64.cxx, DumpToshibaDTI.cxx, DumpToshibaDTI2.cxx, DumpVisusChange.cxx, DuplicatePCDE.cxx, ELSCINT1WaveToText.cxx, ExtractEncapsulatedFile.cs, ExtractEncryptedContent.cxx, ExtractIconFromFile.cxx, Extracting_All_Resolution.cxx, Fake_Image_Using_Stream_Image_Writer.cxx, FileChangeTS.cs, FileChangeTSLossy.cs, FixBrokenJ2K.cxx, FixJAIBugJPEGLS.cxx, GenAllVR.cxx, GenFakeIdentifyFile.cxx, GenFakeImage.cxx, GenLongSeqs.cxx, GenSeqs.cxx, GetJPEGSamplePrecision.cxx, GetSequenceUltrasound.cxx, GetSubSequenceData.cxx, LargeVRDSExplicit.cxx, MpegVideoInfo.cs, NewSequence.cs, ReadAndDumpDICOMDIR.cxx, ReadExplicitLengthSQIVR.cxx, ReadGEMSSDO.cxx, SimplePrint.cs, StreamImageReaderTest.cxx, csa2img.cxx, gdcmrtionplan.cxx, gdcmrtplan.cxx, iU22tomultisc.cxx, pmsct_rgb1.cxx, and rle2img.cxx.

10.73.2 Member Typedef Documentation

10.73.2.1 ValuePtr

```
typedef SmartPointer<Value> gdcm::DataElement::ValuePtr  [protected]
```

10.73.3 Constructor & Destructor Documentation

10.73.3.1 DataElement() [1/2]

```
gdcm::DataElement::DataElement (
            const Tag & t = Tag(0),
            const VL & vl = 0,
            const VR & vr = VR::INVALID )  [inline]
```

10.73.3.2 DataElement() [2/2]

```
gdcm::DataElement::DataElement (
            const DataElement & _val )  [inline]
```

10.73.4 Member Function Documentation

10.73.4.1 Clear()

```
void gdcm::DataElement::Clear ( )  [inline]
```

Clear Data Element (make Value empty and invalidate Tag & VR)

10.73.4.2 Empty()

```
void gdcm::DataElement::Empty ( )  [inline]
```

Make Data Element empty (no Value)

10.73.4.3 GetByteValue()

```
const ByteValue * gdcm::DataElement::GetByteValue ( ) const  [inline]
```

Return the Value of DataElement as a ByteValue (if possible)

Warning

: You need to check for NULL return value

Examples

DumpADAC.cxx, DumpExamCard.cxx, DumpGEMSMovieGroup.cxx, DumpImageHeaderInfo.cxx, DumpPhilipsECHO.cxx, DumpToshibaDTI.cxx, DumpToshibaDTI2.cxx, DuplicatePCDE.cxx, ELSCINT1WaveToText.cxx, ExtractEncapsulatedFile.cs, ExtractEncryptedContent.cxx, ExtractIconFromFile.cxx, FixBrokenJ2K.cxx, FixJAIBugJPEGLS.cxx, GetSubSequenceData.cxx, PatchFile.cxx, ReadExplicitLengthSQIVR.cxx, ReadGEMSSDO.cxx, pmsct_rgb1.cxx, and rle2img.cxx.

Referenced by gdcm::Attribute< Group, Element, TVR, TVM >::SetFromDataElement(), gdcm::Attribute< Group, Element, TVR, VM::VM
gdcm::Attribute< Group, Element, TVR, VM::VM1_n >::SetFromDataElement(), gdcm::Element< TVR, TVM >::SetFromDataElement(),
and gdcm::Element< TVR, VM::VM1_n >::SetFromDataElement().

10.73.4.4 GetLength()

```
template<typename TDE >
VL gdcm::DataElement::GetLength ( ) const  [inline]
```

10.73.4.5 GetSequenceOfFragments() [1/2]

```
SequenceOfFragments * gdcm::DataElement::GetSequenceOfFragments ( )
```

10.73.4.6 GetSequenceOfFragments() [2/2]

```
const SequenceOfFragments * gdcm::DataElement::GetSequenceOfFragments ( ) const
```

Return the Value of DataElement as a Sequence Of Fragments (if possible)

Warning

: You need to check for NULL return value

Examples

DecompressImage.cs, FixBrokenJ2K.cxx, FixJAIBugJPEGLS.cxx, and GetJPEGSamplePrecision.cxx.

10.73.4.7 GetTag() [1/2]

```
Tag & gdcm::DataElement::GetTag ( )  [inline]
```

10.73.4.8 GetTag() [2/2]

```
const Tag & gdcm::DataElement::GetTag ( ) const  [inline]
```

Get Tag.

Examples

DumpGEMSMovieGroup.cxx, DumpVisusChange.cxx, DuplicatePCDE.cxx, SimplePrint.cs, and pmsct_rgb1.cxx.

Referenced by gdcm::DataSet::Insert(), gdcm::FileMetaInformation::Insert(), gdcm::CommandDataSet::Insert(), operator<(), gdcm::SequenceOfItems::Read(), gdcm::SequenceOfFragments::ReadValue(), gdcm::FileMetaInformation::Replace(), gdcm::CommandDataSet::Replace(), gdcm::Attribute< Group, Element, TVR, TVM >::SetFromDataElement(), gdcm::Attribute< Group, Element, TVR, VM::VM1 >::SetFromDataElement(), and gdcm::Attribute< Group, Element, TVR, VM::VM1_n >

10.73.4.9 GetValue() [1/2]

```
Value & gdcm::DataElement::GetValue ( )  [inline]
```

References gdcmAssertAlwaysMacro.

10.73.4.10 GetValue() [2/2]

```
Value const & gdcm::DataElement::GetValue ( ) const  [inline]
```

Set/Get Value (bytes array, SQ of items, SQ of fragments):

Examples

ReadAndDumpDICOMDIR.cxx.

References gdcmAssertAlwaysMacro.

Referenced by gdcm::DataSet::InsertDataElement(), gdcm::Element< TVR, TVM >::SetFromDataElement(), and gdcm::Element< TVR, VM::VM1_n >::SetFromDataElement().

10.73.4.11 GetValueAsSQ()

```
SmartPointer< SequenceOfItems > gdcm::DataElement::GetValueAsSQ ( ) const
```

Interpret the Value stored in the DataElement. This is more robust (but also more expensive) to call this function rather than the simplest form: GetSequenceOfItems() It also return NULL when the Value is NOT of type SequenceOfItems

Warning

in case GetSequenceOfItems() succeed the function return this value, otherwise it creates a new SequenceOfItems, you should handle that in your case, for instance: SmartPointer<SequenceOfItems> sqi = de.GetValueAsSQ();

Examples

ChangeSequenceUltrasound.cxx, DumpExamCard.cxx, DumpGEMSMovieGroup.cxx, DumpPhilipsECHO.cxx, DumpToshibaDTI.cxx, DumpToshibaDTI2.cxx, DumpVisusChange.cxx, ExtractEncryptedContent.cxx, GetSequenceUltrasound.cxx, GetSubSequenceData.cxx, LargeVRDSExplicit.cxx, ReadAndDumpDICOMDIR.cxx, SimplePrint.cs, gdcmrtionplan.cxx, and gdcmrtplan.cxx.

10.73.4.12 GetVL() [1/2]

```
VL & gdcm::DataElement::GetVL ( )  [inline]
```

10.73.4.13 GetVL() [2/2]

```
const VL & gdcm::DataElement::GetVL ( ) const  [inline]
```

Get VL.

Examples

SimplePrint.cs.

Referenced by gdcm::DataSet::InsertDataElement(), gdcm::SequenceOfItems::Read(), and gdcm::SequenceOfFragments::ReadValue().

10.73.4.14 GetVR()

```
VR const & gdcm::DataElement::GetVR ( ) const  [inline]
```

Get VR do not set VR::SQ on bytevalue data element

Examples

DuplicatePCDE.cxx, and GenFakeIdentifyFile.cxx.

Referenced by gdcm::Attribute< Group, Element, TVR, TVM >::GetAsDataElement(), gdcm::Attribute< Group, Element, TVR, VM::VM1
gdcm::Attribute< Group, Element, TVR, VM::VM1_n >::GetAsDataElement(), gdcm::Element< TVR, TVM >::GetAsDataElement(),
gdcm::Element< TVR, VM::VM1_n >::GetAsDataElement(), gdcm::Attribute< Group, Element, TVR, TVM >::SetFromDataElement(),
gdcm::Attribute< Group, Element, TVR, VM::VM1 >::SetFromDataElement(), gdcm::Attribute< Group, Element, TVR, VM::VM1_n >::Se
gdcm::Element< TVR, TVM >::SetFromDataElement(), and gdcm::Element< TVR, VM::VM1_n >::SetFromDataElement().

10.73.4.15 IsEmpty()

```
bool gdcm::DataElement::IsEmpty ( ) const  [inline]
```

Check if Data Element is empty.

Examples

DumpADAC.cxx, DumpCSA.cs, DumpExamCard.cxx, DumpGEMSMovieGroup.cxx, DumpImageHeaderInfo.cxx, DumpToshibaDTI.cxx, DumpToshibaDTI2.cxx, ELSCINT1WaveToText.cxx, FixJAIBugJPEGLS.cxx, pmsct_rgb1.cxx, and rle2img.cxx.

Referenced by gdcm::DataSet::InsertDataElement(), gdcm::Attribute< Group, Element, TVR, TVM >::SetFromDataElement(),
gdcm::Attribute< Group, Element, TVR, VM::VM1 >::SetFromDataElement(), and gdcm::Attribute< Group, Element, TVR, VM::VM1_n >

10.73.4.16 IsUndefinedLength()

```
bool gdcm::DataElement::IsUndefinedLength ( ) const  [inline]
```

return if Value Length if of undefined length

10.73.4.17 operator<()

```
bool gdcm::DataElement::operator< (
            const DataElement & de ) const  [inline]
```

References GetTag().

10.73.4.18 operator=()

```
DataElement & gdcm::DataElement::operator= (
            const DataElement &  )  [default]
```

10.73.4.19 operator==()

```
bool gdcm::DataElement::operator== (
            const DataElement & de ) const  [inline]
```

References TagField, ValueField, ValueLengthField, and VRField.

10.73.4.20 Read()

```
template<typename TDE , typename TSwap >
std::istream & gdcm::DataElement::Read (
            std::istream & is )  [inline]
```

Examples

DumpSiemensBase64.cxx.

10.73.4.21 ReadOrSkip()

```
template<typename TDE , typename TSwap >
std::istream & gdcm::DataElement::ReadOrSkip (
             std::istream & is,
             std::set< Tag > const & skiptags )  [inline]
```

10.73.4.22 ReadPreValue()

```
template<typename TDE , typename TSwap >
std::istream & gdcm::DataElement::ReadPreValue (
             std::istream & is,
             std::set< Tag > const & skiptags )  [inline]
```

10.73.4.23 ReadValue()

```
template<typename TDE , typename TSwap >
std::istream & gdcm::DataElement::ReadValue (
             std::istream & is,
             std::set< Tag > const & skiptags )  [inline]
```

10.73.4.24 ReadValueWithLength()

```
template<typename TDE , typename TSwap >
std::istream & gdcm::DataElement::ReadValueWithLength (
             std::istream & is,
             VL & length,
             std::set< Tag > const & skiptags )  [inline]
```

10.73.4.25 ReadWithLength()

```
template<typename TDE , typename TSwap >
std::istream & gdcm::DataElement::ReadWithLength (
             std::istream & is,
             VL & length )  [inline]
```

10.73.4.26 SetByteValue()

```
void gdcm::DataElement::SetByteValue (
            const char * array,
            VL length )  [inline]
```

Set the byte value

Warning

user need to read DICOM standard for an understanding of:

- even padding
- \0 vs space padding By default even padding is achieved using \0 regardless of the of VR

Examples

BasicImageAnonymizer.cs, ChangePrivateTags.cxx, ChangeSequenceUltrasound.cxx, CreateARGBImage.cxx, CreateCMYKImage.cxx, DecompressImage.cs, DecompressImageMultiframe.cs, DecompressJPEGFile.cs, Fake_Image_Using_Stream_Image_Writer.cxx, FileChangeTS.cs, FileChangeTSLossy.cs, FixBrokenJ2K.cxx, FixJAIBugJPEGLS.cxx, GenFakeIdentifyFile.cxx, GenFakeImage.cxx, GenLongSeqs.cxx, GenSeqs.cxx, GetSubSequenceData.cxx, MpegVideoInfo.cs, NewSequence.cs, StreamImageReaderTest.cxx, iU22tomultisc.cxx, and rle2img.cxx.

Referenced by gdcm::Attribute< Group, Element, TVR, TVM >::GetAsDataElement(), gdcm::Attribute< Group, Element, TVR, VM::VM1 gdcm::Attribute< Group, Element, TVR, VM::VM1_n >::GetAsDataElement(), gdcm::Element< TVR, TVM >::GetAsDataElement(), and gdcm::Element< TVR, VM::VM1_n >::GetAsDataElement().

10.73.4.27 SetTag()

```
void gdcm::DataElement::SetTag (
            const Tag & t )  [inline]
```

Set Tag Use with cautious (need to match Part 6)

Examples

Extracting_All_Resolution.cxx, Fake_Image_Using_Stream_Image_Writer.cxx, GenFakeIdentifyFile.cxx, and GetSubSequenceData.cxx.

10.73.4.28 SetValue()

```
void gdcm::DataElement::SetValue (
            Value const & vl )  [inline]
```

Warning

you need to set the ValueLengthField explicitly

Examples

DecompressImageMultiframe.cs, DecompressJPEGFile.cs, DuplicatePCDE.cxx, Fake_Image_Using_Stream_Image_Writer.cxx, FixBrokenJ2K.cxx, GenFakeIdentifyFile.cxx, GenLongSeqs.cxx, GenSeqs.cxx, MpegVideoInfo.cs, and NewSequence.cs.

References gdcm::Value::GetLength().

10.73.4.29 SetValueFieldLength()

```
void gdcm::DataElement::SetValueFieldLength (
            VL vl,
            bool readvalues )  [protected]
```

10.73.4.30 SetVL()

```
void gdcm::DataElement::SetVL (
            const VL & vl )  [inline]
```

Set VL Use with cautious (need to match Part 6), advanced user only

See also

SetByteValue

10.73.4.31 SetVLToUndefined()

```
void gdcm::DataElement::SetVLToUndefined ( )
```

Examples

Fake_Image_Using_Stream_Image_Writer.cxx, GenAllVR.cxx, GenFakeIdentifyFile.cxx, GenLongSeqs.cxx, GenSeqs.cxx, and NewSequence.cs.

10.73.4.32 SetVR()

```
void gdcm::DataElement::SetVR (
            VR const & vr )  [inline]
```

Set VR Use with cautious (need to match Part 6), advanced user only

Precondition

vr is a VR::VRALL (not a dual one such as OB_OW)

Examples

Fake_Image_Using_Stream_Image_Writer.cxx, FixBrokenJ2K.cxx, FixJAIBugJPEGLS.cxx, GenFakeIdentifyFile.cxx, GenLongSeqs.cxx, GenSeqs.cxx, GetSubSequenceData.cxx, NewSequence.cs, StreamImageReaderTest.cxx, iU22tomultisc.cxx, and rle2img.cxx.

References gdcm::VR::IsVRFile().

Referenced by gdcm::Attribute< Group, Element, TVR, TVM >::GetAsDataElement(), gdcm::Attribute< Group, Element, TVR, VM::VM1 gdcm::Attribute< Group, Element, TVR, VM::VM1_n >::GetAsDataElement(), gdcm::Element< TVR, TVM >::GetAsDataElement(), and gdcm::Element< TVR, VM::VM1_n >::GetAsDataElement().

10.73.4.33 Write()

```
template<typename TDE , typename TSwap >
const std::ostream & gdcm::DataElement::Write (
            std::ostream & os ) const  [inline]
```

10.73.5 Friends And Related Function Documentation

10.73.5.1 operator<<

```
std::ostream & operator<< (
            std::ostream & _os,
            const DataElement & _val )  [friend]
```

10.73.6 Member Data Documentation

10.73.6.1 TagField

```
Tag gdcm::DataElement::TagField  [protected]
```

Referenced by operator==().

10.73.6.2 ValueField

```
ValuePtr gdcm::DataElement::ValueField  [protected]
```

Referenced by operator==().

10.73.6.3 ValueLengthField

```
VL gdcm::DataElement::ValueLengthField  [protected]
```

Referenced by operator==().

10.73.6.4 VRField

```
VR gdcm::DataElement::VRField  [protected]
```

Referenced by operator==().

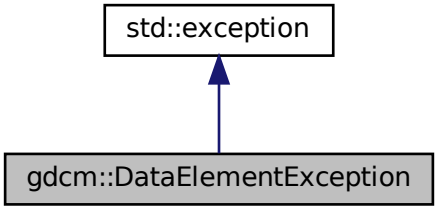
The documentation for this class was generated from the following file:

- gdcmDataElement.h

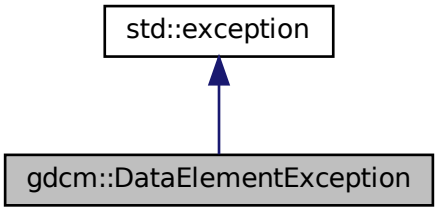
10.74 gdcm::DataElementException Class Reference

`#include <gdcmDataSet.h>`

Inheritance diagram for gdcm::DataElementException:



Collaboration diagram for gdcm::DataElementException:



The documentation for this class was generated from the following file:

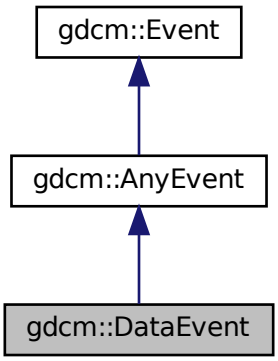
- gdcmDataSet.h

10.75 gdcm::DataEvent Class Reference

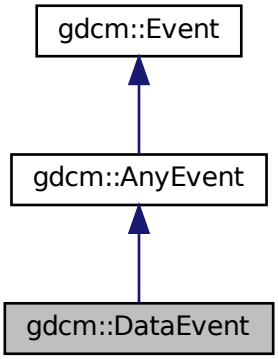
DataEvent.

`#include <gdcmDataEvent.h>`

Inheritance diagram for gdcm::DataEvent:



Collaboration diagram for gdcm::DataEvent:



Public Types

- typedef DataEvent Self
- typedef AnyEvent Superclass

Public Member Functions

- DataEvent (const char *bytes=nullptr, size_t len=0)
- DataEvent (const Self &s)
- ~DataEvent () override=default
- bool CheckEvent (const ::gdcm::Event *e) const override
- const char * GetData () const
- size_t GetDataLength () const
- const char * GetEventName () const override
- ::gdcm::Event * MakeObject () const override
- void operator= (const Self &)=delete
- void SetData (const char *bytes, size_t len)

10.75.1 Detailed Description

DataEvent.

10.75.2 Member Typedef Documentation

10.75.2.1 Self

```
typedef DataEvent gdcm::DataEvent::Self
```

10.75.2.2 Superclass

```
typedef AnyEvent gdcm::DataEvent::Superclass
```

10.75.3 Constructor & Destructor Documentation

10.75.3.1 DataEvent() [1/2]

```
gdcm::DataEvent::DataEvent (
            const char * bytes = nullptr,
            size_t len = 0 )  [inline]
```

10.75.3.2 ~DataEvent()

```
gdcm::DataEvent::~DataEvent ( )  [override], [default]
```

10.75.3.3 DataEvent() [2/2]

```
gdcm::DataEvent::DataEvent (
            const Self & s )  [inline]
```

10.75.4 Member Function Documentation

10.75.4.1 CheckEvent()

```
bool gdcm::DataEvent::CheckEvent (
            const ::gdcm::Event * e ) const  [inline], [override]
```

10.75.4.2 GetData()

```
const char * gdcm::DataEvent::GetData ( ) const  [inline]
```

10.75.4.3 GetDataLength()

```
size_t gdcm::DataEvent::GetDataLength ( ) const  [inline]
```

10.75.4.4 GetEventName()

```
const char * gdcm::DataEvent::GetEventName ( ) const  [inline], [override], [virtual]
```

Return the StringName associated with the event.

Implements gdcm::Event.

10.75.4.5 MakeObject()

```
::gdcm::Event * gdcm::DataEvent::MakeObject ( ) const  [inline], [override], [virtual]
```

Create an Event of this type This method work as a Factory for creating events of each particular type.

Implements gdcm::Event.

10.75.4.6 operator=()

```
void gdcm::DataEvent::operator= (
            const Self &  )  [delete]
```

10.75.4.7 SetData()

```
void gdcm::DataEvent::SetData (
            const char * bytes,
            size_t len )  [inline]
```

The documentation for this class was generated from the following file:

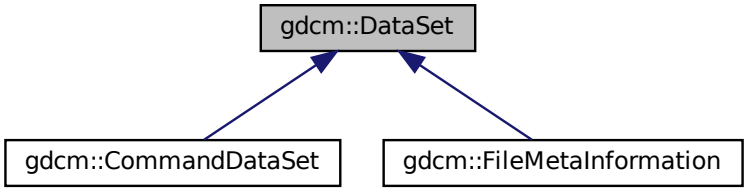
- gdcmDataEvent.h

10.76 gdcm::DataSet Class Reference

Class to represent a Data Set (which contains Data Elements)

```
#include <gdcmDataSet.h>
```

Inheritance diagram for gdcm::DataSet:



Public Types

- typedef DataElementSet::const_iterator ConstIterator
- typedef std::set< DataElement > DataElementSet
- typedef DataElementSet::iterator Iterator
- typedef DataElementSet::size_type SizeType

Public Member Functions

- Iterator Begin ()
- ConstIterator Begin () const
- void Clear ()
- template<typename TDE >
  unsigned int ComputeGroupLength (Tag const &tag) const
- Iterator End ()
- ConstIterator End () const
- bool FindDataElement (const PrivateTag &t) const

  Look up if private tag 't' is present in the dataset:
- bool FindDataElement (const Tag &t) const
- const DataElement & FindNextDataElement (const Tag &t) const
- const DataElement & GetDataElement (const PrivateTag &t) const

  Return the dataelement.
- const DataElement & GetDataElement (const Tag &t) const
- DataElementSet & GetDES ()
- const DataElementSet & GetDES () const
- template<typename TDE >
  VL GetLength () const
- MediaStorage GetMediaStorage () const
- std::string GetPrivateCreator (const Tag &t) const
- PrivateTag GetPrivateTag (const Tag &t) const

  Return the private tag of the private tag 't', private creator will be set to empty if not found.
- void Insert (const DataElement &de)
- bool IsEmpty () const

  Returns if the dataset is empty.
- const DataElement & operator() (uint16_t group, uint16_t element) const
- DataSet & operator= (DataSet const &)=default
- const DataElement & operator[ ] (const Tag &t) const
- void Print (std::ostream &os, std::string const &indent="") const
- template<typename TDE , typename TSwap >
  std::istream & Read (std::istream &is)
- template<typename TDE , typename TSwap >
  std::istream & ReadNested (std::istream &is)
- template<typename TDE , typename TSwap >
  std::istream & ReadSelectedPrivateTags (std::istream &is, const std::set< PrivateTag > &tags, bool readvalues=true)
- template<typename TDE , typename TSwap >
  std::istream & ReadSelectedPrivateTagsWithLength (std::istream &is, const std::set< PrivateTag > &tags, VL &length, bool readvalues=true)
- template<typename TDE , typename TSwap >
  std::istream & ReadSelectedTags (std::istream &is, const std::set< Tag > &tags, bool readvalues=true)

- template<typename TDE , typename TSwap >
  std::istream & ReadSelectedTagsWithLength (std::istream &is, const std::set< Tag > &tags, VL &length, bool readvalues=true)
- template<typename TDE , typename TSwap >
  std::istream & ReadUpToTag (std::istream &is, const Tag &t, std::set< Tag > const &skiptags)
- template<typename TDE , typename TSwap >
  std::istream & ReadUpToTagWithLength (std::istream &is, const Tag &t, std::set< Tag > const &skiptags, VL &length)
- template<typename TDE , typename TSwap >
  std::istream & ReadWithLength (std::istream &is, VL &length)
- SizeType Remove (const Tag &tag)

  Completely remove a dataelement from the dataset.
- void Replace (const DataElement &de)

  Replace a dataelement with another one.
- void ReplaceEmpty (const DataElement &de)

  Only replace a DICOM attribute when it is missing or empty.
- SizeType Size () const
- template<typename TDE , typename TSwap >
  std::ostream const & Write (std::ostream &os) const

Protected Member Functions

- Tag ComputeDataElement (const PrivateTag &t) const
- const DataElement & GetDEEnd () const
- void InsertDataElement (const DataElement &de)

Friends

- class CSAHeader
- std::ostream & operator<< (std::ostream &_os, const DataSet &)

10.76.1 Detailed Description

Class to represent a Data Set (which contains Data Elements)

A Data Set represents an instance of a real world Information Object

Note

DATA SET: Exchanged information consisting of a structured set of Attribute values directly or indirectly related to Information Objects. The value of each Attribute in a Data Set is expressed as a Data Element. A collection of Data Elements ordered by increasing Data Element Tag number that is an encoding of the values of Attributes of a real world object.

Implementation note. If one do: DataSet ds; ds.SetLength(0); ds.Read(is); setting length to 0 actually means try to read is as if it was a root DataSet. Other value are undefined (nested dataset with undefined length) or defined length (different from 0) means nested dataset with defined length.

Warning

a DataSet does not have a Transfer Syntax type, only a File does.

Examples

ChangePrivateTags.cxx, ChangeSequenceUltrasound.cxx, CompressLossyJPEG.cs, CreateFakeRTDOSE.cxx, CreateJPIPDataSet.cxx, DeriveSeries.cxx, DiffFile.cxx, DumpADAC.cxx, DumpCSA.cs, DumpExamCard.cxx, DumpGEMSMovieGroup.cxx, DumpImageHeaderInfo.cxx, DumpPhilipsECHO.cxx, DumpSiemensBase64.cxx, DumpToshibaDTI.cxx, DumpToshibaDTI2.cxx, DumpVisusChange.cxx, DuplicatePCDE.cxx, ELSCINT1WaveToText.cxx, ExtractEncapsulatedFile.cs, ExtractEncryptedContent.cxx, Extracting_All_Resolution.cxx, Fake_Image_Using_Stream_Image_Write FileChangeTS.cs, FileChangeTSLossy.cs, FixOrientation.cxx, GenAllVR.cxx, GenFakeIdentifyFile.cxx, GenLongSeqs.cxx, GenSeqs.cxx, GetJPEGSamplePrecision.cxx, GetSequenceUltrasound.cxx, GetSubSequenceData.cxx, HelloWorld.cxx, LargeVRDSExplicit.cxx, MergeTwoFiles.cxx, MrProtocol.cxx, NewSequence.cs, PatchFile.cxx, ReadAndDumpDICOMDIR.cxx, ReadAndPrintAttributes.cxx, ReadExplicitLengthSQIVR.cxx, ReadGEMSSDO.cxx, SimplePrint.cs, SortImage.cxx, SortImage2.cs, StreamImageReaderTest.cxx, TemplateEmptyImage.cxx, VolumeSorter.cxx, csa2img.cxx, gdcmrtionplan.cxx, gdcmrtplan.cxx, iU22tomultisc.cxx, pmsct_rgb1.cxx, and rle2img.cxx.

10.76.2 Member Typedef Documentation

10.76.2.1 ConstIterator

```
typedef DataElementSet::const_iterator gdcm::DataSet::ConstIterator
```

10.76.2.2 DataElementSet

```
typedef std::set<DataElement> gdcm::DataSet::DataElementSet
```

10.76.2.3 Iterator

```
typedef DataElementSet::iterator gdcm::DataSet::Iterator
```

10.76.2.4 SizeType

```
typedef DataElementSet::size_type gdcm::DataSet::SizeType
```

10.76.3 Member Function Documentation

10.76.3.1 Begin() [1/2]

```
Iterator gdcm::DataSet::Begin ( )  [inline]
```

10.76.3.2 Begin() [2/2]

```
ConstIterator gdcm::DataSet::Begin ( ) const  [inline]
```

Examples

DiffFile.cxx, DumpGEMSMovieGroup.cxx, DumpVisusChange.cxx, and DuplicatePCDE.cxx.

10.76.3.3 Clear()

```
void gdcm::DataSet::Clear ( )  [inline]
```

Referenced by gdcm::Item::Read().

10.76.3.4 ComputeDataElement()

```
Tag gdcm::DataSet::ComputeDataElement (
            const PrivateTag & t ) const  [protected]
```

10.76.3.5 ComputeGroupLength()

```
template<typename TDE >
unsigned int gdcm::DataSet::ComputeGroupLength (
            Tag const & tag ) const  [inline]
```

References gdcm::Tag::GetElement(), and gdcm::Tag::GetGroup().

10.76.3.6 End() [1/2]

```
Iterator gdcm::DataSet::End ( )  [inline]
```

10.76.3.7 End() [2/2]

```
ConstIterator gdcm::DataSet::End ( ) const  [inline]
```

Examples

DiffFile.cxx, DumpGEMSMovieGroup.cxx, DumpVisusChange.cxx, and DuplicatePCDE.cxx.

10.76.3.8 FindDataElement() [1/2]

```
bool gdcm::DataSet::FindDataElement (
            const PrivateTag & t ) const
```

Look up if private tag 't' is present in the dataset:

Examples

ChangePrivateTags.cxx, ChangeSequenceUltrasound.cxx, DumpADAC.cxx, DumpCSA.cs, DumpExamCard.cxx, DumpGEMSMovieGroup.cxx, DumpImageHeaderInfo.cxx, DumpPhilipsECHO.cxx, DumpSiemensBase64.cxx, DumpToshibaDTI.cxx, DumpToshibaDTI2.cxx, DumpVisusChange.cxx, ELSCINT1WaveToText.cxx, ExtractEncapsulatedFile.cs, ExtractEncryptedContent.cxx, GetSequenceUltrasound.cxx, GetSubSequenceData.cxx, LargeVRDSExplicit.cxx, MrProtocol.cxx, ReadAndDumpDICOMDIR.cxx, ReadAndPrintAttributes.cxx, ReadGEMSSDO.cxx, csa2img.cxx, gdcmrtionplan.cxx, gdcmrtplan.cxx, pmsct_rgb1.cxx, and rle2img.cxx.

Referenced by gdcm::Attribute< Group, Element, TVR, TVM >::SetFromDataSet(), gdcm::Attribute< Group, Element, TVR, VM::VM1 >::
and gdcm::Attribute< Group, Element, TVR, VM::VM1_n >::SetFromDataSet().

10.76.3.9 FindDataElement() [2/2]

```
bool gdcm::DataSet::FindDataElement (
            const Tag & t ) const  [inline]
```

10.76.3.10 FindNextDataElement()

```
const DataElement & gdcm::DataSet::FindNextDataElement (
            const Tag & t ) const  [inline]
```

Examples

DuplicatePCDE.cxx.

10.76.3.11 GetDataElement() [1/2]

```
const DataElement & gdcm::DataSet::GetDataElement (
            const PrivateTag & t ) const
```

Return the dataelement.

10.76.3.12 GetDataElement() [2/2]

```
const DataElement & gdcm::DataSet::GetDataElement (
            const Tag & t ) const  [inline]
```

Return the DataElement with Tag 't'

Warning

: This only search at the 'root level' of the DataSet

Examples

ChangePrivateTags.cxx, ChangeSequenceUltrasound.cxx, DecompressImage.cs, DeriveSeries.cxx, DumpADAC.cxx, DumpCSA.cs, DumpExamCard.cxx, DumpGEMSMovieGroup.cxx, DumpImageHeaderInfo.cxx, DumpPhilipsECHO.cxx, DumpSiemensBase64.cxx, DumpToshibaDTI.cxx, DumpToshibaDTI2.cxx, DumpVisusChange.cxx, ELSCINT1WaveToText.cxx, ExtractEncapsulatedFile.cs, ExtractEncryptedContent.cxx, FixBrokenJ2K.cxx, FixJAIBugJPEGLS.cxx, GetJPEGSamplePrecision.cx GetSequenceUltrasound.cxx, GetSubSequenceData.cxx, LargeVRDSExplicit.cxx, MrProtocol.cxx, PatchFile.cxx, ReadAndDumpDICOMDIR.cxx, ReadExplicitLengthSQIVR.cxx, ReadGEMSSDO.cxx, csa2img.cxx, gdcmrtionplan.cxx, gdcmrtplan.cxx, iU22tomultisc.cxx, pmsct_rgb1.cxx, and rle2img.cxx.

Referenced by gdcm::Attribute< Group, Element, TVR, TVM >::Set(), gdcm::Attribute< Group, Element, TVR, VM::VM1 >::Set(), gdcm::Attribute< Group, Element, TVR, VM::VM1_n >::Set(), gdcm::Attribute< Group, Element, TVR, TVM >::SetFromDataSet(), gdcm::Attribute< Group, Element, TVR, VM::VM1 >::SetFromDataSet(), and gdcm::Attribute< Group, Element, TVR, VM::VM1_n >::Se

10.76.3.13 GetDEEnd()

```
const DataElement & gdcm::DataSet::GetDEEnd ( ) const  [protected]
```

10.76.3.14 GetDES() [1/2]

```
DataElementSet & gdcm::DataSet::GetDES ( )  [inline]
```

10.76.3.15 GetDES() [2/2]

```
const DataElementSet & gdcm::DataSet::GetDES ( ) const  [inline]
```

Examples

ReadAndDumpDICOMDIR.cxx.

10.76.3.16 GetLength()

```
template<typename TDE >
VL gdcm::DataSet::GetLength ( ) const  [inline]
```

References gdcm::VL::GetLength().

10.76.3.17 GetMediaStorage()

```
MediaStorage gdcm::DataSet::GetMediaStorage ( ) const
```

10.76.3.18 GetPrivateCreator()

```
std::string gdcm::DataSet::GetPrivateCreator (
            const Tag & t ) const
```

Return the private creator of the private tag 't': or an empty string when not found

Examples

DuplicatePCDE.cxx.

10.76.3.19 GetPrivateTag()

```
PrivateTag gdcm::DataSet::GetPrivateTag (
            const Tag & t ) const
```

Return the private tag of the private tag 't', private creator will be set to empty if not found.

10.76.3.20 Insert()

```
void gdcm::DataSet::Insert (
            const DataElement & de )  [inline]
```

Insert a DataElement in the DataSet.

Warning

: Tag need to be >= 0x8 to be considered valid data element

Examples

CreateJPIPDataSet.cxx, DumpSiemensBase64.cxx, DuplicatePCDE.cxx, Extracting_All_Resolution.cxx, Fake_Image_Using_Stream_Image_Writer.cxx, FileChangeTS.cs, FileChangeTSLossy.cs, GenAllVR.cxx, GenFakeIdentifyFile.cxx, GenLongSeqs.cxx, GenSeqs.cxx, NewSequence.cs, StreamImageReaderTest.cxx, and TemplateEmptyImage.cxx.

References gdcmErrorMacro, gdcm::Tag::GetGroup(), and gdcm::DataElement::GetTag().

10.76.3.21 InsertDataElement()

```
void gdcm::DataSet::InsertDataElement (
            const DataElement & de )  [inline], [protected]
```

References gdcmWarningMacro, gdcm::Value::GetLength(), gdcm::DataElement::GetValue(), gdcm::DataElement::GetVL(), and gdcm::DataElement::IsEmpty().

10.76.3.22 IsEmpty()

```
bool gdcm::DataSet::IsEmpty ( ) const  [inline]
```

Returns if the dataset is empty.

Referenced by gdcm::Item::Read().

10.76.3.23 operator()()

```
const DataElement & gdcm::DataSet::operator() (
            uint16_t group,
            uint16_t element ) const  [inline]
```

10.76.3.24 operator=()

```
DataSet & gdcm::DataSet::operator= (
            DataSet const &  )  [default]
```

10.76.3.25 operator[]()

```
const DataElement & gdcm::DataSet::operator[] (
            const Tag & t ) const  [inline]
```

10.76.3.26 Print()

```
void gdcm::DataSet::Print (
            std::ostream & os,
            std::string const & indent = "" ) const  [inline]
```

10.76.3.27 Read()

```
template<typename TDE , typename TSwap >
std::istream & gdcm::DataSet::Read (
            std::istream & is )
```

Examples

DumpToshibaDTI.cxx, and DumpToshibaDTI2.cxx.

10.76.3.28 ReadNested()

```
template<typename TDE , typename TSwap >
std::istream & gdcm::DataSet::ReadNested (
            std::istream & is )
```

10.76.3.29 ReadSelectedPrivateTags()

```
template<typename TDE , typename TSwap >
std::istream & gdcm::DataSet::ReadSelectedPrivateTags (
            std::istream & is,
            const std::set< PrivateTag > & tags,
            bool readvalues = true )
```

10.76.3.30 ReadSelectedPrivateTagsWithLength()

```
template<typename TDE , typename TSwap >
std::istream & gdcm::DataSet::ReadSelectedPrivateTagsWithLength (
            std::istream & is,
            const std::set< PrivateTag > & tags,
            VL & length,
            bool readvalues = true )
```

10.76.3.31 ReadSelectedTags()

```
template<typename TDE , typename TSwap >
std::istream & gdcm::DataSet::ReadSelectedTags (
            std::istream & is,
            const std::set< Tag > & tags,
            bool readvalues = true )
```

10.76.3.32 ReadSelectedTagsWithLength()

```
template<typename TDE , typename TSwap >
std::istream & gdcm::DataSet::ReadSelectedTagsWithLength (
            std::istream & is,
            const std::set< Tag > & tags,
            VL & length,
            bool readvalues = true )
```

10.76.3.33 ReadUpToTag()

```
template<typename TDE , typename TSwap >
std::istream & gdcm::DataSet::ReadUpToTag (
             std::istream & is,
             const Tag & t,
             std::set< Tag > const & skiptags )
```

10.76.3.34 ReadUpToTagWithLength()

```
template<typename TDE , typename TSwap >
std::istream & gdcm::DataSet::ReadUpToTagWithLength (
             std::istream & is,
             const Tag & t,
             std::set< Tag > const & skiptags,
             VL & length )
```

10.76.3.35 ReadWithLength()

```
template<typename TDE , typename TSwap >
std::istream & gdcm::DataSet::ReadWithLength (
             std::istream & is,
             VL & length )
```

10.76.3.36 Remove()

```
SizeType gdcm::DataSet::Remove (
             const Tag & tag )  [inline]
```

Completely remove a dataelement from the dataset.

Examples

ClinicalTrialIdentificationWorkflow.cs, GenFakeIdentifyFile.cxx, LargeVRDSExplicit.cxx, MergeTwoFiles.cxx, ReformatFile.cs, StandardizeFiles.cs, pmsct_rgb1.cxx, and rle2img.cxx.

10.76.3.37 Replace()

```
void gdcm::DataSet::Replace (
            const DataElement & de )  [inline]
```

Replace a dataelement with another one.

Examples

ChangePrivateTags.cxx, ChangeSequenceUltrasound.cxx, CreateFakeRTDOSE.cxx, DeriveSeries.cxx, FixBrokenJ2K.cxx, FixJAIBugJPEGLS.cxx, FixOrientation.cxx, GenFakeIdentifyFile.cxx, GetSubSequenceData.cxx, HelloWorld.cxx, LargeVRDSExplicit.cxx, PatchFile.cxx, iU22tomultisc.cxx, pmsct_rgb1.cxx, and rle2img.cxx.

References gdcmAssertAlwaysMacro.

10.76.3.38 ReplaceEmpty()

```
void gdcm::DataSet::ReplaceEmpty (
            const DataElement & de )  [inline]
```

Only replace a DICOM attribute when it is missing or empty.

Examples

rle2img.cxx.

References gdcmAssertAlwaysMacro.

10.76.3.39 Size()

```
SizeType gdcm::DataSet::Size ( ) const  [inline]
```

Examples

DumpGEMSMovieGroup.cxx.

Referenced by gdcm::SequenceOfItems::Read().

10.76.3.40 Write()

```
template<typename TDE , typename TSwap >
std::ostream const & gdcm::DataSet::Write (
            std::ostream & os ) const
```

10.76.4 Friends And Related Function Documentation

10.76.4.1 CSAHeader

```
friend class CSAHeader  [friend]
```

10.76.4.2 operator<<

```
std::ostream & operator<< (
            std::ostream & _os,
            const DataSet & val )  [friend]
```

The documentation for this class was generated from the following file:

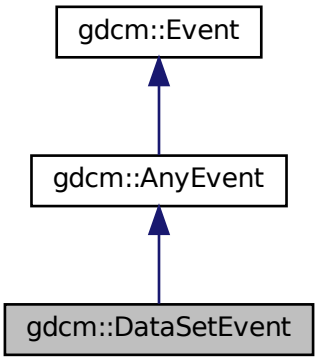
- gdcmDataSet.h

10.77 gdcm::DataSetEvent Class Reference

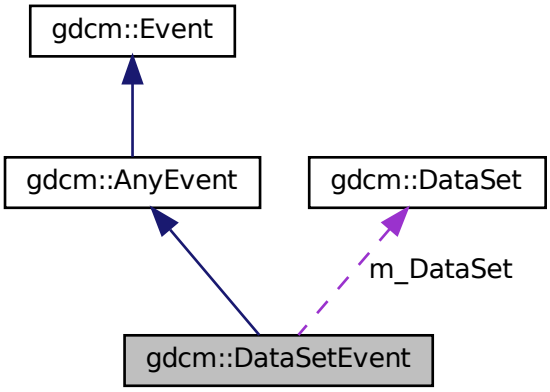
DataSetEvent.

`#include <gdcmDataSetEvent.h>`

Inheritance diagram for gdcm::DataSetEvent:



Collaboration diagram for gdcm::DataSetEvent:



Public Types

- typedef DataSetEvent Self
- typedef AnyEvent Superclass

Public Member Functions

- DataSetEvent (const Self &s)
- DataSetEvent (DataSet const *ds=nullptr)
- ~DataSetEvent () override=default
- bool CheckEvent (const ::gdcm::Event *e) const override
- DataSet const & GetDataSet () const
- const char * GetEventName () const override
- ::gdcm::Event * MakeObject () const override
- void operator= (const Self &)=delete

Public Attributes

- const DataSet * m_DataSet

10.77.1 Detailed Description

DataSetEvent.

Special type of event triggered during the DataSet store/move process

See also

10.77.2 Member Typedef Documentation

10.77.2.1 Self

```
typedef DataSetEvent gdcm::DataSetEvent::Self
```

10.77.2.2 Superclass

```
typedef AnyEvent gdcm::DataSetEvent::Superclass
```

10.77.3 Constructor & Destructor Documentation

10.77.3.1 DataSetEvent() [1/2]

```
gdcm::DataSetEvent::DataSetEvent (
            DataSet const * ds = nullptr )  [inline]
```

10.77.3.2 ~DataSetEvent()

```
gdcm::DataSetEvent::~DataSetEvent ( )  [override], [default]
```

10.77.3.3 DataSetEvent() [2/2]

```
gdcm::DataSetEvent::DataSetEvent (
            const Self & s )  [inline]
```

10.77.4 Member Function Documentation

10.77.4.1 CheckEvent()

```
bool gdcm::DataSetEvent::CheckEvent (
            const ::gdcm::Event * e ) const  [inline], [override]
```

10.77.4.2 GetDataSet()

```
DataSet const & gdcm::DataSetEvent::GetDataSet ( ) const  [inline]
```

References m_DataSet.

10.77.4.3 GetEventName()

```
const char * gdcm::DataSetEvent::GetEventName ( ) const  [inline], [override], [virtual]
```

Return the StringName associated with the event.

Implements gdcm::Event.

10.77.4.4 MakeObject()

```
::gdcm::Event * gdcm::DataSetEvent::MakeObject ( ) const  [inline], [override], [virtual]
```

Create an Event of this type This method work as a Factory for creating events of each particular type.

Implements gdcm::Event.

10.77.4.5 operator=()

```
void gdcm::DataSetEvent::operator= (
            const Self &  )  [delete]
```

10.77.5 Member Data Documentation

10.77.5.1 m_DataSet

```
const DataSet* gdcm::DataSetEvent::m_DataSet
```

Referenced by GetDataSet().

The documentation for this class was generated from the following file:

- gdcmDataSetEvent.h

10.78 gdcm::DataSetHelper Class Reference

DataSetHelper (internal class, not intended for user level)

```
#include <gdcmDataSetHelper.h>
```

Static Public Member Functions

- static VR ComputeVR (File const &file, DataSet const &ds, const Tag &tag)

10.78.1 Detailed Description

DataSetHelper (internal class, not intended for user level)

Examples

SimplePrint.cs.

10.78.2 Member Function Documentation

10.78.2.1 ComputeVR()

```
static VR gdcm::DataSetHelper::ComputeVR (
             File const & file,
             DataSet const & ds,
             const Tag & tag )  [static]
```

ds -> current dataset, which is not the same as the root dataset return VR::INVALID in case of error

Examples

SimplePrint.cs.

The documentation for this class was generated from the following file:

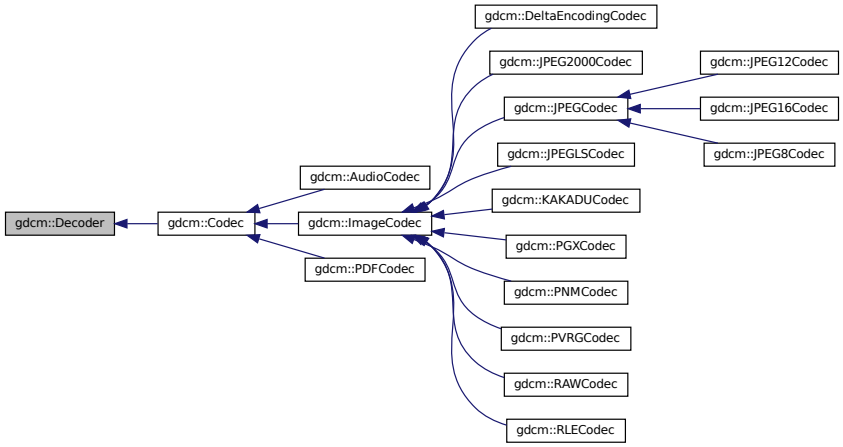
- gdcmDataSetHelper.h

10.79 gdcm::Decoder Class Reference

Decoder.

`#include <gdcmDecoder.h>`

Inheritance diagram for gdcm::Decoder:



Public Member Functions

- virtual ~Decoder ()=default
- virtual bool CanDecode (TransferSyntax const &) const =0

  Return whether this decoder support this transfer syntax (can decode it)
- virtual bool Decode (DataElement const &, DataElement &)

  Decode.

Protected Member Functions

- virtual bool DecodeByStreams (std::istream &, std::ostream &)

10.79.1 Detailed Description

Decoder.

10.79.2 Constructor & Destructor Documentation

10.79.2.1 ~Decoder()

```
virtual gdcm::Decoder::~Decoder ( )  [virtual], [default]
```

10.79.3 Member Function Documentation

10.79.3.1 CanDecode()

```
virtual bool gdcm::Decoder::CanDecode (
            TransferSyntax const &  ) const  [pure virtual]
```

Return whether this decoder support this transfer syntax (can decode it)

Implemented in gdcm::AudioCodec, gdcm::ImageCodec, gdcm::PDFCodec, gdcm::JPEG2000Codec, gdcm::JPEGCodec, gdcm::JPEGLSCodec, gdcm::KAKADUCodec, gdcm::PGXCodec, gdcm::PNMCodec, gdcm::PVRGCodec, gdcm::RAWCodec, and gdcm::RLECodec.

10.79.3.2 Decode()

```
virtual bool gdcm::Decoder::Decode (
            DataElement const & ,
            DataElement &  )  [inline], [virtual]
```

Decode.

Reimplemented in gdcm::DeltaEncodingCodec, gdcm::AudioCodec, gdcm::JPEG2000Codec, gdcm::JPEGCodec, gdcm::JPEGLSCodec, gdcm::KAKADUCodec, gdcm::PDFCodec, gdcm::PVRGCodec, gdcm::RAWCodec, gdcm::RLECodec, and gdcm::ImageCodec.

10.79.3.3 DecodeByStreams()

```
virtual bool gdcm::Decoder::DecodeByStreams (
            std::istream & ,
            std::ostream &  )  [inline], [protected], [virtual]
```

Reimplemented in gdcm::JPEG12Codec, gdcm::JPEG16Codec, gdcm::JPEG2000Codec, gdcm::JPEG8Codec, gdcm::JPEGCodec, gdcm::RAWCodec, gdcm::RLECodec, and gdcm::ImageCodec.

The documentation for this class was generated from the following file:

- gdcmDecoder.h

10.80 gdcm::DefinedTerms Class Reference

Defined Terms are used when the specified explicit Values may be extended by implementors to include additional new Values. These new Values shall be specified in the Conformance Statement (see PS 3.2) and shall not have the same meaning as currently defined Values in this standard. A Data Element with Defined Terms that does not contain a Value equivalent to one of the Values currently specified in this standard shall not be considered to have an invalid value. Note: Interpretation Type ID (4008,0210) is an example of a Data Element having Defined Terms. It is defined to have a Value that may be one of the set of standard Values; REPORT or AMENDMENT (see PS 3.3). Because this Data Element has Defined Terms other Interpretation Type IDs may be defined by the implementor.

`#include <gdcmDefinedTerms.h>`

Public Member Functions

- DefinedTerms ()=default

10.80.1 Detailed Description

Defined Terms are used when the specified explicit Values may be extended by implementors to include additional new Values. These new Values shall be specified in the Conformance Statement (see PS 3.2) and shall not have the same meaning as currently defined Values in this standard. A Data Element with Defined Terms that does not contain a Value equivalent to one of the Values currently specified in this standard shall not be considered to have an invalid value. Note: Interpretation Type ID (4008,0210) is an example of a Data Element having Defined Terms. It is defined to have a Value that may be one of the set of standard Values; REPORT or AMENDMENT (see PS 3.3). Because this Data Element has Defined Terms other Interpretation Type IDs may be defined by the implementor.

10.80.2 Constructor & Destructor Documentation

10.80.2.1 DefinedTerms()

`gdcm::DefinedTerms::DefinedTerms ( )  [default]`

The documentation for this class was generated from the following file:

- gdcmDefinedTerms.h

10.81 gdcm::Defs Class Reference

FIXME I do not like the name 'Defs'.

`#include <gdcmDefs.h>`

Public Member Functions

- Defs ()
- Defs (const Defs &val)=delete
- ~Defs ()
- const IOD & GetIODFromFile (const File &file) const
- IODs & GetIODs ()
- const IODs & GetIODs () const
- Macros & GetMacros ()
- const Macros & GetMacros () const
- Modules & GetModules ()
- const Modules & GetModules () const
- Type GetTypeFromTag (const File &file, const Tag &tag) const
- bool IsEmpty () const
- Defs & operator= (const Defs &val)=delete
- bool Verify (const DataSet &ds) const
- bool Verify (const File &file) const

Static Public Member Functions

- static const char * GetIODNameFromMediaStorage (MediaStorage const &ms)

Protected Member Functions

- void LoadDefaults ()
- void LoadFromFile (const char *filename)

Friends

- class Global

10.81.1 Detailed Description

FIXME I do not like the name 'Defs'.

Note

bla

Examples

GenerateStandardSOPClasses.cxx, and TraverseModules.cxx.

10.81.2 Constructor & Destructor Documentation

10.81.2.1 Defs() [1/2]

```
gdcm::Defs::Defs ( )
```

10.81.2.2 ~Defs()

```
gdcm::Defs::~Defs ( )
```

10.81.2.3 Defs() [2/2]

```
gdcm::Defs::Defs (
            const Defs & val )  [delete]
```

10.81.3 Member Function Documentation

10.81.3.1 GetIODFromFile()

```
const IOD & gdcm::Defs::GetIODFromFile (
            const File & file ) const
```

10.81.3.2 GetIODNameFromMediaStorage()

```
static const char * gdcm::Defs::GetIODNameFromMediaStorage (
            MediaStorage const & ms )  [static]
```

Examples

GenerateStandardSOPClasses.cxx.

10.81.3.3 GetIODs() [1/2]

```
IODs & gdcm::Defs::GetIODs ( )  [inline]
```

10.81.3.4 GetIODs() [2/2]

```
const IODs & gdcm::Defs::GetIODs ( ) const  [inline]
```

Examples

TraverseModules.cxx.

10.81.3.5 GetMacros() [1/2]

```
Macros & gdcm::Defs::GetMacros ( )  [inline]
```

10.81.3.6 GetMacros() [2/2]

```
const Macros & gdcm::Defs::GetMacros ( ) const  [inline]
```

Users should not directly use Macro. Macro are simply a way for DICOM WG to re-use Tables. Macros are conviently wrapped within Modules. See gdcm::Module API directly

Examples

TraverseModules.cxx.

10.81.3.7 GetModules() [1/2]

```
Modules & gdcm::Defs::GetModules ( )  [inline]
```

10.81.3.8 GetModules() [2/2]

```
const Modules & gdcm::Defs::GetModules ( ) const  [inline]
```

Examples

TraverseModules.cxx.

10.81.3.9 GetTypeFromTag()

```
Type gdcm::Defs::GetTypeFromTag (
            const File & file,
            const Tag & tag ) const
```

10.81.3.10 IsEmpty()

```
bool gdcm::Defs::IsEmpty ( ) const  [inline]
```

10.81.3.11 LoadDefaults()

```
void gdcm::Defs::LoadDefaults ( )  [protected]
```

10.81.3.12 LoadFromFile()

```
void gdcm::Defs::LoadFromFile (
            const char * filename )  [protected]
```

10.81.3.13 operator=()

```
Defs & gdcm::Defs::operator= (
            const Defs & val )  [delete]
```

10.81.3.14 Verify() [1/2]

```
bool gdcm::Defs::Verify (
            const DataSet & ds ) const
```

10.81.3.15 Verify() [2/2]

```
bool gdcm::Defs::Verify (
            const File & file ) const
```

10.81.4 Friends And Related Function Documentation

10.81.4.1 Global

```
friend class Global  [friend]
```

The documentation for this class was generated from the following file:

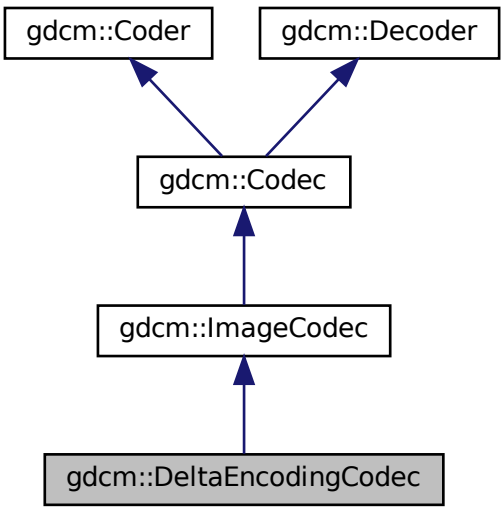
- gdcmDefs.h

10.82 gdcm::DeltaEncodingCodec Class Reference

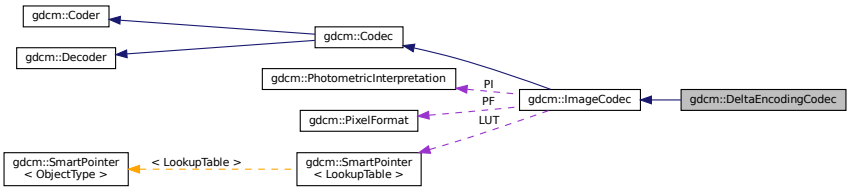
DeltaEncodingCodec compression used by some private vendor.

```
#include <gdcmDeltaEncodingCodec.h>
```

Inheritance diagram for gdcm::DeltaEncodingCodec:



Collaboration diagram for gdcm::DeltaEncodingCodec:



Public Member Functions

- DeltaEncodingCodec ()
- ~DeltaEncodingCodec ()
- bool CanDecode (TransferSyntax const &ts)
- bool Decode (DataElement const &is, DataElement &os)

  Decode.

Protected Member Functions

- bool Decode (std::istream &is, std::ostream &os)

Additional Inherited Members

10.82.1 Detailed Description

DeltaEncodingCodec compression used by some private vendor.

10.82.2 Constructor & Destructor Documentation

10.82.2.1 DeltaEncodingCodec()

```
gdcm::DeltaEncodingCodec::DeltaEncodingCodec ( )
```

10.82.2.2 ~DeltaEncodingCodec()

```
gdcm::DeltaEncodingCodec::~DeltaEncodingCodec ( )
```

10.82.3 Member Function Documentation

10.82.3.1 CanDecode()

```
bool gdcm::DeltaEncodingCodec::CanDecode (
            TransferSyntax const & ts )
```

10.82.3.2 Decode() [1/2]

```
bool gdcm::DeltaEncodingCodec::Decode (
            DataElement const & ,
            DataElement &  )  [virtual]
```

Decode.

Reimplemented from gdcm::Decoder.

10.82.3.3 Decode() [2/2]

```
bool gdcm::DeltaEncodingCodec::Decode (
            std::istream & is,
            std::ostream & os )  [protected]
```

The documentation for this class was generated from the following file:

- gdcmDeltaEncodingCodec.h

10.83 gdcm::DICOMDIR Class Reference

DICOMDIR class.

```
#include <gdcmDICOMDIR.h>
```

Public Member Functions

- DICOMDIR ()=default
- DICOMDIR (FileSet fs)

10.83.1 Detailed Description

DICOMDIR class.

Structured for handling DICOMDIR

10.83.2 Constructor & Destructor Documentation

10.83.2.1 DICOMDIR() [1/2]

```
gdcm::DICOMDIR::DICOMDIR ( )  [default]
```

10.83.2.2 DICOMDIR() [2/2]

```
gdcm::DICOMDIR::DICOMDIR (
            FileSet fs )  [inline]
```

The documentation for this class was generated from the following file:

- gdcmDICOMDIR.h

10.84 gdcm::DICOMDIRGenerator Class Reference

DICOMDIRGenerator class.

```
#include <gdcmDICOMDIRGenerator.h>
```

Public Types

- typedef Directory::FilenamesType FilenamesType
- typedef Directory::FilenameType FilenameType

Public Member Functions

- DICOMDIRGenerator ()
- ~DICOMDIRGenerator ()
- bool Generate ()

  Main function to generate the DICOMDIR.
- File & GetFile ()
- void SetDescriptor (const char *d)
- void SetFile (const File &f)

  Set/Get file. The DICOMDIR file will be valid once a call to Generate has been done.
- void SetFilenames (FilenamesType const &fns)

  Set the list of filenames from which the DICOMDIR should be generated from.
- void SetRootDirectory (FilenameType const &root)

  Set the root directory from which the filenames should be considered.

Protected Member Functions

- bool AddImageDirectoryRecord ()
- bool AddPatientDirectoryRecord ()
- bool AddSeriesDirectoryRecord ()
- bool AddStudyDirectoryRecord ()
- Scanner & GetScanner ()

10.84.1 Detailed Description

DICOMDIRGenerator class.

This is a STD-GEN-CD DICOMDIR generator. ref: PS 3.11-2008 Annex D (Normative) - General Purpose CD-R and DVD Interchange Profiles

Note

PS 3.11 - 2008 / D.3.2 Physical Medium And Medium Format The STD-GEN-CD and STD-GEN-SEC-CD application profiles require the 120 mm CD-R physical medium with the ISO/IEC 9660 Media Format, as defined in PS3.12. See also PS 3.12 - 2008 / Annex F 120mm CD-R Medium (Normative) and PS 3.10 - 2008 / 8 DICOM File Service / 8.1 FILE-SET

Warning

: PS 3.11 - 2008 / D.3.1 SOP Classes and Transfer Syntaxes Composite Image & Stand-alone Storage are required to be stored as Explicit VR Little Endian Uncompressed (1.2.840.10008.1.2.1). When a DICOM file is found using another Transfer Syntax the generator will simply stops.

- Input files should be Explicit VR Little Endian
- filenames should be valid VR::CS value (16 bytes, upper case ...)

Bug : There is a current limitation of not handling Referenced SOP Class UID / Referenced SOP Instance UID simply because the Scanner does not allow us See PS 3.11 / Table D.3-2 STD-GEN Additional DICOMDIR Keys

Examples

GenerateDICOMDIR.cs.

10.84.2 Member Typedef Documentation

10.84.2.1 FilenamesType

```
typedef Directory::FilenamesType gdcm::DICOMDIRGenerator::FilenamesType
```

10.84.2.2 FilenameType

```
typedef Directory::FilenameType gdcm::DICOMDIRGenerator::FilenameType
```

10.84.3 Constructor & Destructor Documentation

10.84.3.1 DICOMDIRGenerator()

```
gdcm::DICOMDIRGenerator::DICOMDIRGenerator ( )
```

10.84.3.2 ~DICOMDIRGenerator()

```
gdcm::DICOMDIRGenerator::~DICOMDIRGenerator ( )
```

10.84.4 Member Function Documentation

10.84.4.1 AddImageDirectoryRecord()

```
bool gdcm::DICOMDIRGenerator::AddImageDirectoryRecord ( )  [protected]
```

10.84.4.2 AddPatientDirectoryRecord()

```
bool gdcm::DICOMDIRGenerator::AddPatientDirectoryRecord ( )  [protected]
```

10.84.4.3 AddSeriesDirectoryRecord()

```
bool gdcm::DICOMDIRGenerator::AddSeriesDirectoryRecord ( )  [protected]
```

10.84.4.4 AddStudyDirectoryRecord()

```
bool gdcm::DICOMDIRGenerator::AddStudyDirectoryRecord ( )  [protected]
```

10.84.4.5 Generate()

```
bool gdcm::DICOMDIRGenerator::Generate ( )
```

Main function to generate the DICOMDIR.

Examples

GenerateDICOMDIR.cs.

10.84.4.6 GetFile()

```
File & gdcm::DICOMDIRGenerator::GetFile ( )
```

Examples

GenerateDICOMDIR.cs.

10.84.4.7 GetScanner()

```
Scanner & gdcm::DICOMDIRGenerator::GetScanner ( )  [protected]
```

10.84.4.8 SetDescriptor()

```
void gdcm::DICOMDIRGenerator::SetDescriptor (
            const char * d )
```

Set the File Set ID.

Warning

this need to be a valid VR::CS value

Examples

GenerateDICOMDIR.cs.

10.84.4.9 SetFile()

```
void gdcm::DICOMDIRGenerator::SetFile (
            const File & f )
```

Set/Get file. The DICOMDIR file will be valid once a call to Generate has been done.

10.84.4.10 SetFilenames()

```
void gdcm::DICOMDIRGenerator::SetFilenames (
            FilenamesType const & fns )
```

Set the list of filenames from which the DICOMDIR should be generated from.

Examples

GenerateDICOMDIR.cs.

10.84.4.11 SetRootDirectory()

```
void gdcm::DICOMDIRGenerator::SetRootDirectory (
            FilenameType const & root )
```

Set the root directory from which the filenames should be considered.

The documentation for this class was generated from the following file:

- gdcmDICOMDIRGenerator.h

10.85 gdcm::Dict Class Reference

Class to represent a map of DictEntry.

`#include <gdcmDict.h>`

Public Types

- typedef MapDictEntry::const_iterator ConstIterator
- typedef MapDictEntry::iterator Iterator
- typedef std::map< Tag, DictEntry > MapDictEntry

Public Member Functions

- Dict ()
- Dict (const Dict &_val)=delete
- void AddDictEntry (const Tag &tag, const DictEntry &de)
- ConstIterator Begin () const
- ConstIterator End () const
- const DictEntry & GetDictEntry (const Tag &tag) const
- const DictEntry & GetDictEntryByKeyword (const char *keyword, Tag &tag) const
- const DictEntry & GetDictEntryByName (const char *name, Tag &tag) const
- const char * GetKeywordFromTag (Tag const &tag) const

  Function to return the Keyword from a Tag.
- bool IsEmpty () const
- Dict & operator= (const Dict &_val)=delete

Protected Member Functions

- void LoadDefault ()

Friends

- class Dicts
- std::ostream & operator<< (std::ostream &_os, const Dict &_val)

10.85.1 Detailed Description

Class to represent a map of DictEntry.

Note

bla TODO FIXME: For Element == 0x0 need to return Name = Group Length ValueRepresentation = UL Value↩ Multiplicity = 1

Examples

GenAllVR.cxx, GenFakeIdentifyFile.cxx, PublicDict.cxx, and ReadAndPrintAttributes.cxx.

10.85.2 Member Typedef Documentation

10.85.2.1 ConstIterator

```
typedef MapDictEntry::const_iterator gdcm::Dict::ConstIterator
```

10.85.2.2 Iterator

```
typedef MapDictEntry::iterator gdcm::Dict::Iterator
```

10.85.2.3 MapDictEntry

```
typedef std::map<Tag, DictEntry> gdcm::Dict::MapDictEntry
```

10.85.3 Constructor & Destructor Documentation

10.85.3.1 Dict() [1/2]

```
gdcm::Dict::Dict ( )  [inline]
```

10.85.3.2 Dict() [2/2]

```
gdcm::Dict::Dict (
            const Dict & _val )  [delete]
```

10.85.4 Member Function Documentation

10.85.4.1 AddDictEntry()

```
void gdcm::Dict::AddDictEntry (
            const Tag & tag,
            const DictEntry & de )  [inline]
```

10.85.4.2 Begin()

```
ConstIterator gdcm::Dict::Begin ( ) const  [inline]
```

Examples

GenAllVR.cxx, and GenFakeIdentifyFile.cxx.

10.85.4.3 End()

```
ConstIterator gdcm::Dict::End ( ) const  [inline]
```

Examples

GenAllVR.cxx, and GenFakeIdentifyFile.cxx.

10.85.4.4 GetDictEntry()

```
const DictEntry & gdcm::Dict::GetDictEntry (
            const Tag & tag ) const  [inline]
```

Examples

GenFakeIdentifyFile.cxx, and PublicDict.cxx.

10.85.4.5 GetDictEntryByKeyword()

```
const DictEntry & gdcm::Dict::GetDictEntryByKeyword (
            const char * keyword,
            Tag & tag ) const  [inline]
```

Lookup DictEntry by keyword. Even if DICOM standard defines keyword as being unique. The lookup table is built on Tag. Therefore looking up a DictEntry by Keyword is more inefficient than looking up by Tag.

10.85.4.6 GetDictEntryByName()

```
const DictEntry & gdcm::Dict::GetDictEntryByName (
            const char * name,
            Tag & tag ) const  [inline]
```

Inefficient way of looking up tag by name. Technically DICOM does not guarantee uniqueness (and Curve / Overlay are there to prove it). But most of the time name is in fact uniq and can be uniquely link to a tag

Examples

ReadAndPrintAttributes.cxx.

10.85.4.7 GetKeywordFromTag()

```
const char * gdcm::Dict::GetKeywordFromTag (
            Tag const & tag ) const  [inline]
```

Function to return the Keyword from a Tag.

10.85.4.8 IsEmpty()

```
bool gdcm::Dict::IsEmpty ( ) const  [inline]
```

10.85.4.9 LoadDefault()

```
void gdcm::Dict::LoadDefault ( )  [protected]
```

10.85.4.10 operator=()

```
Dict & gdcm::Dict::operator= (
            const Dict & _val )  [delete]
```

10.85.5 Friends And Related Function Documentation

10.85.5.1 Dicts

```
friend class Dicts  [friend]
```

10.85.5.2 operator<<

```
std::ostream & operator<< (
             std::ostream & _os,
             const Dict & _val )  [friend]
```

The documentation for this class was generated from the following file:

- gdcmDict.h

10.86 gdcm::DictConverter Class Reference

Class to convert a .dic file into something else:

```
#include <gdcmDictConverter.h>
```

Public Types

- enum OutputTypes {
  DICT_DEFAULT = 0 ,
  DICT_DEBUG ,
  DICT_XML }

Public Member Functions

- DictConverter ()
- ~DictConverter ()
- void Convert ()
- const std::string & GetDictName () const
- const std::string & GetInputFilename () const
- const std::string & GetOutputFilename () const
- int GetOutputType () const
- void SetDictName (const char *name)
- void SetInputFileName (const char *filename)
- void SetOutputFileName (const char *filename)
- void SetOutputType (int type)

Static Public Member Functions

- static bool Readuint16 (const char *raw, uint16_t &ov)
- static bool ReadVM (const char *raw, VM::VMType &type)
- static bool ReadVR (const char *raw, VR::VRType &type)

Protected Member Functions

- void AddGroupLength ()
- bool ConvertToCXX (const char *raw, std::string &cxx)
- bool ConvertToXML (const char *raw, std::string &cxx)
- void WriteFooter ()
- void WriteHeader ()

10.86.1 Detailed Description

Class to convert a .dic file into something else:

- CXX code : embed dict into shared lib (DICT_DEFAULT)
- Debug mode (DICT_DEBUG)
- XML dict (DICT_XML)

Note

10.86.2 Member Enumeration Documentation

10.86.2.1 OutputTypes

`enum gdcm::DictConverter::OutputTypes`

Enumerator

DICT_DEFAULT	
DICT_DEBUG	
DICT_XML	

10.86.3 Constructor & Destructor Documentation

10.86.3.1 DictConverter()

```
gdcm::DictConverter::DictConverter ( )
```

10.86.3.2 ~DictConverter()

```
gdcm::DictConverter::~DictConverter ( )
```

10.86.4 Member Function Documentation

10.86.4.1 AddGroupLength()

```
void gdcm::DictConverter::AddGroupLength ( )  [protected]
```

10.86.4.2 Convert()

```
void gdcm::DictConverter::Convert ( )
```

10.86.4.3 ConvertToCXX()

```
bool gdcm::DictConverter::ConvertToCXX (
            const char * raw,
            std::string & cxx )  [protected]
```

10.86.4.4 ConvertToXML()

```
bool gdcm::DictConverter::ConvertToXML (
            const char * raw,
            std::string & cxx )  [protected]
```

10.86.4.5 GetDictName()

```
const std::string & gdcm::DictConverter::GetDictName ( ) const
```

10.86.4.6 GetInputFilename()

```
const std::string & gdcm::DictConverter::GetInputFilename ( ) const
```

10.86.4.7 GetOutputFilename()

```
const std::string & gdcm::DictConverter::GetOutputFilename ( ) const
```

10.86.4.8 GetOutputType()

```
int gdcm::DictConverter::GetOutputType ( ) const  [inline]
```

10.86.4.9 Readuint16()

```
static bool gdcm::DictConverter::Readuint16 (
            const char * raw,
            uint16_t & ov )  [static]
```

10.86.4.10 ReadVM()

```
static bool gdcm::DictConverter::ReadVM (
            const char * raw,
            VM::VMType & type )  [static]
```

10.86.4.11 ReadVR()

```
static bool gdcm::DictConverter::ReadVR (
            const char * raw,
            VR::VRType & type )  [static]
```

10.86.4.12 SetDictName()

```
void gdcm::DictConverter::SetDictName (
            const char * name )
```

10.86.4.13 SetInputFileName()

```
void gdcm::DictConverter::SetInputFileName (
            const char * filename )
```

10.86.4.14 SetOutputFileName()

```
void gdcm::DictConverter::SetOutputFileName (
            const char * filename )
```

10.86.4.15 SetOutputType()

```
void gdcm::DictConverter::SetOutputType (
            int type )  [inline]
```

10.86.4.16 WriteFooter()

```
void gdcm::DictConverter::WriteFooter ( )  [protected]
```

10.86.4.17 WriteHeader()

```
void gdcm::DictConverter::WriteHeader ( )  [protected]
```

The documentation for this class was generated from the following file:

- gdcmDictConverter.h

10.87 gdcm::DictEntry Class Reference

Class to represent an Entry in the Dict.

```
#include <gdcmDictEntry.h>
```

Public Member Functions

- DictEntry (const char *name="", const char *keyword="", VR const &vr=VR::INVALID, VM const &vm=VM::VM0, bool ret=false)
- const char * GetKeyword () const

  same as GetName but without spaces...
- const char * GetName () const

  Set/Get Name.
- bool GetRetired () const

  Set/Get Retired flag.
- const VM & GetVM () const

  Set/Get VM.
- const VR & GetVR () const

  Set/Get VR.
- bool IsUnique () const
- void SetElementXX (bool v)

  Set whether element is shared in multiple elements (Source Image IDs typically)
- void SetGroupXX (bool v)

  Set whether element is shared in multiple groups (Curve/Overlay typically)
- void SetKeyword (const char *keyword)
- void SetName (const char *name)
- void SetRetired (bool retired)
- void SetVM (VM const &vm)
- void SetVR (const VR &vr)

Friends

- class Dict
- std::ostream & operator<< (std::ostream &_os, const DictEntry &_val)

10.87.1 Detailed Description

Class to represent an Entry in the Dict.

Does not really exist within the DICOM definition, just a way to minimize storage and have a mapping from gdcm::Tag to the needed information

Note

bla TODO FIXME: Need a PublicDictEntry...indeed DictEntry has a notion of retired which does not exist in PrivateDictEntry...

See also

gdcm::Dict

Examples

GenAllVR.cxx, GenFakeIdentifyFile.cxx, PublicDict.cxx, and TraverseModules.cxx.

10.87.2 Constructor & Destructor Documentation

10.87.2.1 DictEntry()

```
gdcm::DictEntry::DictEntry (
            const char * name = "",
            const char * keyword = "",
            VR const & vr = VR::INVALID,
            VM const & vm = VM::VM0,
            bool ret = false )  [inline]
```

10.87.3 Member Function Documentation

10.87.3.1 GetKeyword()

```
const char * gdcm::DictEntry::GetKeyword ( ) const  [inline]
```

same as GetName but without spaces...

10.87.3.2 GetName()

```
const char * gdcm::DictEntry::GetName ( ) const  [inline]
```

Set/Get Name.

Referenced by gdcm::PrivateDict::PrintXML().

10.87.3.3 GetRetired()

```
bool gdcm::DictEntry::GetRetired ( ) const  [inline]
```

Set/Get Retired flag.

Examples

GenAllVR.cxx.

10.87.3.4 GetVM()

```
const VM & gdcm::DictEntry::GetVM ( ) const  [inline]
```

Set/Get VM.

Referenced by gdcm::PrivateDict::AddDictEntry(), and gdcm::PrivateDict::PrintXML().

10.87.3.5 GetVR()

```
const VR & gdcm::DictEntry::GetVR ( ) const  [inline]
```

Set/Get VR.

Examples

GenAllVR.cxx, and GenFakeIdentifyFile.cxx.

Referenced by gdcm::PrivateDict::AddDictEntry(), and gdcm::PrivateDict::PrintXML().

10.87.3.6 IsUnique()

```
bool gdcm::DictEntry::IsUnique ( ) const  [inline]
```

Return whether the name of the DataElement can be considered to be unique. As of 2008 all elements name were unique (except the expclitely 'XX' ones)

10.87.3.7 SetElementXX()

```
void gdcm::DictEntry::SetElementXX (
            bool v )  [inline]
```

Set whether element is shared in multiple elements (Source Image IDs typically)

10.87.3.8 SetGroupXX()

```
void gdcm::DictEntry::SetGroupXX (
            bool v )  [inline]
```

Set whether element is shared in multiple groups (Curve/Overlay typically)

10.87.3.9 SetKeyword()

```
void gdcm::DictEntry::SetKeyword (
            const char * keyword )  [inline]
```

10.87.3.10 SetName()

```
void gdcm::DictEntry::SetName (
            const char * name )  [inline]
```

10.87.3.11 SetRetired()

```
void gdcm::DictEntry::SetRetired (
            bool retired )  [inline]
```

10.87.3.12 SetVM()

```
void gdcm::DictEntry::SetVM (
            VM const & vm )  [inline]
```

Referenced by gdcm::PrivateDict::AddDictEntry().

10.87.3.13 SetVR()

```
void gdcm::DictEntry::SetVR (
            const VR & vr )  [inline]
```

Referenced by gdcm::PrivateDict::AddDictEntry().

10.87.4 Friends And Related Function Documentation

10.87.4.1 Dict

```
friend class Dict  [friend]
```

10.87.4.2 operator<<

```
std::ostream & operator<< (
             std::ostream & _os,
             const DictEntry & _val )  [friend]
```

The documentation for this class was generated from the following file:

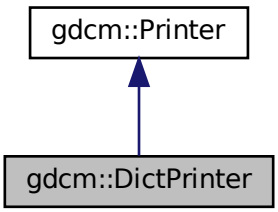
- gdcmDictEntry.h

10.88 gdcm::DictPrinter Class Reference

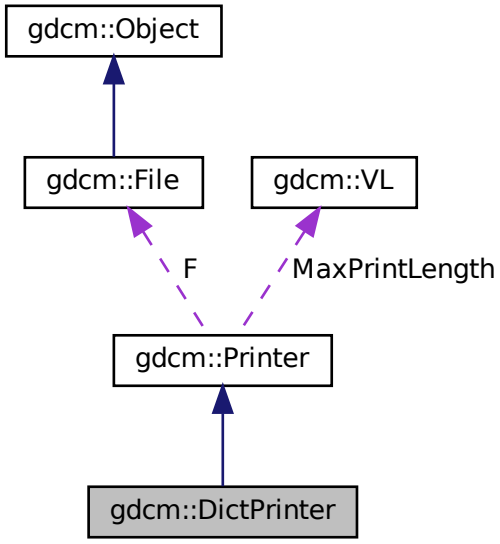
DictPrinter class.

```
#include <gdcmDictPrinter.h>
```

Inheritance diagram for gdcm::DictPrinter:



Collaboration diagram for gdcm::DictPrinter:



Public Member Functions

- DictPrinter ()
- ~DictPrinter ()
- void Print (std::ostream &os)

Protected Member Functions

- void PrintDataElement2 (std::ostream &os, const DataSet &ds, const DataElement &ide)
- void PrintDataSet2 (std::ostream &os, const DataSet &ds)

Additional Inherited Members

10.88.1 Detailed Description

DictPrinter class.

10.88.2 Constructor & Destructor Documentation

10.88.2.1 DictPrinter()

```
gdcm::DictPrinter::DictPrinter ( )
```

10.88.2.2 ~DictPrinter()

```
gdcm::DictPrinter::~DictPrinter ( )
```

10.88.3 Member Function Documentation

10.88.3.1 Print()

```
void gdcm::DictPrinter::Print (
            std::ostream & os )
```

10.88.3.2 PrintDataElement2()

```
void gdcm::DictPrinter::PrintDataElement2 (
            std::ostream & os,
            const DataSet & ds,
            const DataElement & ide )  [protected]
```

10.88.3.3 PrintDataSet2()

```
void gdcm::DictPrinter::PrintDataSet2 (
            std::ostream & os,
            const DataSet & ds )  [protected]
```

The documentation for this class was generated from the following file:

- gdcmDictPrinter.h

10.89 gdcm::Dicts Class Reference

Class to manipulate the sum of knowledge (all the dict user load)

`#include <gdcmDicts.h>`

Public Member Functions

- Dicts ()
- Dicts (const Dicts &_val)=delete
- ~Dicts ()
- const CSAHeaderDict & GetCSAHeaderDict () const
- const DictEntry & GetDictEntry (const PrivateTag &tag) const
- const DictEntry & GetDictEntry (const Tag &tag, const char *owner=nullptr) const

  THREAD SAFE.
- PrivateDict & GetPrivateDict ()
- const PrivateDict & GetPrivateDict () const
- const Dict & GetPublicDict () const
- bool IsEmpty () const
- Dicts & operator= (const Dicts &_val)=delete

Protected Types

- enum ConstructorType {
  PHILIPS ,
  GEMS ,
  SIEMENS }

Protected Member Functions

- void LoadDefaults ()

Static Protected Member Functions

- static const char * GetConstructorString (ConstructorType type)

Friends

- class Global
- std::ostream & operator<< (std::ostream &_os, const Dicts &d)

10.89.1 Detailed Description

Class to manipulate the sum of knowledge (all the dict user load)

Note

bla

Examples

GenAllVR.cxx, GenFakeIdentifyFile.cxx, PublicDict.cxx, ReadAndPrintAttributes.cxx, and TraverseModules.cxx.

10.89.2 Member Enumeration Documentation

10.89.2.1 ConstructorType

```
enum gdcm::Dicts::ConstructorType  [protected]
```

Enumerator

PHILIPS	
GEMS	
SIEMENS	

10.89.3 Constructor & Destructor Documentation

10.89.3.1 Dicts() [1/2]

```
gdcm::Dicts::Dicts ( )
```

10.89.3.2 ~Dicts()

```
gdcm::Dicts::~Dicts ( )
```

10.89.3.3 Dicts() [2/2]

```
gdcm::Dicts::Dicts (
            const Dicts & _val )  [delete]
```

10.89.4 Member Function Documentation

10.89.4.1 GetConstructorString()

```
static const char * gdcm::Dicts::GetConstructorString (
            ConstructorType type )  [static], [protected]
```

10.89.4.2 GetCSAHeaderDict()

```
const CSAHeaderDict & gdcm::Dicts::GetCSAHeaderDict ( ) const
```

Examples

MrProtocol.cxx.

10.89.4.3 GetDictEntry() [1/2]

```
const DictEntry & gdcm::Dicts::GetDictEntry (
            const PrivateTag & tag ) const
```

10.89.4.4 GetDictEntry() [2/2]

```
const DictEntry & gdcm::Dicts::GetDictEntry (
            const Tag & tag,
            const char * owner = nullptr ) const
```

THREAD SAFE.

works for both public and private dicts: owner is null for public dict

Warning

owner need to be set to appropriate owner for call to work. see

Examples

PublicDict.cxx, and TraverseModules.cxx.

10.89.4.5 GetPrivateDict() [1/2]

```
PrivateDict & gdcm::Dicts::GetPrivateDict ( )
```

10.89.4.6 GetPrivateDict() [2/2]

```
const PrivateDict & gdcm::Dicts::GetPrivateDict ( ) const
```

10.89.4.7 GetPublicDict()

```
const Dict & gdcm::Dicts::GetPublicDict ( ) const
```

Examples

GenAllVR.cxx, GenFakeIdentifyFile.cxx, PublicDict.cxx, and ReadAndPrintAttributes.cxx.

10.89.4.8 IsEmpty()

```
bool gdcm::Dicts::IsEmpty ( ) const  [inline]
```

10.89.4.9 LoadDefaults()

```
void gdcm::Dicts::LoadDefaults ( )  [protected]
```

10.89.4.10 operator=()

```
Dicts & gdcm::Dicts::operator= (
            const Dicts & _val )  [delete]
```

10.89.5 Friends And Related Function Documentation

10.89.5.1 Global

```
friend class Global  [friend]
```

10.89.5.2 operator<<

```
std::ostream & operator<< (
            std::ostream & _os,
            const Dicts & d )  [friend]
```

The documentation for this class was generated from the following file:

- gdcmDicts.h

10.90 gdcm::network::DIMSE Class Reference

DIMSE.

```
#include <gdcmDIMSE.h>
```

Public Types

- enum CommandTypes {
  C_STORE_RQ = 0x0001 ,
  C_STORE_RSP = 0x8001 ,
  C_GET_RQ = 0x0010 ,
  C_GET_RSP = 0x8010 ,
  C_FIND_RQ = 0x0020 ,
  C_FIND_RSP = 0x8020 ,
  C_MOVE_RQ = 0x0021 ,
  C_MOVE_RSP = 0x8021 ,
  C_ECHO_RQ = 0x0030 ,
  C_ECHO_RSP = 0x8030 ,
  N_EVENT_REPORT_RQ = 0x0100 ,
  N_EVENT_REPORT_RSP = 0x8100 ,
  N_GET_RQ = 0x0110 ,
  N_GET_RSP = 0x8110 ,
  N_SET_RQ = 0x0120 ,
  N_SET_RSP = 0x8120 ,
  N_ACTION_RQ = 0x0130 ,
  N_ACTION_RSP = 0x8130 ,
  N_CREATE_RQ = 0x0140 ,
  N_CREATE_RSP = 0x8140 ,
  N_DELETE_RQ = 0x0150 ,
  N_DELETE_RSP = 0x8150 ,
  C_CANCEL_RQ = 0x0FFF }

10.90.1 Detailed Description

DIMSE.

PS 3.7 - 2009 Annex E Command Dictionary (Normative) E.1 REGISTRY OF DICOM COMMAND ELEMENTS Table E.1-1 COMMAND FIELDS (PART 1)

10.90.2 Member Enumeration Documentation

10.90.2.1 CommandTypes

enum gdcm::network::DIMSE::CommandTypes

Enumerator

C_STORE_RQ	
C_STORE_RSP	
C_GET_RQ	
C_GET_RSP	
C_FIND_RQ	
C_FIND_RSP	
C_MOVE_RQ	
C_MOVE_RSP	
C_ECHO_RQ	
C_ECHO_RSP	
N_EVENT_REPORT_RQ	
N_EVENT_REPORT_RSP	
N_GET_RQ	
N_GET_RSP	
N_SET_RQ	
N_SET_RSP	
N_ACTION_RQ	
N_ACTION_RSP	
N_CREATE_RQ	
N_CREATE_RSP	
N_DELETE_RQ	
N_DELETE_RSP	
C_CANCEL_RQ	

The documentation for this class was generated from the following file:

- gdcmDIMSE.h

10.91 gdcm::DirectionCosines Class Reference

class to handle DirectionCosines

`#include <gdcmDirectionCosines.h>`

Public Member Functions

- DirectionCosines ()
- DirectionCosines (const double dircos[6])
- ~DirectionCosines ()
- double ComputeDistAlongNormal (const double ipp[3]) const

  Compute the distance along the normal.
- void Cross (double z[3]) const

  Compute Cross product.
- double CrossDot (DirectionCosines const &dc) const

  Compute the Dot product of the two cross vector of both DirectionCosines object.
- double Dot () const

  Compute Dot.
- bool IsValid () const

  Return whether or not this is a valid direction cosines.
- void Normalize ()

  Normalize in-place.
- operator const double * () const

  *Make the class behave like a const double *.*
- void Print (std::ostream &) const

  Print.
- bool SetFromString (const char *str)

Static Public Member Functions

- static double Dot (const double x[3], const double y[3])

  Compute Dot.
- static void Normalize (double v[3])

  Normalize in-place.

10.91.1 Detailed Description

class to handle DirectionCosines

Examples

DiscriminateVolume.cxx.

10.91.2 Constructor & Destructor Documentation

10.91.2.1 DirectionCosines() [1/2]

```
gdcm::DirectionCosines::DirectionCosines ( )
```

10.91.2.2 DirectionCosines() [2/2]

```
gdcm::DirectionCosines::DirectionCosines (
            const double dircos[6] )
```

10.91.2.3 ~DirectionCosines()

```
gdcm::DirectionCosines::~DirectionCosines ( )
```

10.91.3 Member Function Documentation

10.91.3.1 ComputeDistAlongNormal()

```
double gdcm::DirectionCosines::ComputeDistAlongNormal (
            const double ipp[3] ) const
```

Compute the distance along the normal.

10.91.3.2 Cross()

```
void gdcm::DirectionCosines::Cross (
            double z[3] ) const
```

Compute Cross product.

10.91.3.3 CrossDot()

```
double gdcm::DirectionCosines::CrossDot (
            DirectionCosines const & dc ) const
```

Compute the Dot product of the two cross vector of both DirectionCosines object.

Examples

DiscriminateVolume.cxx.

10.91.3.4 Dot() [1/2]

```
double gdcm::DirectionCosines::Dot ( ) const
```

Compute Dot.

10.91.3.5 Dot() [2/2]

```
static double gdcm::DirectionCosines::Dot (
            const double x[3],
            const double y[3] )  [static]
```

Compute Dot.

10.91.3.6 IsValid()

```
bool gdcm::DirectionCosines::IsValid ( ) const
```

Return whether or not this is a valid direction cosines.

10.91.3.7 Normalize() [1/2]

```
void gdcm::DirectionCosines::Normalize ( )
```

Normalize in-place.

10.91.3.8 Normalize() [2/2]

```
static void gdcm::DirectionCosines::Normalize (
            double v[3] )  [static]
```

Normalize in-place.

10.91.3.9 operator const double *()

```
gdcm::DirectionCosines::operator const double * ( ) const  [inline]
```

Make the class behave like a const double *.

10.91.3.10 Print()

```
void gdcm::DirectionCosines::Print (
            std::ostream &  ) const
```

Print.

10.91.3.11 SetFromString()

```
bool gdcm::DirectionCosines::SetFromString (
            const char * str )
```

Initialize from string str. It requires 6 floating point separated by a backslash character.

Examples

DiscriminateVolume.cxx.

The documentation for this class was generated from the following file:

- gdcmDirectionCosines.h

10.92 gdcm::Directory Class Reference

Class for manipulation directories.

```
#include <gdcmDirectory.h>
```

Public Types

- typedef std::vector< FilenameType > FilenamesType
- typedef std::string FilenameType

Public Member Functions

- Directory ()=default
- ~Directory ()=default
- FilenamesType const & GetDirectories () const

  Return the Directories traversed.
- FilenamesType const & GetFilenames () const

  Set/Get the file names within the directory.
- FilenameType const & GetToplevel () const

  Get the name of the toplevel directory.
- unsigned int Load (FilenameType const &name, bool recursive=false)
- void Print (std::ostream &os=std::cout) const

  Print.

Protected Member Functions

- unsigned int Explore (FilenameType const &name, bool recursive)

  Return number of file found when 'recursive'ly exploring directory `name`

Friends

- std::ostream & operator<< (std::ostream &_os, const Directory &d)

10.92.1 Detailed Description

Class for manipulation directories.

Note

This implementation provide a cross platform implementation for manipulating directories: basically traversing directories and harvesting files

will not take into account unix type hidden file recursive option will not look into UNIX type hidden directory (those starting with a '.')

Since python or C# provide there own equivalent implementation, in which case gdcm::Directory does not make much sense.

Examples

ClinicalTrialIdentificationWorkflow.cs, DecompressImageMultiframe.cs, DiscriminateVolume.cxx, DumpToSQLITE3.cxx, DumpVisusChange.cxx, GenerateDICOMDIR.cs, GenerateRTSTRUCT.cxx, ReadUTF8QtDir.cxx, ScanDirectory.cs, SortImage.cxx, StandardizeFiles.cs, VolumeSorter.cxx, gdcmorthoplanes.cxx, reslicesphere.cxx, and threadgdcm.cxx.

10.92.2 Member Typedef Documentation

10.92.2.1 FilenamesType

```
typedef std::vector<FilenameType> gdcm::Directory::FilenamesType
```

Examples

DiscriminateVolume.cxx.

10.92.2.2 FilenameType

```
typedef std::string gdcm::Directory::FilenameType
```

10.92.3 Constructor & Destructor Documentation

10.92.3.1 Directory()

```
gdcm::Directory::Directory ( )  [default]
```

10.92.3.2 ~Directory()

```
gdcm::Directory::~Directory ( )  [default]
```

10.92.4 Member Function Documentation

10.92.4.1 Explore()

```
unsigned int gdcm::Directory::Explore (
            FilenameType const & name,
            bool recursive )  [protected]
```

Return number of file found when 'recursive'ly exploring directory `name`

10.92.4.2 GetDirectories()

```
FilenamesType const & gdcm::Directory::GetDirectories ( ) const  [inline]
```

Return the Directories traversed.

10.92.4.3 GetFilenames()

```
FilenamesType const & gdcm::Directory::GetFilenames ( ) const  [inline]
```

Set/Get the file names within the directory.

Examples

ClinicalTrialIdentificationWorkflow.cs, DecompressImageMultiframe.cs, DiscriminateVolume.cxx, DumpToSQLITE3.cxx, DumpVisusChange.cxx, GenerateDICOMDIR.cs, GenerateRTSTRUCT.cxx, ReadUTF8QtDir.cxx, ScanDirectory.cs, SortImage.cxx, StandardizeFiles.cs, VolumeSorter.cxx, gdcmorthoplanes.cxx, reslicesphere.cxx, and threadgdcm.cxx.

10.92.4.4 GetToplevel()

```
FilenameType const & gdcm::Directory::GetToplevel ( ) const  [inline]
```

Get the name of the toplevel directory.

10.92.4.5 Load()

```
unsigned int gdcm::Directory::Load (
            FilenameType const & name,
            bool recursive = false )
```

construct a list of filenames and subdirectory beneath directory: name

Warning

: hidden file and hidden directory are not loaded.

Examples

ClinicalTrialIdentificationWorkflow.cs, DecompressImageMultiframe.cs, DiscriminateVolume.cxx, DumpToSQLITE3.cxx, DumpVisusChange.cxx, GenerateDICOMDIR.cs, GenerateRTSTRUCT.cxx, ReadUTF8QtDir.cxx, ScanDirectory.cs, SortImage.cxx, StandardizeFiles.cs, VolumeSorter.cxx, gdcmorthoplanes.cxx, reslicesphere.cxx, and threadgdcm.cxx.

10.92.4.6 Print()

```
void gdcm::Directory::Print (
            std::ostream & os = std::cout ) const
```

Print.

Examples

SortImage.cxx.

10.92.5 Friends And Related Function Documentation

10.92.5.1 operator<<

```
std::ostream & operator<< (
            std::ostream & _os,
            const Directory & d )  [friend]
```

The documentation for this class was generated from the following file:

- gdcmDirectory.h

10.93 gdcm::DirectoryHelper Class Reference

DirectoryHelper.

```
#include <gdcmDirectoryHelper.h>
```

Static Public Member Functions

- static Directory::FilenamesType GetCTImageSeriesUIDs (const std::string &inDirectory)
- static Directory::FilenamesType GetFilenamesFromSeriesUIDs (const std::string &inDirectory, const std::string &inSeriesUID)
- static std::string GetFrameOfReference (const std::vector< DataSet > &inDS)
- static Directory::FilenamesType GetMRImageSeriesUIDs (const std::string &inDirectory)
- static Directory::FilenamesType GetRTStructSeriesUIDs (const std::string &inDirectory)
- static Directory::FilenamesType GetSeriesUIDsBySOPClassUID (const std::string &inDirectory, const std::string &inSOPClassUID)
- static std::string GetSOPClassUID (const std::vector< DataSet > &inDS)
- static std::string GetStringValueFromTag (const Tag &t, const DataSet &ds)
- static std::vector< DataSet > LoadImageFromFiles (const std::string &inDirectory, const std::string &inSeriesUID)
- static std::string RetrieveSOPInstanceUIDFromIndex (int inIndex, const std::vector< DataSet > &inDS)
- static std::string RetrieveSOPInstanceUIDFromZPosition (double inZPos, const std::vector< DataSet > &inDS)

10.93.1 Detailed Description

DirectoryHelper.

this class is designed to help mitigate some of the commonly performed operations on directories. namely: 1) the ability to determine the number of series in a directory by what type of series is present 2) the ability to find all ct series in a directory 3) the ability to find all mr series in a directory 4) to load a set of DataSets from a series that's already been sorted by the IPP sorter 5) For rtstruct stuff, you need to know the sopinstanceuid of each z plane, so there's a retrieval function for that 6) then a few other functions for rtstruct writeouts

10.93.2 Member Function Documentation

10.93.2.1 GetCTImageSeriesUIDs()

```
static Directory::FilenamesType gdcm::DirectoryHelper::GetCTImageSeriesUIDs (
            const std::string & inDirectory )  [static]
```

10.93.2.2 GetFilenamesFromSeriesUIDs()

```
static Directory::FilenamesType gdcm::DirectoryHelper::GetFilenamesFromSeriesUIDs (
            const std::string & inDirectory,
            const std::string & inSeriesUID )  [static]
```

10.93.2.3 GetFrameOfReference()

```
static std::string gdcm::DirectoryHelper::GetFrameOfReference (
            const std::vector< DataSet > & inDS )  [static]
```

10.93.2.4 GetMRImageSeriesUIDs()

```
static Directory::FilenamesType gdcm::DirectoryHelper::GetMRImageSeriesUIDs (
            const std::string & inDirectory )  [static]
```

10.93.2.5 GetRTStructSeriesUIDs()

```
static Directory::FilenamesType gdcm::DirectoryHelper::GetRTStructSeriesUIDs (
            const std::string & inDirectory )  [static]
```

10.93.2.6 GetSeriesUIDsBySOPClassUID()

```
static Directory::FilenamesType gdcm::DirectoryHelper::GetSeriesUIDsBySOPClassUID (
            const std::string & inDirectory,
            const std::string & inSOPClassUID )  [static]
```

10.93.2.7 GetSOPClassUID()

```
static std::string gdcm::DirectoryHelper::GetSOPClassUID (
            const std::vector< DataSet > & inDS )  [static]
```

10.93.2.8 GetStringValueFromTag()

```
static std::string gdcm::DirectoryHelper::GetStringValueFromTag (
            const Tag & t,
            const DataSet & ds )  [static]
```

10.93.2.9 LoadImageFromFiles()

```
static std::vector< DataSet > gdcm::DirectoryHelper::LoadImageFromFiles (
            const std::string & inDirectory,
            const std::string & inSeriesUID )  [static]
```

10.93.2.10 RetrieveSOPInstanceUIDFromIndex()

```
static std::string gdcm::DirectoryHelper::RetrieveSOPInstanceUIDFromIndex (
            int inIndex,
            const std::vector< DataSet > & inDS )  [static]
```

10.93.2.11 RetrieveSOPInstanceUIDFromZPosition()

```
static std::string gdcm::DirectoryHelper::RetrieveSOPInstanceUIDFromZPosition (
            double inZPos,
            const std::vector< DataSet > & inDS )  [static]
```

The documentation for this class was generated from the following file:

- gdcmDirectoryHelper.h

10.94 gdcm::DPath Class Reference

class to handle a DICOM path While supp 118 did introduced a notion of XPath for XML Native model this convention is too XML-centric. Instead prefer DCMTK style notation https://groups.google.com/g/comp.↩protocols.dicom/c/IyIH0IOBMPA

```
#include <gdcmDPath.h>
```

Public Member Functions

- DPath ()
- ~DPath ()
- bool ConstructFromString (const char *path)
- bool Match (DPath const &other) const

  Return whether or not 'other' match the template DPath.
- bool operator< (const DPath &rhs) const
- void Print (std::ostream &) const

Static Public Member Functions

- static bool IsValid (const char *path)

  Return if path is valid or not.

Friends

- std::ostream & operator<< (std::ostream &_os, const DPath &_val)

10.94.1 Detailed Description

class to handle a DICOM path While supp 118 did introduced a notion of XPath for XML Native model this convention is too XML-centric. Instead prefer DCMTK style notation `https://groups.google.com/g/comp.protocols.dicom/c/IyIH0IOBMPA`

10.94.2 Constructor & Destructor Documentation

10.94.2.1 DPath()

```
gdcm::DPath::DPath ( )
```

10.94.2.2 ~DPath()

```
gdcm::DPath::~DPath ( )
```

10.94.3 Member Function Documentation

10.94.3.1 ConstructFromString()

```
bool gdcm::DPath::ConstructFromString (
            const char * path )
```

10.94.3.2 IsValid()

```
static bool gdcm::DPath::IsValid (
            const char * path )  [static]
```

Return if path is valid or not.

10.94.3.3 Match()

```
bool gdcm::DPath::Match (
            DPath const & other ) const
```

Return whether or not 'other' match the template DPath.

10.94.3.4 operator<()

```
bool gdcm::DPath::operator< (
            const DPath & rhs ) const
```

10.94.3.5 Print()

```
void gdcm::DPath::Print (
            std::ostream &  ) const
```

10.94.4 Friends And Related Function Documentation

10.94.4.1 operator<<

```
std::ostream & operator<< (
            std::ostream & _os,
            const DPath & _val )  [friend]
```

The documentation for this class was generated from the following file:

- gdcmDPath.h

10.95 gdcm::DummyValueGenerator Class Reference

Class for generating dummy value.

`#include <gdcmDummyValueGenerator.h>`

Static Public Member Functions

- static const char * Generate (const char *input)

10.95.1 Detailed Description

Class for generating dummy value.

See also

Anonymizer

10.95.2 Member Function Documentation

10.95.2.1 Generate()

```
static const char * gdcm::DummyValueGenerator::Generate (
            const char * input )  [static]
```

Generate a dummy value from an input value. This is guarantee to always return the same output value when input is identical. Return an array of bytes that can be used for anonymization purpose, return NULL on error NOT THREAD SAFE

The documentation for this class was generated from the following file:

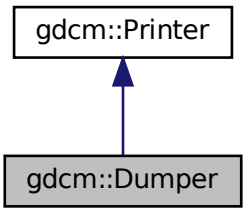
- gdcmDummyValueGenerator.h

10.96 gdcm::Dumper Class Reference

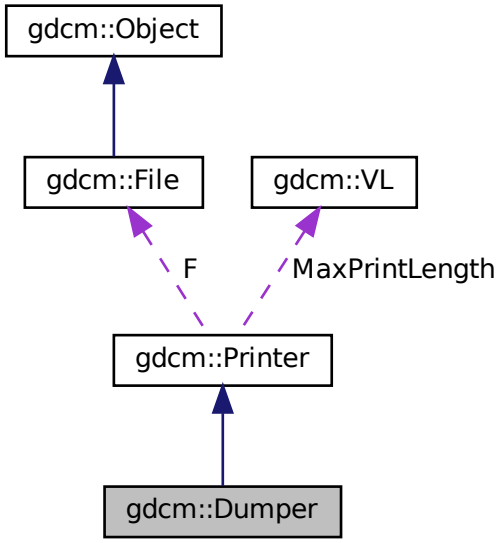
Codec class.

`#include <gdcmDumper.h>`

Inheritance diagram for gdcm::Dumper:



Collaboration diagram for gdcm::Dumper:



Public Member Functions

- Dumper ()
- ~Dumper ()=default

Additional Inherited Members

10.96.1 Detailed Description

Codec class.

Note

Use it to simply dump value read from the file. No interpretation is done. But it is real fast ! Almost no overhead

10.96.2 Constructor & Destructor Documentation

10.96.2.1 Dumper()

```
gdcm::Dumper::Dumper ( )  [inline]
```

10.96.2.2 ~Dumper()

```
gdcm::Dumper::~Dumper ( )  [default]
```

The documentation for this class was generated from the following file:

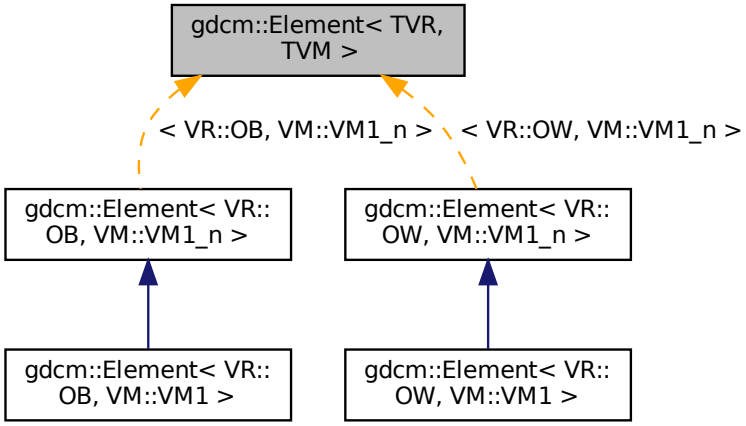
- gdcmDumper.h

10.97 gdcm::Element< TVR, TVM > Class Template Reference

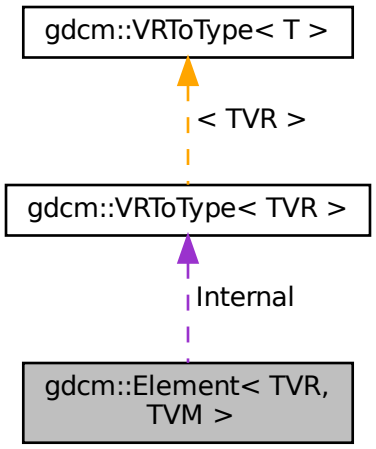
Element class.

```
#include <gdcmElement.h>
```

Inheritance diagram for gdcm::Element< TVR, TVM >:



Collaboration diagram for gdcm::Element< TVR, TVM >:



Public Types

- typedef VRToType< TVR >::Type Type

Public Member Functions

- DataElement GetAsDataElement () const
- unsigned long GetLength () const
- VRToType< TVR >::Type & GetValue (unsigned int idx=0)
- const VRToType< TVR >::Type & GetValue (unsigned int idx=0) const
- const VRToType< TVR >::Type * GetValues () const
- VRToType< TVR >::Type operator[] (unsigned int idx) const
- void Print (std::ostream &_os) const
- void Read (std::istream &_is)
- void Set (Value const &v)
- void SetFromDataElement (DataElement const &de)
- void SetValue (typename VRToType< TVR >::Type v, unsigned int idx=0)
- void Write (std::ostream &_os) const

Static Public Member Functions

- static VM GetVM ()
- static VR GetVR ()

Public Attributes

- VRToType< TVR >::Type Internal [VMToLength< TVM >::Length]

Protected Member Functions

- void SetNoSwap (Value const &v)

10.97.1 Detailed Description

template<long long TVR, int TVM>
class gdcm::Element< TVR, TVM >

Element class.

Note

TODO

Examples

DumpADAC.cxx, DumpExamCard.cxx, DumpGEMSMovieGroup.cxx, DumpPhilipsECHO.cxx, Extracting_All_Resolution.cxx, Fake_Image_Using_Stream_Image_Writer.cxx, GetSubSequenceData.cxx, csa2img.cxx, and iU22tomultisc.cxx.

10.97.2 Member Typedef Documentation

10.97.2.1 Type

```
template<long long TVR, int TVM>
typedef VRToType<TVR>::Type gdcm::Element< TVR, TVM >::Type
```

10.97.3 Member Function Documentation

10.97.3.1 GetAsDataElement()

```
template<long long TVR, int TVM>
DataElement gdcm::Element< TVR, TVM >::GetAsDataElement ( ) const  [inline]
```

Examples

Extracting_All_Resolution.cxx, and Fake_Image_Using_Stream_Image_Writer.cxx.

References gdcm::DataElement::GetVR(), gdcm::DataElement::SetByteValue(), and gdcm::DataElement::SetVR().

10.97.3.2 GetLength()

```
template<long long TVR, int TVM>
unsigned long gdcm::Element< TVR, TVM >::GetLength ( ) const  [inline]
```

Examples

DumpGEMSMovieGroup.cxx.

10.97.3.3 GetValue() [1/2]

```
template<long long TVR, int TVM>
VRToType< TVR >::Type & gdcm::Element< TVR, TVM >::GetValue (
            unsigned int idx = 0 )  [inline]
```

10.97.3.4 GetValue() [2/2]

```
template<long long TVR, int TVM>
const VRToType< TVR >::Type & gdcm::Element< TVR, TVM >::GetValue (
            unsigned int idx = 0 ) const  [inline]
```

Examples

DumpADAC.cxx, DumpExamCard.cxx, DumpGEMSMovieGroup.cxx, DumpPhilipsECHO.cxx, GetSubSequenceData.cxx, and csa2img.cxx.

10.97.3.5 GetValues()

```
template<long long TVR, int TVM>
const VRToType< TVR >::Type * gdcm::Element< TVR, TVM >::GetValues ( ) const  [inline]
```

10.97.3.6 GetVM()

```
template<long long TVR, int TVM>
static VM gdcm::Element< TVR, TVM >::GetVM ( )  [inline], [static]
```

10.97.3.7 GetVR()

```
template<long long TVR, int TVM>
static VR gdcm::Element< TVR, TVM >::GetVR ( )  [inline], [static]
```

10.97.3.8 operator[]()

```
template<long long TVR, int TVM>
VRToType< TVR >::Type gdcm::Element< TVR, TVM >::operator[] (
            unsigned int idx ) const  [inline]
```

10.97.3.9 Print()

```
template<long long TVR, int TVM>
void gdcm::Element< TVR, TVM >::Print (
            std::ostream & _os ) const  [inline]
```

Examples

DumpGEMSMovieGroup.cxx.

10.97.3.10 Read()

```
template<long long TVR, int TVM>
void gdcm::Element< TVR, TVM >::Read (
            std::istream & _is )  [inline]
```

10.97.3.11 Set()

```
template<long long TVR, int TVM>
void gdcm::Element< TVR, TVM >::Set (
            Value const & v )  [inline]
```

Examples

csa2img.cxx.

References gdcm::ByteValue::GetLength(), and gdcm::ByteValue::GetPointer().

10.97.3.12 SetFromDataElement()

```
template<long long TVR, int TVM>
void gdcm::Element< TVR, TVM >::SetFromDataElement (
            DataElement< TVR, TVM > const & de )  [inline]
```

Examples

DumpADAC.cxx, DumpExamCard.cxx, DumpGEMSMovieGroup.cxx, DumpPhilipsECHO.cxx, GetSubSequenceData.cxx, and iU22tomultisc.cxx.

References gdcm::DataElement::GetByteValue(), gdcm::DataElement::GetValue(), and gdcm::DataElement::GetVR().

10.97.3.13 SetNoSwap()

```
template<long long TVR, int TVM>
void gdcm::Element< TVR, TVM >::SetNoSwap (
             Value const & v )  [inline], [protected]
```

References gdcm::ByteValue::GetLength(), and gdcm::ByteValue::GetPointer().

10.97.3.14 SetValue()

```
template<long long TVR, int TVM>
void gdcm::Element< TVR, TVM >::SetValue (
             typename VRToType< TVR >::Type v,
             unsigned int idx = 0 )  [inline]
```

Examples

Extracting_All_Resolution.cxx, and Fake_Image_Using_Stream_Image_Writer.cxx.

10.97.3.15 Write()

```
template<long long TVR, int TVM>
void gdcm::Element< TVR, TVM >::Write (
             std::ostream & _os ) const  [inline]
```

10.97.4 Member Data Documentation

10.97.4.1 Internal

```
template<long long TVR, int TVM>
VRToType<TVR>::Type gdcm::Element< TVR, TVM >::Internal[VMToLength< TVM >::Length]
```

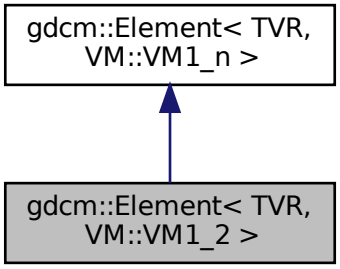
The documentation for this class was generated from the following file:

- gdcmElement.h

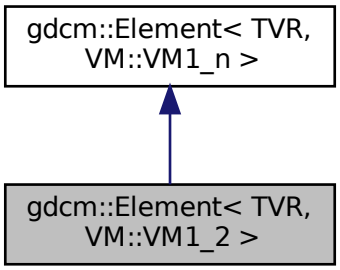
10.98 gdcm::Element< TVR, VM::VM1_2 > Class Template Reference

`#include <gdcmElement.h>`

Inheritance diagram for gdcm::Element< TVR, VM::VM1_2 >:



Collaboration diagram for gdcm::Element< TVR, VM::VM1_2 >:



Public Types

- typedef Element< TVR, VM::VM1_n > Parent

Public Member Functions

- void SetLength (int len)

Additional Inherited Members

10.98.1 Member Typedef Documentation

10.98.1.1 Parent

```
template<long long TVR>
typedef Element<TVR, VM::VM1_n> gdcm::Element< TVR, VM::VM1_2 >::Parent
```

10.98.2 Member Function Documentation

10.98.2.1 SetLength()

```
template<long long TVR>
void gdcm::Element< TVR, VM::VM1_2 >::SetLength (
            int len )  [inline]
```

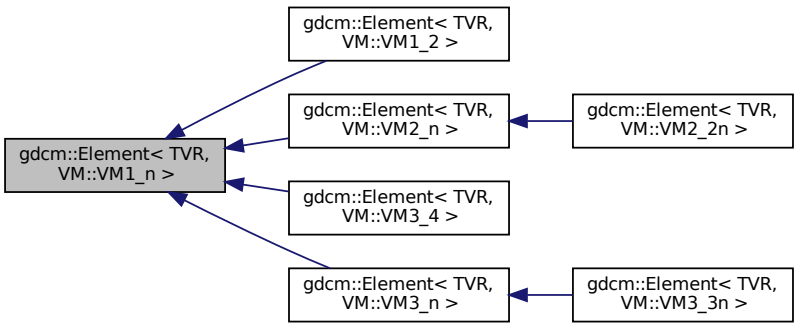
The documentation for this class was generated from the following file:

- gdcmElement.h

10.99 gdcm::Element< TVR, VM::VM1_n > Class Template Reference

```
#include <gdcmElement.h>
```

Inheritance diagram for gdcm::Element< TVR, VM::VM1_n >:



Public Types

- typedef VRToType< TVR >::Type Type

Public Member Functions

- Element ()
- Element (const Element &_val)
- ~Element ()
- DataElement GetAsDataElement () const
- unsigned long GetLength () const
- VRToType< TVR >::Type & GetValue (unsigned int idx=0)
- const VRToType< TVR >::Type & GetValue (unsigned int idx=0) const
- Element & operator= (const Element &_val)
- VRToType< TVR >::Type operator[ ] (unsigned int idx) const
- void Print (std::ostream &_os) const
- void Read (std::istream &_is)
- void Set (Value const &v)
- void SetArray (const Type *array, unsigned long len, bool save=false)
- void SetFromDataElement (DataElement const &de)
- void SetLength (unsigned long len)
- void SetValue (typename VRToType< TVR >::Type v, unsigned int idx=0)
- void Write (std::ostream &_os) const
- void WriteASCII (std::ostream &os) const

Static Public Member Functions

- static VM GetVM ()
- static VR GetVR ()

Protected Member Functions

- void SetNoSwap (Value const &v)

10.99.1 Member Typedef Documentation

10.99.1.1 Type

```
template<long long TVR>
typedef VRToType<TVR>::Type gdcm::Element< TVR, VM::VM1_n >::Type
```

10.99.2 Constructor & Destructor Documentation

10.99.2.1 Element() [1/2]

```
template<long long TVR>
gdcm::Element< TVR, VM::VM1_n >::Element ( )  [inline], [explicit]
```

10.99.2.2 ~Element()

```
template<long long TVR>
gdcm::Element< TVR, VM::VM1_n >::~Element ( )  [inline]
```

10.99.2.3 Element() [2/2]

```
template<long long TVR>
gdcm::Element< TVR, VM::VM1_n >::Element (
            const Element< TVR, VM::VM1_n > & _val )  [inline]
```

10.99.3 Member Function Documentation

10.99.3.1 GetAsDataElement()

```
template<long long TVR>
DataElement gdcm::Element< TVR, VM::VM1_n >::GetAsDataElement ( ) const  [inline]
```

References gdcm::DataElement::GetVR(), gdcm::DataElement::SetByteValue(), and gdcm::DataElement::SetVR().

10.99.3.2 GetLength()

```
template<long long TVR>
unsigned long gdcm::Element< TVR, VM::VM1_n >::GetLength ( ) const  [inline]
```

10.99.3.3 GetValue() [1/2]

```
template<long long TVR>
VRToType< TVR >::Type & gdcm::Element< TVR, VM::VM1_n >::GetValue (
            unsigned int idx = 0 )  [inline]
```

10.99.3.4 GetValue() [2/2]

```
template<long long TVR>
const VRToType< TVR >::Type & gdcm::Element< TVR, VM::VM1_n >::GetValue (
            unsigned int idx = 0 ) const  [inline]
```

10.99.3.5 GetVM()

```
template<long long TVR>
static VM gdcm::Element< TVR, VM::VM1_n >::GetVM ( )  [inline], [static]
```

10.99.3.6 GetVR()

```
template<long long TVR>
static VR gdcm::Element< TVR, VM::VM1_n >::GetVR ( )  [inline], [static]
```

10.99.3.7 operator=()

```
template<long long TVR>
Element & gdcm::Element< TVR, VM::VM1_n >::operator= (
            const Element< TVR, VM::VM1_n > & _val )  [inline]
```

10.99.3.8 operator[]()

```
template<long long TVR>
VRToType< TVR >::Type gdcm::Element< TVR, VM::VM1_n >::operator[] (
            unsigned int idx ) const  [inline]
```

10.99.3.9 Print()

```
template<long long TVR>
void gdcm::Element< TVR, VM::VM1_n >::Print (
            std::ostream & _os ) const  [inline]
```

10.99.3.10 Read()

```
template<long long TVR>
void gdcm::Element< TVR, VM::VM1_n >::Read (
            std::istream & _is )  [inline]
```

10.99.3.11 Set()

```
template<long long TVR>
void gdcm::Element< TVR, VM::VM1_n >::Set (
            Value const & v )  [inline]
```

References gdcm::ByteValue::GetLength(), gdcm::ByteValue::GetPointer(), gdcm::ByteValue::GetVoidPointer(), and gdcm::VRBINARY.

10.99.3.12 SetArray()

```
template<long long TVR>
void gdcm::Element< TVR, VM::VM1_n >::SetArray (
            const Type * array,
            unsigned long len,
            bool save = false )  [inline]
```

10.99.3.13 SetFromDataElement()

```
template<long long TVR>
void gdcm::Element< TVR, VM::VM1_n >::SetFromDataElement (
            DataElement< TVR, VM::VM1_n > const & de )  [inline]
```

References gdcm::DataElement::GetByteValue(), gdcm::DataElement::GetValue(), and gdcm::DataElement::GetVR().

10.99.3.14 SetLength()

```
template<long long TVR>
void gdcm::Element< TVR, VM::VM1_n >::SetLength (
            unsigned long len )  [inline]
```

10.99.3.15 SetNoSwap()

```
template<long long TVR>
void gdcm::Element< TVR, VM::VM1_n >::SetNoSwap (
            Value const & v )  [inline], [protected]
```

References gdcm::ByteValue::GetLength(), gdcm::ByteValue::GetPointer(), and gdcm::VRBINARY.

10.99.3.16 SetValue()

```
template<long long TVR>
void gdcm::Element< TVR, VM::VM1_n >::SetValue (
            typename VRToType< TVR >::Type v,
            unsigned int idx = 0 )  [inline]
```

10.99.3.17 Write()

```
template<long long TVR>
void gdcm::Element< TVR, VM::VM1_n >::Write (
            std::ostream & _os ) const  [inline]
```

10.99.3.18 WriteASCII()

```
template<long long TVR>
void gdcm::Element< TVR, VM::VM1_n >::WriteASCII (
            std::ostream & os ) const  [inline]
```

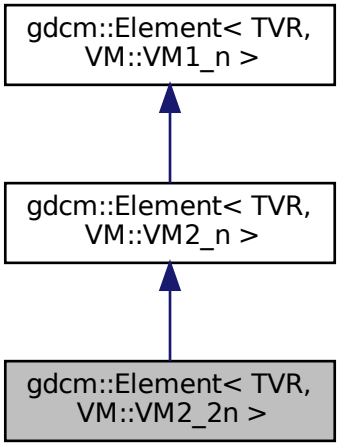
The documentation for this class was generated from the following file:

- gdcmElement.h

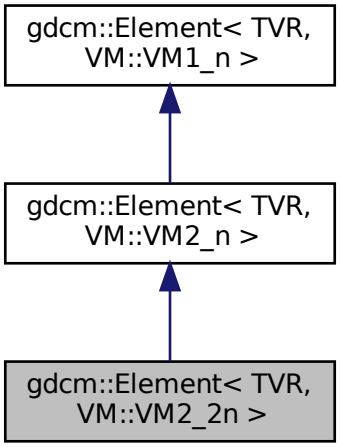
10.100 gdcm::Element< TVR, VM::VM2_2n > Class Template Reference

`#include <gdcmElement.h>`

Inheritance diagram for gdcm::Element< TVR, VM::VM2_2n >:



Collaboration diagram for gdcm::Element< TVR, VM::VM2_2n >:



Public Types

- typedef Element< TVR, VM::VM2_n > Parent

Public Member Functions

- void SetLength (int len)

Additional Inherited Members

10.100.1 Member Typedef Documentation

10.100.1.1 Parent

```
template<long long TVR>
typedef Element<TVR, VM::VM2_n> gdcm::Element< TVR, VM::VM2_2n >::Parent
```

10.100.2 Member Function Documentation

10.100.2.1 SetLength()

```
template<long long TVR>
void gdcm::Element< TVR, VM::VM2_2n >::SetLength (
            int len )  [inline]
```

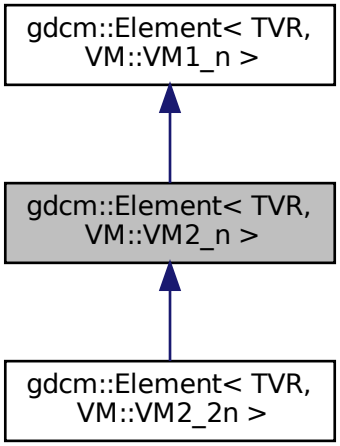
The documentation for this class was generated from the following file:

- gdcmElement.h

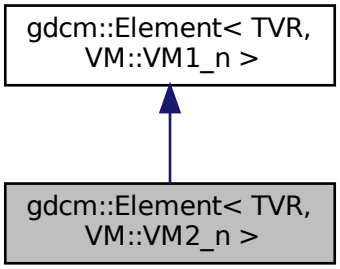
10.101 gdcm::Element< TVR, VM::VM2_n > Class Template Reference

`#include <gdcmElement.h>`

Inheritance diagram for gdcm::Element< TVR, VM::VM2_n >:



Collaboration diagram for gdcm::Element< TVR, VM::VM2_n >:



Public Types

- typedef Element< TVR, VM::VM1_n > Parent

Public Member Functions

- void SetLength (int len)

Additional Inherited Members

10.101.1 Member Typedef Documentation

10.101.1.1 Parent

```
template<long long TVR>
typedef Element<TVR, VM::VM1_n> gdcm::Element< TVR, VM::VM2_n >::Parent
```

10.101.2 Member Function Documentation

10.101.2.1 SetLength()

```
template<long long TVR>
void gdcm::Element< TVR, VM::VM2_n >::SetLength (
            int len )  [inline]
```

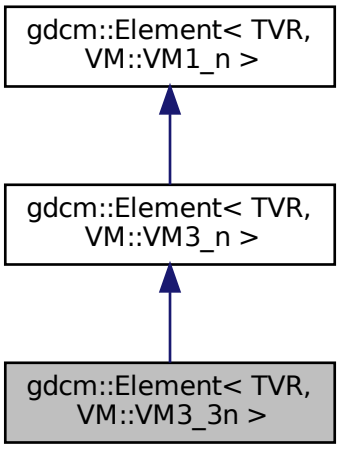
The documentation for this class was generated from the following file:

- gdcmElement.h

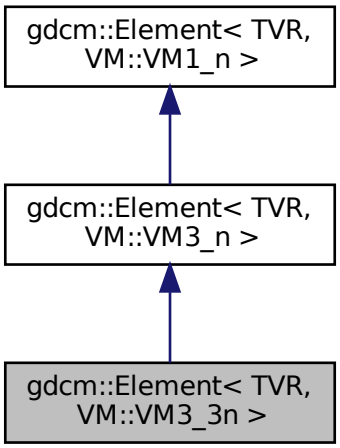
10.102 gdcm::Element< TVR, VM::VM3_3n > Class Template Reference

```
#include <gdcmElement.h>
```

Inheritance diagram for gdcm::Element< TVR, VM::VM3_3n >:



Collaboration diagram for gdcm::Element< TVR, VM::VM3_3n >:



Public Types

- typedef Element< TVR, VM::VM3_n > Parent

Public Member Functions

- void SetLength (int len)

Additional Inherited Members

10.102.1 Member Typedef Documentation

10.102.1.1 Parent

```
template<long long TVR>
typedef Element<TVR, VM::VM3_n> gdcm::Element< TVR, VM::VM3_3n >::Parent
```

10.102.2 Member Function Documentation

10.102.2.1 SetLength()

```
template<long long TVR>
void gdcm::Element< TVR, VM::VM3_3n >::SetLength (
            int len )  [inline]
```

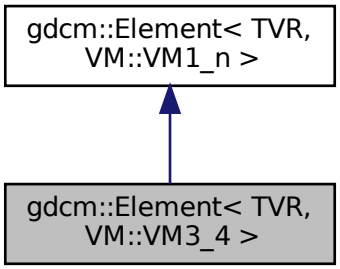
The documentation for this class was generated from the following file:

- gdcmElement.h

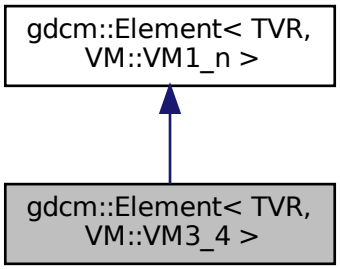
10.103 gdcm::Element< TVR, VM::VM3_4 > Class Template Reference

```
#include <gdcmElement.h>
```

Inheritance diagram for gdcm::Element< TVR, VM::VM3_4 >:



Collaboration diagram for gdcm::Element< TVR, VM::VM3_4 >:



Public Types

- typedef Element< TVR, VM::VM1_n > Parent

Public Member Functions

- void SetLength (int len)

Additional Inherited Members

10.103.1 Member Typedef Documentation

10.103.1.1 Parent

```
template<long long TVR>
typedef Element<TVR, VM::VM1_n> gdcm::Element< TVR, VM::VM3_4 >::Parent
```

10.103.2 Member Function Documentation

10.103.2.1 SetLength()

```
template<long long TVR>
void gdcm::Element< TVR, VM::VM3_4 >::SetLength (
            int len )  [inline]
```

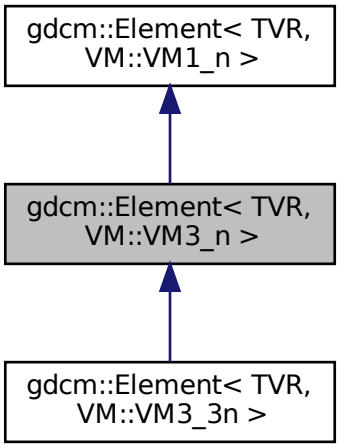
The documentation for this class was generated from the following file:

- gdcmElement.h

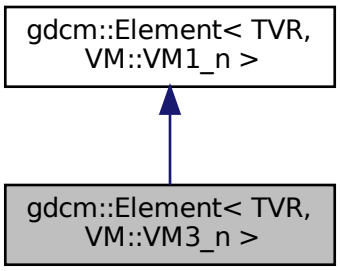
10.104 gdcm::Element< TVR, VM::VM3_n > Class Template Reference

```
#include <gdcmElement.h>
```

Inheritance diagram for gdcm::Element< TVR, VM::VM3_n >:



Collaboration diagram for gdcm::Element< TVR, VM::VM3_n >:



Public Types

- typedef Element< TVR, VM::VM1_n > Parent

Public Member Functions

- void SetLength (int len)

Additional Inherited Members

10.104.1 Member Typedef Documentation

10.104.1.1 Parent

```
template<long long TVR>
typedef Element<TVR, VM::VM1_n> gdcm::Element< TVR, VM::VM3_n >::Parent
```

10.104.2 Member Function Documentation

10.104.2.1 SetLength()

```
template<long long TVR>
void gdcm::Element< TVR, VM::VM3_n >::SetLength (
            int len )  [inline]
```

The documentation for this class was generated from the following file:

- gdcmElement.h

10.105 gdcm::Element< VR::AS, VM::VM5 > Class Reference

```
#include <gdcmElement.h>
```

Public Member Functions

- unsigned long GetLength () const
- void Print (std::ostream &_os) const

Public Attributes

- char Internal [VMToLength< VM::VM5 >::Length *sizeof(VRToType< VR::AS >::Type)]

10.105.1 Member Function Documentation

10.105.1.1 GetLength()

```
unsigned long gdcm::Element< VR::AS, VM::VM5 >::GetLength ( ) const  [inline]
```

10.105.1.2 Print()

```
void gdcm::Element< VR::AS, VM::VM5 >::Print (
            std::ostream & _os ) const  [inline]
```

10.105.2 Member Data Documentation

10.105.2.1 Internal

```
char gdcm::Element< VR::AS, VM::VM5 >::Internal[VMToLength< VM::VM5 >::Length *sizeof(VRToType< VR::AS >::Type)]
```

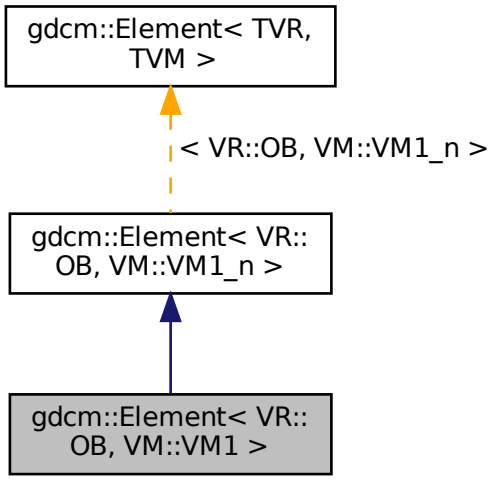
The documentation for this class was generated from the following file:

- gdcmElement.h

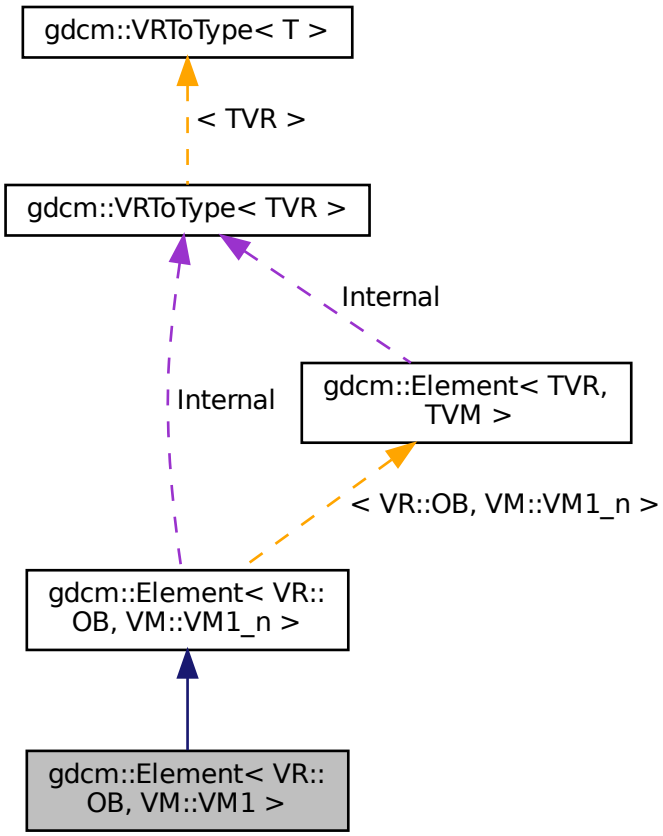
10.106 gdcm::Element< VR::OB, VM::VM1 > Class Reference

```
#include <gdcmElement.h>
```

Inheritance diagram for gdcm::Element< VR::OB, VM::VM1 >:



Collaboration diagram for gdcm::Element< VR::OB, VM::VM1 >:



Additional Inherited Members

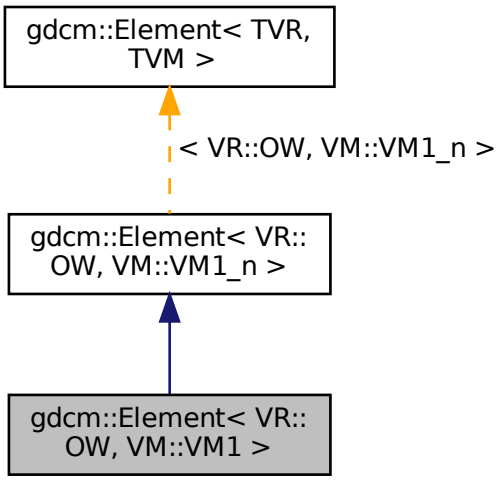
The documentation for this class was generated from the following file:

- gdcmElement.h

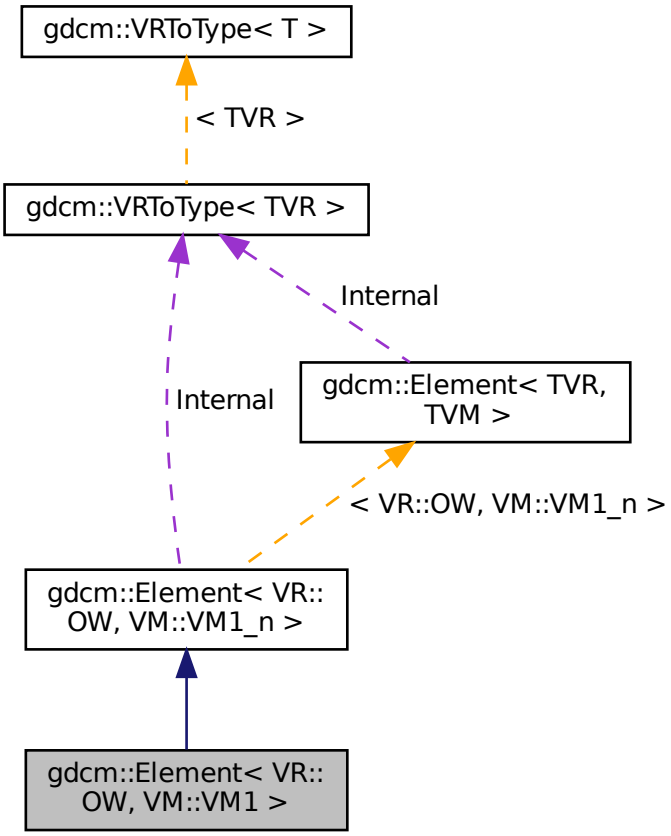
10.107 gdcm::Element< VR::OW, VM::VM1 > Class Reference

`#include <gdcmElement.h>`

Inheritance diagram for gdcm::Element< VR::OW, VM::VM1 >:



Collaboration diagram for gdcm::Element< VR::OW, VM::VM1 >:



Additional Inherited Members

The documentation for this class was generated from the following file:

- gdcmElement.h

10.108 gdcm::ElementDisableCombinations< TVR, TVM > Class Template Reference

A class which is used to produce compile errors for an invalid combination of template parameters.

`#include <gdcmElement.h>`

10.108.1 Detailed Description

template<long long TVR, int TVM>
class gdcm::ElementDisableCombinations< TVR, TVM >

A class which is used to produce compile errors for an invalid combination of template parameters.

Invalid combinations have specialized declarations with no definition.

The documentation for this class was generated from the following file:

- gdcmElement.h

10.109 gdcm::ElementDisableCombinations< VR::OB, VM::VM1_n > Class Reference

`#include <gdcmElement.h>`

The documentation for this class was generated from the following file:

- gdcmElement.h

10.110 gdcm::ElementDisableCombinations< VR::OW, VM::VM1_n > Class Reference

`#include <gdcmElement.h>`

The documentation for this class was generated from the following file:

- gdcmElement.h

10.111 gdcm::EmptyMaskGenerator Class Reference

EmptyMaskGenerator Main class to generate a Empty Mask Series from an input Series. This class takes an input folder and generates a series of DICOM files in the specified output directory. This class handles multiples DICOM Series within the same input directory.

`#include <gdcmEmptyMaskGenerator.h>`

Public Types

- enum SOPClassUIDMode {
  UseOriginalSOPClassUID = 0 ,
  UseGrayscaleSecondaryImageStorage }

Public Member Functions

- EmptyMaskGenerator ()
- ~EmptyMaskGenerator ()
- bool Execute ()

  Main loop.
- void SetInputDirectory (const char *dirname)

  Specify input directory.
- void SetOutputDirectory (const char *dirname)

  Specify output directory.
- void SetSOPClassUIDMode (SOPClassUIDMode mode)

10.111.1 Detailed Description

EmptyMaskGenerator Main class to generate a Empty Mask Series from an input Series. This class takes an input folder and generates a series of DICOM files in the specified output directory. This class handles multiples DICOM Series within the same input directory.

The class allow two mode of operations:

- UseOriginalSOPClassUID
- UseGrayscaleSecondaryImageStorage

UseOriginalSOPClassUID is the mode where original attributes are copied from the original DICOM instance.

UseGrayscaleSecondaryImageStorage is the mode where attributes are generated so as to create a Multiframe↩ GrayscaleByteSecondaryCaptureImageStorage (MultiframeGrayscaleWordSecondaryCaptureImageStorage) instance.

In both mode:

- the Study references (StudyInstanceUID and StudyID) are preserved.
- the PatientID reference is preserved.
- the Image Type attribute will be setup so that the fourth element is set to 'MASK'.
- a new Series Instance UID is generated. It is thus required to run the process over all files using the same input Series Instance UID so that a proper mapping from the old Series UID is done to the new one. Since a new Series Instance UID is generated, there is no sense to preserve the original Frame of Reference UID, although it would have made sense here.

Examples

EmptyMask.cxx.

10.111.2 Member Enumeration Documentation

10.111.2.1 SOPClassUIDMode

```
enum gdcm::EmptyMaskGenerator::SOPClassUIDMode
```

Enumerator

UseOriginalSOPClassUID	
UseGrayscaleSecondaryImageStorage	

10.111.3 Constructor & Destructor Documentation

10.111.3.1 EmptyMaskGenerator()

```
gdcm::EmptyMaskGenerator::EmptyMaskGenerator ( )
```

10.111.3.2 ~EmptyMaskGenerator()

```
gdcm::EmptyMaskGenerator::~EmptyMaskGenerator ( )
```

10.111.4 Member Function Documentation

10.111.4.1 Execute()

```
bool gdcm::EmptyMaskGenerator::Execute ( )
```

Main loop.

Examples

EmptyMask.cxx.

10.111.4.2 SetInputDirectory()

```
void gdcm::EmptyMaskGenerator::SetInputDirectory (
            const char * dirname )
```

Specify input directory.

Examples

EmptyMask.cxx.

10.111.4.3 SetOutputDirectory()

```
void gdcm::EmptyMaskGenerator::SetOutputDirectory (
            const char * dirname )
```

Specify output directory.

Examples

EmptyMask.cxx.

10.111.4.4 SetSOPClassUIDMode()

```
void gdcm::EmptyMaskGenerator::SetSOPClassUIDMode (
            SOPClassUIDMode mode )
```

Select generation of SOP Class UID method: Default is UseOriginalSOPClassUID

Examples

EmptyMask.cxx.

The documentation for this class was generated from the following file:

- gdcmEmptyMaskGenerator.h

10.112 gdcm::EncapsulatedDocument Class Reference

EncapsulatedDocument.

```
#include <gdcmEncapsulatedDocument.h>
```

Public Member Functions

- EncapsulatedDocument ()=default

10.112.1 Detailed Description

EncapsulatedDocument.

10.112.2 Constructor & Destructor Documentation

10.112.2.1 EncapsulatedDocument()

```
gdcm::EncapsulatedDocument::EncapsulatedDocument ( )  [default]
```

The documentation for this class was generated from the following file:

- gdcmEncapsulatedDocument.h

10.113 gdcm::EncodingImplementation< T > Class Template Reference

EncodingImplementation.

10.113.1 Detailed Description

template<long long T>
class gdcm::EncodingImplementation< T >

EncodingImplementation.

Note

TODO

The documentation for this class was generated from the following file:

- gdcmElement.h

10.114 gdcm::EncodingImplementation< VR::VRASCII > Class Reference

`#include <gdcmElement.h>`

Public Member Functions

- template<> void Write (const double *data, unsigned long length, std::ostream &_os)

Static Public Member Functions

- template<typename T >
  static void Read (T *data, unsigned long length, std::istream &_is)
- template<typename T >
  static void ReadComputeLength (T *data, unsigned int &length, std::istream &_is)
- template<typename T >
  static void ReadNoSwap (T *data, unsigned long length, std::istream &_is)
- template<typename T >
  static void Write (const T *data, unsigned long length, std::ostream &_os)

10.114.1 Member Function Documentation

10.114.1.1 Read()

```
template<typename T >
static void gdcm::EncodingImplementation< VR::VRASCII >::Read (
            T * data,
            unsigned long length,
            std::istream & _is )  [inline], [static]
```

10.114.1.2 ReadComputeLength()

```
template<typename T >
static void gdcm::EncodingImplementation< VR::VRASCII >::ReadComputeLength (
            T * data,
            unsigned int & length,
            std::istream & _is )  [inline], [static]
```

References gdcm::backslash().

10.114.1.3 ReadNoSwap()

```
template<typename T >
static void gdcm::EncodingImplementation< VR::VRASCII >::ReadNoSwap (
            T * data,
            unsigned long length,
            std::istream & _is )  [inline], [static]
```

10.114.1.4 Write() [1/2]

```
template<>
void gdcm::EncodingImplementation< VR::VRASCII >::Write (
            const double * data,
            unsigned long length,
            std::ostream & _os )  [inline]
```

References gdcm::x16printf().

10.114.1.5 Write() [2/2]

```
template<typename T >
static void gdcm::EncodingImplementation< VR::VRASCII >::Write (
            const T * data,
            unsigned long length,
            std::ostream & _os )  [inline], [static]
```

The documentation for this class was generated from the following file:

- gdcmElement.h

10.115 gdcm::EncodingImplementation< VR::VRBINARY > Class Reference

`#include <gdcmElement.h>`

Static Public Member Functions

- template<typename T >
  static void Read (T *data, unsigned long length, std::istream &_is)
- template<typename T >
  static void ReadComputeLength (T *data, unsigned int &length, std::istream &_is)
- template<typename T >
  static void ReadNoSwap (T *data, unsigned long length, std::istream &_is)
- template<typename T >
  static void Write (const T *data, unsigned long length, std::ostream &_os)

10.115.1 Member Function Documentation

10.115.1.1 Read()

```
template<typename T >
static void gdcm::EncodingImplementation< VR::VRBINARY >::Read (
            T * data,
            unsigned long length,
            std::istream & _is )  [inline], [static]
```

10.115.1.2 ReadComputeLength()

```
template<typename T >
static void gdcm::EncodingImplementation< VR::VRBINARY >::ReadComputeLength (
            T * data,
            unsigned int & length,
            std::istream & _is )  [inline], [static]
```

10.115.1.3 ReadNoSwap()

```
template<typename T >
static void gdcm::EncodingImplementation< VR::VRBINARY >::ReadNoSwap (
            T * data,
            unsigned long length,
            std::istream & _is )  [inline], [static]
```

10.115.1.4 Write()

```
template<typename T >
static void gdcm::EncodingImplementation< VR::VRBINARY >::Write (
            const T * data,
            unsigned long length,
            std::ostream & _os )  [inline], [static]
```

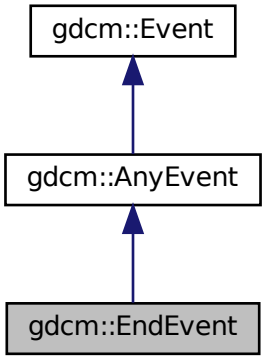
The documentation for this class was generated from the following file:

- gdcmElement.h

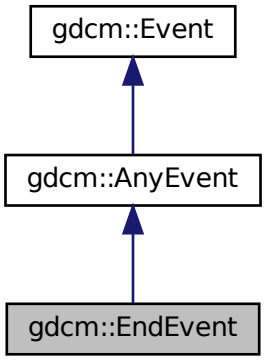
10.116 gdcm::EndEvent Class Reference

`#include <gdcmEvent.h>`

Inheritance diagram for gdcm::EndEvent:



Collaboration diagram for gdcm::EndEvent:



Additional Inherited Members

The documentation for this class was generated from the following file:

- gdcmEvent.h

10.117 gdcm::EnumeratedValues Class Reference

Element. A Data Element with Enumerated Values that does not have a Value equivalent to one of the Values specified in this standard has an invalid value within the scope of a specific Information Object/SOP Class definition. Note:

`#include <gdcmEnumeratedValues.h>`

Public Member Functions

- EnumeratedValues ()=default

10.117.1 Detailed Description

Element. A Data Element with Enumerated Values that does not have a Value equivalent to one of the Values specified in this standard has an invalid value within the scope of a specific Information Object/SOP Class definition. Note:

1. Patient Sex (0010, 0040) is an example of a Data Element having Enumerated Values. It is defined to have a Value that is either "M", "F", or "O" (see PS 3.3). No other Value shall be given to this Data Element.
2. Future modifications of this standard may add to the set of allowed values for Data Elements with Enumerated Values. Such additions by themselves may or may not require a change in SOP Class UIDs, depending on the semantics of the Data Element.

10.117.2 Constructor & Destructor Documentation

10.117.2.1 EnumeratedValues()

`gdcm::EnumeratedValues::EnumeratedValues ( ) [default]`

The documentation for this class was generated from the following file:

- gdcmEnumeratedValues.h

10.118 gdcm::EquipmentManufacturer Class Reference

`#include <gdcmEquipmentManufacturer.h>`

Public Types

- enum Type {
  UNKNOWN = 0 ,
  FUJI ,
  GEMS ,
  HITACHI ,
  KODAK ,
  MARCONI ,
  PMS ,
  SIEMENS ,
  TOSHIBA }

Static Public Member Functions

- static Type Compute (DataSet const &ds)
- static const char * TypeToString (Type type)

10.118.1 Detailed Description

The intent is for private tags handling. This class is not meant to handle all possible vendors in the world, simply those well known where we intend to read private tags afterwards (typically SIEMENS+CSA, GEMS+PDB ...)

10.118.2 Member Enumeration Documentation

10.118.2.1 Type

`enum gdcm::EquipmentManufacturer::Type`

Enumerator

UNKNOWN	
FUJI	
GEMS	
HITACHI	
KODAK	
MARCONI	
PMS	
SIEMENS	
TOSHIBA	

10.118.3 Member Function Documentation

10.118.3.1 Compute()

```
static Type gdcm::EquipmentManufacturer::Compute (
            DataSet const & ds )  [static]
```

10.118.3.2 TypeToString()

```
static const char * gdcm::EquipmentManufacturer::TypeToString (
            Type type )  [static]
```

The documentation for this class was generated from the following file:

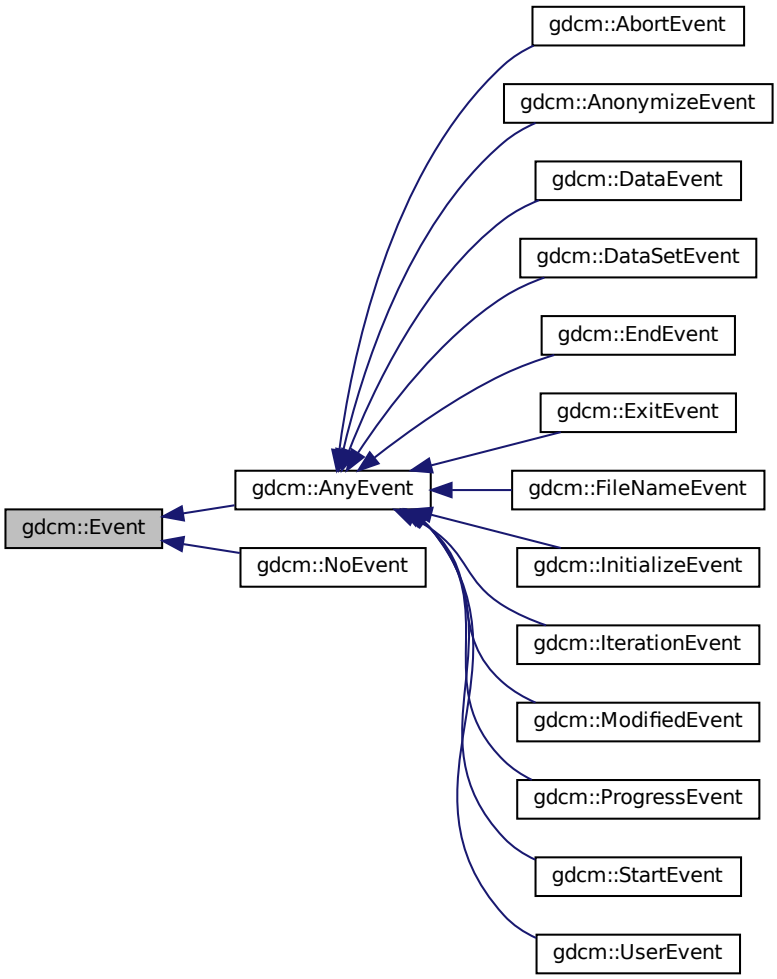
- gdcmEquipmentManufacturer.h

10.119 gdcm::Event Class Reference

superclass for callback/observer methods

```
#include <gdcmEvent.h>
```

Inheritance diagram for gdcm::Event:



Public Member Functions

- Event ()
- Event (const Event &)
- virtual ~Event ()
- virtual bool CheckEvent (const Event *) const =0
- virtual const char * GetEventName () const =0
- virtual Event * MakeObject () const =0
- void operator= (const Event &)=delete
- virtual void Print (std::ostream &os) const

10.119.1 Detailed Description

superclass for callback/observer methods

See also

Command Subject

Examples

BasicAnonymizer.cs, ClinicalTrialIdentificationWorkflow.cs, ScanDirectory.cs, and SimpleScanner.cxx.

10.119.2 Constructor & Destructor Documentation

10.119.2.1 Event() [1/2]

```
gdcm::Event::Event ( )
```

10.119.2.2 ~Event()

```
virtual gdcm::Event::~Event ( )  [virtual]
```

10.119.2.3 Event() [2/2]

```
gdcm::Event::Event (
            const Event &  )
```

10.119.3 Member Function Documentation

10.119.3.1 CheckEvent()

```
virtual bool gdcm::Event::CheckEvent (
            const Event *  ) const  [pure virtual]
```

Check if given event matches or derives from this event.

10.119.3.2 GetEventName()

```
virtual const char * gdcm::Event::GetEventName ( ) const  [pure virtual]
```

Return the StringName associated with the event.

Implemented in gdcm::DataEvent, gdcm::FileNameEvent, gdcm::ProgressEvent, gdcm::DataSetEvent, and gdcm::AnonymizeEvent.

Examples

BasicAnonymizer.cs, ClinicalTrialIdentificationWorkflow.cs, and ScanDirectory.cs.

10.119.3.3 MakeObject()

```
virtual Event * gdcm::Event::MakeObject ( ) const  [pure virtual]
```

Create an Event of this type This method work as a Factory for creating events of each particular type.

Implemented in gdcm::DataEvent, gdcm::FileNameEvent, gdcm::ProgressEvent, gdcm::DataSetEvent, and gdcm::AnonymizeEvent.

10.119.3.4 operator=()

```
void gdcm::Event::operator= (
            const Event &  )  [delete]
```

10.119.3.5 Print()

```
virtual void gdcm::Event::Print (
            std::ostream & os ) const  [virtual]
```

Print Event information. This method can be overridden by specific Event subtypes. The default is to print out the type of the event.

Referenced by gdcm::operator<<().

The documentation for this class was generated from the following file:

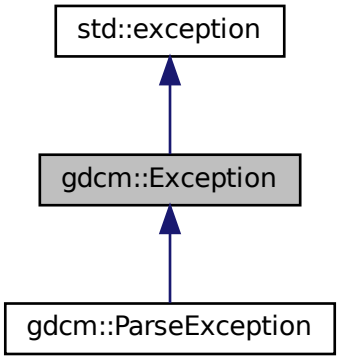
- gdcmEvent.h

10.120 gdcm::Exception Class Reference

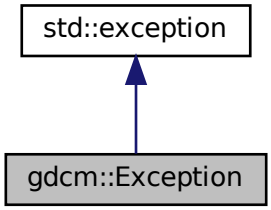
Exception.

`#include <gdcmException.h>`

Inheritance diagram for gdcm::Exception:



Collaboration diagram for gdcm::Exception:



Public Member Functions

- Exception (const char *desc="None", const char *file=__FILE__, unsigned int lineNumber=__LINE__, const char *func="")
- ~Exception () override throw ()
- const char * GetDescription () const

  Return the Description.
- const char * what () const override throw ()

  what implementation

10.120.1 Detailed Description

Exception.

Standard exception handling object.

Note

Its copy-constructor and assignment operator are generated by the compiler.

Examples

ExtractImageRegion.cs, ExtractImageRegionWithLUT.cs, ExtractOneFrame.cs, FileChangeTS.cs, and FileChangeTSLossy.cs.

10.120.2 Constructor & Destructor Documentation

10.120.2.1 Exception()

```
gdcm::Exception::Exception (
            const char * desc = "None",
            const char * file = __FILE__,
            unsigned int lineNumber = __LINE__,
            const char * func = "" )  [inline], [explicit]
```

Explicit constructor, initializing the description and the text returned by what().

Note

The last parameter is ignored for the time being. It may be used to specify the function where the exception was thrown.

10.120.2.2 ~Exception()

```
gdcm::Exception::~Exception ( ) throw ( )   [inline], [override]
```

10.120.3 Member Function Documentation

10.120.3.1 GetDescription()

```
const char * gdcm::Exception::GetDescription ( ) const  [inline]
```

Return the Description.

Referenced by gdcm::SequenceOfItems::Read().

10.120.3.2 what()

```
const char * gdcm::Exception::what ( ) const throw ( )  [inline], [override]
```

what implementation

Referenced by gdcm::SequenceOfFragments::ReadValue().

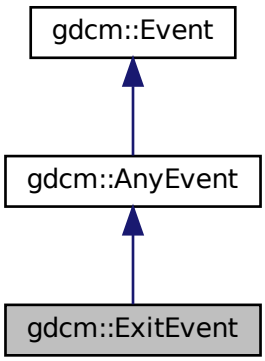
The documentation for this class was generated from the following file:

- gdcmException.h

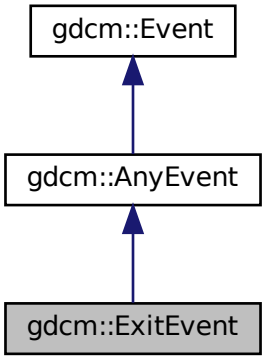
10.121 gdcm::ExitEvent Class Reference

```
#include <gdcmEvent.h>
```

Inheritance diagram for gdcm::ExitEvent:



Collaboration diagram for gdcm::ExitEvent:



Additional Inherited Members

The documentation for this class was generated from the following file:

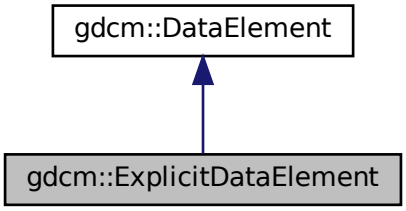
- gdcmEvent.h

10.122 gdcm::ExplicitDataElement Class Reference

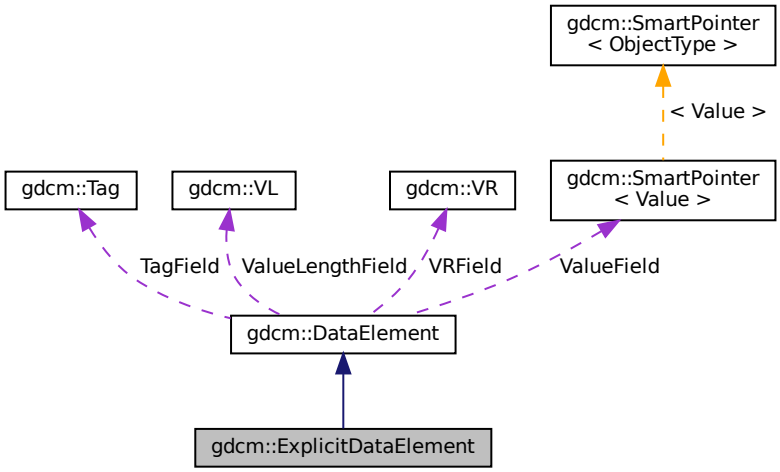
Class to read/write a DataElement as Explicit Data Element.

`#include <gdcmExplicitDataElement.h>`

Inheritance diagram for gdcm::ExplicitDataElement:



Collaboration diagram for gdcm::ExplicitDataElement:



Public Member Functions

- VL GetLength () const
- template<typename TSwap >
  std::istream & Read (std::istream &is)
- template<typename TSwap >
  std::istream & ReadPreValue (std::istream &is)
- template<typename TSwap >
  std::istream & ReadValue (std::istream &is, bool readvalues=true)
- template<typename TSwap >
  std::istream & ReadWithLength (std::istream &is, VL &length)
- template<typename TSwap >
  const std::ostream & Write (std::ostream &os) const

Additional Inherited Members

10.122.1 Detailed Description

Class to read/write a DataElement as Explicit Data Element.

Note

bla

Examples

DumpSiemensBase64.cxx, DumpToshibaDTI.cxx, DumpToshibaDTI2.cxx, and ReadAndDumpDICOMDIR2.cxx.

10.122.2 Member Function Documentation

10.122.2.1 GetLength()

```
VL gdcm::ExplicitDataElement::GetLength ( ) const
```

10.122.2.2 Read()

```
template<typename TSwap >
std::istream & gdcm::ExplicitDataElement::Read (
             std::istream & is )
```

10.122.2.3 ReadPreValue()

```
template<typename TSwap >
std::istream & gdcm::ExplicitDataElement::ReadPreValue (
             std::istream & is )
```

10.122.2.4 ReadValue()

```
template<typename TSwap >
std::istream & gdcm::ExplicitDataElement::ReadValue (
             std::istream & is,
             bool readvalues = true )
```

10.122.2.5 ReadWithLength()

```
template<typename TSwap >
std::istream & gdcm::ExplicitDataElement::ReadWithLength (
             std::istream & is,
             VL & length )
```

10.122.2.6 Write()

```
template<typename TSwap >
const std::ostream & gdcm::ExplicitDataElement::Write (
             std::ostream & os ) const
```

The documentation for this class was generated from the following file:

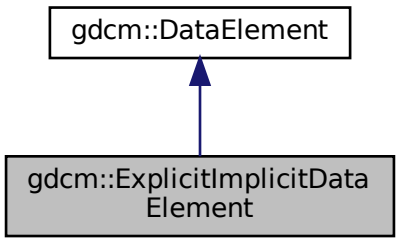
- gdcmExplicitDataElement.h

10.123 gdcm::ExplicitImplicitDataElement Class Reference

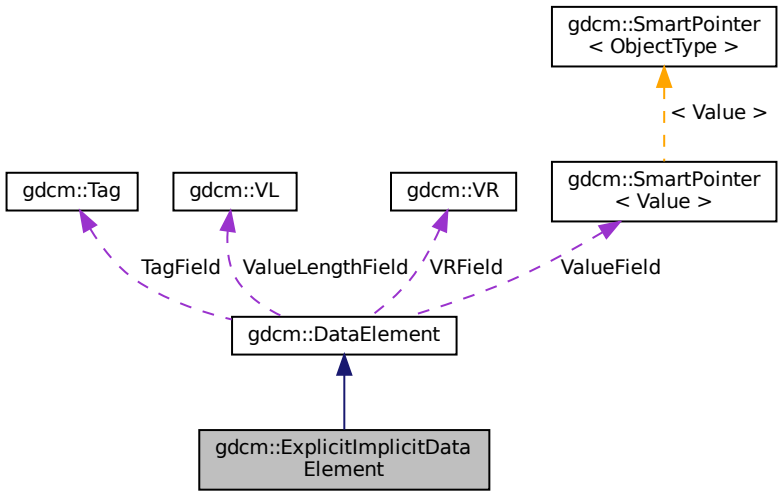
Class to read/write a DataElement as ExplicitImplicit Data Element.

```
#include <gdcmExplicitImplicitDataElement.h>
```

Inheritance diagram for gdcm::ExplicitImplicitDataElement:



Collaboration diagram for gdcm::ExplicitImplicitDataElement:



Public Member Functions

- VL GetLength () const
- template<typename TSwap >
  std::istream & Read (std::istream &is)
- template<typename TSwap >
  std::istream & ReadPreValue (std::istream &is)
- template<typename TSwap >
  std::istream & ReadValue (std::istream &is, bool readvalues=true)
- template<typename TSwap >
  std::istream & ReadWithLength (std::istream &is, VL &length)

Additional Inherited Members

10.123.1 Detailed Description

Class to read/write a DataElement as ExplicitImplicit Data Element.

Note

This only happen for some Philips images Should I derive from ExplicitDataElement instead ? This is the class that is the closest the GDCM1.x parser. At each element we try first to read it as explicit, if this fails, then we try again as an implicit element.

10.123.2 Member Function Documentation

10.123.2.1 GetLength()

```
VL gdcm::ExplicitImplicitDataElement::GetLength ( ) const
```

10.123.2.2 Read()

```
template<typename TSwap >
std::istream & gdcm::ExplicitImplicitDataElement::Read (
            std::istream & is )
```

10.123.2.3 ReadPreValue()

```
template<typename TSwap >
std::istream & gdcm::ExplicitImplicitDataElement::ReadPreValue (
            std::istream & is )
```

10.123.2.4 ReadValue()

```
template<typename TSwap >
std::istream & gdcm::ExplicitImplicitDataElement::ReadValue (
            std::istream & is,
            bool readvalues = true )
```

10.123.2.5 ReadWithLength()

```
template<typename TSwap >
std::istream & gdcm::ExplicitImplicitDataElement::ReadWithLength (
            std::istream & is,
            VL & length )  [inline]
```

The documentation for this class was generated from the following file:

- gdcmExplicitImplicitDataElement.h

10.124 gdcm::Fiducials Class Reference

Fiducials.

`#include <gdcmFiducials.h>`

Public Member Functions

- Fiducials ()=default

10.124.1 Detailed Description

Fiducials.

10.124.2 Constructor & Destructor Documentation

10.124.2.1 Fiducials()

`gdcm::Fiducials::Fiducials ( ) [default]`

The documentation for this class was generated from the following file:

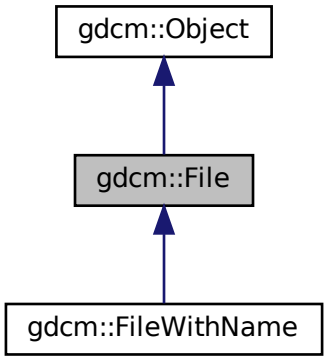
- gdcmFiducials.h

10.125 gdcm::File Class Reference

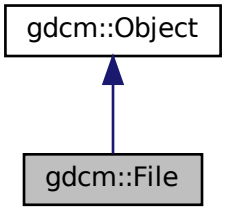
a DICOM File

`#include <gdcmFile.h>`

Inheritance diagram for gdcm::File:



Collaboration diagram for gdcm::File:



Public Member Functions

- File ()
- ~File () override
- DataSet & GetDataSet ()

  Get Data Set.
- const DataSet & GetDataSet () const

  Get Data Set.
- FileMetaInformation & GetHeader ()

  Get File Meta Information.
- const FileMetaInformation & GetHeader () const

  Get File Meta Information.

- std::istream & Read (std::istream &is)

  Read.
- void SetDataSet (const DataSet &ds)

  Set Data Set.
- void SetHeader (const FileMetaInformation &fmi)

  Set File Meta Information.
- std::ostream const & Write (std::ostream &os) const

  Write.

Friends

- std::ostream & operator<< (std::ostream &os, const File &val)

Additional Inherited Members

10.125.1 Detailed Description

a DICOM File

See PS 3.10 File: A File is an ordered string of zero or more bytes, where the first byte is at the beginning of the file and the last byte at the end of the File. Files are identified by a unique File ID and may by written, read and/or deleted.

See also

Reader Writer

Examples

ChangePrivateTags.cxx, ChangeSequenceUltrasound.cxx, CompressLossyJPEG.cs, CreateFakeRTDOSE.cxx, CreateJPIPDataSet.cxx, DeriveSeries.cxx, DiffFile.cxx, DumpCSA.cs, DumpGEMSMovieGroup.cxx, DumpPhilipsECHO.cxx, DumpSiemensBase64.cxx, DumpToshibaDTI.cxx, DumpToshibaDTI2.cxx, DumpVisusChange.cxx, DuplicatePCDE.cxx, EncapsulateFileInRawData.cxx, ExtractEncapsulatedFile.cs, ExtractEncryptedContent.cxx, ExtractImageRegion.cs, ExtractImageRegionWithLUT.cs, ExtractOneFrame.cs, Extracting_All_Resolution.cxx, Fake_Image_Using_Stream_Image_Writer.cx FileChangeTS.cs, FileChangeTSLossy.cs, FixBrokenJ2K.cxx, FixOrientation.cxx, GenAllVR.cxx, GenFakeIdentifyFile.cxx, GenFakeImage.cxx, GenLongSeqs.cxx, GenSeqs.cxx, GetJPEGSamplePrecision.cxx, GetSequenceUltrasound.cxx, GetSubSequenceData.cxx, HelloWorld.cxx, LargeVRDSExplicit.cxx, MakeTemplate.cxx, MpegVideoInfo.cs, NewSequence.cs, PatchFile.cxx, QIDO-RS.cxx, ReadAndDumpDICOMDIR.cxx, ReadAndPrintAttributes.cxx, ReadGEMSSDO.cxx, SimplePrint.cs, SimplePrintPatientName.cs, StreamImageReaderTest.cxx, TemplateEmptyImage.cxx, and iU22tomultisc.cxx.

10.125.2 Constructor & Destructor Documentation

10.125.2.1 File()

```
gdcm::File::File ( )
```

10.125.2.2 ~File()

```
gdcm::File::~File ( )  [override]
```

10.125.3 Member Function Documentation

10.125.3.1 GetDataSet() [1/2]

```
DataSet & gdcm::File::GetDataSet ( )  [inline]
```

Get Data Set.

10.125.3.2 GetDataSet() [2/2]

```
const DataSet & gdcm::File::GetDataSet ( ) const  [inline]
```

Get Data Set.

Examples

ChangePrivateTags.cxx, ChangeSequenceUltrasound.cxx, CompressLossyJPEG.cs, CreateFakeRTDOSE.cxx, CreateJPIPDataSet.cxx, DecompressImage.cs, DeriveSeries.cxx, DiffFile.cxx, DumpADAC.cxx, DumpCSA.cs, DumpExamCard.cxx, DumpGEMSMovieGroup.cxx, DumpImageHeaderInfo.cxx, DumpPhilipsECHO.cxx, DumpSiemensBase64.cxx, DumpToshibaDTI.cxx, DumpToshibaDTI2.cxx, DumpVisusChange.cxx, DuplicatePCDE.cxx, ELSCINT1WaveToText.cxx, ExtractEncapsulatedFile.cs, ExtractEncryptedContent.cxx, Extracting_All_Resolution.cxx, Fake_Image_Using_Stream_Image_Writer.cxx, FileChangeTS.cs, FileChangeTSLossy.cs, FixBrokenJ2K.cxx, FixJAIBugJPEGLS.cxx, FixOrientation.cxx, GenAllVR.cxx, GenFakeIdentifyFile.cxx, GenLongSeqs.cxx, GenSeqs.cxx, GetJPEGSamplePrecision.cxx, GetSequenceUltrasound.cxx, GetSubSequenceData.cxx, HelloWorld.cxx, LargeVRDSExplicit.cxx, MergeTwoFiles.cxx, MrProtocol.cxx, NewSequence.cs, PatchFile.cxx, QIDO-RS.cxx, ReadAndDumpDICOMDIR.cxx, ReadAndDumpDICOMDIR2.cxx, ReadAndPrintAttributes.cxx, ReadExplicitLengthSQIVR.cxx, ReadGEMSSDO.cxx, SimplePrint.cs, StreamImageReaderTest.cxx, TemplateEmptyImage.cxx, csa2img.cxx, gdcmrtionplan.cxx, gdcmrtplan.cxx, iU22tomultisc.cxx, pmsct_rgb1.cxx, and rle2img.cxx.

10.125.3.3 GetHeader() [1/2]

```
FileMetaInformation & gdcm::File::GetHeader ( )  [inline]
```

Get File Meta Information.

10.125.3.4 GetHeader() [2/2]

```
const FileMetaInformation & gdcm::File::GetHeader ( ) const  [inline]
```

Get File Meta Information.

Examples

CreateJPIPDataSet.cxx, DumpToshibaDTI.cxx, DumpToshibaDTI2.cxx, EncapsulateFileInRawData.cxx, Extracting_All_Resolution.cxx, Fake_Image_Using_Stream_Image_Writer.cxx, FixJAIBugJPEGLS.cxx, GenAllVR.cxx, GenFakeIdentifyFile.cxx, GetJPEGSamplePrecision.cxx, LargeVRDSExplicit.cxx, MakeTemplate.cxx, MergeTwoFiles.cxx, MpegVideoInfo.cs, QIDO-RS.cxx, ReadAndDumpDICOMDIR.cxx, ReformatFile.cs, StandardizeFiles.cs, StreamImageReaderTest.cxx, pmsct_rgb1.cxx, and rle2img.cxx.

10.125.3.5 Read()

```
std::istream & gdcm::File::Read (
            std::istream & is )
```

Read.

10.125.3.6 SetDataSet()

```
void gdcm::File::SetDataSet (
            const DataSet & ds )  [inline]
```

Set Data Set.

10.125.3.7 SetHeader()

```
void gdcm::File::SetHeader (
            const FileMetaInformation & fmi )  [inline]
```

Set File Meta Information.

10.125.3.8 Write()

```
std::ostream const & gdcm::File::Write (
            std::ostream & os ) const
```

Write.

10.125.4 Friends And Related Function Documentation

10.125.4.1 operator<<

```
std::ostream & operator<< (
            std::ostream & os,
            const File & val )  [friend]
```

The documentation for this class was generated from the following file:

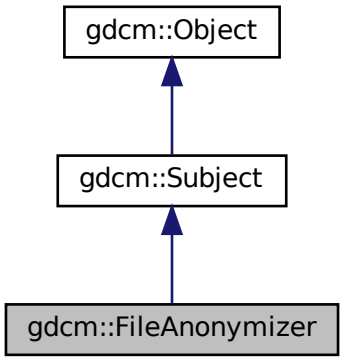
- gdcmFile.h

10.126 gdcm::FileAnonymizer Class Reference

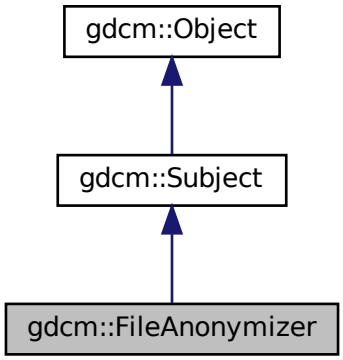
FileAnonymizer.

```
#include <gdcmFileAnonymizer.h>
```

Inheritance diagram for gdcm::FileAnonymizer:



Collaboration diagram for gdcm::FileAnonymizer:



Public Member Functions

- FileAnonymizer ()
- ~FileAnonymizer () override
- void Empty (Tag const &t)
- void Remove (Tag const &t)

  remove a tag (even a SQ can be removed)
- void Replace (Tag const &t, const char *value_data, VL const &vl)
- void Replace (Tag const &t, const char *value_str)
- void SetInputFileName (const char *filename_native)

  Set input filename.
- void SetOutputFileName (const char *filename_native)

  Set output filename.
- bool Write ()

  Write the output file.

Additional Inherited Members

10.126.1 Detailed Description

FileAnonymizer.

This Anonymizer is a file-based Anonymizer. It requires a valid DICOM file and will use the Value Length to skip over any information.

It will not load the DICOM dataset taken from SetInputFileName() into memory and should consume much less memory than Anonymizer.

Warning

: Each time you call Replace() with a value. This value will copied, and stored in memory. The behavior is not ideal for extremely large data (larger than memory size). This class is really meant to take a large DICOM input file and then only changed some small attribute.

caveats:

- This class will NOT work with unordered attributes in a DICOM File,
- This class does neither recompute nor update the Group Length element,
- This class currently does not update the File Meta Information header.
- Only strict inplace Replace operation is supported when input and output file are the same.

Examples

FileAnonymize.cs, and MakeTemplate.cxx.

10.126.2 Constructor & Destructor Documentation

10.126.2.1 FileAnonymizer()

```
gdcm::FileAnonymizer::FileAnonymizer ( )
```

10.126.2.2 ~FileAnonymizer()

```
gdcm::FileAnonymizer::~FileAnonymizer ( )  [override]
```

10.126.3 Member Function Documentation

10.126.3.1 Empty()

```
void gdcm::FileAnonymizer::Empty (
            Tag const & t )
```

Make Tag t empty Warning: does not handle SQ element

Examples

FileAnonymize.cs, and MakeTemplate.cxx.

10.126.3.2 Remove()

```
void gdcm::FileAnonymizer::Remove (
            Tag const & t )
```

remove a tag (even a SQ can be removed)

Examples

FileAnonymize.cs.

10.126.3.3 Replace() [1/2]

```
void gdcm::FileAnonymizer::Replace (
            Tag const & t,
            const char * value_data,
            VL const & vl )
```

when the value contains \0, it is a good idea to specify the length. This function is required when dealing with VRBINARY tag

10.126.3.4 Replace() [2/2]

```
void gdcm::FileAnonymizer::Replace (
            Tag const & t,
            const char * value_str )
```

Replace tag with another value, if tag is not found it will be created: WARNING: this function can only execute if tag is a VRASCII WARNING: Do not ever try to write a value in a SQ Data Element !

Examples

FileAnonymize.cs.

10.126.3.5 SetInputFileName()

```
void gdcm::FileAnonymizer::SetInputFileName (
            const char * filename_native )
```

Set input filename.

Examples

FileAnonymize.cs, and MakeTemplate.cxx.

10.126.3.6 SetOutputFileName()

```
void gdcm::FileAnonymizer::SetOutputFileName (
            const char * filename_native )
```

Set output filename.

Examples

FileAnonymize.cs, and MakeTemplate.cxx.

10.126.3.7 Write()

```
bool gdcm::FileAnonymizer::Write ( )
```

Write the output file.

Examples

FileAnonymize.cs, and MakeTemplate.cxx.

The documentation for this class was generated from the following file:

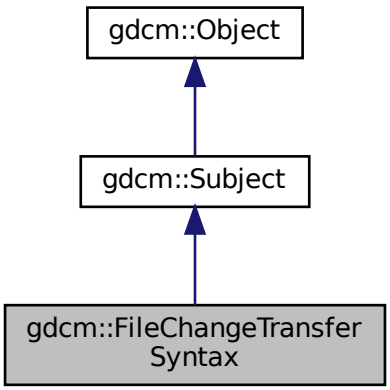
- gdcmFileAnonymizer.h

10.127 gdcm::FileChangeTransferSyntax Class Reference

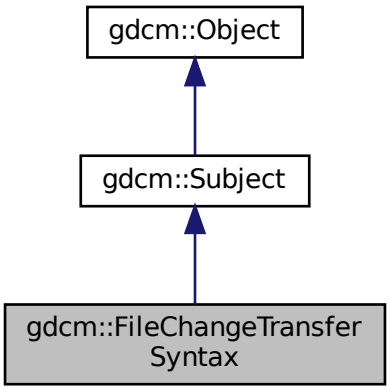
FileChangeTransferSyntax.

`#include <gdcmFileChangeTransferSyntax.h>`

Inheritance diagram for gdcm::FileChangeTransferSyntax:



Collaboration diagram for gdcm::FileChangeTransferSyntax:



Public Member Functions

- FileChangeTransferSyntax ()
- ~FileChangeTransferSyntax () override
- bool Change ()

  Change the transfer syntax.
- ImageCodec * GetCodec ()
- void SetInputFileName (const char *filename_native)

  Set input filename (raw DICOM)
- void SetOutputFileName (const char *filename_native)

  Set output filename (target compressed DICOM)
- void SetTransferSyntax (TransferSyntax const &ts)

  Specify the Target Transfer Syntax.

Static Public Member Functions

- static SmartPointer< FileChangeTransferSyntax > New ()

  for wrapped language: instantiate a reference counted object

Additional Inherited Members

10.127.1 Detailed Description

FileChangeTransferSyntax.

This class is a file-based (limited) replacement of the in-memory ImageChangeTransferSyntax.

This class provide a file-based compression-only mechanism. It will take in an uncompressed DICOM image file (Pixel Data element). Then produced as output a compressed DICOM file (Transfer Syntax will be updated).

Currently it supports the following transfer syntax:

- JPEGLosslessProcess14_1

Examples

FileChangeTS.cs, and FileChangeTSLossy.cs.

10.127.2 Constructor & Destructor Documentation

10.127.2.1 FileChangeTransferSyntax()

```
gdcm::FileChangeTransferSyntax::FileChangeTransferSyntax ( )
```

10.127.2.2 ~FileChangeTransferSyntax()

```
gdcm::FileChangeTransferSyntax::~FileChangeTransferSyntax ( )  [override]
```

10.127.3 Member Function Documentation

10.127.3.1 Change()

```
bool gdcm::FileChangeTransferSyntax::Change ( )
```

Change the transfer syntax.

Examples

FileChangeTS.cs, and FileChangeTSLossy.cs.

10.127.3.2 GetCodec()

```
ImageCodec * gdcm::FileChangeTransferSyntax::GetCodec ( )
```

Retrieve the actual codec (valid after calling SetTransferSyntax) Only advanced users should call this function.

Examples

FileChangeTSLossy.cs.

10.127.3.3 New()

```
static SmartPointer< FileChangeTransferSyntax > gdcm::FileChangeTransferSyntax::New ( )  [inline],
[static]
```

for wrapped language: instantiate a reference counted object

Examples

FileChangeTS.cs, and FileChangeTSLossy.cs.

10.127.3.4 SetInputFileName()

```
void gdcm::FileChangeTransferSyntax::SetInputFileName (
            const char * filename_native )
```

Set input filename (raw DICOM)

Examples

FileChangeTS.cs, and FileChangeTSLossy.cs.

10.127.3.5 SetOutputFileName()

```
void gdcm::FileChangeTransferSyntax::SetOutputFileName (
            const char * filename_native )
```

Set output filename (target compressed DICOM)

Examples

FileChangeTS.cs, and FileChangeTSLossy.cs.

10.127.3.6 SetTransferSyntax()

```
void gdcm::FileChangeTransferSyntax::SetTransferSyntax (
            TransferSyntax const & ts )
```

Specify the Target Transfer Syntax.

Examples

FileChangeTS.cs, and FileChangeTSLossy.cs.

The documentation for this class was generated from the following file:

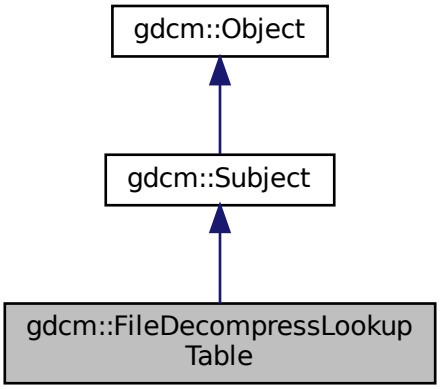
- gdcmFileChangeTransferSyntax.h

10.128 gdcm::FileDecompressLookupTable Class Reference

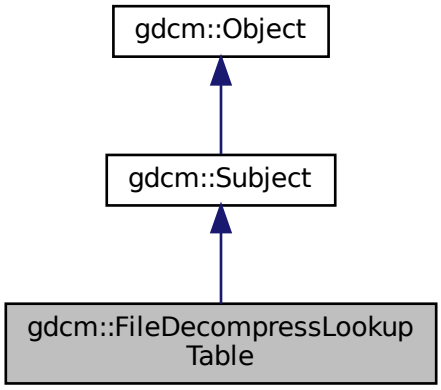
FileDecompressLookupTable class.

```
#include <gdcmFileDecompressLookupTable.h>
```

Inheritance diagram for gdcm::FileDecompressLookupTable:



Collaboration diagram for gdcm::FileDecompressLookupTable:



Public Member Functions

- FileDecompressLookupTable ()=default
- ~FileDecompressLookupTable () override=default
- bool Change ()

  Decompress.
- File & GetFile ()
- Pixmap & GetPixmap ()
- const Pixmap & GetPixmap () const
- void SetFile (const File &f)

  Set/Get File.
- void SetPixmap (Pixmap const &img)

Additional Inherited Members

10.128.1 Detailed Description

FileDecompressLookupTable class.

It decompress the segmented LUT into linearized one (only PALETTE_COLOR images) Output will be a PhotometricInterpretation=RGB image

10.128.2 Constructor & Destructor Documentation

10.128.2.1 FileDecompressLookupTable()

```
gdcm::FileDecompressLookupTable::FileDecompressLookupTable ( )  [default]
```

10.128.2.2 ~FileDecompressLookupTable()

```
gdcm::FileDecompressLookupTable::~FileDecompressLookupTable ( )  [override], [default]
```

10.128.3 Member Function Documentation

10.128.3.1 Change()

```
bool gdcm::FileDecompressLookupTable::Change ( )
```

Decompress.

10.128.3.2 GetFile()

```
File & gdcm::FileDecompressLookupTable::GetFile ( )  [inline]
```

10.128.3.3 GetPixmap() [1/2]

```
Pixmap & gdcm::FileDecompressLookupTable::GetPixmap ( )  [inline]
```

10.128.3.4 GetPixmap() [2/2]

```
const Pixmap & gdcm::FileDecompressLookupTable::GetPixmap ( ) const  [inline]
```

10.128.3.5 SetFile()

```
void gdcm::FileDecompressLookupTable::SetFile (
            const File & f )  [inline]
```

Set/Get File.

10.128.3.6 SetPixmap()

```
void gdcm::FileDecompressLookupTable::SetPixmap (
            Pixmap const & img )  [inline]
```

The documentation for this class was generated from the following file:

- gdcmFileDecompressLookupTable.h

10.129 gdcm::FileDerivation Class Reference

FileDerivation class.

```
#include <gdcmFileDerivation.h>
```

Public Member Functions

- FileDerivation ()
- ~FileDerivation ()
- bool AddReference (const char *referencedsopclassuid, const char *referencedsopinstanceuid)
- bool Derive ()

  Change.
- File & GetFile ()
- const File & GetFile () const
- void SetAppendDerivationHistory (bool b)
- void SetDerivationCodeSequenceCodeValue (unsigned int codevalue)

  Specify the Derivation Code Sequence Code Value. Eg 113040.
- void SetDerivationDescription (const char *dd)

  Specify the Derivation Description. Eg "lossy conversion".
- void SetFile (const File &f)

  Set/Get File.
- void SetPurposeOfReferenceCodeSequenceCodeValue (unsigned int codevalue)

  Specify the Purpose Of Reference Code Value. Eg. 121320.

Protected Member Functions

- bool AddDerivationDescription ()
- bool AddPurposeOfReferenceCodeSequence (DataSet &ds)
- bool AddSourceImageSequence ()

10.129.1 Detailed Description

FileDerivation class.

See PS 3.16 - 2008 For the list of Code Value that can be used for in Derivation Code Sequence

URL: `http://medical.nema.org/medical/dicom/2008/08_16pu.pdf`

DICOM Part 16 has two Context Groups CID 7202 and CID 7203 which contain a set of codes defining reason for a source image reference (ie. reason code for referenced image sequence) and a coded description of the deriation applied to the new image data from the original. Both these context groups are extensible.

File Derivation is compulsory when creating a lossy derived image.

Examples

DeriveSeries.cxx, GenFakeImage.cxx, and ReformatFile.cs.

10.129.2 Constructor & Destructor Documentation

10.129.2.1 FileDerivation()

```
gdcm::FileDerivation::FileDerivation ( )
```

10.129.2.2 ~FileDerivation()

```
gdcm::FileDerivation::~FileDerivation ( )
```

10.129.3 Member Function Documentation

10.129.3.1 AddDerivationDescription()

```
bool gdcm::FileDerivation::AddDerivationDescription ( )  [protected]
```

10.129.3.2 AddPurposeOfReferenceCodeSequence()

```
bool gdcm::FileDerivation::AddPurposeOfReferenceCodeSequence (
            DataSet & ds )  [protected]
```

10.129.3.3 AddReference()

```
bool gdcm::FileDerivation::AddReference (
            const char * referencedsopclassuid,
            const char * referencedsopinstanceuid )
```

Create the proper reference. Need to pass the original SOP Class UID and the original SOP Instance UID, so that those value can be used as Reference.

Warning

referencedsopclassuid and referencedsopinstanceuid needs to be \0 padded. This is not compatible with how ByteValue->GetPointer works.

Examples

DeriveSeries.cxx, GenFakeImage.cxx, and ReformatFile.cs.

10.129.3.4 AddSourceImageSequence()

```
bool gdcm::FileDerivation::AddSourceImageSequence ( )  [protected]
```

10.129.3.5 Derive()

```
bool gdcm::FileDerivation::Derive ( )
```

Change.

Examples

DeriveSeries.cxx, GenFakeImage.cxx, and ReformatFile.cs.

10.129.3.6 GetFile() [1/2]

```
File & gdcm::FileDerivation::GetFile ( )  [inline]
```

Examples

GenFakeImage.cxx, and ReformatFile.cs.

10.129.3.7 GetFile() [2/2]

```
const File & gdcm::FileDerivation::GetFile ( ) const  [inline]
```

10.129.3.8 SetAppendDerivationHistory()

```
void gdcm::FileDerivation::SetAppendDerivationHistory (
            bool b )
```

Specify if Derivation history should be appended (default false) When false, this is an error if input already has a derivation history When true, both Purpose of Reference Code Value and Derivation Code Sequence Code Value can have their history appended.

10.129.3.9 SetDerivationCodeSequenceCodeValue()

```
void gdcm::FileDerivation::SetDerivationCodeSequenceCodeValue (
            unsigned int codevalue )
```

Specify the Derivation Code Sequence Code Value. Eg 113040.

Examples

DeriveSeries.cxx, GenFakeImage.cxx, and ReformatFile.cs.

10.129.3.10 SetDerivationDescription()

```
void gdcm::FileDerivation::SetDerivationDescription (
            const char * dd )
```

Specify the Derivation Description. Eg "lossy conversion".

10.129.3.11 SetFile()

```
void gdcm::FileDerivation::SetFile (
            const File & f )  [inline]
```

Set/Get File.

Examples

DeriveSeries.cxx, GenFakeImage.cxx, and ReformatFile.cs.

10.129.3.12 SetPurposeOfReferenceCodeSequenceCodeValue()

```
void gdcm::FileDerivation::SetPurposeOfReferenceCodeSequenceCodeValue (
            unsigned int codevalue )
```

Specify the Purpose Of Reference Code Value. Eg. 121320.

Examples

DeriveSeries.cxx, GenFakeImage.cxx, and ReformatFile.cs.

The documentation for this class was generated from the following file:

- gdcmFileDerivation.h

10.130 gdcm::FileExplicitFilter Class Reference

FileExplicitFilter class.

```
#include <gdcmFileExplicitFilter.h>
```

Public Member Functions

- FileExplicitFilter ()
- ~FileExplicitFilter ()=default
- bool Change ()

  Set FMI Transfer Syntax.
- File & GetFile ()
- void SetChangePrivateTags (bool b)

  Decide whether or not to VR'ify private tags.
- void SetFile (const File &f)

  Set/Get File.
- void SetRecomputeItemLength (bool b)

  By default set Sequence & Item length to Undefined to avoid recomputing length:
- void SetRecomputeSequenceLength (bool b)
- void SetUseVRUN (bool b)

  When VR=16bits in explicit but Implicit has a 32bits length, use VR=UN.

Protected Member Functions

- bool ChangeFMI ()
- bool ProcessDataSet (DataSet &ds, Dicts const &dicts)

10.130.1 Detailed Description

FileExplicitFilter class.

After changing a file from Implicit to Explicit representation (see ImageChangeTransferSyntax) one operation is to make sure the VR of each DICOM attribute are accurate and do match the one from PS 3.6. Indeed when a file is written in Implicit representation, the VR is not stored directly in the file.

Warning

changing an implicit dataset to an explicit dataset is NOT a trivial task of simply changing the VR to the dict one:

- One has to make sure SQ is properly set
- One has to recompute the explicit length SQ
- One has to make sure that VR is valid for the encoding
- One has to make sure that VR 16bits can store the original value length

Examples

GenAllVR.cxx, and LargeVRDSExplicit.cxx.

10.130.2 Constructor & Destructor Documentation

10.130.2.1 FileExplicitFilter()

```
gdcm::FileExplicitFilter::FileExplicitFilter ( )  [inline]
```

10.130.2.2 ~FileExplicitFilter()

```
gdcm::FileExplicitFilter::~FileExplicitFilter ( )  [default]
```

10.130.3 Member Function Documentation

10.130.3.1 Change()

```
bool gdcm::FileExplicitFilter::Change ( )
```

Set FMI Transfer Syntax.

Change

Examples

GenAllVR.cxx, and LargeVRDSExplicit.cxx.

10.130.3.2 ChangeFMI()

```
bool gdcm::FileExplicitFilter::ChangeFMI ( )  [protected]
```

10.130.3.3 GetFile()

```
File & gdcm::FileExplicitFilter::GetFile ( )  [inline]
```

10.130.3.4 ProcessDataSet()

```
bool gdcm::FileExplicitFilter::ProcessDataSet (
            DataSet & ds,
            Dicts const & dicts )  [protected]
```

10.130.3.5 SetChangePrivateTags()

```
void gdcm::FileExplicitFilter::SetChangePrivateTags (
            bool b )  [inline]
```

Decide whether or not to VR'ify private tags.

10.130.3.6 SetFile()

```
void gdcm::FileExplicitFilter::SetFile (
            const File & f )  [inline]
```

Set/Get File.

Examples

GenAllVR.cxx, and LargeVRDSExplicit.cxx.

10.130.3.7 SetRecomputeItemLength()

```
void gdcm::FileExplicitFilter::SetRecomputeItemLength (
            bool b )
```

By default set Sequence & Item length to Undefined to avoid recomputing length:

10.130.3.8 SetRecomputeSequenceLength()

```
void gdcm::FileExplicitFilter::SetRecomputeSequenceLength (
            bool b )
```

10.130.3.9 SetUseVRUN()

```
void gdcm::FileExplicitFilter::SetUseVRUN (
            bool b )  [inline]
```

When VR=16bits in explicit but Implicit has a 32bits length, use VR=UN.

The documentation for this class was generated from the following file:

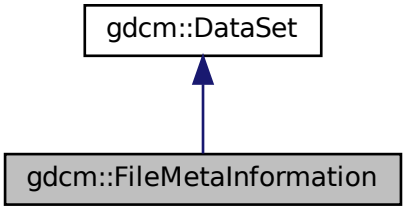
- gdcmFileExplicitFilter.h

10.131 gdcm::FileMetaInformation Class Reference

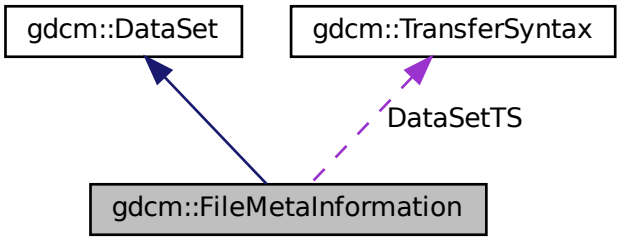
Class to represent a File Meta Information.

`#include <gdcmFileMetaInformation.h>`

Inheritance diagram for gdcm::FileMetaInformation:



Collaboration diagram for gdcm::FileMetaInformation:



Public Member Functions

- FileMetaInformation ()
- FileMetaInformation (FileMetaInformation const &fmi)
- ~FileMetaInformation ()
- void FillFromDataSet (DataSet const &ds)

  Construct a FileMetaInformation from an already existing DataSet:
- const TransferSyntax & GetDataSetTransferSyntax () const
- VL GetFullLength () const

- MediaStorage GetMediaStorage () const
- std::string GetMediaStorageAsString () const
- TransferSyntax::NegociatedType GetMetaInformationTS () const
- Preamble & GetPreamble ()
- const Preamble & GetPreamble () const

  Get Preamble.
- void Insert (const DataElement &de)
- bool IsValid () const
- FileMetaInformation & operator= (const FileMetaInformation &fmi)
- std::istream & Read (std::istream &is)

  Read.
- std::istream & ReadCompat (std::istream &is)
- void Replace (const DataElement &de)
- void SetDataSetTransferSyntax (const TransferSyntax &ts)
- void SetPreamble (const Preamble &p)
- std::ostream & Write (std::ostream &os) const

  Write.

Static Public Member Functions

- static void AppendImplementationClassUID (const char *imp)
- static const char * GetImplementationClassUID ()
- static const char * GetImplementationVersionName ()
- static const char * GetSourceApplicationEntityTitle ()
- static void SetImplementationClassUID (const char *imp)

  Override the GDCM default values:
- static void SetImplementationVersionName (const char *version)
- static void SetSourceApplicationEntityTitle (const char *title)

Protected Member Functions

- void ComputeDataSetMediaStorageSOPClass ()
- void ComputeDataSetTransferSyntax ()
- void Default ()
- template<typename TSwap >
  std::istream & ReadCompatInternal (std::istream &is)

Static Protected Member Functions

- static const char * GetFileMetaInformationVersion ()
- static const char * GetGDCMImplementationClassUID ()
- static const char * GetGDCMImplementationVersionName ()
- static const char * GetGDCMSourceApplicationEntityTitle ()

Protected Attributes

- MediaStorage::MSType DataSetMS
- TransferSyntax DataSetTS
- TransferSyntax::NegociatedType MetaInformationTS

Friends

- std::ostream & operator<< (std::ostream &_os, const FileMetaInformation &_val)

Additional Inherited Members

10.131.1 Detailed Description

Class to represent a File Meta Information.

FileMetaInformation is a Explicit Structured Set. Whenever the file contains an ImplicitDataElement DataSet, a conversion will take place.

Definition: The File Meta Information includes identifying information on the encapsulated Data Set. This header consists of a 128 byte File Preamble, followed by a 4 byte DICOM prefix, followed by the File Meta Elements shown in Table 7.1-1. This header shall be present in every DICOM file.

See also

Writer Reader

Examples

ClinicalTrialIdentificationWorkflow.cs, DumpToshibaDTI.cxx, DumpToshibaDTI2.cxx, GenAllVR.cxx, GenFakeIdentifyFile.cxx, LargeVRDSExplicit.cxx, MakeTemplate.cxx, ReadAndDumpDICOMDIR.cxx, ReformatFile.cs, and StandardizeFiles.cs.

10.131.2 Constructor & Destructor Documentation

10.131.2.1 FileMetaInformation() [1/2]

```
gdcm::FileMetaInformation::FileMetaInformation ( )
```

10.131.2.2 ~FileMetaInformation()

```
gdcm::FileMetaInformation::~FileMetaInformation ( )
```

10.131.2.3 FileMetaInformation() [2/2]

```
gdcm::FileMetaInformation::FileMetaInformation (
            FileMetaInformation const & fmi )  [inline]
```

References DataSetMS, DataSetTS, and MetaInformationTS.

10.131.3 Member Function Documentation

10.131.3.1 AppendImplementationClassUID()

```
static void gdcm::FileMetaInformation::AppendImplementationClassUID (
            const char * imp )  [static]
```

10.131.3.2 ComputeDataSetMediaStorageSOPClass()

```
void gdcm::FileMetaInformation::ComputeDataSetMediaStorageSOPClass ( )  [protected]
```

10.131.3.3 ComputeDataSetTransferSyntax()

```
void gdcm::FileMetaInformation::ComputeDataSetTransferSyntax ( )  [protected]
```

10.131.3.4 Default()

```
void gdcm::FileMetaInformation::Default ( )  [protected]
```

10.131.3.5 FillFromDataSet()

```
void gdcm::FileMetaInformation::FillFromDataSet (
            DataSet const & ds )
```

Construct a FileMetaInformation from an already existing DataSet:

10.131.3.6 GetDataSetTransferSyntax()

```
const TransferSyntax & gdcm::FileMetaInformation::GetDataSetTransferSyntax ( ) const  [inline]
```

Examples

GetJPEGSamplePrecision.cxx, and MergeTwoFiles.cxx.

10.131.3.7 GetFileMetaInformationVersion()

```
static const char * gdcm::FileMetaInformation::GetFileMetaInformationVersion ( )  [static], [protected]
```

10.131.3.8 GetFullLength()

```
VL gdcm::FileMetaInformation::GetFullLength ( ) const  [inline]
```

References gdcm::VL::GetLength().

10.131.3.9 GetGDCMImplementationClassUID()

```
static const char * gdcm::FileMetaInformation::GetGDCMImplementationClassUID ( )  [static], [protected]
```

10.131.3.10 GetGDCMImplementationVersionName()

```
static const char * gdcm::FileMetaInformation::GetGDCMImplementationVersionName ( )  [static],
[protected]
```

10.131.3.11 GetGDCMSourceApplicationEntityTitle()

```
static const char * gdcm::FileMetaInformation::GetGDCMSourceApplicationEntityTitle ( )  [static],
[protected]
```

10.131.3.12 GetImplementationClassUID()

```
static const char * gdcm::FileMetaInformation::GetImplementationClassUID ( )  [static]
```

10.131.3.13 GetImplementationVersionName()

```
static const char * gdcm::FileMetaInformation::GetImplementationVersionName ( )  [static]
```

10.131.3.14 GetMediaStorage()

```
MediaStorage gdcm::FileMetaInformation::GetMediaStorage ( ) const
```

10.131.3.15 GetMediaStorageAsString()

```
std::string gdcm::FileMetaInformation::GetMediaStorageAsString ( ) const
```

10.131.3.16 GetMetaInformationTS()

```
TransferSyntax::NegociatedType gdcm::FileMetaInformation::GetMetaInformationTS ( ) const  [inline]
```

10.131.3.17 GetPreamble() [1/2]

```
Preamble & gdcm::FileMetaInformation::GetPreamble ( )  [inline]
```

10.131.3.18 GetPreamble() [2/2]

```
const Preamble & gdcm::FileMetaInformation::GetPreamble ( ) const  [inline]
```

Get Preamble.

10.131.3.19 GetSourceApplicationEntityTitle()

```
static const char * gdcm::FileMetaInformation::GetSourceApplicationEntityTitle ( )  [static]
```

10.131.3.20 Insert()

```
void gdcm::FileMetaInformation::Insert (
            const DataElement & de )  [inline]
```

References gdcmErrorMacro, gdcm::Tag::GetGroup(), and gdcm::DataElement::GetTag().

10.131.3.21 IsValid()

```
bool gdcm::FileMetaInformation::IsValid ( ) const  [inline]
```

10.131.3.22 operator=()

```
FileMetaInformation & gdcm::FileMetaInformation::operator= (
            const FileMetaInformation & fmi )  [inline]
```

References DataSetMS, DataSetTS, and MetaInformationTS.

10.131.3.23 Read()

```
std::istream & gdcm::FileMetaInformation::Read (
            std::istream & is )
```

Read.

10.131.3.24 ReadCompat()

```
std::istream & gdcm::FileMetaInformation::ReadCompat (
            std::istream & is )
```

10.131.3.25 ReadCompatInternal()

```
template<typename TSwap >
std::istream & gdcm::FileMetaInformation::ReadCompatInternal (
            std::istream & is )  [protected]
```

10.131.3.26 Replace()

```
void gdcm::FileMetaInformation::Replace (
            const DataElement & de )  [inline]
```

Examples

LargeVRDSExplicit.cxx.

References gdcm::DataElement::GetTag().

10.131.3.27 SetDataSetTransferSyntax()

```
void gdcm::FileMetaInformation::SetDataSetTransferSyntax (
            const TransferSyntax & ts )
```

Examples

CreateJPIPDataSet.cxx, DumpToshibaDTI.cxx, DumpToshibaDTI2.cxx, EncapsulateFileInRawData.cxx, Extracting_All_Resolution.cxx, Fake_Image_Using_Stream_Image_Writer.cxx, FixJAIBugJPEGLS.cxx, GenAllVR.cxx, GenFakeIdentifyFile.cxx, LargeVRDSExplicit.cxx, MakeTemplate.cxx, MpegVideoInfo.cs, QIDO-RS.cxx, StreamImageReaderTest.cxx, pmsct_rgb1.cxx, and rle2img.cxx.

10.131.3.28 SetImplementationClassUID()

```
static void gdcm::FileMetaInformation::SetImplementationClassUID (
            const char * imp )  [static]
```

Override the GDCM default values:

10.131.3.29 SetImplementationVersionName()

```
static void gdcm::FileMetaInformation::SetImplementationVersionName (
            const char * version )  [static]
```

10.131.3.30 SetPreamble()

```
void gdcm::FileMetaInformation::SetPreamble (
            const Preamble & p )  [inline]
```

10.131.3.31 SetSourceApplicationEntityTitle()

```
static void gdcm::FileMetaInformation::SetSourceApplicationEntityTitle (
            const char * title )  [static]
```

Examples

ClinicalTrialIdentificationWorkflow.cs, FixJAIBugJPEGLS.cxx, GenerateDICOMDIR.cs, ReformatFile.cs, and StandardizeFiles.cs.

10.131.3.32 Write()

```
std::ostream & gdcm::FileMetaInformation::Write (
            std::ostream & os ) const
```

Write.

10.131.4 Friends And Related Function Documentation

10.131.4.1 operator<<

```
std::ostream & operator<< (
            std::ostream & _os,
            const FileMetaInformation & _val )  [friend]
```

10.131.5 Member Data Documentation

10.131.5.1 DataSetMS

```
MediaStorage::MSType gdcm::FileMetaInformation::DataSetMS  [protected]
```

Referenced by FileMetaInformation(), and operator=().

10.131.5.2 DataSetTS

```
TransferSyntax gdcm::FileMetaInformation::DataSetTS  [protected]
```

Referenced by FileMetaInformation(), and operator=().

10.131.5.3 MetaInformationTS

```
TransferSyntax::NegociatedType gdcm::FileMetaInformation::MetaInformationTS  [protected]
```

Referenced by FileMetaInformation(), and operator=().

The documentation for this class was generated from the following file:

- gdcmFileMetaInformation.h

10.132 gdcm::Filename Class Reference

Class to manipulate file name's.

```
#include <gdcmFilename.h>
```

Public Member Functions

- Filename (const char *filename="")
- bool EndWith (const char ending[ ]) const

  Does the filename ends with a particular string ?
- const char * GetExtension ()

  return only the extension part of a filename
- const char * GetFileName () const

  Return the full filename.
- const char * GetName ()

  return only the name part of a filename
- const char * GetPath ()

  Return only the path component of a filename.
- bool IsEmpty () const

  return whether the filename is empty
- bool IsIdentical (Filename const &fn) const
- operator const char * () const
- const char * ToUnixSlashes ()

  Convert backslash (windows style) to UNIX style slash.
- const char * ToWindowsSlashes ()

  Convert forward slash (UNIX style) to windows style slash.

Static Public Member Functions

- static const char * Join (const char *path, const char *filename)

10.132.1 Detailed Description

Class to manipulate file name's.

Note

OS independent representation of a filename (to query path, name and extension from a filename)

Examples

ClinicalTrialIdentificationWorkflow.cs.

10.132.2 Constructor & Destructor Documentation

10.132.2.1 Filename()

```
gdcm::Filename::Filename (
            const char * filename = "" )  [inline]
```

10.132.3 Member Function Documentation

10.132.3.1 EndWith()

```
bool gdcm::Filename::EndWith (
            const char ending[] ) const
```

Does the filename ends with a particular string ?

10.132.3.2 GetExtension()

```
const char * gdcm::Filename::GetExtension ( )
```

return only the extension part of a filename

10.132.3.3 GetFileName()

```
const char * gdcm::Filename::GetFileName ( ) const  [inline]
```

Return the full filename.

10.132.3.4 GetName()

```
const char * gdcm::Filename::GetName ( )
```

return only the name part of a filename

10.132.3.5 GetPath()

```
const char * gdcm::Filename::GetPath ( )
```

Return only the path component of a filename.

Examples

ClinicalTrialIdentificationWorkflow.cs.

10.132.3.6 IsEmpty()

```
bool gdcm::Filename::IsEmpty ( ) const  [inline]
```

return whether the filename is empty

10.132.3.7 IsIdentical()

```
bool gdcm::Filename::IsIdentical (
            Filename const & fn ) const
```

10.132.3.8 Join()

```
static const char * gdcm::Filename::Join (
            const char * path,
            const char * filename )  [static]
```

Join two paths NOT THREAD SAFE

Examples

BasicAnonymizer.cs, and ClinicalTrialIdentificationWorkflow.cs.

10.132.3.9 operator const char *()

```
gdcm::Filename::operator const char * ( ) const  [inline]
```

Simple operator to allow Filename myfilename("..."); const char * s = myfilename;

10.132.3.10 ToUnixSlashes()

```
const char * gdcm::Filename::ToUnixSlashes ( )
```

Convert backslash (windows style) to UNIX style slash.

10.132.3.11 ToWindowsSlashes()

```
const char * gdcm::Filename::ToWindowsSlashes ( )
```

Convert forward slash (UNIX style) to windows style slash.

The documentation for this class was generated from the following file:

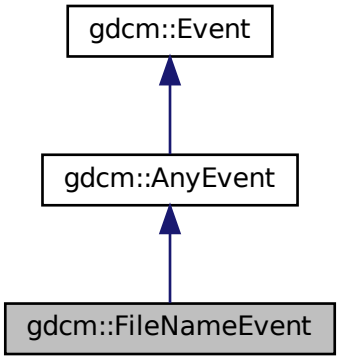
- gdcmFilename.h

10.133 gdcm::FileNameEvent Class Reference

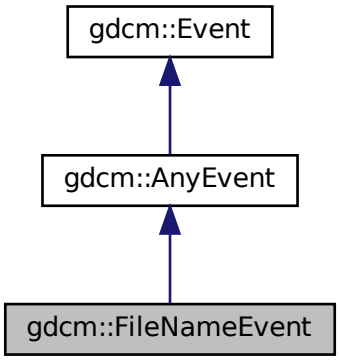
FileNameEvent.

```
#include <gdcmFileNameEvent.h>
```

Inheritance diagram for gdcm::FileNameEvent:



Collaboration diagram for gdcm::FileNameEvent:



Public Types

- typedef FileNameEvent Self
- typedef AnyEvent Superclass

Public Member Functions

- FileNameEvent (const char *s="")
- FileNameEvent (const Self &s)
- ~FileNameEvent () override=default
- bool CheckEvent (const ::gdcm::Event *e) const override
- const char * GetEventName () const override
- const char * GetFileName () const
- ::gdcm::Event * MakeObject () const override
- void operator= (const Self &)=delete
- void SetFileName (const char *f)

10.133.1 Detailed Description

FileNameEvent.

Special type of event triggered during processing of FileSet

See also

AnyEvent

Examples

ScanDirectory.cs, and SimpleScanner.cxx.

10.133.2 Member Typedef Documentation

10.133.2.1 Self

```
typedef FileNameEvent gdcm::FileNameEvent::Self
```

10.133.2.2 Superclass

```
typedef AnyEvent gdcm::FileNameEvent::Superclass
```

10.133.3 Constructor & Destructor Documentation

10.133.3.1 FileNameEvent() [1/2]

```
gdcm::FileNameEvent::FileNameEvent (
            const char * s = "" )  [inline]
```

10.133.3.2 ~FileNameEvent()

```
gdcm::FileNameEvent::~FileNameEvent ( )  [override], [default]
```

10.133.3.3 FileNameEvent() [2/2]

```
gdcm::FileNameEvent::FileNameEvent (
            const Self & s )  [inline]
```

10.133.4 Member Function Documentation

10.133.4.1 CheckEvent()

```
bool gdcm::FileNameEvent::CheckEvent (
            const ::gdcm::Event * e ) const  [inline], [override]
```

10.133.4.2 GetEventName()

```
const char * gdcm::FileNameEvent::GetEventName ( ) const  [inline], [override], [virtual]
```

Return the StringName associated with the event.

Implements gdcm::Event.

10.133.4.3 GetFileName()

```
const char * gdcm::FileNameEvent::GetFileName ( ) const  [inline]
```

Examples

ScanDirectory.cs, and SimpleScanner.cxx.

10.133.4.4 MakeObject()

```
::gdcm::Event * gdcm::FileNameEvent::MakeObject ( ) const  [inline], [override], [virtual]
```

Create an Event of this type This method work as a Factory for creating events of each particular type.

Implements gdcm::Event.

10.133.4.5 operator=()

```
void gdcm::FileNameEvent::operator= (
            const Self &  )  [delete]
```

10.133.4.6 SetFileName()

```
void gdcm::FileNameEvent::SetFileName (
            const char * f )  [inline]
```

The documentation for this class was generated from the following file:

- gdcmFileNameEvent.h

10.134 gdcm::FilenameGenerator Class Reference

FilenameGenerator.

```
#include <gdcmFilenameGenerator.h>
```

Public Types

- typedef std::vector< FilenameType > FilenamesType
- typedef std::string FilenameType
- typedef FilenamesType::size_type SizeType

Public Member Functions

- FilenameGenerator ()
- ~FilenameGenerator ()=default
- bool Generate ()

  Generate (return success)
- const char * GetFilename (SizeType n) const

  Get a particular filename (call after Generate)
- FilenamesType const & GetFilenames () const

  Return all filenames.
- SizeType GetNumberOfFilenames () const
- const char * GetPattern () const
- const char * GetPrefix () const
- void SetNumberOfFilenames (SizeType nfiles)

  Set/Get the number of filenames to generate.
- void SetPattern (const char *pattern)

  Set/Get pattern.
- void SetPrefix (const char *prefix)

  Set/Get prefix.

10.134.1 Detailed Description

FilenameGenerator.

class to generate filenames based on a pattern (C-style)

Output will be:

for i = 0, number of filenames: outfilename[i] = prefix + (pattern % i)

where pattern % i means C-style snprintf of Pattern using value 'i'

Examples

ConvertMultiFrameToSingleFrame.cxx, and CreateFakePET.cxx.

10.134.2 Member Typedef Documentation

10.134.2.1 FilenamesType

```
typedef std::vector<FilenameType> gdcm::FilenameGenerator::FilenamesType
```

10.134.2.2 FilenameType

```
typedef std::string gdcm::FilenameGenerator::FilenameType
```

10.134.2.3 SizeType

```
typedef FilenamesType::size_type gdcm::FilenameGenerator::SizeType
```

10.134.3 Constructor & Destructor Documentation

10.134.3.1 FilenameGenerator()

```
gdcm::FilenameGenerator::FilenameGenerator ( )  [inline]
```

10.134.3.2 ~FilenameGenerator()

```
gdcm::FilenameGenerator::~FilenameGenerator ( )  [default]
```

10.134.4 Member Function Documentation

10.134.4.1 Generate()

```
bool gdcm::FilenameGenerator::Generate ( )
```

Generate (return success)

Examples

ConvertMultiFrameToSingleFrame.cxx, and CreateFakePET.cxx.

10.134.4.2 GetFilename()

```
const char * gdcm::FilenameGenerator::GetFilename (
            SizeType n ) const
```

Get a particular filename (call after Generate)

Examples

ConvertMultiFrameToSingleFrame.cxx, and CreateFakePET.cxx.

10.134.4.3 GetFilenames()

```
FilenamesType const & gdcm::FilenameGenerator::GetFilenames ( ) const  [inline]
```

Return all filenames.

10.134.4.4 GetNumberOfFilenames()

```
SizeType gdcm::FilenameGenerator::GetNumberOfFilenames ( ) const
```

Examples

ConvertMultiFrameToSingleFrame.cxx, and CreateFakePET.cxx.

10.134.4.5 GetPattern()

```
const char * gdcm::FilenameGenerator::GetPattern ( ) const  [inline]
```

10.134.4.6 GetPrefix()

```
const char * gdcm::FilenameGenerator::GetPrefix ( ) const  [inline]
```

10.134.4.7 SetNumberOfFilenames()

```
void gdcm::FilenameGenerator::SetNumberOfFilenames (
            SizeType nfiles )
```

Set/Get the number of filenames to generate.

Examples

ConvertMultiFrameToSingleFrame.cxx, and CreateFakePET.cxx.

10.134.4.8 SetPattern()

```
void gdcm::FilenameGenerator::SetPattern (
            const char * pattern )  [inline]
```

Set/Get pattern.

Examples

ConvertMultiFrameToSingleFrame.cxx, and CreateFakePET.cxx.

10.134.4.9 SetPrefix()

```
void gdcm::FilenameGenerator::SetPrefix (
            const char * prefix )  [inline]
```

Set/Get prefix.

The documentation for this class was generated from the following file:

- gdcmFilenameGenerator.h

10.135 gdcm::FileSet Class Reference

```
#include <gdcmFileSet.h>
```

Public Types

- typedef std::vector< FileType > FilesType
- typedef std::string FileType

Public Member Functions

- FileSet ()
- bool AddFile (const char *filename)
- void AddFile (File const &)
- FilesType const & GetFiles () const
- void SetFiles (FilesType const &files)

Friends

- std::ostream & operator<< (std::ostream &_os, const FileSet &d)

10.135.1 Detailed Description

File-set: A File-set is a collection of DICOM Files (and possibly non-DICOM Files) that share a common naming space within which File IDs are unique.

10.135.2 Member Typedef Documentation

10.135.2.1 FilesType

```
typedef std::vector<FileType> gdcm::FileSet::FilesType
```

10.135.2.2 FileType

```
typedef std::string gdcm::FileSet::FileType
```

10.135.3 Constructor & Destructor Documentation

10.135.3.1 FileSet()

```
gdcm::FileSet::FileSet ( )  [inline]
```

10.135.4 Member Function Documentation

10.135.4.1 AddFile() [1/2]

```
bool gdcm::FileSet::AddFile (
            const char * filename )
```

Add a file 'filename' to the list of files. Return true on success, false in case filename could not be found on system.

10.135.4.2 AddFile() [2/2]

```
void gdcm::FileSet::AddFile (
            File const &  )  [inline]
```

Deprecated . Does nothing

10.135.4.3 GetFiles()

```
FilesType const & gdcm::FileSet::GetFiles ( ) const  [inline]
```

10.135.4.4 SetFiles()

```
void gdcm::FileSet::SetFiles (
            FilesType const & files )
```

10.135.5 Friends And Related Function Documentation

10.135.5.1 operator<<

```
std::ostream & operator<< (
            std::ostream & _os,
            const FileSet & d )  [friend]
```

The documentation for this class was generated from the following file:

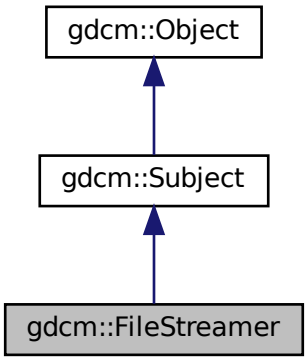
- gdcmFileSet.h

10.136 gdcm::FileStreamer Class Reference

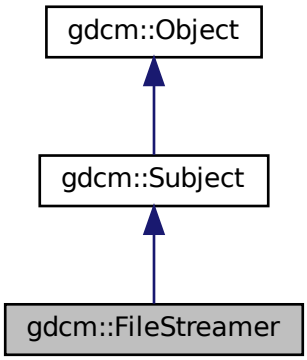
FileStreamer.

`#include <gdcmFileStreamer.h>`

Inheritance diagram for gdcm::FileStreamer:



Collaboration diagram for gdcm::FileStreamer:



Public Member Functions

- FileStreamer ()
- ~FileStreamer () override
- bool AppendToDataElement (const Tag &t, const char *array, size_t len)

  Append to previously started Tag t.
- bool AppendToGroupDataElement (const PrivateTag &pt, const char *array, size_t len)

  Append to previously started private creator.
- bool CheckDataElement (const Tag &t)
- void CheckTemplateFileName (bool check)
- bool ReserveDataElement (size_t len)
- bool ReserveGroupDataElement (unsigned short ndataelement)
- void SetOutputFileName (const char *filename_native)

  Set output filename (target file)
- void SetTemplateFileName (const char *filename_native)

  Set input DICOM template filename.
- bool StartDataElement (const Tag &t)
- bool StartGroupDataElement (const PrivateTag &pt, size_t maxsizede=0, uint8_t startoffset=0)
- bool StopDataElement (const Tag &t)

  Stop appending to tag t. This will compute the proper attribute length.
- bool StopGroupDataElement (const PrivateTag &pt)

  Stop appending to private creator.

Static Public Member Functions

- static SmartPointer< FileStreamer > New ()

  for wrapped language: instantiate a reference counted object

Additional Inherited Members

10.136.1 Detailed Description

FileStreamer.

This class let a user create a massive DICOM DataSet from a template DICOM file, by appending chunks of data.

This class support two mode of operation:

1. Creating a single DataElement by appending chunk after chunk of data.
2. Creating a set of DataElement within the same group, using a private creator for start. New DataElement are added any time the user defined maximum size for data element is reached.

Warning

any existing DataElement is removed, pick carefully which DataElement to add.

Examples

FileStreaming.cs.

10.136.2 Constructor & Destructor Documentation

10.136.2.1 FileStreamer()

```
gdcm::FileStreamer::FileStreamer ( )
```

10.136.2.2 ~FileStreamer()

```
gdcm::FileStreamer::~FileStreamer ( )  [override]
```

10.136.3 Member Function Documentation

10.136.3.1 AppendToDataElement()

```
bool gdcm::FileStreamer::AppendToDataElement (
            const Tag & t,
            const char * array,
            size_t len )
```

Append to previously started Tag t.

10.136.3.2 AppendToGroupDataElement()

```
bool gdcm::FileStreamer::AppendToGroupDataElement (
            const PrivateTag & pt,
            const char * array,
            size_t len )
```

Append to previously started private creator.

Examples

FileStreaming.cs.

10.136.3.3 CheckDataElement()

```
bool gdcm::FileStreamer::CheckDataElement (
            const Tag & t )
```

Decide to check the Data Element to be written (default: off) The implementation has default strategy for checking validity of DataElement. Currently it only support checking for the following tags:

- (7fe0,0010) Pixel Data

10.136.3.4 CheckTemplateFileName()

```
void gdcm::FileStreamer::CheckTemplateFileName (
            bool check )
```

Instead of simply blindly copying the input DICOM Template file, GDCM will be used to check the input file, and correct any issues recognized within the file. Only use if you do not have control over the input template file.

10.136.3.5 New()

```
static SmartPointer< FileStreamer > gdcm::FileStreamer::New ( )  [inline], [static]
```

for wrapped language: instantiate a reference counted object

10.136.3.6 ReserveDataElement()

```
bool gdcm::FileStreamer::ReserveDataElement (
            size_t len )
```

Add a hint on the final size of the dataelement. When optimally chosen, this reduce the number of file in-place copying. Should be called before StartDataElement

10.136.3.7 ReserveGroupDataElement()

```
bool gdcm::FileStreamer::ReserveGroupDataElement (
            unsigned short ndataelement )
```

Optimisation: pre-allocate the number of dataelement within the private group (ndataelement <= 256). Should be called before StartGroupDataElement

10.136.3.8 SetOutputFileName()

```
void gdcm::FileStreamer::SetOutputFileName (
            const char * filename_native )
```

Set output filename (target file)

Examples

FileStreaming.cs.

10.136.3.9 SetTemplateFileName()

```
void gdcm::FileStreamer::SetTemplateFileName (
            const char * filename_native )
```

Set input DICOM template filename.

Examples

FileStreaming.cs.

10.136.3.10 StartDataElement()

```
bool gdcm::FileStreamer::StartDataElement (
            const Tag & t )
```

Start Single Data Element Operation This will delete any existing Tag t. Need to call it only once.

10.136.3.11 StartGroupDataElement()

```
bool gdcm::FileStreamer::StartGroupDataElement (
            const PrivateTag & pt,
            size_t maxsizede = 0,
            uint8_t startoffset = 0 )
```

Start Private Group (multiple DataElement) Operation. Each newly added DataElement will have a length lower than

Parameters

maxsizede	. When not specified, maxsizede is set to maximum size allowed by DICOM (= 2^32). startoffset can be used to specify the very first element you want to start with (instead of the first possible). Value should be in [0x0, 0xff] This will find the first available private creator.

Bug maxsizede should be a value lower than the actual total size of the buffer to be copied

Examples

FileStreaming.cs.

10.136.3.12 StopDataElement()

```
bool gdcm::FileStreamer::StopDataElement (
            const Tag & t )
```

Stop appending to tag t. This will compute the proper attribute length.

10.136.3.13 StopGroupDataElement()

```
bool gdcm::FileStreamer::StopGroupDataElement (
            const PrivateTag & pt )
```

Stop appending to private creator.

Examples

FileStreaming.cs.

The documentation for this class was generated from the following file:

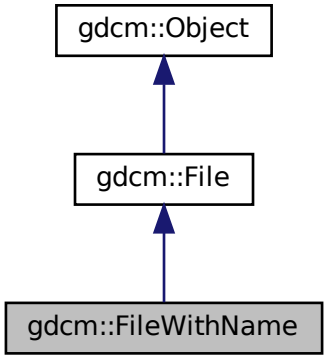
- gdcmFileStreamer.h

10.137 gdcm::FileWithName Class Reference

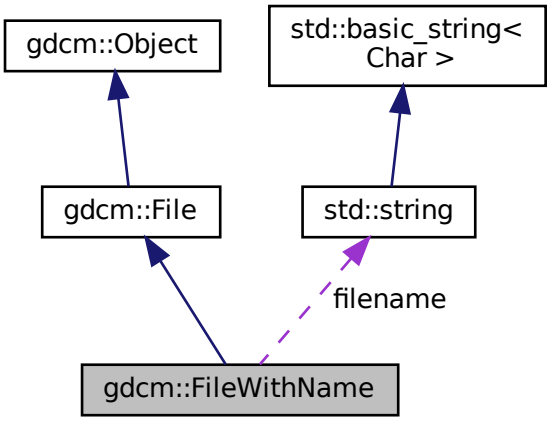
FileWithName.

`#include <gdcmSerieHelper.h>`

Inheritance diagram for gdcm::FileWithName:



Collaboration diagram for gdcm::FileWithName:



Public Member Functions

- FileWithName (File &f)

Public Attributes

- std::string filename

Additional Inherited Members

10.137.1 Detailed Description

FileWithName.

Backward only class do not use in newer code

10.137.2 Constructor & Destructor Documentation

10.137.2.1 FileWithName()

```
gdcm::FileWithName::FileWithName (
            File & f )  [inline]
```

10.137.3 Member Data Documentation

10.137.3.1 filename

```
std::string gdcm::FileWithName::filename
```

The documentation for this class was generated from the following file:

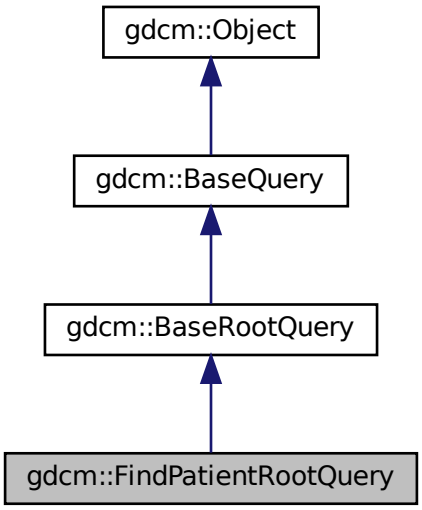
- gdcmSerieHelper.h

10.138 gdcm::FindPatientRootQuery Class Reference

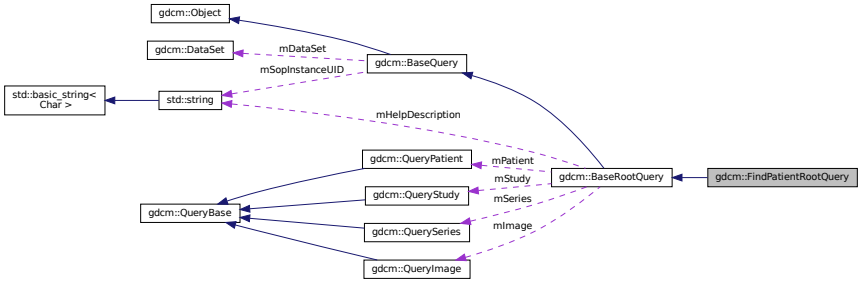
PatientRootQuery.

`#include <gdcmFindPatientRootQuery.h>`

Inheritance diagram for gdcm::FindPatientRootQuery:



Collaboration diagram for gdcm::FindPatientRootQuery:



Public Member Functions

- FindPatientRootQuery ()
- UIDs::TSName GetAbstractSyntaxUID () const override
- std::vector< Tag > GetTagListByLevel (const EQueryLevel &inQueryLevel) override
- void InitializeDataSet (const EQueryLevel &inQueryLevel) override
- bool ValidateQuery (bool inStrict=true) const override

Friends

- class QueryFactory

Additional Inherited Members

10.138.1 Detailed Description

PatientRootQuery.

contains: the class which will produce a dataset for c-find with patient root

10.138.2 Constructor & Destructor Documentation

10.138.2.1 FindPatientRootQuery()

```
gdcm::FindPatientRootQuery::FindPatientRootQuery ( )
```

10.138.3 Member Function Documentation

10.138.3.1 GetAbstractSyntaxUID()

```
UIDs::TSName gdcm::FindPatientRootQuery::GetAbstractSyntaxUID ( ) const  [override], [virtual]
```

Implements gdcm::BaseQuery.

10.138.3.2 GetTagListByLevel()

```
std::vector< Tag > gdcm::FindPatientRootQuery::GetTagListByLevel (
            const EQueryLevel & inQueryLevel )  [override], [virtual]
```

this function will return all tags at a given query level, so that they maybe selected for searching. The boolean forFind is true if the query is a find query, or false for a move query.

Implements gdcm::BaseRootQuery.

10.138.3.3 InitializeDataSet()

```
void gdcm::FindPatientRootQuery::InitializeDataSet (
            const EQueryLevel & inQueryLevel )  [override], [virtual]
```

this function sets tag 8,52 to the appropriate value based on query level also fills in the right unique tags, as per the standard's requirements should allow for connection with dcmtk

Implements gdcm::BaseRootQuery.

10.138.3.4 ValidateQuery()

```
bool gdcm::FindPatientRootQuery::ValidateQuery (
            bool inStrict = true ) const  [override], [virtual]
```

have to be able to ensure that 0x8,0x52 is set (which will be true if InitializeDataSet is called...) that the level is appropriate (ie, not setting PATIENT for a study query that the tags in the query match the right level (either required, unique, optional) by default, this function checks to see if the query is for finding, which is more permissive than for moving. For moving, only the unique tags are allowed. 10 Jan 2011: adding in the 'strict' mode. according to the standard (at least, how I've read it), only tags for a particular level should be allowed in a particular query (ie, just series level tags in a series level query). However, it seems that dcm4chee doesn't share that interpretation. So, if 'inStrict' is false, then tags from the current level and all higher levels are now considered valid. So, if you're doing a non-strict series-level query, tags from the patient and study level can be passed along as well.

Implements gdcm::BaseRootQuery.

10.138.4 Friends And Related Function Documentation

10.138.4.1 QueryFactory

```
friend class QueryFactory  [friend]
```

The documentation for this class was generated from the following file:

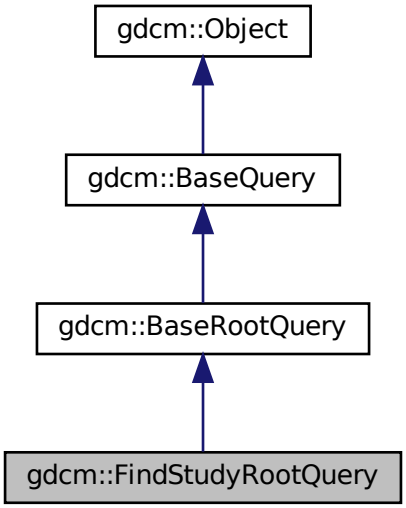
- gdcmFindPatientRootQuery.h

10.139 gdcm::FindStudyRootQuery Class Reference

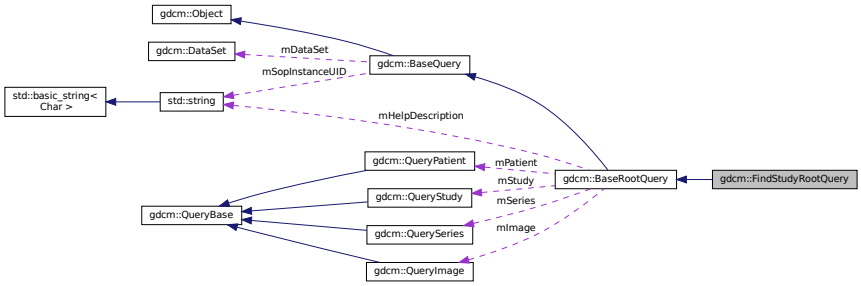
FindStudyRootQuery.

`#include <gdcmFindStudyRootQuery.h>`

Inheritance diagram for gdcm::FindStudyRootQuery:



Collaboration diagram for gdcm::FindStudyRootQuery:



Public Member Functions

- FindStudyRootQuery ()
- UIDs::TSName GetAbstractSyntaxUID () const override
- std::vector< Tag > GetTagListByLevel (const EQueryLevel &inQueryLevel) override
- void InitializeDataSet (const EQueryLevel &inQueryLevel) override
- bool ValidateQuery (bool inStrict=true) const override

Friends

- class QueryFactory

Additional Inherited Members

10.139.1 Detailed Description

FindStudyRootQuery.

contains: the class which will produce a dataset for C-FIND with study root

10.139.2 Constructor & Destructor Documentation

10.139.2.1 FindStudyRootQuery()

```
gdcm::FindStudyRootQuery::FindStudyRootQuery ( )
```

10.139.3 Member Function Documentation

10.139.3.1 GetAbstractSyntaxUID()

```
UIDs::TSName gdcm::FindStudyRootQuery::GetAbstractSyntaxUID ( ) const  [override], [virtual]
```

Implements gdcm::BaseQuery.

10.139.3.2 GetTagListByLevel()

```
std::vector< Tag > gdcm::FindStudyRootQuery::GetTagListByLevel (
            const EQueryLevel & inQueryLevel )  [override], [virtual]
```

this function will return all tags at a given query level, so that they maybe selected for searching. The boolean forFind is true if the query is a find query, or false for a move query.

Implements gdcm::BaseRootQuery.

10.139.3.3 InitializeDataSet()

```
void gdcm::FindStudyRootQuery::InitializeDataSet (
            const EQueryLevel & inQueryLevel )  [override], [virtual]
```

this function sets tag 8,52 to the appropriate value based on query level also fills in the right unique tags, as per the standard's requirements should allow for connection with dcmtk

Implements gdcm::BaseRootQuery.

10.139.3.4 ValidateQuery()

```
bool gdcm::FindStudyRootQuery::ValidateQuery (
            bool inStrict = true ) const  [override], [virtual]
```

have to be able to ensure that (0008,0052) is set that the level is appropriate (ie, not setting PATIENT for a study query that the tags in the query match the right level (either required, unique, optional)

Implements gdcm::BaseRootQuery.

10.139.4 Friends And Related Function Documentation

10.139.4.1 QueryFactory

```
friend class QueryFactory  [friend]
```

The documentation for this class was generated from the following file:

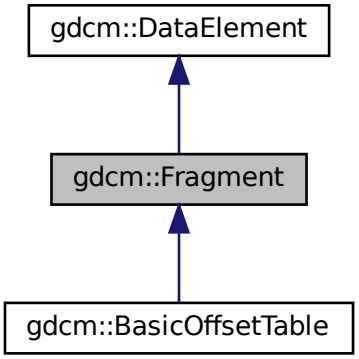
- gdcmFindStudyRootQuery.h

10.140 gdcm::Fragment Class Reference

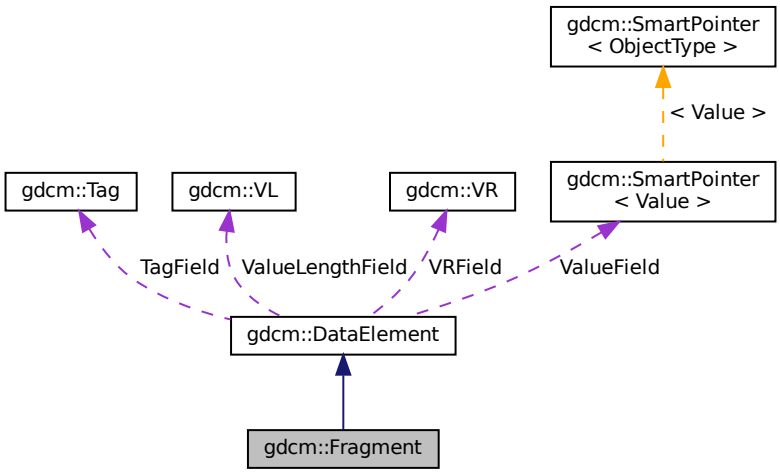
Class to represent a Fragment.

`#include <gdcmFragment.h>`

Inheritance diagram for gdcm::Fragment:



Collaboration diagram for gdcm::Fragment:



Public Member Functions

- Fragment ()
- VL ComputeLength () const
- VL GetLength () const
- template<typename TSwap >
  std::istream & Read (std::istream &is)
- template<typename TSwap >
  std::istream & ReadBacktrack (std::istream &is)
- template<typename TSwap >
  std::istream & ReadPreValue (std::istream &is)
- template<typename TSwap >
  std::istream & ReadValue (std::istream &is)
- template<typename TSwap >
  std::ostream & Write (std::ostream &os) const

Friends

- std::ostream & operator<< (std::ostream &os, const Fragment &val)

Additional Inherited Members

10.140.1 Detailed Description

Class to represent a Fragment.

Examples

DecompressImageMultiframe.cs, DecompressJPEGFile.cs, FixBrokenJ2K.cxx, FixJAIBugJPEGLS.cxx, and MpegVideoInfo.cs.

10.140.2 Constructor & Destructor Documentation

10.140.2.1 Fragment()

```
gdcm::Fragment::Fragment ( )  [inline]
```

10.140.3 Member Function Documentation

10.140.3.1 ComputeLength()

```
VL gdcm::Fragment::ComputeLength ( ) const
```

10.140.3.2 GetLength()

```
VL gdcm::Fragment::GetLength ( ) const
```

10.140.3.3 Read()

```
template<typename TSwap >
std::istream & gdcm::Fragment::Read (
            std::istream & is )  [inline]
```

Referenced by gdcm::SequenceOfFragments::ReadValue().

10.140.3.4 ReadBacktrack()

```
template<typename TSwap >
std::istream & gdcm::Fragment::ReadBacktrack (
            std::istream & is )  [inline]
```

References gdcmErrorMacro, gdcmWarningMacro, and gdcm::ParseException::SetLastElement().

Referenced by gdcm::SequenceOfFragments::ReadValue().

10.140.3.5 ReadPreValue()

```
template<typename TSwap >
std::istream & gdcm::Fragment::ReadPreValue (
            std::istream & is )  [inline]
```

10.140.3.6 ReadValue()

```
template<typename TSwap >
std::istream & gdcm::Fragment::ReadValue (
             std::istream & is )  [inline]
```

References gdcmWarningMacro, and gdcm::ParseException::SetLastElement().

10.140.3.7 Write()

```
template<typename TSwap >
std::ostream & gdcm::Fragment::Write (
             std::ostream & os ) const  [inline]
```

References gdcm::ByteValue::ComputeLength(), gdcm::ByteValue::GetLength(), gdcm::ByteValue::Write(), and gdcm::VL::Write().

10.140.4 Friends And Related Function Documentation

10.140.4.1 operator<<

```
std::ostream & operator<< (
             std::ostream & os,
             const Fragment & val )  [friend]
```

The documentation for this class was generated from the following file:

- gdcmFragment.h

10.141 gdcm::Global Class Reference

Global.

```
#include <gdcmGlobal.h>
```

Public Member Functions

- Global ()
- Global (const Global &_val)=delete
- ~Global ()
- bool Append (const char *path)
- Defs const & GetDefs () const
- Dicts & GetDicts ()
- Dicts const & GetDicts () const
- bool LoadResourcesFiles ()
- Global & operator= (const Global &_val)=delete
- bool Prepend (const char *path)

Static Public Member Functions

- static Global & GetInstance ()

  return the singleton instance

Protected Member Functions

- const char * Locate (const char *resfile) const

  Locate a resource file.

Friends

- std::ostream & operator<< (std::ostream &_os, const Global &g)

10.141.1 Detailed Description

Global.

Note

Global should be included in any translation unit that will use Dict or that implements the singleton pattern. It makes sure that the Dict singleton is created before and destroyed after all other singletons in GDCM.

Examples

BasicAnonymizer.cs, ClinicalTrialIdentificationWorkflow.cs, GenAllVR.cxx, GenFakeIdentifyFile.cxx, GenerateStandardSOPClasses.
PublicDict.cxx, ReadAndPrintAttributes.cxx, and TraverseModules.cxx.

10.141.2 Constructor & Destructor Documentation

10.141.2.1 Global() [1/2]

```
gdcm::Global::Global ( )
```

10.141.2.2 ~Global()

```
gdcm::Global::~Global ( )
```

10.141.2.3 Global() [2/2]

```
gdcm::Global::Global (
            const Global & _val )  [delete]
```

10.141.3 Member Function Documentation

10.141.3.1 Append()

```
bool gdcm::Global::Append (
            const char * path )
```

Append path at the end of the path list

Warning

not thread safe !

10.141.3.2 GetDefs()

```
Defs const & gdcm::Global::GetDefs ( ) const
```

retrieve the default/internal (Part 3) You need to explicitly call LoadResourcesFiles before

Examples

GenerateStandardSOPClasses.cxx, and TraverseModules.cxx.

10.141.3.3 GetDicts() [1/2]

```
Dicts & gdcm::Global::GetDicts ( )
```

10.141.3.4 GetDicts() [2/2]

```
Dicts const & gdcm::Global::GetDicts ( ) const
```

retrieve the default/internal dicts (Part 6) This dict is filled up at load time

Examples

GenAllVR.cxx, GenFakeIdentifyFile.cxx, MrProtocol.cxx, PublicDict.cxx, ReadAndPrintAttributes.cxx, and TraverseModules.cxx.

10.141.3.5 GetInstance()

```
static Global & gdcm::Global::GetInstance ( )  [static]
```

return the singleton instance

Examples

BasicAnonymizer.cs, ClinicalTrialIdentificationWorkflow.cs, GenAllVR.cxx, GenFakeIdentifyFile.cxx, GenerateStandardSOPClasses.
MrProtocol.cxx, PublicDict.cxx, and ReadAndPrintAttributes.cxx.

10.141.3.6 LoadResourcesFiles()

```
bool gdcm::Global::LoadResourcesFiles ( )
```

Load all internal XML files, resource path need to have been set before calling this member function (see Append/↩ Prepend members func)

Warning

not thread safe !

Examples

BasicAnonymizer.cs, ClinicalTrialIdentificationWorkflow.cs, GenerateStandardSOPClasses.cxx, and TraverseModules.cxx.

10.141.3.7 Locate()

```
const char * gdcm::Global::Locate (
            const char * resfile ) const  [protected]
```

Locate a resource file.

10.141.3.8 operator=()

```
Global & gdcm::Global::operator= (
            const Global & _val )  [delete]
```

10.141.3.9 Prepend()

```
bool gdcm::Global::Prepend (
            const char * path )
```

Prepend path at the beginning of the path list

Warning

not thread safe !

10.141.4 Friends And Related Function Documentation

10.141.4.1 operator<<

```
std::ostream & operator<< (
            std::ostream & _os,
            const Global & g )  [friend]
```

The documentation for this class was generated from the following file:

- gdcmGlobal.h

10.142 gdcm::GroupDict Class Reference

Class to represent the mapping from group number to its abbreviation and name.

```
#include <gdcmGroupDict.h>
```

Public Types

- typedef std::vector< std::string > GroupStringVector

Public Member Functions

- GroupDict ()
- ~GroupDict ()=default
- std::string const & GetAbbreviation (uint16_t num) const
- std::string const & GetName (uint16_t num) const
- size_t Size () const

Protected Member Functions

- void Add (std::string const &abbreviation, std::string const &name)
- void Insert (uint16_t num, std::string const &abbreviation, std::string const &name)

Friends

- std::ostream & operator<< (std::ostream &_os, const GroupDict &_val)

10.142.1 Detailed Description

Class to represent the mapping from group number to its abbreviation and name.

Note

Should I rewrite this class to use a std::map instead of std::vector for problem of memory consumption ?

10.142.2 Member Typedef Documentation

10.142.2.1 GroupStringVector

```
typedef std::vector<std::string> gdcm::GroupDict::GroupStringVector
```

10.142.3 Constructor & Destructor Documentation

10.142.3.1 GroupDict()

```
gdcm::GroupDict::GroupDict ( )  [inline]
```

10.142.3.2 ~GroupDict()

```
gdcm::GroupDict::~GroupDict ( )  [default]
```

10.142.4 Member Function Documentation

10.142.4.1 Add()

```
void gdcm::GroupDict::Add (
            std::string const & abbreviation,
            std::string const & name )  [protected]
```

10.142.4.2 GetAbbreviation()

```
std::string const & gdcm::GroupDict::GetAbbreviation (
            uint16_t num ) const
```

10.142.4.3 GetName()

```
std::string const & gdcm::GroupDict::GetName (
            uint16_t num ) const
```

10.142.4.4 Insert()

```
void gdcm::GroupDict::Insert (
            uint16_t num,
            std::string const & abbreviation,
            std::string const & name )  [protected]
```

10.142.4.5 Size()

```
size_t gdcm::GroupDict::Size ( ) const  [inline]
```

10.142.5 Friends And Related Function Documentation

10.142.5.1 operator<<

```
std::ostream & operator<< (
            std::ostream & _os,
            const GroupDict & _val )  [friend]
```

The documentation for this class was generated from the following file:

- gdcmGroupDict.h

10.143 gdcm::IconImageFilter Class Reference

IconImageFilter.

```
#include <gdcmIconImageFilter.h>
```

Public Member Functions

- IconImageFilter ()
- ~IconImageFilter ()
- bool Extract ()

  Extract all Icon found in File.
- File & GetFile ()
- const File & GetFile () const
- IconImage & GetIconImage (unsigned int i) const
- unsigned int GetNumberOfIconImages () const

  Retrieve extract IconImage (need to call Extract first)
- void SetFile (const File &f)

  Set/Get File.

Protected Member Functions

- void ExtractIconImages ()
- void ExtractVeproIconImages ()

10.143.1 Detailed Description

IconImageFilter.

This filter will extract icons from a File This filter will loop over all known sequence (public and private) that may contains an IconImage and retrieve them. The filter will fails with a value of false if no icon can be found Since it handle both public and private icon type, one should not assume the icon is in uncompress form, some private vendor store private icon in JPEG8/JPEG12

Implementation details: This filter supports the following Icons:

- (0088,0200) Icon Image Sequence
- (0009,10,GEIIS) GE IIS Thumbnail Sequence
- (6003,10,GEMS_Ultrasound_ImageGroup_001) GEMS Image Thumbnail Sequence
- (0055,30,VEPRO VIF 3.0 DATA) Icon Data
- (0055,30,VEPRO VIM 5.0 DATA) ICONDATA2

Warning

the icon stored in those private attribute do not conform to definition of Icon Image Sequence (do not simply copy/paste). For example some private icon can be expressed as 12bits pixel, while the DICOM standard only allow 8bits icons.

See also

ImageReader

Examples

ExtractIconFromFile.cxx.

10.143.2 Constructor & Destructor Documentation

10.143.2.1 IconImageFilter()

```
gdcm::IconImageFilter::IconImageFilter ( )
```

10.143.2.2 ~IconImageFilter()

```
gdcm::IconImageFilter::~IconImageFilter ( )
```

10.143.3 Member Function Documentation

10.143.3.1 Extract()

```
bool gdcm::IconImageFilter::Extract ( )
```

Extract all Icon found in File.

Examples

ExtractIconFromFile.cxx.

10.143.3.2 ExtractIconImages()

```
void gdcm::IconImageFilter::ExtractIconImages ( )  [protected]
```

10.143.3.3 ExtractVeproIconImages()

```
void gdcm::IconImageFilter::ExtractVeproIconImages ( )  [protected]
```

10.143.3.4 GetFile() [1/2]

```
File & gdcm::IconImageFilter::GetFile ( )  [inline]
```

10.143.3.5 GetFile() [2/2]

```
const File & gdcm::IconImageFilter::GetFile ( ) const  [inline]
```

10.143.3.6 GetIconImage()

```
IconImage & gdcm::IconImageFilter::GetIconImage (
            unsigned int i ) const
```

Examples

ExtractIconFromFile.cxx.

10.143.3.7 GetNumberOfIconImages()

```
unsigned int gdcm::IconImageFilter::GetNumberOfIconImages ( ) const
```

Retrieve extract IconImage (need to call Extract first)

Examples

ExtractIconFromFile.cxx.

10.143.3.8 SetFile()

```
void gdcm::IconImageFilter::SetFile (
            const File & f )  [inline]
```

Set/Get File.

Examples

ExtractIconFromFile.cxx.

The documentation for this class was generated from the following file:

- gdcmIconImageFilter.h

10.144 gdcm::IconImageGenerator Class Reference

IconImageGenerator.

```
#include <gdcmIconImageGenerator.h>
```

Public Member Functions

- IconImageGenerator ()
- ~IconImageGenerator ()
- void AutoPixelMinMax (bool b)
- void ConvertRGBToPaletteColor (bool b)
- bool Generate ()

  Generate Icon.
- const IconImage & GetIconImage () const

  Retrieve generated Icon.
- Pixmap & GetPixmap ()
- const Pixmap & GetPixmap () const
- void SetOutputDimensions (const unsigned int dims[2])

  Set Target dimension of output Icon.
- void SetOutsideValuePixel (double v)
- void SetPixelMinMax (double min, double max)
- void SetPixmap (const Pixmap &p)

  Set/Get File.

10.144.1 Detailed Description

IconImageGenerator.

This filter will generate a valid Icon from the Pixel Data element (an instance of Pixmap). To generate a valid Icon, one is only allowed the following Photometric Interpretation:

- MONOCHROME1
- MONOCHROME2
- PALETTE_COLOR

The Pixel Bits Allocated is restricted to 8bits, therefore 16 bits image needs to be rescaled. By default the filter will use the full scalar range of 16bits image to rescale to unsigned 8bits. This may not be ideal for some situation, in which case the API SetPixelMinMax can be used to overwrite the default min,max interval used.

See also

ImageReader

Examples

ExtractIconFromFile.cxx.

10.144.2 Constructor & Destructor Documentation

10.144.2.1 IconImageGenerator()

```
gdcm::IconImageGenerator::IconImageGenerator ( )
```

10.144.2.2 ~IconImageGenerator()

```
gdcm::IconImageGenerator::~IconImageGenerator ( )
```

10.144.3 Member Function Documentation

10.144.3.1 AutoPixelMinMax()

```
void gdcm::IconImageGenerator::AutoPixelMinMax (
            bool b )
```

Instead of explicitly specifying the min/max value for the rescale operation, let the internal mechanism compute the min/max of icon and rescale to best appropriate.

Examples

ExtractIconFromFile.cxx.

10.144.3.2 ConvertRGBToPaletteColor()

```
void gdcm::IconImageGenerator::ConvertRGBToPaletteColor (
            bool b )
```

Converting from RGB to PALETTE_COLOR can be a slow operation. However DICOM standard requires that color icon be described as palette. Set this boolean to false only if you understand the consequences. default value is true, false generates invalid Icon Image Sequence

10.144.3.3 Generate()

```
bool gdcm::IconImageGenerator::Generate ( )
```

Generate Icon.

Examples

ExtractIconFromFile.cxx.

10.144.3.4 GetIconImage()

```
const IconImage & gdcm::IconImageGenerator::GetIconImage ( ) const  [inline]
```

Retrieve generated Icon.

Examples

ExtractIconFromFile.cxx.

10.144.3.5 GetPixmap() [1/2]

```
Pixmap & gdcm::IconImageGenerator::GetPixmap ( )  [inline]
```

10.144.3.6 GetPixmap() [2/2]

```
const Pixmap & gdcm::IconImageGenerator::GetPixmap ( ) const  [inline]
```

10.144.3.7 SetOutputDimensions()

```
void gdcm::IconImageGenerator::SetOutputDimensions (
            const unsigned int dims[2] )
```

Set Target dimension of output Icon.

Examples

ExtractIconFromFile.cxx.

10.144.3.8 SetOutsideValuePixel()

```
void gdcm::IconImageGenerator::SetOutsideValuePixel (
            double v )
```

Set a pixel value that should be discarded. This happen typically for CT image, where a pixel has been used to pad outside the image (see Pixel Padding Value). Requires AutoPixelMinMax(true)

10.144.3.9 SetPixelMinMax()

```
void gdcm::IconImageGenerator::SetPixelMinMax (
             double min,
             double max )
```

Override default min/max to compute best rescale for 16bits -> 8bits downscale. Typically those value can be read from the SmallestImagePixelValue LargestImagePixelValue DICOM attribute.

10.144.3.10 SetPixmap()

```
void gdcm::IconImageGenerator::SetPixmap (
             const Pixmap & p )  [inline]
```

Set/Get File.

Examples

ExtractIconFromFile.cxx.

The documentation for this class was generated from the following file:

- gdcmIconImageGenerator.h

10.145 gdcm::ignore_char Struct Reference

```
#include <gdcmElement.h>
```

Public Member Functions

- ignore_char (char c)

Public Attributes

- char m_char

10.145.1 Constructor & Destructor Documentation

10.145.1.1 ignore_char()

```
gdcm::ignore_char::ignore_char (
            char c )  [inline]
```

10.145.2 Member Data Documentation

10.145.2.1 m_char

```
char gdcm::ignore_char::m_char
```

Referenced by gdcm::operator>>().

The documentation for this struct was generated from the following file:

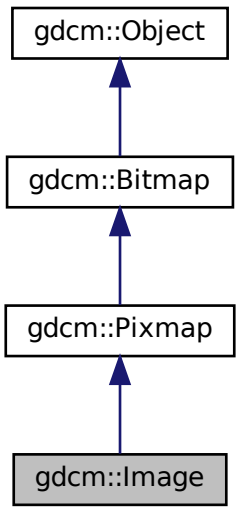
- gdcmElement.h

10.146 gdcm::Image Class Reference

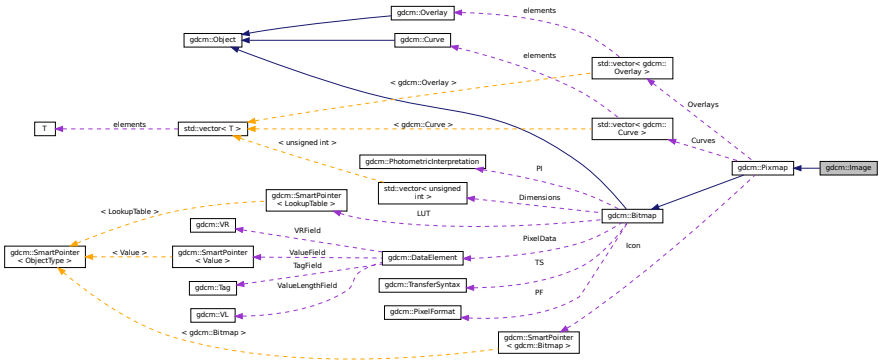
Image.

```
#include <gdcmImage.h>
```

Inheritance diagram for gdcm::Image:



Collaboration diagram for gdcm::Image:



Public Member Functions

- Image ()
- ~Image () override=default
- const double * GetDirectionCosines () const
- double GetDirectionCosines (unsigned int idx) const
- double GetIntercept () const
- const double * GetOrigin () const
- double GetOrigin (unsigned int idx) const
- double GetSlope () const
- const double * GetSpacing () const
- double GetSpacing (unsigned int idx) const
- void Print (std::ostream &os) const override

  print
- void SetDirectionCosines (const double dircos[6])
- void SetDirectionCosines (const float dircos[6])
- void SetDirectionCosines (unsigned int idx, double dircos)
- void SetIntercept (double intercept)

  intercept
- void SetOrigin (const double origin[3])
- void SetOrigin (const float origin[3])
- void SetOrigin (unsigned int idx, double ori)
- void SetSlope (double slope)

  slope
- void SetSpacing (const double spacing[3])
- void SetSpacing (unsigned int idx, double spacing)

Additional Inherited Members

10.146.1 Detailed Description

Image.

This is the container for an Image in the general sense. From this container you should be able to request information like:

- Origin
- Dimension
- PixelFormat ... But also to retrieve the image as a raw buffer (char *) Since we have to deal with both RAW data and JPEG stream (which internally encode all the above information) this API might seems redundant. One way to solve that would be to subclass Image with JPEGImage which would from the stream extract the header info and fill it to please Image...well except origin for instance

Basically you can see it as a storage for the Pixel Data element (7fe0,0010).

Warning

This class does some heuristics to guess the Spacing but is not compatible with DICOM CP-586. In case of doubt use PixmapReader instead

See also

ImageReader PixmapReader

Examples

BasicImageAnonymizer.cs, CompressImage.cxx, CompressLossyJPEG.cs, ConvertToQImage.cxx, CreateARGBImage.cxx, CreateCMYKImage.cxx, DecompressImage.cs, DecompressImageMultiframe.cs, DecompressJPEGFile.cs, ExtractIconFromFile.cxx, FixJAIBugJPEGLS.cxx, GenFakeImage.cxx, GetArray.cs, GetJPEGSamplePrecision.cxx, GetSubSequenceData.cxx, HelloVizWorld.cxx, MpegVideoInfo.cs, PatchFile.cxx, PrintLUT.cxx, ReadMultiTimesException.cxx, RescaleImage.cs, TemplateEmptyImage.cxx, csa2img.cxx, iU22tomultisc.cxx, and threadgdcm.cxx.

10.146.2 Constructor & Destructor Documentation

10.146.2.1 Image()

```
gdcm::Image::Image ( )  [inline]
```

10.146.2.2 ~Image()

```
gdcm::Image::~Image ( )  [override], [default]
```

10.146.3 Member Function Documentation

10.146.3.1 GetDirectionCosines() [1/2]

```
const double * gdcm::Image::GetDirectionCosines ( ) const
```

Return a 6-tuples specifying the direction cosines A default value of (1,0,0,0,1,0) will be return when the direction cosines was not specified.

10.146.3.2 GetDirectionCosines() [2/2]

```
double gdcm::Image::GetDirectionCosines (
            unsigned int idx ) const
```

10.146.3.3 GetIntercept()

```
double gdcm::Image::GetIntercept ( ) const  [inline]
```

10.146.3.4 GetOrigin() [1/2]

```
const double * gdcm::Image::GetOrigin ( ) const
```

Return a 3-tuples specifying the origin Will return (0,0,0) if the origin was not specified.

Examples

HelloVizWorld.cxx.

10.146.3.5 GetOrigin() [2/2]

```
double gdcm::Image::GetOrigin (
            unsigned int idx ) const
```

10.146.3.6 GetSlope()

```
double gdcm::Image::GetSlope ( ) const  [inline]
```

10.146.3.7 GetSpacing() [1/2]

```
const double * gdcm::Image::GetSpacing ( ) const
```

Return a 3-tuples specifying the spacing NOTE: 3rd value can be an arbitrary 1 value when the spacing was not specified (ex. 2D image). WARNING: when the spacing is not specifier, a default value of 1 will be returned

10.146.3.8 GetSpacing() [2/2]

```
double gdcm::Image::GetSpacing (
            unsigned int idx ) const
```

10.146.3.9 Print()

```
void gdcm::Image::Print (
            std::ostream & os ) const  [override], [virtual]
```

print

Reimplemented from gdcm::Bitmap.

Examples

CompressImage.cxx, and PatchFile.cxx.

10.146.3.10 SetDirectionCosines() [1/3]

```
void gdcm::Image::SetDirectionCosines (
            const double dircos[6] )
```

10.146.3.11 SetDirectionCosines() [2/3]

```
void gdcm::Image::SetDirectionCosines (
            const float dircos[6] )
```

10.146.3.12 SetDirectionCosines() [3/3]

```
void gdcm::Image::SetDirectionCosines (
            unsigned int idx,
            double dircos )
```

10.146.3.13 SetIntercept()

```
void gdcm::Image::SetIntercept (
            double intercept )  [inline]
```

intercept

Examples

TemplateEmptyImage.cxx.

10.146.3.14 SetOrigin() [1/3]

```
void gdcm::Image::SetOrigin (
            const double origin[3] )
```

10.146.3.15 SetOrigin() [2/3]

```
void gdcm::Image::SetOrigin (
            const float origin[3] )
```

10.146.3.16 SetOrigin() [3/3]

```
void gdcm::Image::SetOrigin (
            unsigned int idx,
            double ori )
```

10.146.3.17 SetSlope()

```
void gdcm::Image::SetSlope (
            double slope )  [inline]
```

slope

Examples

TemplateEmptyImage.cxx.

10.146.3.18 SetSpacing() [1/2]

```
void gdcm::Image::SetSpacing (
            const double spacing[3] )
```

Examples

csa2img.cxx, and iU22tomultisc.cxx.

10.146.3.19 SetSpacing() [2/2]

```
void gdcm::Image::SetSpacing (
            unsigned int idx,
            double spacing )
```

The documentation for this class was generated from the following file:

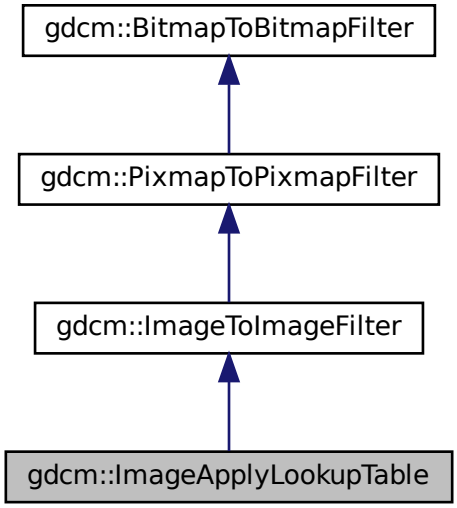
- gdcmImage.h

10.147 gdcm::ImageApplyLookupTable Class Reference

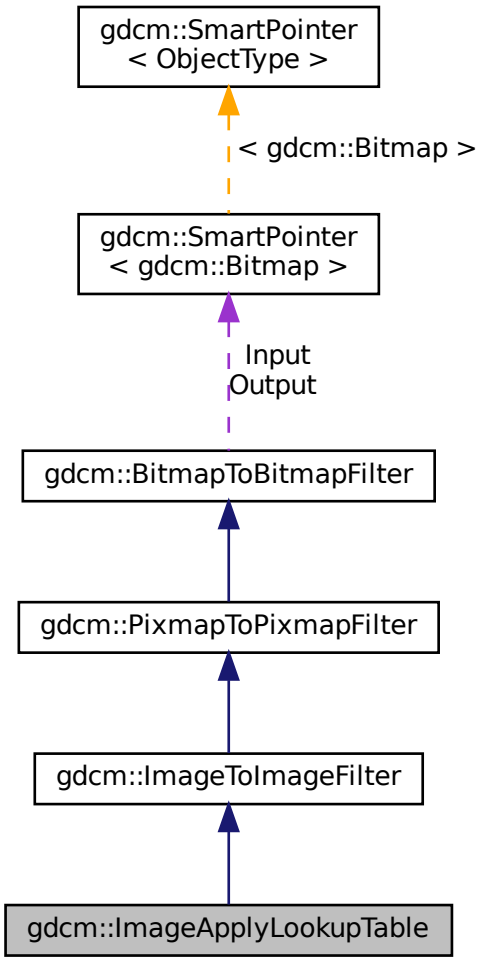
ImageApplyLookupTable class.

```
#include <gdcmImageApplyLookupTable.h>
```

Inheritance diagram for gdcm::ImageApplyLookupTable:



Collaboration diagram for gdcm::ImageApplyLookupTable:



Public Member Functions

- ImageApplyLookupTable ()
- ~ImageApplyLookupTable ()
- bool Apply ()

  Apply.
- void SetRGB8 (bool b)

  RGB8 ?

Additional Inherited Members

10.147.1 Detailed Description

ImageApplyLookupTable class.

It applies the LUT the PixelData (only PALETTE_COLOR images) Output will be a PhotometricInterpretation=RGB image

10.147.2 Constructor & Destructor Documentation

10.147.2.1 ImageApplyLookupTable()

```
gdcm::ImageApplyLookupTable::ImageApplyLookupTable ( )
```

10.147.2.2 ~ImageApplyLookupTable()

```
gdcm::ImageApplyLookupTable::~ImageApplyLookupTable ( )
```

10.147.3 Member Function Documentation

10.147.3.1 Apply()

```
bool gdcm::ImageApplyLookupTable::Apply ( )
```

Apply.

10.147.3.2 SetRGB8()

```
void gdcm::ImageApplyLookupTable::SetRGB8 (
            bool b )
```

RGB8 ?

The documentation for this class was generated from the following file:

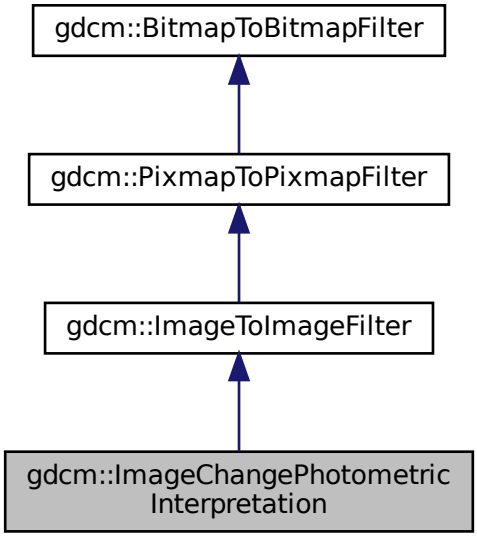
- gdcmImageApplyLookupTable.h

10.148 gdcm::ImageChangePhotometricInterpretation Class Reference

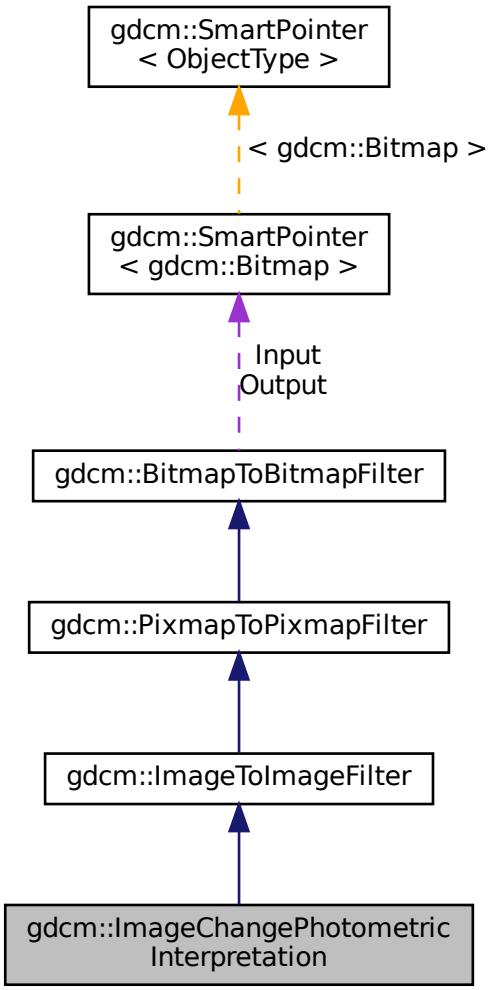
ImageChangePhotometricInterpretation class.

`#include <gdcmImageChangePhotometricInterpretation.h>`

Inheritance diagram for gdcm::ImageChangePhotometricInterpretation:



Collaboration diagram for gdcm::ImageChangePhotometricInterpretation:



Public Member Functions

- ImageChangePhotometricInterpretation ()
- ~ImageChangePhotometricInterpretation ()=default
- bool Change ()

  Change.
- const PhotometricInterpretation & GetPhotometricInterpretation () const
- void SetPhotometricInterpretation (PhotometricInterpretation const &pi)

  Set/Get requested PhotometricInterpretation.

Static Public Member Functions

- template<typename T >
  static void RGB2YBR (T ybr[3], const T rgb[3], unsigned short storedbits=8)
- template<typename T >
  static void YBR2RGB (T rgb[3], const T ybr[3], unsigned short storedbits=8)

Protected Member Functions

- bool ChangeMonochrome ()
- bool ChangeRGB2YBR ()
- bool ChangeYBR2RGB ()

Additional Inherited Members

10.148.1 Detailed Description

ImageChangePhotometricInterpretation class.

Class to change the Photometric Interpretation of an input DICOM

10.148.2 Constructor & Destructor Documentation

10.148.2.1 ImageChangePhotometricInterpretation()

```
gdcm::ImageChangePhotometricInterpretation::ImageChangePhotometricInterpretation ( )  [inline]
```

10.148.2.2 ~ImageChangePhotometricInterpretation()

```
gdcm::ImageChangePhotometricInterpretation::~ImageChangePhotometricInterpretation ( )  [default]
```

10.148.3 Member Function Documentation

10.148.3.1 Change()

```
bool gdcm::ImageChangePhotometricInterpretation::Change ( )
```

Change.

10.148.3.2 ChangeMonochrome()

```
bool gdcm::ImageChangePhotometricInterpretation::ChangeMonochrome ( )  [protected]
```

10.148.3.3 ChangeRGB2YBR()

```
bool gdcm::ImageChangePhotometricInterpretation::ChangeRGB2YBR ( )  [protected]
```

10.148.3.4 ChangeYBR2RGB()

```
bool gdcm::ImageChangePhotometricInterpretation::ChangeYBR2RGB ( )  [protected]
```

10.148.3.5 GetPhotometricInterpretation()

```
const PhotometricInterpretation & gdcm::ImageChangePhotometricInterpretation::GetPhotometric↩
Interpretation ( ) const  [inline]
```

10.148.3.6 RGB2YBR()

```
template<typename T >
void gdcm::ImageChangePhotometricInterpretation::RGB2YBR (
            T ybr[3],
            const T rgb[3],
            unsigned short storedbits = 8 )  [static]
```

colorspace conversion (based on CCIR Recommendation 601-2) -> T.871

References gdcm::Round().

10.148.3.7 SetPhotometricInterpretation()

```
void gdcm::ImageChangePhotometricInterpretation::SetPhotometricInterpretation (
            PhotometricInterpretation const & pi )  [inline]
```

Set/Get requested PhotometricInterpretation.

10.148.3.8 YBR2RGB()

```
template<typename T >
void gdcm::ImageChangePhotometricInterpretation::YBR2RGB (
            T rgb[3],
            const T ybr[3],
            unsigned short storedbits = 8 )  [static]
```

References gdcm::Round().

The documentation for this class was generated from the following file:

- gdcmImageChangePhotometricInterpretation.h

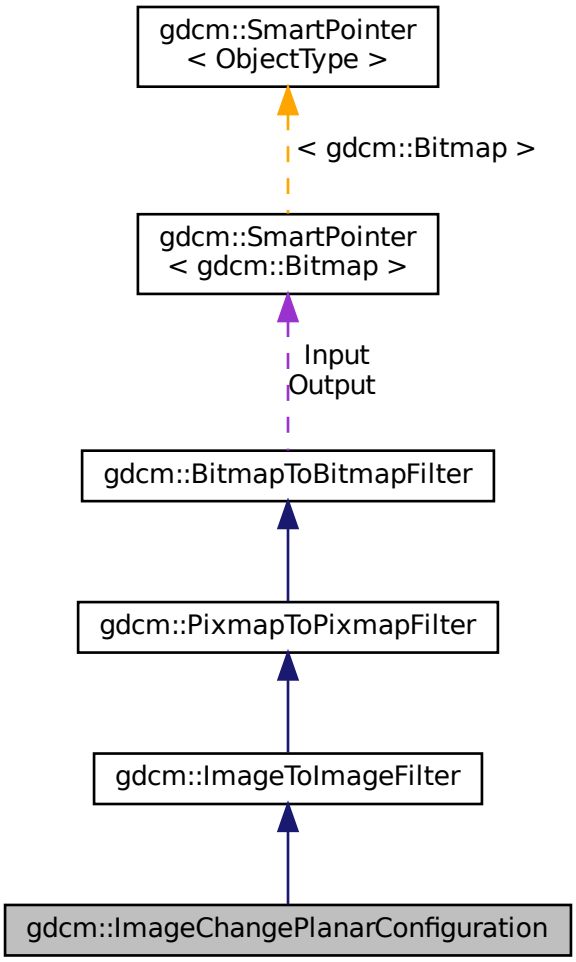
10.149 gdcm::ImageChangePlanarConfiguration Class Reference

ImageChangePlanarConfiguration class.

`#include <gdcmImageChangePlanarConfiguration.h>`

Inheritance diagram for gdcm::ImageChangePlanarConfiguration:

Collaboration diagram for gdcm::ImageChangePlanarConfiguration:



Public Member Functions

- ImageChangePlanarConfiguration ()
- ~ImageChangePlanarConfiguration ()=default
- bool Change ()

  Change.
- unsigned int GetPlanarConfiguration () const
- void SetPlanarConfiguration (unsigned int pc)

  Set/Get requested PlanarConfigation.

Static Public Member Functions

- template<typename T >
  static size_t RGBPixelsToRGBPlanes (T *r, T *g, T *b, const T *rgb, size_t s)
- template<typename T >
  static size_t RGBPlanesToRGBPixels (T *out, const T *r, const T *g, const T *b, size_t s)

Additional Inherited Members

10.149.1 Detailed Description

ImageChangePlanarConfiguration class.

Class to change the Planar configuration of an input DICOM By default it will change into the more usual representation: PlanarConfiguration = 0

10.149.2 Constructor & Destructor Documentation

10.149.2.1 ImageChangePlanarConfiguration()

```
gdcm::ImageChangePlanarConfiguration::ImageChangePlanarConfiguration ( )  [inline]
```

10.149.2.2 ~ImageChangePlanarConfiguration()

```
gdcm::ImageChangePlanarConfiguration::~ImageChangePlanarConfiguration ( )  [default]
```

10.149.3 Member Function Documentation

10.149.3.1 Change()

```
bool gdcm::ImageChangePlanarConfiguration::Change ( )
```

Change.

10.149.3.2 GetPlanarConfiguration()

```
unsigned int gdcm::ImageChangePlanarConfiguration::GetPlanarConfiguration ( ) const  [inline]
```

10.149.3.3 RGBPixelsToRGBPlanes()

```
template<typename T >
size_t gdcm::ImageChangePlanarConfiguration::RGBPixelsToRGBPlanes (
            T * r,
            T * g,
            T * b,
            const T * rgb,
            size_t s )  [static]
```

Convert a regular RGB pixel image (R,G,B,R,G,B...) into a planar R,G,B image (R,R..,G,G...B,B)

Warning

this works on a frame basis, you need to loop over all frames in multiple frames image to apply this function

10.149.3.4 RGBPlanesToRGBPixels()

```
template<typename T >
size_t gdcm::ImageChangePlanarConfiguration::RGBPlanesToRGBPixels (
            T * out,
            const T * r,
            const T * g,
            const T * b,
            size_t s )  [static]
```

s is the size of one plane (r,g or b). Thus the output buffer needs to be at least 3∗s bytes long s can be seen as the number of RGB pixels in the output

10.149.3.5 SetPlanarConfiguration()

```
void gdcm::ImageChangePlanarConfiguration::SetPlanarConfiguration (
            unsigned int pc )  [inline]
```

Set/Get requested PlanarConfigation.

The documentation for this class was generated from the following file:

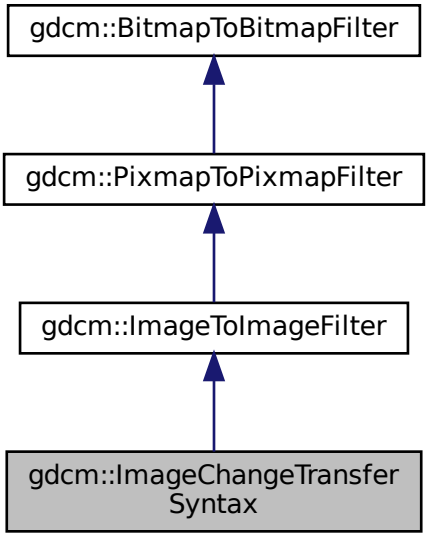
- gdcmImageChangePlanarConfiguration.h

10.150 gdcm::ImageChangeTransferSyntax Class Reference

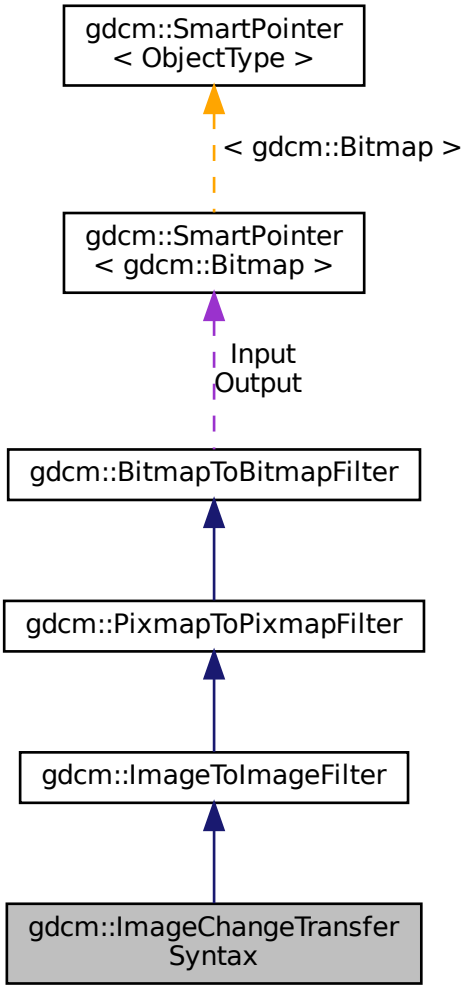
ImageChangeTransferSyntax class.

`#include <gdcmImageChangeTransferSyntax.h>`

Inheritance diagram for gdcm::ImageChangeTransferSyntax:



Collaboration diagram for gdcm::ImageChangeTransferSyntax:



Public Member Functions

- ImageChangeTransferSyntax ()
- ~ImageChangeTransferSyntax ()=default
- bool Change ()

  Change.
- const TransferSyntax & GetTransferSyntax () const

  Get Transfer Syntax.
- void SetCompressIconImage (bool b)
- void SetForce (bool f)
- void SetTransferSyntax (const TransferSyntax &ts)

Set target Transfer Syntax.

- void SetUserCodec (ImageCodec *ic)

Protected Member Functions

- bool TryJPEG2000Codec (const DataElement &pixelde, Bitmap const &input, Bitmap &output)
- bool TryJPEGCodec (const DataElement &pixelde, Bitmap const &input, Bitmap &output)
- bool TryJPEGLSCodec (const DataElement &pixelde, Bitmap const &input, Bitmap &output)
- bool TryRAWCodec (const DataElement &pixelde, Bitmap const &input, Bitmap &output)
- bool TryRLECodec (const DataElement &pixelde, Bitmap const &input, Bitmap &output)

Additional Inherited Members

10.150.1 Detailed Description

ImageChangeTransferSyntax class.

Class to change the transfer syntax of an input DICOM

If only Force param is set but no input TransferSyntax is set, it is assumed that user only wants to inspect encapsulated stream (advanced dev. option).

When using UserCodec it is very important that the TransferSyntax (as set in SetTransferSyntax) is actually understood by UserCodec (ie. UserCodec->CanCode(TransferSyntax)). Otherwise the behavior is to use a default codec.

See also

JPEGCodec JPEGLSCodec JPEG2000Codec

Examples

BasicImageAnonymizer.cs, CompressImage.cxx, CompressLossyJPEG.cs, and StandardizeFiles.cs.

10.150.2 Constructor & Destructor Documentation

10.150.2.1 ImageChangeTransferSyntax()

```
gdcm::ImageChangeTransferSyntax::ImageChangeTransferSyntax ( )  [inline]
```

10.150.2.2 ~ImageChangeTransferSyntax()

```
gdcm::ImageChangeTransferSyntax::~ImageChangeTransferSyntax ( )  [default]
```

10.150.3 Member Function Documentation

10.150.3.1 Change()

```
bool gdcm::ImageChangeTransferSyntax::Change ( )
```

Change.

Examples

BasicImageAnonymizer.cs, CompressImage.cxx, CompressLossyJPEG.cs, and StandardizeFiles.cs.

10.150.3.2 GetTransferSyntax()

```
const TransferSyntax & gdcm::ImageChangeTransferSyntax::GetTransferSyntax ( ) const  [inline]
```

Get Transfer Syntax.

10.150.3.3 SetCompressIconImage()

```
void gdcm::ImageChangeTransferSyntax::SetCompressIconImage (
            bool b )  [inline]
```

Decide whether or not to also compress the Icon Image using the same Transfer Syntax. Default is to simply decompress icon image

Examples

StandardizeFiles.cs.

10.150.3.4 SetForce()

```
void gdcm::ImageChangeTransferSyntax::SetForce (
            bool f )  [inline]
```

When target Transfer Syntax is identical to input target syntax, no operation is actually done. This is an issue when someone wants to re-compress using GDCM internal implementation a JPEG (for example) image

Examples

StandardizeFiles.cs.

10.150.3.5 SetTransferSyntax()

```
void gdcm::ImageChangeTransferSyntax::SetTransferSyntax (
            const TransferSyntax & ts )  [inline]
```

Set target Transfer Syntax.

Examples

BasicImageAnonymizer.cs, CompressImage.cxx, CompressLossyJPEG.cs, and StandardizeFiles.cs.

10.150.3.6 SetUserCodec()

```
void gdcm::ImageChangeTransferSyntax::SetUserCodec (
            ImageCodec * ic )  [inline]
```

Allow user to specify exactly which codec to use. this is needed to specify special qualities or compression option.

Warning

if the codec 'ic' is not compatible with the TransferSyntax requested, it will not be used. It is the user responsibility to check that UserCodec->CanCode(TransferSyntax)

Examples

CompressLossyJPEG.cs.

10.150.3.7 TryJPEG2000Codec()

```
bool gdcm::ImageChangeTransferSyntax::TryJPEG2000Codec (
            const DataElement & pixelde,
            Bitmap const & input,
            Bitmap & output )  [protected]
```

10.150.3.8 TryJPEGCodec()

```
bool gdcm::ImageChangeTransferSyntax::TryJPEGCodec (
            const DataElement & pixelde,
            Bitmap const & input,
            Bitmap & output )  [protected]
```

10.150.3.9 TryJPEGLSCodec()

```
bool gdcm::ImageChangeTransferSyntax::TryJPEGLSCodec (
            const DataElement & pixelde,
            Bitmap const & input,
            Bitmap & output )  [protected]
```

10.150.3.10 TryRAWCodec()

```
bool gdcm::ImageChangeTransferSyntax::TryRAWCodec (
            const DataElement & pixelde,
            Bitmap const & input,
            Bitmap & output )  [protected]
```

10.150.3.11 TryRLECodec()

```
bool gdcm::ImageChangeTransferSyntax::TryRLECodec (
            const DataElement & pixelde,
            Bitmap const & input,
            Bitmap & output )  [protected]
```

The documentation for this class was generated from the following file:

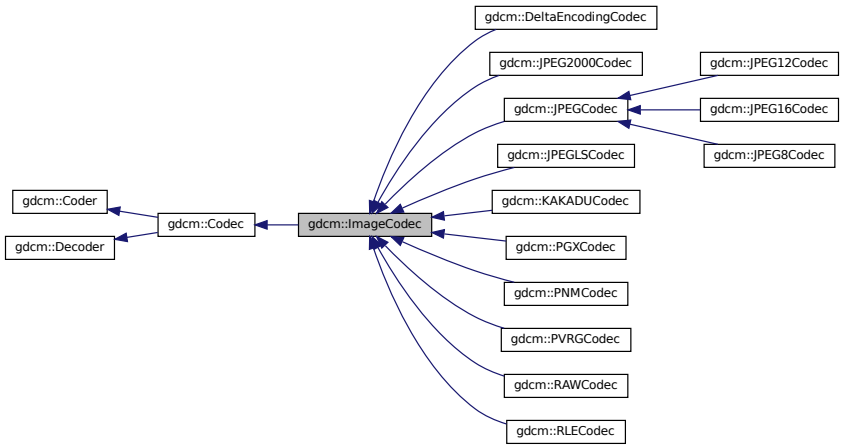
- gdcmImageChangeTransferSyntax.h

10.151 gdcm::ImageCodec Class Reference

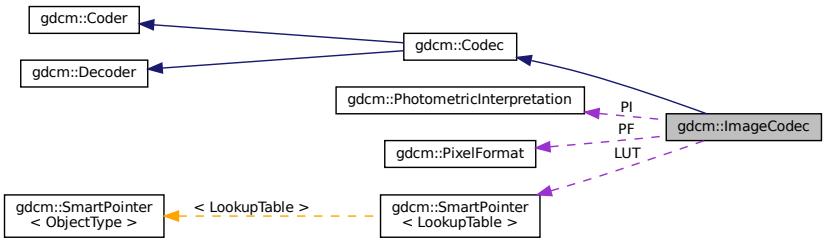
ImageCodec.

`#include <gdcmImageCodec.h>`

Inheritance diagram for gdcm::ImageCodec:



Collaboration diagram for gdcm::ImageCodec:



Public Member Functions

- ImageCodec ()
- ~ImageCodec () override
- bool CanCode (TransferSyntax const &) const override

  Return whether this coder support this transfer syntax (can code it)
- bool CanDecode (TransferSyntax const &) const override

  Return whether this decoder support this transfer syntax (can decode it)
- bool CleanupUnusedBits (char *data, size_t datalen)
- virtual ImageCodec * Clone () const =0
- bool Decode (DataElement const &is_, DataElement &os) override

  Decode.
- const unsigned int * GetDimensions () const
- virtual bool GetHeaderInfo (std::istream &is_, TransferSyntax &ts)
- bool GetLossyFlag () const
- const LookupTable & GetLUT () const
- bool GetNeedByteSwap () const
- unsigned int GetNumberOfDimensions () const
- const PhotometricInterpretation & GetPhotometricInterpretation () const
- PixelFormat & GetPixelFormat ()
- const PixelFormat & GetPixelFormat () const
- unsigned int GetPlanarConfiguration () const
- bool IsLossy () const
- void SetDimensions (const std::vector< unsigned int > &d)
- void SetDimensions (const unsigned int d[3])
- void SetLossyFlag (bool l)
- void SetLUT (LookupTable const &lut)
- void SetNeedByteSwap (bool b)
- void SetNeedOverlayCleanup (bool b)
- void SetNumberOfDimensions (unsigned int dim)
- void SetPhotometricInterpretation (PhotometricInterpretation const &pi)
- virtual void SetPixelFormat (PixelFormat const &pf)
- void SetPlanarConfiguration (unsigned int pc)

Protected Types

- typedef SmartPointer< LookupTable > LUTPtr

Protected Member Functions

- virtual bool AppendFrameEncode (std::ostream &out, const char *data, size_t datalen)
- virtual bool AppendRowEncode (std::ostream &out, const char *data, size_t datalen)
- bool DecodeByStreams (std::istream &is_, std::ostream &os) override
- bool DoByteSwap (std::istream &is_, std::ostream &os)
- bool DoInvertMonochrome (std::istream &is_, std::ostream &os)
- bool DoOverlayCleanup (std::istream &is_, std::ostream &os)
- bool DoPaddedCompositePixelCode (std::istream &is_, std::ostream &os)
- bool DoPlanarConfiguration (std::istream &is_, std::ostream &os)
- bool DoSimpleCopy (std::istream &is_, std::ostream &os)
- bool DoYBR (std::istream &is_, std::ostream &os)
- bool DoYBRFull422 (std::istream &is_, std::ostream &os)
- virtual bool IsFrameEncoder ()
- virtual bool IsRowEncoder ()
- virtual bool IsValid (PhotometricInterpretation const &pi)
- virtual bool StartEncode (std::ostream &os)
- virtual bool StopEncode (std::ostream &os)

Protected Attributes

- unsigned int Dimensions [3]
- bool LossyFlag
- LUTPtr LUT
- bool NeedByteSwap
- bool NeedOverlayCleanup
- unsigned int NumberOfDimensions
- PixelFormat PF
- PhotometricInterpretation PI
- unsigned int PlanarConfiguration
- bool RequestPaddedCompositePixelCode
- bool RequestPlanarConfiguration

Friends

- class FileChangeTransferSyntax
- class ImageChangePhotometricInterpretation

10.151.1 Detailed Description

ImageCodec.

Note

Main codec, this is a central place for all implementation

Examples

FileChangeTSLossy.cs.

10.151.2 Member Typedef Documentation

10.151.2.1 LUTPtr

```
typedef SmartPointer<LookupTable> gdcm::ImageCodec::LUTPtr  [protected]
```

10.151.3 Constructor & Destructor Documentation

10.151.3.1 ImageCodec()

```
gdcm::ImageCodec::ImageCodec ( )
```

10.151.3.2 ~ImageCodec()

```
gdcm::ImageCodec::~ImageCodec ( )  [override]
```

10.151.4 Member Function Documentation

10.151.4.1 AppendFrameEncode()

```
virtual bool gdcm::ImageCodec::AppendFrameEncode (
            std::ostream & out,
            const char * data,
            size_t datalen )  [protected], [virtual]
```

Reimplemented in gdcm::JPEG2000Codec, gdcm::JPEGCodec, gdcm::JPEGLSCodec, and gdcm::RLECodec.

10.151.4.2 AppendRowEncode()

```
virtual bool gdcm::ImageCodec::AppendRowEncode (
            std::ostream & out,
            const char * data,
            size_t datalen )  [protected], [virtual]
```

Reimplemented in gdcm::JPEG2000Codec, gdcm::JPEGCodec, gdcm::JPEGLSCodec, and gdcm::RLECodec.

10.151.4.3 CanCode()

```
bool gdcm::ImageCodec::CanCode (
            TransferSyntax const &  ) const  [inline], [override], [virtual]
```

Return whether this coder support this transfer syntax (can code it)

Implements gdcm::Coder.

Reimplemented in gdcm::JPEG2000Codec, gdcm::JPEGCodec, gdcm::JPEGLSCodec, gdcm::KAKADUCodec, gdcm::PGXCodec, gdcm::PNMCodec, gdcm::PVRGCodec, gdcm::RAWCodec, and gdcm::RLECodec.

10.151.4.4 CanDecode()

```
bool gdcm::ImageCodec::CanDecode (
            TransferSyntax const &  ) const  [inline], [override], [virtual]
```

Return whether this decoder support this transfer syntax (can decode it)

Implements gdcm::Decoder.

Reimplemented in gdcm::JPEG2000Codec, gdcm::JPEGCodec, gdcm::JPEGLSCodec, gdcm::KAKADUCodec, gdcm::PGXCodec, gdcm::PNMCodec, gdcm::PVRGCodec, gdcm::RAWCodec, and gdcm::RLECodec.

10.151.4.5 CleanupUnusedBits()

```
bool gdcm::ImageCodec::CleanupUnusedBits (
            char * data,
            size_t datalen )
```

10.151.4.6 Clone()

```
virtual ImageCodec * gdcm::ImageCodec::Clone ( ) const  [pure virtual]
```

Implemented in gdcm::JPEG2000Codec, gdcm::JPEGCodec, gdcm::JPEGLSCodec, gdcm::KAKADUCodec, gdcm::PGXCodec, gdcm::PNMCodec, gdcm::PVRGCodec, gdcm::RAWCodec, and gdcm::RLECodec.

10.151.4.7 Decode()

```
bool gdcm::ImageCodec::Decode (
            DataElement const & ,
            DataElement &  )  [override], [virtual]
```

Decode.

Reimplemented from gdcm::Decoder.

Reimplemented in gdcm::JPEG2000Codec, gdcm::JPEGCodec, gdcm::JPEGLSCodec, gdcm::KAKADUCodec, gdcm::PVRGCodec, gdcm::RAWCodec, and gdcm::RLECodec.

10.151.4.8 DecodeByStreams()

```
bool gdcm::ImageCodec::DecodeByStreams (
            std::istream & is_,
            std::ostream & os )  [override], [protected], [virtual]
```

Reimplemented from gdcm::Decoder.

Reimplemented in gdcm::JPEG12Codec, gdcm::JPEG16Codec, gdcm::JPEG2000Codec, gdcm::JPEG8Codec, gdcm::JPEGCodec, gdcm::RAWCodec, and gdcm::RLECodec.

10.151.4.9 DoByteSwap()

```
bool gdcm::ImageCodec::DoByteSwap (
            std::istream & is_,
            std::ostream & os )  [protected]
```

10.151.4.10 DoInvertMonochrome()

```
bool gdcm::ImageCodec::DoInvertMonochrome (
            std::istream & is_,
            std::ostream & os )  [protected]
```

10.151.4.11 DoOverlayCleanup()

```
bool gdcm::ImageCodec::DoOverlayCleanup (
            std::istream & is_,
            std::ostream & os )  [protected]
```

10.151.4.12 DoPaddedCompositePixelCode()

```
bool gdcm::ImageCodec::DoPaddedCompositePixelCode (
            std::istream & is_,
            std::ostream & os )  [protected]
```

10.151.4.13 DoPlanarConfiguration()

```
bool gdcm::ImageCodec::DoPlanarConfiguration (
            std::istream & is_,
            std::ostream & os )  [protected]
```

10.151.4.14 DoSimpleCopy()

```
bool gdcm::ImageCodec::DoSimpleCopy (
            std::istream & is_,
            std::ostream & os )  [protected]
```

10.151.4.15 DoYBR()

```
bool gdcm::ImageCodec::DoYBR (
            std::istream & is_,
            std::ostream & os )  [protected]
```

10.151.4.16 DoYBRFull422()

```
bool gdcm::ImageCodec::DoYBRFull422 (
            std::istream & is_,
            std::ostream & os )  [protected]
```

10.151.4.17 GetDimensions()

```
const unsigned int * gdcm::ImageCodec::GetDimensions ( ) const  [inline]
```

10.151.4.18 GetHeaderInfo()

```
virtual bool gdcm::ImageCodec::GetHeaderInfo (
            std::istream & is_,
            TransferSyntax & ts )  [virtual]
```

Reimplemented in gdcm::JPEG12Codec, gdcm::JPEG16Codec, gdcm::JPEG2000Codec, gdcm::JPEG8Codec, gdcm::JPEGCodec, gdcm::JPEGLSCodec, gdcm::PGXCodec, gdcm::PNMCodec, gdcm::RAWCodec, and gdcm::RLECodec.

10.151.4.19 GetLossyFlag()

```
bool gdcm::ImageCodec::GetLossyFlag ( ) const
```

10.151.4.20 GetLUT()

```
const LookupTable & gdcm::ImageCodec::GetLUT ( ) const  [inline]
```

10.151.4.21 GetNeedByteSwap()

```
bool gdcm::ImageCodec::GetNeedByteSwap ( ) const  [inline]
```

10.151.4.22 GetNumberOfDimensions()

```
unsigned int gdcm::ImageCodec::GetNumberOfDimensions ( ) const
```

10.151.4.23 GetPhotometricInterpretation()

```
const PhotometricInterpretation & gdcm::ImageCodec::GetPhotometricInterpretation ( ) const
```

10.151.4.24 GetPixelFormat() [1/2]

```
PixelFormat & gdcm::ImageCodec::GetPixelFormat ( )  [inline]
```

Examples

GetJPEGSamplePrecision.cxx.

10.151.4.25 GetPixelFormat() [2/2]

```
const PixelFormat & gdcm::ImageCodec::GetPixelFormat ( ) const  [inline]
```

10.151.4.26 GetPlanarConfiguration()

```
unsigned int gdcm::ImageCodec::GetPlanarConfiguration ( ) const  [inline]
```

10.151.4.27 IsFrameEncoder()

```
virtual bool gdcm::ImageCodec::IsFrameEncoder ( )  [protected], [virtual]
```

Reimplemented in gdcm::JPEG2000Codec, gdcm::JPEGCodec, gdcm::JPEGLSCodec, and gdcm::RLECodec.

10.151.4.28 IsLossy()

```
bool gdcm::ImageCodec::IsLossy ( ) const
```

10.151.4.29 IsRowEncoder()

```
virtual bool gdcm::ImageCodec::IsRowEncoder ( )  [protected], [virtual]
```

Reimplemented in gdcm::JPEG2000Codec, gdcm::JPEGCodec, gdcm::JPEGLSCodec, and gdcm::RLECodec.

10.151.4.30 IsValid()

```
virtual bool gdcm::ImageCodec::IsValid (
            PhotometricInterpretation const & pi )  [protected], [virtual]
```

Reimplemented in gdcm::JPEGCodec.

10.151.4.31 SetDimensions() [1/2]

```
void gdcm::ImageCodec::SetDimensions (
            const std::vector< unsigned int > & d )
```

10.151.4.32 SetDimensions() [2/2]

```
void gdcm::ImageCodec::SetDimensions (
            const unsigned int d[3] )
```

Examples

ExtractIconFromFile.cxx.

10.151.4.33 SetLossyFlag()

```
void gdcm::ImageCodec::SetLossyFlag (
            bool l )
```

10.151.4.34 SetLUT()

```
void gdcm::ImageCodec::SetLUT (
            LookupTable const & lut )  [inline]
```

Examples

ExtractIconFromFile.cxx.

10.151.4.35 SetNeedByteSwap()

```
void gdcm::ImageCodec::SetNeedByteSwap (
            bool b )  [inline]
```

10.151.4.36 SetNeedOverlayCleanup()

```
void gdcm::ImageCodec::SetNeedOverlayCleanup (
            bool b )  [inline]
```

10.151.4.37 SetNumberOfDimensions()

```
void gdcm::ImageCodec::SetNumberOfDimensions (
            unsigned int dim )
```

10.151.4.38 SetPhotometricInterpretation()

```
void gdcm::ImageCodec::SetPhotometricInterpretation (
            PhotometricInterpretation const & pi )
```

Examples

ExtractIconFromFile.cxx.

10.151.4.39 SetPixelFormat()

```
virtual void gdcm::ImageCodec::SetPixelFormat (
            PixelFormat const & pf )  [inline], [virtual]
```

Reimplemented in gdcm::JPEGCodec.

Examples

ExtractIconFromFile.cxx.

10.151.4.40 SetPlanarConfiguration()

```
void gdcm::ImageCodec::SetPlanarConfiguration (
            unsigned int pc )  [inline]
```

10.151.4.41 StartEncode()

```
virtual bool gdcm::ImageCodec::StartEncode (
            std::ostream & os )  [protected], [virtual]
```

Reimplemented in gdcm::JPEG2000Codec, gdcm::JPEGCodec, gdcm::JPEGLSCodec, and gdcm::RLECodec.

10.151.4.42 StopEncode()

```
virtual bool gdcm::ImageCodec::StopEncode (
            std::ostream & os )  [protected], [virtual]
```

Reimplemented in gdcm::JPEG2000Codec, gdcm::JPEGCodec, gdcm::JPEGLSCodec, and gdcm::RLECodec.

10.151.5 Friends And Related Function Documentation

10.151.5.1 FileChangeTransferSyntax

```
friend class FileChangeTransferSyntax  [friend]
```

This is a high level API to encode in a streaming fashion. Each plugin will handle differently the caching mechanism so that a limited memory is used when compressing dataset. Codec will fall into two categories:

- Full row encoder: only a single scanline (row) of data is needed to be loaded at a time;
- Full frame encoder (default): a complete frame (row x col) is needed to be loaded at a time

10.151.5.2 ImageChangePhotometricInterpretation

```
friend class ImageChangePhotometricInterpretation  [friend]
```

10.151.6 Member Data Documentation

10.151.6.1 Dimensions

```
unsigned int gdcm::ImageCodec::Dimensions[3]  [protected]
```

10.151.6.2 LossyFlag

```
bool gdcm::ImageCodec::LossyFlag  [protected]
```

10.151.6.3 LUT

```
LUTPtr gdcm::ImageCodec::LUT  [protected]
```

10.151.6.4 NeedByteSwap

```
bool gdcm::ImageCodec::NeedByteSwap  [protected]
```

10.151.6.5 NeedOverlayCleanup

```
bool gdcm::ImageCodec::NeedOverlayCleanup  [protected]
```

10.151.6.6 NumberOfDimensions

```
unsigned int gdcm::ImageCodec::NumberOfDimensions  [protected]
```

10.151.6.7 PF

```
PixelFormat gdcm::ImageCodec::PF  [protected]
```

10.151.6.8 PI

```
PhotometricInterpretation gdcm::ImageCodec::PI  [protected]
```

10.151.6.9 PlanarConfiguration

```
unsigned int gdcm::ImageCodec::PlanarConfiguration  [protected]
```

10.151.6.10 RequestPaddedCompositePixelCode

```
bool gdcm::ImageCodec::RequestPaddedCompositePixelCode  [protected]
```

10.151.6.11 RequestPlanarConfiguration

```
bool gdcm::ImageCodec::RequestPlanarConfiguration  [protected]
```

The documentation for this class was generated from the following file:

- gdcmImageCodec.h

10.152 gdcm::ImageConverter Class Reference

Image Converter.

```
#include <gdcmImageConverter.h>
```

Public Member Functions

- ImageConverter ()
- ~ImageConverter ()
- void Convert ()
- const Image & GetOuput () const
- void SetInput (Image const &input)

10.152.1 Detailed Description

Image Converter.

Note

This is the class used to convert from on Image to another This is typically used to convert let say YBR JPEG compressed Image to a RAW RGB Image. So that the buffer can be directly pass to third party application. This filter is application level and not integrated directly in GDCM

10.152.2 Constructor & Destructor Documentation

10.152.2.1 ImageConverter()

```
gdcm::ImageConverter::ImageConverter ( )
```

10.152.2.2 ~ImageConverter()

```
gdcm::ImageConverter::~ImageConverter ( )
```

10.152.3 Member Function Documentation

10.152.3.1 Convert()

```
void gdcm::ImageConverter::Convert ( )
```

10.152.3.2 GetOuput()

```
const Image & gdcm::ImageConverter::GetOuput ( ) const
```

10.152.3.3 SetInput()

```
void gdcm::ImageConverter::SetInput (
            Image const & input )
```

The documentation for this class was generated from the following file:

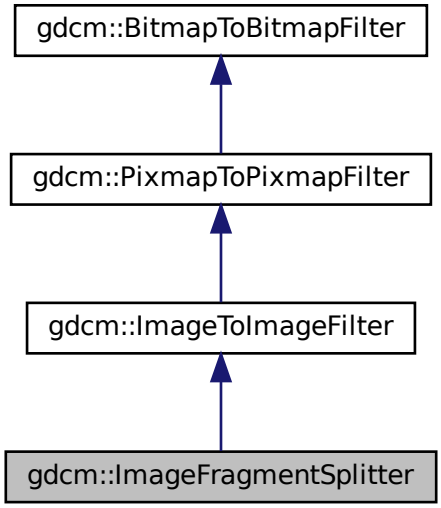
- gdcmImageConverter.h

10.153 gdcm::ImageFragmentSplitter Class Reference

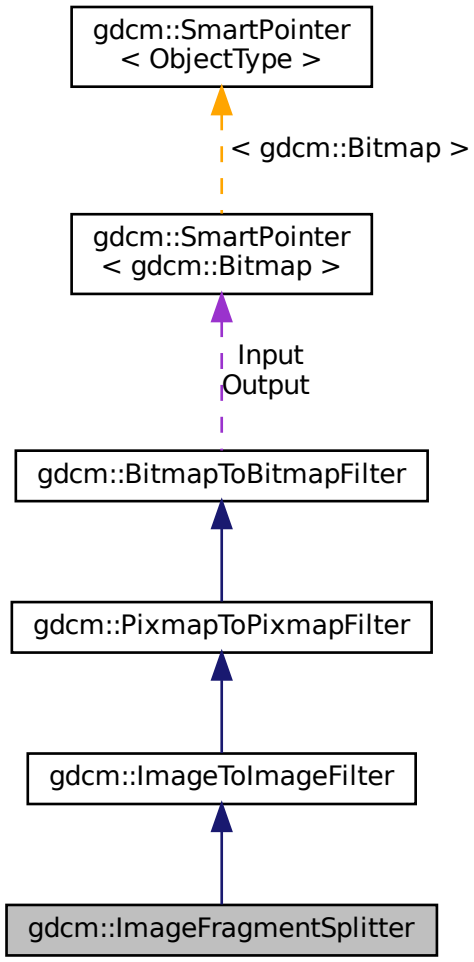
ImageFragmentSplitter class.

`#include <gdcmImageFragmentSplitter.h>`

Inheritance diagram for gdcm::ImageFragmentSplitter:



Collaboration diagram for gdcm::ImageFragmentSplitter:



Public Member Functions

- ImageFragmentSplitter ()
- ~ImageFragmentSplitter ()=default
- unsigned int GetFragmentSizeMax () const
- void SetForce (bool f)
- void SetFragmentSizeMax (unsigned int fragsize)

  FragmentSizeMax needs to be an even number.
- bool Split ()

  Split.

Additional Inherited Members

10.153.1 Detailed Description

ImageFragmentSplitter class.

For single frame image, DICOM standard allow splitting the frame into multiple fragments

10.153.2 Constructor & Destructor Documentation

10.153.2.1 ImageFragmentSplitter()

```
gdcm::ImageFragmentSplitter::ImageFragmentSplitter ( )  [inline]
```

10.153.2.2 ~ImageFragmentSplitter()

```
gdcm::ImageFragmentSplitter::~ImageFragmentSplitter ( )  [default]
```

10.153.3 Member Function Documentation

10.153.3.1 GetFragmentSizeMax()

```
unsigned int gdcm::ImageFragmentSplitter::GetFragmentSizeMax ( ) const  [inline]
```

10.153.3.2 SetForce()

```
void gdcm::ImageFragmentSplitter::SetForce (
            bool f )  [inline]
```

When file already has all it's segment < FragmentSizeMax there is not need to run the filter. Unless the user explicitly say 'force' recomputation !

10.153.3.3 SetFragmentSizeMax()

```
void gdcm::ImageFragmentSplitter::SetFragmentSizeMax (
            unsigned int fragsize )
```

FragmentSizeMax needs to be an even number.

10.153.3.4 Split()

```
bool gdcm::ImageFragmentSplitter::Split ( )
```

Split.

The documentation for this class was generated from the following file:

- gdcmImageFragmentSplitter.h

10.154 gdcm::ImageHelper Class Reference

ImageHelper (internal class, not intended for user level)

```
#include <gdcmImageHelper.h>
```

Static Public Member Functions

- static MediaStorage ComputeMediaStorageFromModality (const char *modality, unsigned int dimension=2, PixelFormat const &pf=PixelFormat(), PhotometricInterpretation const &pi=PhotometricInterpretation(), double rescaleintercept=0, double rescaleslope=1)

  Moved from MediaStorage here, since we need extra info stored in PixelFormat & PhotometricInterpretation.
- static bool ComputeSpacingFromImagePositionPatient (const std::vector< double > &imageposition, std::vector< double > &spacing)

  DO NOT USE.
- static std::vector< unsigned int > GetDimensionsValue (const File &f)
- static bool GetDirectionCosinesFromDataSet (DataSet const &ds, std::vector< double > &dircos)
- static std::vector< double > GetDirectionCosinesValue (File const &f)
- static bool GetForcePixelSpacing ()
- static bool GetForceRescaleInterceptSlope ()
- static SmartPointer< LookupTable > GetLUT (File const &f)

  returns the lookup table of an image file
- static std::vector< double > GetOriginValue (File const &f)

  Set/Get Origin (IPP) from/to a file.
- static PhotometricInterpretation GetPhotometricInterpretationValue (File const &f)
- static PixelFormat GetPixelFormatValue (const File &f)
- static unsigned int GetPlanarConfigurationValue (const File &f)

- static bool GetPMSRescaleInterceptSlope ()
- static const ByteValue * GetPointerFromElement (Tag const &tag, File const &f)
- static bool GetRealWorldValueMappingContent (File const &f, RealWorldValueMappingContent &rwvmc)
- static std::vector< double > GetRescaleInterceptSlopeValue (File const &f)
- static std::vector< double > GetSpacingValue (File const &f)

  Set/Get Spacing from/to a File.
- static void SetDimensionsValue (File &f, const Pixmap &img)
- static void SetDirectionCosinesValue (DataSet &ds, const std::vector< double > &dircos)
- static void SetForcePixelSpacing (bool)
- static void SetForceRescaleInterceptSlope (bool)
- static void SetOriginValue (DataSet &ds, const Image &img)
- static void SetPMSRescaleInterceptSlope (bool)
- static void SetRescaleInterceptSlopeValue (File &f, const Image &img)
- static void SetSpacingValue (DataSet &ds, const std::vector< double > &spacing)

Static Protected Member Functions

- static Tag GetSpacingTagFromMediaStorage (MediaStorage const &ms)
- static Tag GetZSpacingTagFromMediaStorage (MediaStorage const &ms)

10.154.1 Detailed Description

ImageHelper (internal class, not intended for user level)

Helper for writing World images in DICOM. DICOM has a 'template' approach to image where MR Image Storage are distinct object from Enhanced MR Image Storage. For example the Pixel Spacing in one object is not at the same position (ie Tag) as in the other this class is the central (read: fragile) place where all the dispatching is done from a unified view of a world image (typically VTK or ITK point of view) down to the low level DICOM point of view.

Warning

: do not expect the API of this class to be maintained at any point, since as Modalities are added the API might have to be augmented or behavior changed to cope with new modalities.

Examples

ExtractImageRegion.cs, ExtractImageRegionWithLUT.cs, and ExtractOneFrame.cs.

10.154.2 Member Function Documentation

10.154.2.1 ComputeMediaStorageFromModality()

```
static MediaStorage gdcm::ImageHelper::ComputeMediaStorageFromModality (
            const char * modality,
            unsigned int dimension = 2,
            PixelFormat const & pf = PixelFormat(),
            PhotometricInterpretation const & pi = PhotometricInterpretation(),
            double rescaleintercept = 0,
            double rescaleslope = 1 )  [static]
```

Moved from MediaStorage here, since we need extra info stored in PixelFormat & PhotometricInterpretation.

10.154.2.2 ComputeSpacingFromImagePositionPatient()

```
static bool gdcm::ImageHelper::ComputeSpacingFromImagePositionPatient (
            const std::vector< double > & imageposition,
            std::vector< double > & spacing )  [static]
```

DO NOT USE.

10.154.2.3 GetDimensionsValue()

```
static std::vector< unsigned int > gdcm::ImageHelper::GetDimensionsValue (
            const File & f )  [static]
```

This function checks tags (0x0028, 0x0010) and (0x0028, 0x0011) for the rows and columns of the image in pixels (as opposed to actual distances). The output is {col , row}

Examples

ExtractImageRegion.cs, ExtractImageRegionWithLUT.cs, ExtractOneFrame.cs, Extracting_All_Resolution.cxx, Fake_Image_Using_Stream_Image_Writer.cxx, and StreamImageReaderTest.cxx.

10.154.2.4 GetDirectionCosinesFromDataSet()

```
static bool gdcm::ImageHelper::GetDirectionCosinesFromDataSet (
            DataSet const & ds,
            std::vector< double > & dircos )  [static]
```

10.154.2.5 GetDirectionCosinesValue()

```
static std::vector< double > gdcm::ImageHelper::GetDirectionCosinesValue (
            File const & f )  [static]
```

Get Direction Cosines (IOP) from/to a file Requires a file because mediastorage must be known

10.154.2.6 GetForcePixelSpacing()

```
static bool gdcm::ImageHelper::GetForcePixelSpacing ( )  [static]
```

10.154.2.7 GetForceRescaleInterceptSlope()

```
static bool gdcm::ImageHelper::GetForceRescaleInterceptSlope ( )  [static]
```

10.154.2.8 GetLUT()

```
static SmartPointer< LookupTable > gdcm::ImageHelper::GetLUT (
            File const & f )  [static]
```

returns the lookup table of an image file

10.154.2.9 GetOriginValue()

```
static std::vector< double > gdcm::ImageHelper::GetOriginValue (
            File const & f )  [static]
```

Set/Get Origin (IPP) from/to a file.

10.154.2.10 GetPhotometricInterpretationValue()

```
static PhotometricInterpretation gdcm::ImageHelper::GetPhotometricInterpretationValue (
            File const & f )  [static]
```

Examples

ExtractImageRegion.cs.

10.154.2.11 GetPixelFormatValue()

```
static PixelFormat gdcm::ImageHelper::GetPixelFormatValue (
            const File & f )  [static]
```

This function returns pixel information about an image from its dataset That includes samples per pixel and bit depth (in that order)

10.154.2.12 GetPlanarConfigurationValue()

```
static unsigned int gdcm::ImageHelper::GetPlanarConfigurationValue (
            const File & f )  [static]
```

10.154.2.13 GetPMSRescaleInterceptSlope()

```
static bool gdcm::ImageHelper::GetPMSRescaleInterceptSlope ( )  [static]
```

10.154.2.14 GetPointerFromElement()

```
static const ByteValue * gdcm::ImageHelper::GetPointerFromElement (
            Tag const & tag,
            File const & f )  [static]
```

10.154.2.15 GetRealWorldValueMappingContent()

```
static bool gdcm::ImageHelper::GetRealWorldValueMappingContent (
            File const & f,
            RealWorldValueMappingContent & rwvmc )  [static]
```

10.154.2.16 GetRescaleInterceptSlopeValue()

```
static std::vector< double > gdcm::ImageHelper::GetRescaleInterceptSlopeValue (
            File const & f )  [static]
```

Set/Get shift/scale from/to a file

Warning

this function reads/sets the Slope/Intercept in appropriate class storage, but also Grid Scaling in RT Dose Storage Can't take a dataset because the mediastorage of the file must be known

10.154.2.17 GetSpacingTagFromMediaStorage()

```
static Tag gdcm::ImageHelper::GetSpacingTagFromMediaStorage (
            MediaStorage const & ms )  [static], [protected]
```

10.154.2.18 GetSpacingValue()

```
static std::vector< double > gdcm::ImageHelper::GetSpacingValue (
            File const & f )  [static]
```

Set/Get Spacing from/to a File.

10.154.2.19 GetZSpacingTagFromMediaStorage()

```
static Tag gdcm::ImageHelper::GetZSpacingTagFromMediaStorage (
            MediaStorage const & ms )  [static], [protected]
```

10.154.2.20 SetDimensionsValue()

```
static void gdcm::ImageHelper::SetDimensionsValue (
            File & f,
            const Pixmap & img )  [static]
```

10.154.2.21 SetDirectionCosinesValue()

```
static void gdcm::ImageHelper::SetDirectionCosinesValue (
            DataSet & ds,
            const std::vector< double > & dircos )  [static]
```

Set Direction Cosines (IOP) from/to a file When IOD does not defines what is IOP (eg. typically Secondary Capture Image Storage) this call will simply remove the IOP attribute. Else in case of MR/CT image storage, this call will properly lookup the correct attribute to store the IOP.

10.154.2.22 SetForcePixelSpacing()

```
static void gdcm::ImageHelper::SetForcePixelSpacing (
            bool  )  [static]
```

GDCM 1.x compatibility issue: When using ReWrite an MR Image Storage would be rewritten as Secondary Capture Object while still having a Pixel Spacing tag (0028,0030). If you have deal with those files, use this very special flag to handle them Unless explicitly set elsewhere by the standard, it will use value from 0028,0030 / 0018,0088 for the Pixel Spacing of the Image

10.154.2.23 SetForceRescaleInterceptSlope()

```
static void gdcm::ImageHelper::SetForceRescaleInterceptSlope (
            bool  )  [static]
```

GDCM 1.x compatibility issue: Do not use anymore. This hack was used for some MR Image Storage generated by Philips Modality. When "Combine MR Rescaling" is set to TRUE, rescaling is removed. But when set to FALSE, the Modality LUT was exported. Internally GDCM now handles this gracefully.

10.154.2.24 SetOriginValue()

```
static void gdcm::ImageHelper::SetOriginValue (
            DataSet & ds,
            const Image & img )  [static]
```

10.154.2.25 SetPMSRescaleInterceptSlope()

```
static void gdcm::ImageHelper::SetPMSRescaleInterceptSlope (
            bool  )  [static]
```

Since GDCM 2.6.1 Philips Medical System are read using the Private Field For Rescale Slope/Intercept by default. This mechanism can be deactivated using the following API: This option has no effect when ForceRescaleInterceptSlope is set to true GDCM will only read those private attribute but never write them out.

10.154.2.26 SetRescaleInterceptSlopeValue()

```
static void gdcm::ImageHelper::SetRescaleInterceptSlopeValue (
            File & f,
            const Image & img )  [static]
```

10.154.2.27 SetSpacingValue()

```
static void gdcm::ImageHelper::SetSpacingValue (
             DataSet & ds,
             const std::vector< double > & spacing )  [static]
```

The documentation for this class was generated from the following file:

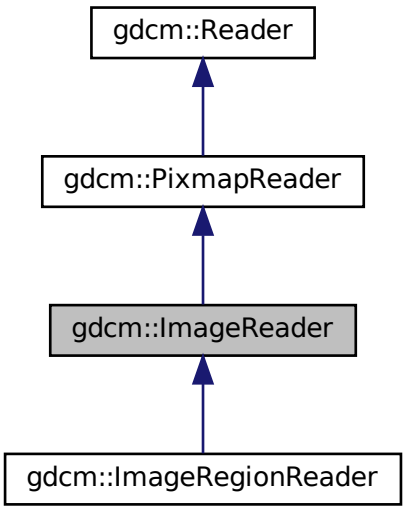
- gdcmImageHelper.h

10.155 gdcm::ImageReader Class Reference

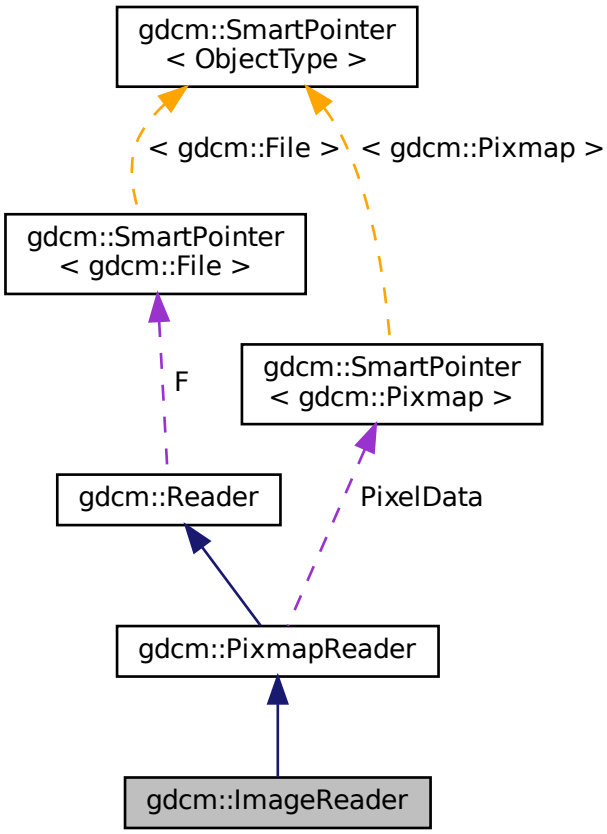
ImageReader.

```
#include <gdcmImageReader.h>
```

Inheritance diagram for gdcm::ImageReader:



Collaboration diagram for gdcm::ImageReader:



Public Member Functions

- ImageReader ()
- ~ImageReader () override
- Image & GetImage ()
- const Image & GetImage () const

  Return the read image.
- bool Read () override

Protected Member Functions

- bool ReadACRNEMAImage () override
- bool ReadImage (MediaStorage const &ms) override

Additional Inherited Members

10.155.1 Detailed Description

ImageReader.

Note

its role is to convert the DICOM DataSet into a Image representation Image is different from Pixmap has it has a position and a direction in Space.

See also

Image

Examples

BasicImageAnonymizer.cs, CheckBigEndianBug.cxx, CompressImage.cxx, CompressLossyJPEG.cs, ConvertToQImage.cxx, DecompressImage.cs, ExtractIconFromFile.cxx, FixBrokenJ2K.cxx, FixJAIBugJPEGLS.cxx, GetArray.cs, GetJPEGSamplePrecision.cxx, HelloVizWorld.cxx, MergeTwoFiles.cxx, MpegVideoInfo.cs, MrProtocol.cxx, PatchFile.cxx, PrintLUT.cxx, ReadMultiTimesException.cxx, RescaleImage.cs, and threadgdcm.cxx.

10.155.2 Constructor & Destructor Documentation

10.155.2.1 ImageReader()

```
gdcm::ImageReader::ImageReader ( )
```

10.155.2.2 ~ImageReader()

```
gdcm::ImageReader::~ImageReader ( )  [override]
```

10.155.3 Member Function Documentation

10.155.3.1 GetImage() [1/2]

```
Image & gdcm::ImageReader::GetImage ( )
```

10.155.3.2 GetImage() [2/2]

```
const Image & gdcm::ImageReader::GetImage ( ) const
```

Return the read image.

Examples

BasicImageAnonymizer.cs, CompressImage.cxx, CompressLossyJPEG.cs, ConvertToQImage.cxx, DecompressImage.cs, ExtractIconFromFile.cxx, ExtractImageRegionWithLUT.cs, FixJAIBugJPEGLS.cxx, GetArray.cs, GetJPEGSamplePrecision.cxx, HelloVizWorld.cxx, MergeTwoFiles.cxx, MpegVideoInfo.cs, PatchFile.cxx, PrintLUT.cxx, ReadMultiTimesException.cxx, RescaleImage.cs, TemplateEmptyImage.cxx, and threadgdcm.cxx.

10.155.3.3 Read()

```
bool gdcm::ImageReader::Read ( )  [override], [virtual]
```

Read the DICOM image. There are two reason for failure:

1. The input filename is not DICOM
2. The input DICOM file does not contains an Image.

Reimplemented from gdcm::Reader.

Reimplemented in gdcm::ImageRegionReader.

Examples

BasicImageAnonymizer.cs, CheckBigEndianBug.cxx, CompressImage.cxx, CompressLossyJPEG.cs, ConvertToQImage.cxx, DecompressImage.cs, ExtractIconFromFile.cxx, FixBrokenJ2K.cxx, FixJAIBugJPEGLS.cxx, GetArray.cs, GetJPEGSamplePrecision.cxx, HelloVizWorld.cxx, MergeTwoFiles.cxx, MrProtocol.cxx, PatchFile.cxx, PrintLUT.cxx, ReadMultiTimesException.cxx, RescaleImage.cs, and threadgdcm.cxx.

10.155.3.4 ReadACRNEMAImage()

```
bool gdcm::ImageReader::ReadACRNEMAImage ( )  [override], [protected], [virtual]
```

Reimplemented from gdcm::PixmapReader.

10.155.3.5 ReadImage()

```
bool gdcm::ImageReader::ReadImage (
            MediaStorage const & ms )  [override], [protected], [virtual]
```

Reimplemented from gdcm::PixmapReader.

The documentation for this class was generated from the following file:

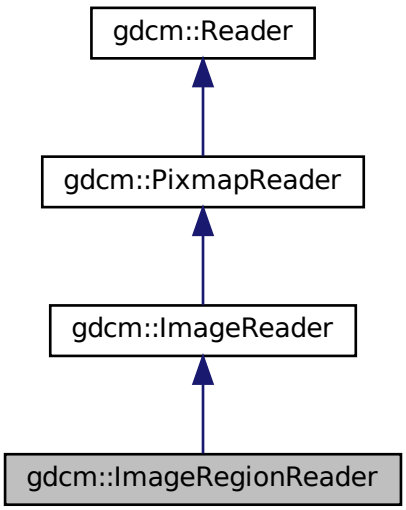
- gdcmImageReader.h

10.156 gdcm::ImageRegionReader Class Reference

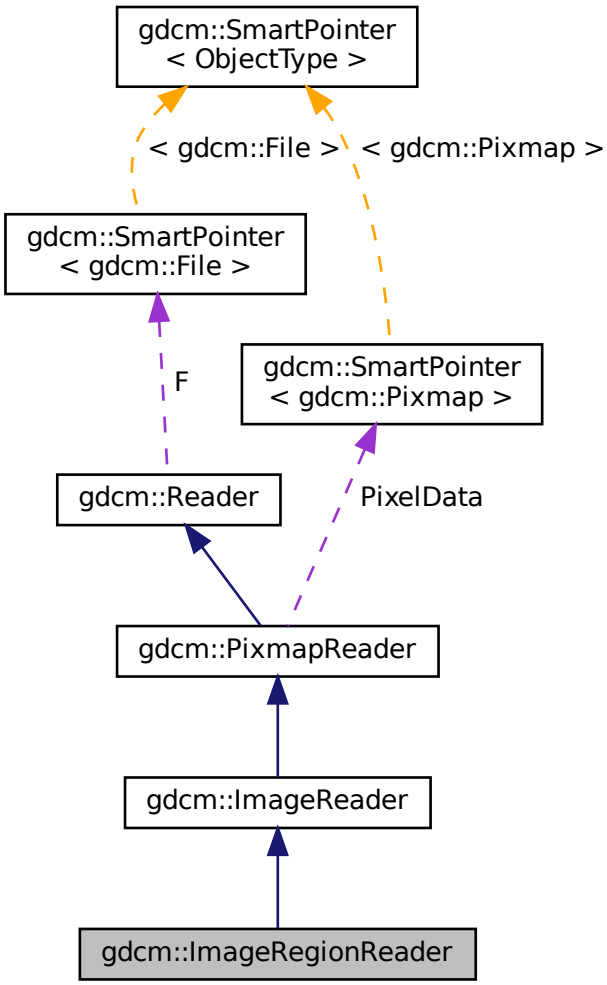
ImageRegionReader.

```
#include <gdcmImageRegionReader.h>
```

Inheritance diagram for gdcm::ImageRegionReader:



Collaboration diagram for gdcm::ImageRegionReader:



Public Member Functions

- ImageRegionReader ()
- ~ImageRegionReader () override
- size_t ComputeBufferLength () const
- Region const & GetRegion () const
- bool ReadInformation ()
- bool ReadIntoBuffer (char *inreadbuffer, size_t buflen)
- void SetRegion (Region const &region)

  Set/Get Region to be read.

Protected Member Functions

- bool Read () override

  To prevent user from calling super class Read() function.

Additional Inherited Members

10.156.1 Detailed Description

ImageRegionReader.

This class is able to read a region from a DICOM file containing an image. This implementation requires that the information stored in the DICOM header are consistent with what is in the encapsulated Pixel Data. This is technically not required by DICOM standard, which makes this implementation illegal with regards to the famous JPEG note: `http://dicom.nema.org/medical/dicom/current/output/chtml/part05/sect_8.↩2.html#para_4bcb841e-c6bf-4e26-82a5-3fad3c942da0`

See also

ImageReader

Examples

ExtractImageRegion.cs, ExtractImageRegionWithLUT.cs, and TemplateEmptyImage.cxx.

10.156.2 Constructor & Destructor Documentation

10.156.2.1 ImageRegionReader()

```
gdcm::ImageRegionReader::ImageRegionReader ( )
```

10.156.2.2 ~ImageRegionReader()

```
gdcm::ImageRegionReader::~ImageRegionReader ( )  [override]
```

10.156.3 Member Function Documentation

10.156.3.1 ComputeBufferLength()

```
size_t gdcm::ImageRegionReader::ComputeBufferLength ( ) const
```

Explicit call which will compute the minimal buffer length that can hold the whole uncompressed image as defined by Region `region`.

Returns

0 upon error

10.156.3.2 GetRegion()

```
Region const & gdcm::ImageRegionReader::GetRegion ( ) const
```

10.156.3.3 Read()

```
bool gdcm::ImageRegionReader::Read ( )  [override], [protected], [virtual]
```

To prevent user from calling super class Read() function.

Reimplemented from gdcm::ImageReader.

10.156.3.4 ReadInformation()

```
bool gdcm::ImageRegionReader::ReadInformation ( )
```

Read meta information (not Pixel Data) from the DICOM file.

Returns

false upon error

Examples

ExtractImageRegion.cs, ExtractImageRegionWithLUT.cs, and TemplateEmptyImage.cxx.

10.156.3.5 ReadIntoBuffer()

```
bool gdcm::ImageRegionReader::ReadIntoBuffer (
            char * inreadbuffer,
            size_t buflen )
```

Read into buffer: For Python, the `buflen` param is deduced directly from the input bytearray passed as parameter (function only takes one param).

Returns

false upon error

Examples

ExtractImageRegion.cs, and ExtractImageRegionWithLUT.cs.

10.156.3.6 SetRegion()

```
void gdcm::ImageRegionReader::SetRegion (
            Region const & region )
```

Set/Get Region to be read.

Examples

ExtractImageRegion.cs, and ExtractImageRegionWithLUT.cs.

The documentation for this class was generated from the following file:

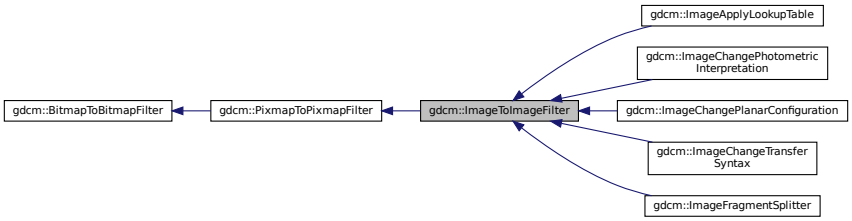
- gdcmImageRegionReader.h

10.157 gdcm::ImageToImageFilter Class Reference

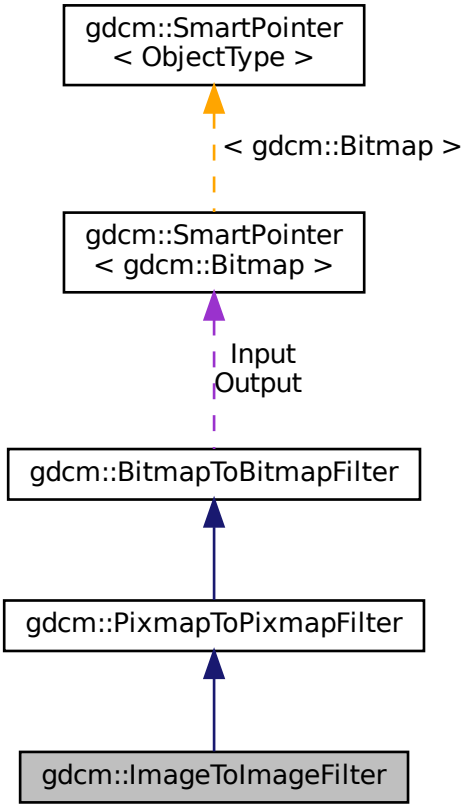
ImageToImageFilter class.

```
#include <gdcmImageToImageFilter.h>
```

Inheritance diagram for gdcm::ImageToImageFilter:



Collaboration diagram for gdcm::ImageToImageFilter:



Public Member Functions

- ImageToImageFilter ()

- ~ImageToImageFilter ()=default
- Image & GetInput ()
- const Image & GetOutput () const

  Get Output image.

Additional Inherited Members

10.157.1 Detailed Description

ImageToImageFilter class.

Super class for all filter taking an image and producing an output image

10.157.2 Constructor & Destructor Documentation

10.157.2.1 ImageToImageFilter()

```
gdcm::ImageToImageFilter::ImageToImageFilter ( )
```

10.157.2.2 ~ImageToImageFilter()

```
gdcm::ImageToImageFilter::~ImageToImageFilter ( )  [default]
```

10.157.3 Member Function Documentation

10.157.3.1 GetInput()

```
Image & gdcm::ImageToImageFilter::GetInput ( )
```

10.157.3.2 GetOutput()

```
const Image & gdcm::ImageToImageFilter::GetOutput ( ) const
```

Get Output image.

Examples

BasicImageAnonymizer.cs, CompressImage.cxx, and CompressLossyJPEG.cs.

The documentation for this class was generated from the following file:

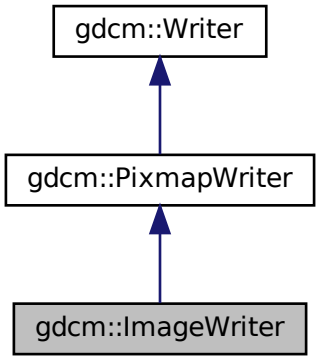
- gdcmImageToImageFilter.h

10.158 gdcm::ImageWriter Class Reference

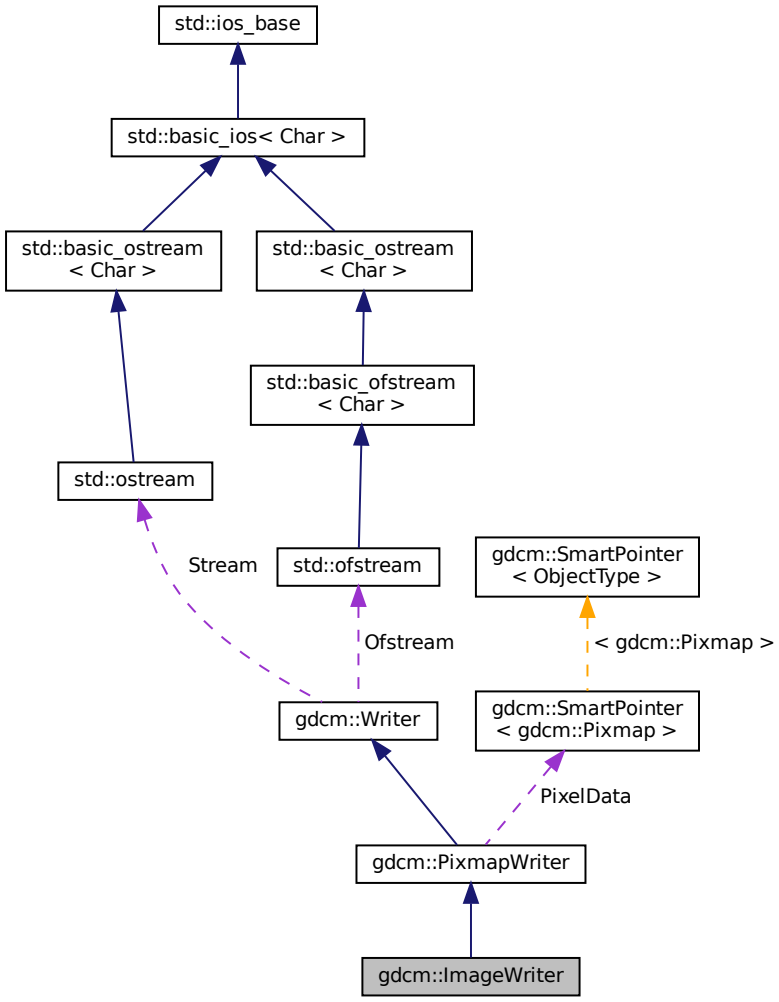
ImageWriter.

```
#include <gdcmImageWriter.h>
```

Inheritance diagram for gdcm::ImageWriter:



Collaboration diagram for gdcm::ImageWriter:



Public Member Functions

- ImageWriter ()
- ~ImageWriter () override
- MediaStorage ComputeTargetMediaStorage ()
- const Image & GetImage () const override
- Image & GetImage () override
- bool Write () override

  Write.

Additional Inherited Members

10.158.1 Detailed Description

ImageWriter.

This is an extended version of the PixmapWriter. Pay attention that:

1. It will populate missing attribute for Secondary Capture Image Storage instances,
2. It may also change an input MR Image Storage instance into a pseudo Enhanced MR Image Storage instance whenever Modality LUT is required.
3. Some DataElement related to gdcm::Image may be slightly altered.

Examples

BasicImageAnonymizer.cs, CompressImage.cxx, CompressLossyJPEG.cs, CreateARGBImage.cxx, CreateCMYKImage.cxx, DecompressImage.cs, GenFakeImage.cxx, GetSubSequenceData.cxx, HelloVizWorld.cxx, MergeTwoFiles.cxx, MpegVideoInfo.cs, TemplateEmptyImage.cxx, csa2img.cxx, and iU22tomultisc.cxx.

10.158.2 Constructor & Destructor Documentation

10.158.2.1 ImageWriter()

```
gdcm::ImageWriter::ImageWriter ( )
```

10.158.2.2 ~ImageWriter()

```
gdcm::ImageWriter::~ImageWriter ( )  [override]
```

10.158.3 Member Function Documentation

10.158.3.1 ComputeTargetMediaStorage()

```
MediaStorage gdcm::ImageWriter::ComputeTargetMediaStorage ( )
```

internal function used to compute a target MediaStorage the most appropriate User may want to call this function ahead of time (before Write)

Examples

TemplateEmptyImage.cxx.

10.158.3.2 GetImage() [1/2]

```
const Image & gdcm::ImageWriter::GetImage ( ) const  [inline], [override], [virtual]
```

Set/Get Image to be written It will overwrite anything Image infos found in DataSet (see parent class to see how to pass dataset)

Reimplemented from gdcm::PixmapWriter.

Examples

CreateARGBImage.cxx, CreateCMYKImage.cxx, csa2img.cxx, and iU22tomultisc.cxx.

10.158.3.3 GetImage() [2/2]

```
Image & gdcm::ImageWriter::GetImage ( )  [inline], [override], [virtual]
```

Reimplemented from gdcm::PixmapWriter.

10.158.3.4 Write()

```
bool gdcm::ImageWriter::Write ( )  [override], [virtual]
```

Write.

Reimplemented from gdcm::Writer.

Examples

BasicImageAnonymizer.cs, CompressImage.cxx, CompressLossyJPEG.cs, CreateARGBImage.cxx, CreateCMYKImage.cxx, DecompressImage.cs, GenFakeImage.cxx, GetSubSequenceData.cxx, HelloVizWorld.cxx, MergeTwoFiles.cxx, MpegVideoInfo.cs, TemplateEmptyImage.cxx, csa2img.cxx, and iU22tomultisc.cxx.

The documentation for this class was generated from the following file:

- gdcmImageWriter.h

10.159 gdcm::network::ImplementationClassUIDSub Class Reference

ImplementationClassUIDSub.

`#include <gdcmImplementationClassUIDSub.h>`

Public Member Functions

- ImplementationClassUIDSub ()
- void Print (std::ostream &os) const
- std::istream & Read (std::istream &is)
- size_t Size () const
- const std::ostream & Write (std::ostream &os) const

10.159.1 Detailed Description

ImplementationClassUIDSub.

PS 3.7 Table D.3-1 IMPLEMENTATION CLASS UID SUB-ITEM FIELDS (A-ASSOCIATE-RQ)

10.159.2 Constructor & Destructor Documentation

10.159.2.1 ImplementationClassUIDSub()

```
gdcm::network::ImplementationClassUIDSub::ImplementationClassUIDSub ( )
```

10.159.3 Member Function Documentation

10.159.3.1 Print()

```
void gdcm::network::ImplementationClassUIDSub::Print (
            std::ostream & os ) const
```

10.159.3.2 Read()

```
std::istream & gdcm::network::ImplementationClassUIDSub::Read (
             std::istream & is )
```

10.159.3.3 Size()

```
size_t gdcm::network::ImplementationClassUIDSub::Size ( ) const
```

10.159.3.4 Write()

```
const std::ostream & gdcm::network::ImplementationClassUIDSub::Write (
             std::ostream & os ) const
```

The documentation for this class was generated from the following file:

- gdcmImplementationClassUIDSub.h

10.160 gdcm::network::ImplementationUIDSub Class Reference

ImplementationUIDSub.

```
#include <gdcmImplementationUIDSub.h>
```

Public Member Functions

- ImplementationUIDSub ()
- const std::ostream & Write (std::ostream &os) const

10.160.1 Detailed Description

ImplementationUIDSub.

Table D.3-2 IMPLEMENTATION UID SUB-ITEM FIELDS (A-ASSOCIATE-AC)

10.160.2 Constructor & Destructor Documentation

10.160.2.1 ImplementationUIDSub()

```
gdcm::network::ImplementationUIDSub::ImplementationUIDSub ( )
```

10.160.3 Member Function Documentation

10.160.3.1 Write()

```
const std::ostream & gdcm::network::ImplementationUIDSub::Write (
            std::ostream & os ) const
```

The documentation for this class was generated from the following file:

- gdcmImplementationUIDSub.h

10.161 gdcm::network::ImplementationVersionNameSub Class Reference

ImplementationVersionNameSub.

```
#include <gdcmImplementationVersionNameSub.h>
```

Public Member Functions

- ImplementationVersionNameSub ()
- void Print (std::ostream &os) const
- std::istream & Read (std::istream &is)
- size_t Size () const
- const std::ostream & Write (std::ostream &os) const

10.161.1 Detailed Description

ImplementationVersionNameSub.

Table D.3-3 IMPLEMENTATION VERSION NAME SUB-ITEM FIELDS (A-ASSOCIATE-RQ)

10.161.2 Constructor & Destructor Documentation

10.161.2.1 ImplementationVersionNameSub()

```
gdcm::network::ImplementationVersionNameSub::ImplementationVersionNameSub ( )
```

10.161.3 Member Function Documentation

10.161.3.1 Print()

```
void gdcm::network::ImplementationVersionNameSub::Print (
            std::ostream & os ) const
```

10.161.3.2 Read()

```
std::istream & gdcm::network::ImplementationVersionNameSub::Read (
            std::istream & is )
```

10.161.3.3 Size()

```
size_t gdcm::network::ImplementationVersionNameSub::Size ( ) const
```

10.161.3.4 Write()

```
const std::ostream & gdcm::network::ImplementationVersionNameSub::Write (
            std::ostream & os ) const
```

The documentation for this class was generated from the following file:

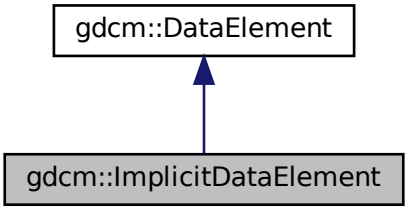
- gdcmImplementationVersionNameSub.h

10.162 gdcm::ImplicitDataElement Class Reference

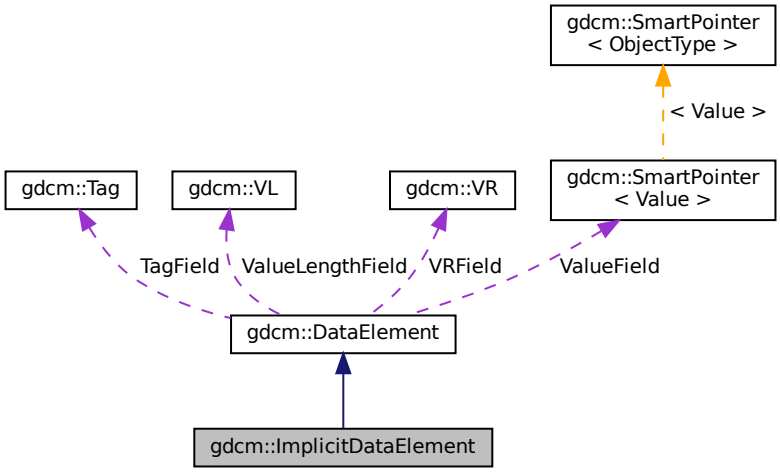
Class to represent an *Implicit VR* Data Element.

`#include <gdcmImplicitDataElement.h>`

Inheritance diagram for gdcm::ImplicitDataElement:



Collaboration diagram for gdcm::ImplicitDataElement:



Public Member Functions

- VL GetLength () const
- template<typename TSwap >
  std::istream & Read (std::istream &is)

- template<typename TSwap >
  std::istream & ReadPreValue (std::istream &is)
- template<typename TSwap >
  std::istream & ReadValue (std::istream &is, bool readvalues=true)
- template<typename TSwap >
  std::istream & ReadValueWithLength (std::istream &is, VL &length, bool readvalues=true)
- template<typename TSwap >
  std::istream & ReadWithLength (std::istream &is, VL &length, bool readvalues=true)
- template<typename TSwap >
  const std::ostream & Write (std::ostream &os) const

Additional Inherited Members

10.162.1 Detailed Description

Class to represent an *Implicit VR* Data Element.

Note

bla

Examples

ReadExplicitLengthSQIVR.cxx.

10.162.2 Member Function Documentation

10.162.2.1 GetLength()

```
VL gdcm::ImplicitDataElement::GetLength ( ) const
```

10.162.2.2 Read()

```
template<typename TSwap >
std::istream & gdcm::ImplicitDataElement::Read (
            std::istream & is )
```

10.162.2.3 ReadPreValue()

```
template<typename TSwap >
std::istream & gdcm::ImplicitDataElement::ReadPreValue (
             std::istream & is )
```

10.162.2.4 ReadValue()

```
template<typename TSwap >
std::istream & gdcm::ImplicitDataElement::ReadValue (
             std::istream & is,
             bool readvalues = true )
```

10.162.2.5 ReadValueWithLength()

```
template<typename TSwap >
std::istream & gdcm::ImplicitDataElement::ReadValueWithLength (
             std::istream & is,
             VL & length,
             bool readvalues = true )
```

10.162.2.6 ReadWithLength()

```
template<typename TSwap >
std::istream & gdcm::ImplicitDataElement::ReadWithLength (
             std::istream & is,
             VL & length,
             bool readvalues = true )
```

10.162.2.7 Write()

```
template<typename TSwap >
const std::ostream & gdcm::ImplicitDataElement::Write (
             std::ostream & os ) const
```

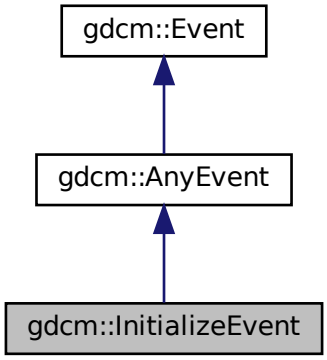
The documentation for this class was generated from the following file:

- gdcmImplicitDataElement.h

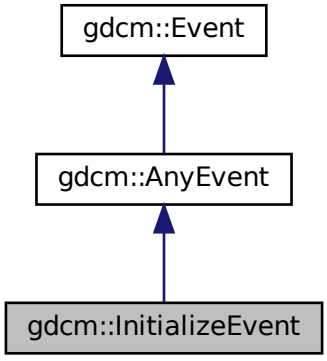
10.163 gdcm::InitializeEvent Class Reference

`#include <gdcmEvent.h>`

Inheritance diagram for gdcm::InitializeEvent:



Collaboration diagram for gdcm::InitializeEvent:



Additional Inherited Members

The documentation for this class was generated from the following file:

- gdcmEvent.h

10.164 gdcm::IOD Class Reference

Class for representing a IOD.

`#include <gdcmIOD.h>`

Public Types

- typedef std::vector< IODEntry > MapIODEntry
- typedef MapIODEntry::size_type SizeType

Public Member Functions

- IOD ()=default
- void AddIODEntry (const IODEntry &iode)
- void Clear ()
- const IODEntry & GetIODEntry (SizeType idx) const
- SizeType GetNumberOfIODs () const
- Type GetTypeFromTag (const Defs &defs, const Tag &tag) const

Friends

- std::ostream & operator<< (std::ostream &_os, const IOD &_val)

10.164.1 Detailed Description

Class for representing a IOD.

Note

bla

See also

Dict

Examples

TraverseModules.cxx.

10.164.2 Member Typedef Documentation

10.164.2.1 MapIODEntry

```
typedef std::vector<IODEntry> gdcm::IOD::MapIODEntry
```

10.164.2.2 SizeType

```
typedef MapIODEntry::size_type gdcm::IOD::SizeType
```

10.164.3 Constructor & Destructor Documentation

10.164.3.1 IOD()

```
gdcm::IOD::IOD ( )  [default]
```

10.164.4 Member Function Documentation

10.164.4.1 AddIODEntry()

```
void gdcm::IOD::AddIODEntry (
            const IODEntry & iode )  [inline]
```

10.164.4.2 Clear()

```
void gdcm::IOD::Clear ( )  [inline]
```

10.164.4.3 GetIODEntry()

```
const IODEntry & gdcm::IOD::GetIODEntry (
            SizeType idx ) const  [inline]
```

Examples

TraverseModules.cxx.

10.164.4.4 GetNumberOfIODs()

```
SizeType gdcm::IOD::GetNumberOfIODs ( ) const  [inline]
```

Examples

TraverseModules.cxx.

10.164.4.5 GetTypeFromTag()

```
Type gdcm::IOD::GetTypeFromTag (
            const Defs & defs,
            const Tag & tag ) const
```

10.164.5 Friends And Related Function Documentation

10.164.5.1 operator<<

```
std::ostream & operator<< (
            std::ostream & _os,
            const IOD & _val )  [friend]
```

The documentation for this class was generated from the following file:

- gdcmIOD.h

10.165 gdcm::IODEntry Class Reference

Class for representing a IODEntry.

`#include <gdcmIODEntry.h>`

Public Member Functions

- IODEntry (const char *name="", const char *ref="", const char *usag="")
- const char * GetIE () const
- const char * GetName () const
- const char * GetRef () const
- const char * GetUsage () const
- Usage::UsageType GetUsageType () const
- void SetIE (const char *ie)
- void SetName (const char *name)
- void SetRef (const char *ref)
- void SetUsage (const char *usag)

Friends

- std::ostream & operator<< (std::ostream &_os, const IODEntry &_val)

10.165.1 Detailed Description

Class for representing a IODEntry.

Note

A.1.3 IOD Module Table and Functional Group Macro Table This Section of each IOD defines in a tabular form the Modules comprising the IOD. The following information must be specified for each Module in the table:

- The name of the Module or Functional Group
- A reference to the Section in Annex C which defines the Module or Functional Group
- The usage of the Module or Functional Group; whether it is:
- Mandatory (see A.1.3.1) , abbreviated M
- Conditional (see A.1.3.2) , abbreviated C
- User Option (see A.1.3.3) , abbreviated U The Modules referenced are defined in Annex C. A.1.3.1 MANDATORY MODULES For each IOD, Mandatory Modules shall be supported per the definitions, semantics and requirements defined in Annex C. PS 3.3 - 2008 Page 96
- Standard - A.1.3.2 CONDITIONAL MODULES Conditional Modules are Mandatory Modules if specific conditions are met. If the specified conditions are not met, this Module shall not be supported; that is, no information defined in that Module shall be sent. A.1.3.3 USER OPTION MODULES User Option Modules may or may not be supported. If an optional Module is supported, the Attribute Types specified in the Modules in Annex C shall be supported.

See also

DictEntry

Examples

TraverseModules.cxx.

10.165.2 Constructor & Destructor Documentation

10.165.2.1 IODEntry()

```
gdcm::IODEntry::IODEntry (
            const char * name = "",
            const char * ref = "",
            const char * usag = "" )  [inline]
```

10.165.3 Member Function Documentation

10.165.3.1 GetIE()

```
const char * gdcm::IODEntry::GetIE ( ) const  [inline]
```

10.165.3.2 GetName()

```
const char * gdcm::IODEntry::GetName ( ) const  [inline]
```

10.165.3.3 GetRef()

```
const char * gdcm::IODEntry::GetRef ( ) const  [inline]
```

Examples

TraverseModules.cxx.

10.165.3.4 GetUsage()

```
const char * gdcm::IODEntry::GetUsage ( ) const  [inline]
```

10.165.3.5 GetUsageType()

```
Usage::UsageType gdcm::IODEntry::GetUsageType ( ) const
```

10.165.3.6 SetIE()

```
void gdcm::IODEntry::SetIE (
            const char * ie )  [inline]
```

10.165.3.7 SetName()

```
void gdcm::IODEntry::SetName (
            const char * name )  [inline]
```

10.165.3.8 SetRef()

```
void gdcm::IODEntry::SetRef (
            const char * ref )  [inline]
```

10.165.3.9 SetUsage()

```
void gdcm::IODEntry::SetUsage (
            const char * usag )  [inline]
```

10.165.4 Friends And Related Function Documentation

10.165.4.1 operator<<

```
std::ostream & operator<< (
            std::ostream & _os,
            const IODEntry & _val )  [friend]
```

The documentation for this class was generated from the following file:

- gdcmIODEntry.h

10.166 gdcm::IODs Class Reference

Class for representing a IODs.

```
#include <gdcmIODs.h>
```

Public Types

- typedef std::map< IODName, IOD > IODMapType
- typedef IODMapType::const_iterator IODMapTypeConstIterator
- typedef std::string IODName

Public Member Functions

- IODs ()=default
- void AddIOD (const char *name, const IOD &module)
- IODMapTypeConstIterator Begin () const
- void Clear ()
- IODMapTypeConstIterator End () const
- const IOD & GetIOD (const char *name) const

Friends

- std::ostream & operator<< (std::ostream &_os, const IODs &_val)

10.166.1 Detailed Description

Class for representing a IODs.

Note

bla

See also

IOD

Examples

TraverseModules.cxx.

10.166.2 Member Typedef Documentation

10.166.2.1 IODMapType

```
typedef std::map<IODName, IOD> gdcm::IODs::IODMapType
```

10.166.2.2 IODMapTypeConstIterator

```
typedef IODMapType::const_iterator gdcm::IODs::IODMapTypeConstIterator
```

10.166.2.3 IODName

```
typedef std::string gdcm::IODs::IODName
```

10.166.3 Constructor & Destructor Documentation

10.166.3.1 IODs()

```
gdcm::IODs::IODs ( )  [default]
```

10.166.4 Member Function Documentation

10.166.4.1 AddIOD()

```
void gdcm::IODs::AddIOD (
            const char * name,
            const IOD & module )  [inline]
```

10.166.4.2 Begin()

```
IODMapTypeConstIterator gdcm::IODs::Begin ( ) const  [inline]
```

Examples

TraverseModules.cxx.

10.166.4.3 Clear()

```
void gdcm::IODs::Clear ( )  [inline]
```

10.166.4.4 End()

```
IODMapTypeConstIterator gdcm::IODs::End ( ) const  [inline]
```

Examples

TraverseModules.cxx.

10.166.4.5 GetIOD()

```
const IOD & gdcm::IODs::GetIOD (
            const char * name ) const  [inline]
```

10.166.5 Friends And Related Function Documentation

10.166.5.1 operator<<

```
std::ostream & operator<< (
            std::ostream & _os,
            const IODs & _val )  [friend]
```

The documentation for this class was generated from the following file:

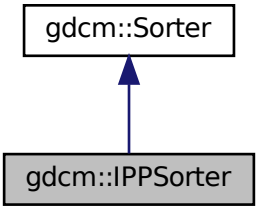
- gdcmIODs.h

10.167 gdcm::IPPSorter Class Reference

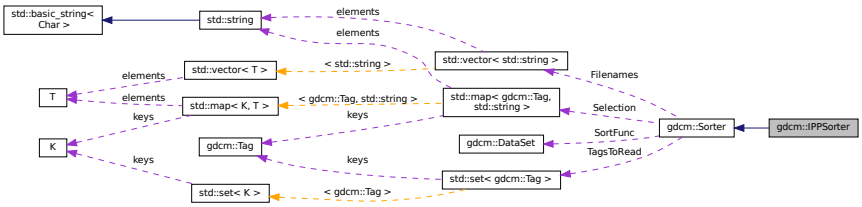
IPPSorter.

`#include <gdcmIPPSorter.h>`

Inheritance diagram for gdcm::IPPSorter:



Collaboration diagram for gdcm::IPPSorter:



Public Member Functions

- IPPSorter ()
- double GetDirectionCosinesTolerance () const
- double GetZSpacing () const
- double GetZSpacingTolerance () const
- void SetComputeZSpacing (bool b)
- void SetDirectionCosinesTolerance (double tol)
- void SetDropDuplicatePositions (bool b)
- void SetZSpacingTolerance (double tol)
- bool Sort (std::vector< std::string > const &filenames) override

Protected Attributes

- bool ComputeZSpacing
- double DirCosTolerance
- bool DropDuplicatePositions
- double ZSpacing
- double ZTolerance

Additional Inherited Members

10.167.1 Detailed Description

IPPSorter.

Implement a simple Image Position (Patient) sorter, along the Image Orientation (Patient) direction. This algorithm does NOT support duplicate and will FAIL in case of duplicate IPP.

Warning

See special note for SetZSpacingTolerance when computing the ZSpacing from the IPP of each DICOM files (default tolerance for consistent spacing is: 1e-6mm)

For more information on Spacing, and how it is defined in DICOM, advanced users may refers to:

```
http://gdcm.sourceforge.net/wiki/index.php/Imager_Pixel_Spacing
```

Bug There are currently a couple of bugs in this implementation:

- Gantry Tilt is not considered (always an error)
- Application programmer should only sort valid DataSet (eg. MRImageStorage, CTImageStorage, PETImage↩ Storage)

Examples

Compute3DSpacing.cxx, VolumeSorter.cxx, gdcmorthoplanes.cxx, and reslicesphere.cxx.

10.167.2 Constructor & Destructor Documentation

10.167.2.1 IPPSorter()

```
gdcm::IPPSorter::IPPSorter ( )
```

10.167.3 Member Function Documentation

10.167.3.1 GetDirectionCosinesTolerance()

```
double gdcm::IPPSorter::GetDirectionCosinesTolerance ( ) const  [inline]
```

10.167.3.2 GetZSpacing()

```
double gdcm::IPPSorter::GetZSpacing ( ) const  [inline]
```

Read-only function to provide access to the computed value for the Z-Spacing The ComputeZSpacing must have been set to true before execution of sort algorithm. Call this function *after* calling Sort(); Z-Spacing will be 0 on 2 occasions:

- Sorting simply failed, potentially duplicate IPP => ZSpacing = 0
- ZSpacing could not be computed (Z-Spacing is not constant, or ZTolerance is too low)

Examples

Compute3DSpacing.cxx, gdcmorthoplanes.cxx, and reslicesphere.cxx.

10.167.3.3 GetZSpacingTolerance()

```
double gdcm::IPPSorter::GetZSpacingTolerance ( ) const  [inline]
```

10.167.3.4 SetComputeZSpacing()

```
void gdcm::IPPSorter::SetComputeZSpacing (
            bool b )  [inline]
```

Functions related to Z-Spacing computation Set to true when sort algorithm should also perform a regular Z-Spacing computation using the Image Position (Patient) Potential reason for failure:

1. ALL slices are taken into account, if one slice if missing then ZSpacing will be set to 0 since the spacing will not be found to be regular along the Series

Examples

Compute3DSpacing.cxx, VolumeSorter.cxx, gdcmorthoplanes.cxx, and reslicesphere.cxx.

10.167.3.5 SetDirectionCosinesTolerance()

```
void gdcm::IPPSorter::SetDirectionCosinesTolerance (
            double tol )  [inline]
```

Sometimes IOP along a series is slightly changing for example: "0.999081\\0.0426953\\0.00369272\\-0.0419025\\0.↩
955059\\0.293439", "0.999081\\0.0426953\\0.00369275\\-0.0419025\\0.955059\\0.293439", "0.999081\\0.0426952\\0.↩
00369272\\-0.0419025\\0.955059\\0.293439", We need an API to define the tolerance which is allowed. Internally the cross vector of each direction cosines is computed. The tolerance then define the distance in between 1.0 to the dot product of those cross vectors. In a perfect world this dot product is of course 1.0 which imply a DirectionCosines tolerance of exactly 0.0 (default).

10.167.3.6 SetDropDuplicatePositions()

```
void gdcm::IPPSorter::SetDropDuplicatePositions (
            bool b )  [inline]
```

Makes the IPPSorter ignore multiple images located at the same position. Only the first occurrence will be kept. Drop↩
DuplicatePositions defaults to false.

10.167.3.7 SetZSpacingTolerance()

```
void gdcm::IPPSorter::SetZSpacingTolerance (
            double tol )  [inline]
```

1. Another reason for failure is that that Z-Spacing is only slightly changing (eg 1e-3) along the series, a human can determine that this is ok and change the tolerance from its default value: 1e-6

Examples

Compute3DSpacing.cxx, gdcmorthoplanes.cxx, and reslicesphere.cxx.

10.167.3.8 Sort()

```
bool gdcm::IPPSorter::Sort (
            std::vector< std::string > const & filenames )  [override], [virtual]
```

Main entry point to the sorter. It will execute the filter, option should be set before running this function (SetZSpacing↩
Tolerance, ...) Return value indicate if sorting could be achieved,. Warning this does *NOT* imply that spacing is consistent, it only means the file are sorted according to IPP You should check if ZSpacing is 0 or not to deduce if file are actually a 3D volume

Reimplemented from gdcm::Sorter.

Examples

Compute3DSpacing.cxx, VolumeSorter.cxx, gdcmorthoplanes.cxx, and reslicesphere.cxx.

10.167.4 Member Data Documentation

10.167.4.1 ComputeZSpacing

```
bool gdcm::IPPSorter::ComputeZSpacing  [protected]
```

10.167.4.2 DirCosTolerance

```
double gdcm::IPPSorter::DirCosTolerance  [protected]
```

10.167.4.3 DropDuplicatePositions

```
bool gdcm::IPPSorter::DropDuplicatePositions  [protected]
```

10.167.4.4 ZSpacing

```
double gdcm::IPPSorter::ZSpacing  [protected]
```

10.167.4.5 ZTolerance

```
double gdcm::IPPSorter::ZTolerance  [protected]
```

The documentation for this class was generated from the following file:

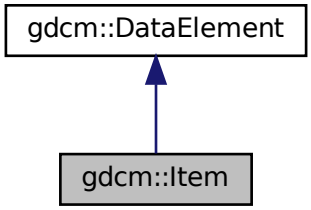
- gdcmIPPSorter.h

10.168 gdcm::Item Class Reference

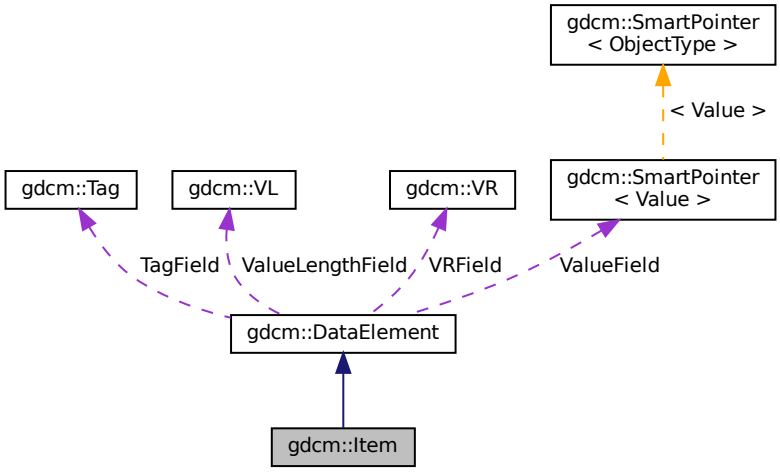
Class to represent an Item.

`#include <gdcmItem.h>`

Inheritance diagram for gdcm::Item:



Collaboration diagram for gdcm::Item:



Public Member Functions

- Item ()
- Item (Item const &val)

- void Clear ()
- bool FindDataElement (const Tag &t) const
- const DataElement & GetDataElement (const Tag &t) const
- template<typename TDE >
  VL GetLength () const
- DataSet & GetNestedDataSet ()
- const DataSet & GetNestedDataSet () const
- void InsertDataElement (const DataElement &de)
- template<typename TDE , typename TSwap >
  std::istream & Read (std::istream &is)
- void SetNestedDataSet (const DataSet &nested)
- template<typename TDE , typename TSwap >
  const std::ostream & Write (std::ostream &os) const

Friends

- std::ostream & operator<< (std::ostream &os, const Item &val)

Additional Inherited Members

10.168.1 Detailed Description

Class to represent an Item.

A component of the value of a Data Element that is of Value Representation Sequence of Items. An Item contains a Data Set . See PS 3.5 7.5.1 Item Encoding Rules Each Item of a Data Element of VR SQ shall be encoded as a DICOM Standard Data Element with a specific Data Element Tag of Value (FFFE,E000). The Item Tag is followed by a 4 byte Item Length field encoded in one of the following two ways Explicit/ Implicit

Note

ITEM: A component of the Value of a Data Element that is of Value Representation Sequence of Items. An Item contains a Data Set.

Examples

ChangeSequenceUltrasound.cxx, DumpExamCard.cxx, DumpGEMSMovieGroup.cxx, DumpPhilipsECHO.cxx, DumpToshibaDTI.cxx, DumpToshibaDTI2.cxx, DumpVisusChange.cxx, ExtractEncryptedContent.cxx, Extracting_All_Resolution.cxx, Fake_Image_Using_Stream_Image_Writer.cxx, GenAllVR.cxx, GenFakeIdentifyFile.cxx, GenLongSeqs.cxx, GenSeqs.cxx, GetSequenceUltrasound.cxx, GetSubSequenceData.cxx, LargeVRDSExplicit.cxx, NewSequence.cs, SimplePrint.cs, gdcmrtionplan.cxx, and gdcmrtplan.cxx.

10.168.2 Constructor & Destructor Documentation

10.168.2.1 Item() [1/2]

```
gdcm::Item::Item ( )  [inline]
```

10.168.2.2 Item() [2/2]

```
gdcm::Item::Item (
            Item const & val )  [inline]
```

10.168.3 Member Function Documentation

10.168.3.1 Clear()

```
void gdcm::Item::Clear ( )  [inline]
```

Referenced by gdcm::SequenceOfItems::Read().

10.168.3.2 FindDataElement()

```
bool gdcm::Item::FindDataElement (
            const Tag & t ) const  [inline]
```

10.168.3.3 GetDataElement()

```
const DataElement & gdcm::Item::GetDataElement (
            const Tag & t ) const  [inline]
```

10.168.3.4 GetLength()

```
template<typename TDE >
VL gdcm::Item::GetLength ( ) const
```

10.168.3.5 GetNestedDataSet() [1/2]

```
DataSet & gdcm::Item::GetNestedDataSet ( )  [inline]
```

10.168.3.6 GetNestedDataSet() [2/2]

```
const DataSet & gdcm::Item::GetNestedDataSet ( ) const  [inline]
```

Examples

ChangeSequenceUltrasound.cxx, DumpExamCard.cxx, DumpGEMSMovieGroup.cxx, DumpPhilipsECHO.cxx, DumpToshibaDTI.cxx, DumpToshibaDTI2.cxx, DumpVisusChange.cxx, ExtractEncryptedContent.cxx, Extracting_All_Resolution.cxx, Fake_Image_Using_Stream_Image_Writer.cxx, GenAllVR.cxx, GenFakeIdentifyFile.cxx, GenSeqs.cxx, GetSequenceUltrasound.cxx, GetSubSequenceData.cxx, LargeVRDSExplicit.cxx, NewSequence.cs, SimplePrint.cs, gdcmrtionplan.cxx, and gdcmrtplan.cxx.

Referenced by gdcm::SequenceOfItems::Read().

10.168.3.7 InsertDataElement()

```
void gdcm::Item::InsertDataElement (
            const DataElement & de )  [inline]
```

10.168.3.8 Read()

```
template<typename TDE , typename TSwap >
std::istream & gdcm::Item::Read (
            std::istream & is )  [inline]
```

References gdcm::ByteSwapFilter::ByteSwap(), gdcm::DataSet::Clear(), gdcmDebugMacro, gdcmErrorMacro, gdcmWarningMacro, gdcm::DataSet::IsEmpty(), and gdcm::ByteSwapFilter::SetByteSwapTag().

Referenced by gdcm::SequenceOfItems::Read().

10.168.3.9 SetNestedDataSet()

```
void gdcm::Item::SetNestedDataSet (
            const DataSet & nested )  [inline]
```

10.168.3.10 Write()

```
template<typename TDE , typename TSwap >
const std::ostream & gdcm::Item::Write (
            std::ostream & os ) const  [inline]
```

References gdcmWarningMacro, gdcm::VL::GetLength(), gdcm::Tag::Write(), and gdcm::VL::Write().

10.168.4 Friends And Related Function Documentation

10.168.4.1 operator<<

```
std::ostream & operator<< (
            std::ostream & os,
            const Item & val )  [friend]
```

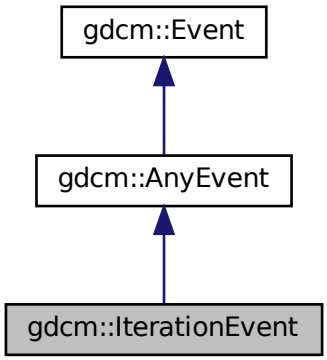
The documentation for this class was generated from the following file:

- gdcmItem.h

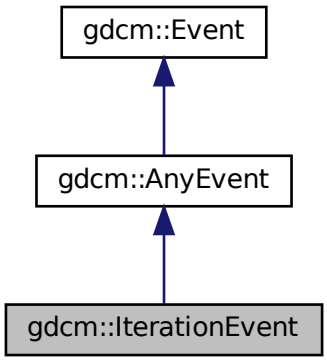
10.169 gdcm::IterationEvent Class Reference

```
#include <gdcmEvent.h>
```

Inheritance diagram for gdcm::IterationEvent:



Collaboration diagram for gdcm::IterationEvent:



Additional Inherited Members

The documentation for this class was generated from the following file:

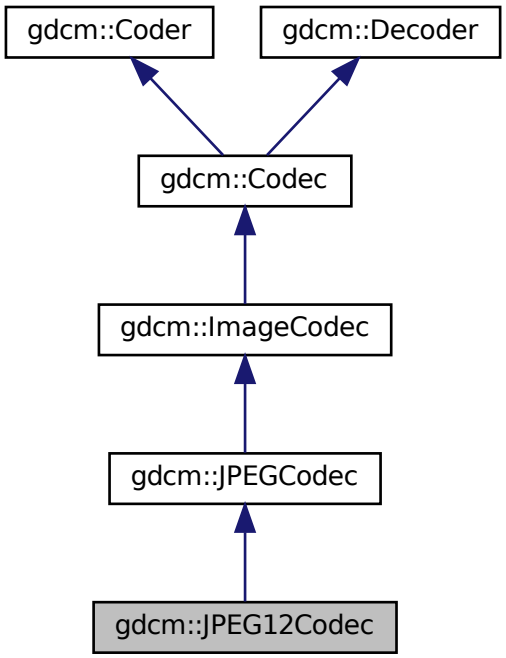
- gdcmEvent.h

10.170 gdcm::JPEG12Codec Class Reference

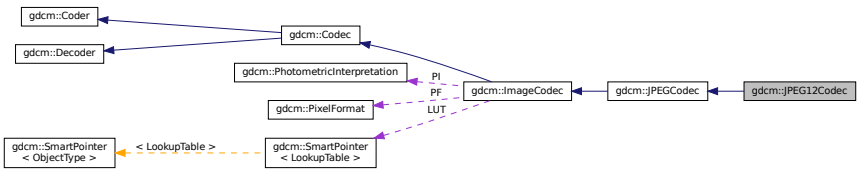
Class to do JPEG 12bits (lossy & lossless)

`#include <gdcmJPEG12Codec.h>`

Inheritance diagram for gdcm::JPEG12Codec:



Collaboration diagram for gdcm::JPEG12Codec:



Public Member Functions

- JPEG12Codec ()
- ~JPEG12Codec () override
- bool DecodeByStreams (std::istream &is, std::ostream &os) override
- bool GetHeaderInfo (std::istream &is, TransferSyntax &ts) override
- bool InternalCode (const char *input, unsigned long len, std::ostream &os) override

Protected Member Functions

- bool EncodeBuffer (std::ostream &os, const char *data, size_t datalen) override
- bool IsStateSuspension () const override

Additional Inherited Members

10.170.1 Detailed Description

Class to do JPEG 12bits (lossy & lossless)

Note

internal class

10.170.2 Constructor & Destructor Documentation

10.170.2.1 JPEG12Codec()

```
gdcm::JPEG12Codec::JPEG12Codec ( )
```

10.170.2.2 ~JPEG12Codec()

```
gdcm::JPEG12Codec::~JPEG12Codec ( )  [override]
```

10.170.3 Member Function Documentation

10.170.3.1 DecodeByStreams()

```
bool gdcm::JPEG12Codec::DecodeByStreams (
            std::istream & is,
            std::ostream & os )  [override], [virtual]
```

Reimplemented from gdcm::ImageCodec.

10.170.3.2 EncodeBuffer()

```
bool gdcm::JPEG12Codec::EncodeBuffer (
            std::ostream & os,
            const char * data,
            size_t datalen )  [override], [protected], [virtual]
```

Reimplemented from gdcm::JPEGCodec.

10.170.3.3 GetHeaderInfo()

```
bool gdcm::JPEG12Codec::GetHeaderInfo (
            std::istream & is,
            TransferSyntax & ts )  [override], [virtual]
```

Reimplemented from gdcm::ImageCodec.

10.170.3.4 InternalCode()

```
bool gdcm::JPEG12Codec::InternalCode (
            const char * input,
            unsigned long len,
            std::ostream & os )  [override], [virtual]
```

Reimplemented from gdcm::Coder.

10.170.3.5 IsStateSuspension()

```
bool gdcm::JPEG12Codec::IsStateSuspension ( ) const  [override], [protected], [virtual]
```

Reimplemented from gdcm::JPEGCodec.

The documentation for this class was generated from the following file:

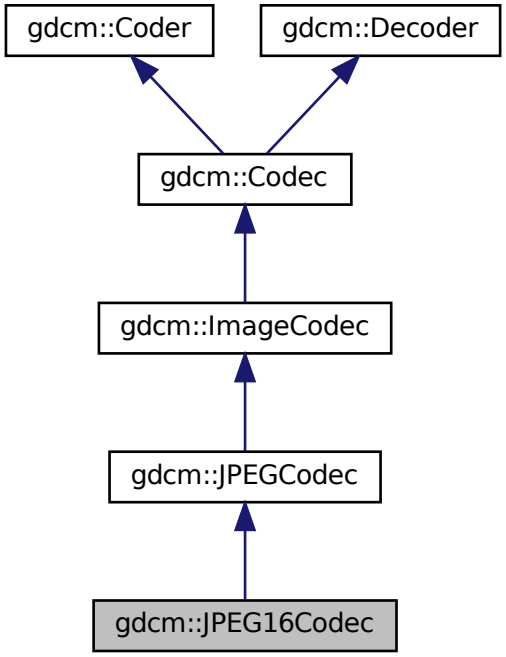
- gdcmJPEG12Codec.h

10.171 gdcm::JPEG16Codec Class Reference

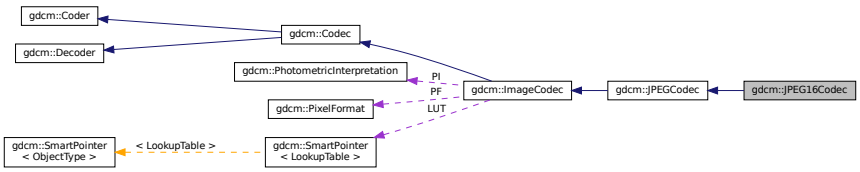
Class to do JPEG 16bits (lossless)

`#include <gdcmJPEG16Codec.h>`

Inheritance diagram for gdcm::JPEG16Codec:



Collaboration diagram for gdcm::JPEG16Codec:



Public Member Functions

- JPEG16Codec ()
- ∼JPEG16Codec () override
- bool DecodeByStreams (std::istream &is, std::ostream &os) override
- bool GetHeaderInfo (std::istream &is, TransferSyntax &ts) override
- bool InternalCode (const char ∗input, unsigned long len, std::ostream &os) override

Protected Member Functions

- bool EncodeBuffer (std::ostream &os, const char ∗data, size_t datalen) override
- bool IsStateSuspension () const override

Additional Inherited Members

10.171.1 Detailed Description

Class to do JPEG 16bits (lossless)

Note

internal class

10.171.2 Constructor & Destructor Documentation

10.171.2.1 JPEG16Codec()

```
gdcm::JPEG16Codec::JPEG16Codec ( )
```

10.171.2.2 ∼JPEG16Codec()

```
gdcm::JPEG16Codec::∼JPEG16Codec ( )  [override]
```

10.171.3 Member Function Documentation

10.171.3.1 DecodeByStreams()

```
bool gdcm::JPEG16Codec::DecodeByStreams (
            std::istream & is,
            std::ostream & os )  [override], [virtual]
```

Reimplemented from gdcm::ImageCodec.

10.171.3.2 EncodeBuffer()

```
bool gdcm::JPEG16Codec::EncodeBuffer (
            std::ostream & os,
            const char * data,
            size_t datalen )  [override], [protected], [virtual]
```

Reimplemented from gdcm::JPEGCodec.

10.171.3.3 GetHeaderInfo()

```
bool gdcm::JPEG16Codec::GetHeaderInfo (
            std::istream & is,
            TransferSyntax & ts )  [override], [virtual]
```

Reimplemented from gdcm::ImageCodec.

10.171.3.4 InternalCode()

```
bool gdcm::JPEG16Codec::InternalCode (
            const char * input,
            unsigned long len,
            std::ostream & os )  [override], [virtual]
```

Reimplemented from gdcm::Coder.

10.171.3.5 IsStateSuspension()

```
bool gdcm::JPEG16Codec::IsStateSuspension ( ) const  [override], [protected], [virtual]
```

Reimplemented from gdcm::JPEGCodec.

The documentation for this class was generated from the following file:

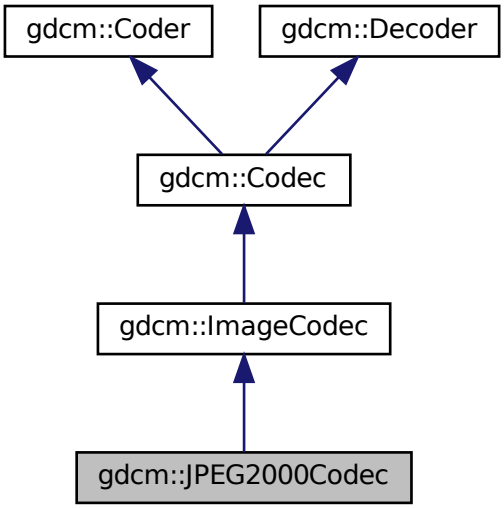
- gdcmJPEG16Codec.h

10.172 gdcm::JPEG2000Codec Class Reference

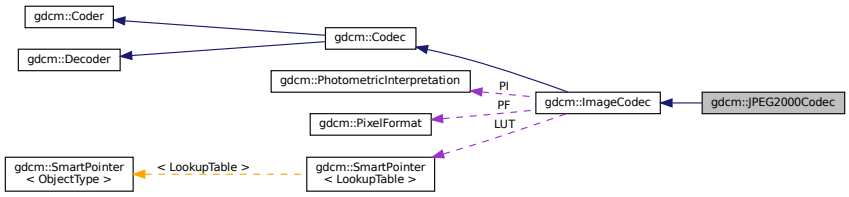
Class to do JPEG 2000.

`#include <gdcmJPEG2000Codec.h>`

Inheritance diagram for gdcm::JPEG2000Codec:



Collaboration diagram for gdcm::JPEG2000Codec:



Public Member Functions

- JPEG2000Codec ()
- ~JPEG2000Codec () override
- bool CanCode (TransferSyntax const &ts) const override

  Return whether this coder support this transfer syntax (can code it)

- bool CanDecode (TransferSyntax const &ts) const override

  Return whether this decoder support this transfer syntax (can decode it)
- ImageCodec * Clone () const override
- bool Code (DataElement const &in, DataElement &out) override

  Code.
- bool Decode (DataElement const &is, DataElement &os) override

  Decode.
- bool GetHeaderInfo (std::istream &is, TransferSyntax &ts) override
- double GetQuality (unsigned int idx=0) const
- double GetRate (unsigned int idx=0) const
- void SetMCT (unsigned int mct)
- void SetNumberOfResolutions (unsigned int nres)
- void SetNumberOfThreadsForDecompression (int nThreads)
- void SetQuality (unsigned int idx, double q)
- void SetRate (unsigned int idx, double rate)
- void SetReversible (bool res)
- void SetTileSize (unsigned int tx, unsigned int ty)

Protected Member Functions

- bool AppendFrameEncode (std::ostream &out, const char *data, size_t datalen) override
- bool AppendRowEncode (std::ostream &out, const char *data, size_t datalen) override
- bool DecodeByStreams (std::istream &is, std::ostream &os) override
- bool DecodeExtent (char *buffer, unsigned int xmin, unsigned int xmax, unsigned int ymin, unsigned int ymax, unsigned int zmin, unsigned int zmax, std::istream &is)
- bool IsFrameEncoder () override
- bool IsRowEncoder () override
- bool StartEncode (std::ostream &) override
- bool StopEncode (std::ostream &) override

Friends

- class Bitmap
- class ImageRegionReader

Additional Inherited Members

10.172.1 Detailed Description

Class to do JPEG 2000.

Note

the class will produce JPC (JPEG 2000 codestream), since some private implementor are using full jp2 file the decoder tolerate jp2 input this is an implementation of an ImageCodec

10.172.2 Constructor & Destructor Documentation

10.172.2.1 JPEG2000Codec()

```
gdcm::JPEG2000Codec::JPEG2000Codec ( )
```

10.172.2.2 ~JPEG2000Codec()

```
gdcm::JPEG2000Codec::~JPEG2000Codec ( )  [override]
```

10.172.3 Member Function Documentation

10.172.3.1 AppendFrameEncode()

```
bool gdcm::JPEG2000Codec::AppendFrameEncode (
            std::ostream & out,
            const char * data,
            size_t datalen )  [override], [protected], [virtual]
```

Reimplemented from gdcm::ImageCodec.

10.172.3.2 AppendRowEncode()

```
bool gdcm::JPEG2000Codec::AppendRowEncode (
            std::ostream & out,
            const char * data,
            size_t datalen )  [override], [protected], [virtual]
```

Reimplemented from gdcm::ImageCodec.

10.172.3.3 CanCode()

```
bool gdcm::JPEG2000Codec::CanCode (
            TransferSyntax const &  ) const  [override], [virtual]
```

Return whether this coder support this transfer syntax (can code it)

Reimplemented from gdcm::ImageCodec.

10.172.3.4 CanDecode()

```
bool gdcm::JPEG2000Codec::CanDecode (
            TransferSyntax const &  ) const  [override], [virtual]
```

Return whether this decoder support this transfer syntax (can decode it)

Reimplemented from gdcm::ImageCodec.

10.172.3.5 Clone()

```
ImageCodec * gdcm::JPEG2000Codec::Clone ( ) const  [override], [virtual]
```

Implements gdcm::ImageCodec.

10.172.3.6 Code()

```
bool gdcm::JPEG2000Codec::Code (
            DataElement const & in_,
            DataElement & out_ )  [override], [virtual]
```

Code.

Reimplemented from gdcm::Coder.

10.172.3.7 Decode()

```
bool gdcm::JPEG2000Codec::Decode (
            DataElement const & ,
            DataElement &  )  [override], [virtual]
```

Decode.

Reimplemented from gdcm::ImageCodec.

10.172.3.8 DecodeByStreams()

```
bool gdcm::JPEG2000Codec::DecodeByStreams (
            std::istream & is,
            std::ostream & os )  [override], [protected], [virtual]
```

Reimplemented from gdcm::ImageCodec.

10.172.3.9 DecodeExtent()

```
bool gdcm::JPEG2000Codec::DecodeExtent (
            char * buffer,
            unsigned int xmin,
            unsigned int xmax,
            unsigned int ymin,
            unsigned int ymax,
            unsigned int zmin,
            unsigned int zmax,
            std::istream & is )  [protected]
```

10.172.3.10 GetHeaderInfo()

```
bool gdcm::JPEG2000Codec::GetHeaderInfo (
            std::istream & is,
            TransferSyntax & ts )  [override], [virtual]
```

Reimplemented from gdcm::ImageCodec.

10.172.3.11 GetQuality()

```
double gdcm::JPEG2000Codec::GetQuality (
            unsigned int idx = 0 ) const
```

10.172.3.12 GetRate()

```
double gdcm::JPEG2000Codec::GetRate (
            unsigned int idx = 0 ) const
```

10.172.3.13 IsFrameEncoder()

```
bool gdcm::JPEG2000Codec::IsFrameEncoder ( )  [override], [protected], [virtual]
```

Reimplemented from gdcm::ImageCodec.

10.172.3.14 IsRowEncoder()

```
bool gdcm::JPEG2000Codec::IsRowEncoder ( )  [override], [protected], [virtual]
```

Reimplemented from gdcm::ImageCodec.

10.172.3.15 SetMCT()

```
void gdcm::JPEG2000Codec::SetMCT (
            unsigned int mct )
```

10.172.3.16 SetNumberOfResolutions()

```
void gdcm::JPEG2000Codec::SetNumberOfResolutions (
            unsigned int nres )
```

10.172.3.17 SetNumberOfThreadsForDecompression()

```
void gdcm::JPEG2000Codec::SetNumberOfThreadsForDecompression (
            int nThreads )
```

Set Number of threads

Parameters

nThreads	: number of threads for decompression codec, if 0 or 1 decompression is done in current thread, if negative value is set determine how many virtual threads are available

10.172.3.18 SetQuality()

```
void gdcm::JPEG2000Codec::SetQuality (
            unsigned int idx,
            double q )
```

10.172.3.19 SetRate()

```
void gdcm::JPEG2000Codec::SetRate (
            unsigned int idx,
            double rate )
```

10.172.3.20 SetReversible()

```
void gdcm::JPEG2000Codec::SetReversible (
            bool res )
```

10.172.3.21 SetTileSize()

```
void gdcm::JPEG2000Codec::SetTileSize (
            unsigned int tx,
            unsigned int ty )
```

10.172.3.22 StartEncode()

```
bool gdcm::JPEG2000Codec::StartEncode (
            std::ostream &  )  [override], [protected], [virtual]
```

Reimplemented from gdcm::ImageCodec.

10.172.3.23 StopEncode()

```
bool gdcm::JPEG2000Codec::StopEncode (
            std::ostream &  )  [override], [protected], [virtual]
```

Reimplemented from gdcm::ImageCodec.

10.172.4 Friends And Related Function Documentation

10.172.4.1 Bitmap

```
friend class Bitmap  [friend]
```

10.172.4.2 ImageRegionReader

```
friend class ImageRegionReader  [friend]
```

The documentation for this class was generated from the following file:

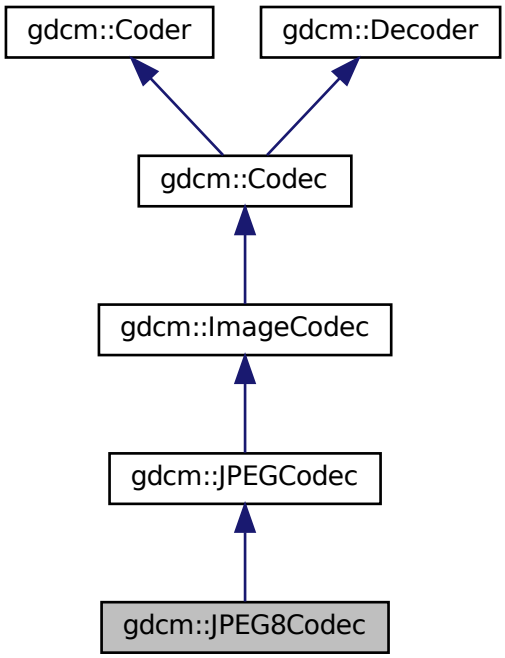
- gdcmJPEG2000Codec.h

10.173 gdcm::JPEG8Codec Class Reference

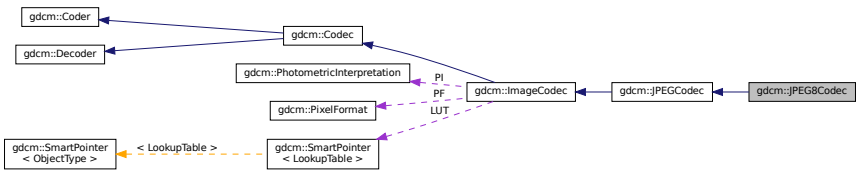
Class to do JPEG 8bits (lossy & lossless)

```
#include <gdcmJPEG8Codec.h>
```

Inheritance diagram for gdcm::JPEG8Codec:



Collaboration diagram for gdcm::JPEG8Codec:



Public Member Functions

- JPEG8Codec ()
- ~JPEG8Codec () override
- bool DecodeByStreams (std::istream &is, std::ostream &os) override
- bool GetHeaderInfo (std::istream &is, TransferSyntax &ts) override
- bool InternalCode (const char *input, unsigned long len, std::ostream &os) override

Protected Member Functions

- bool EncodeBuffer (std::ostream &os, const char *data, size_t datalen) override
- bool IsStateSuspension () const override

Additional Inherited Members

10.173.1 Detailed Description

Class to do JPEG 8bits (lossy & lossless)

Note

internal class

10.173.2 Constructor & Destructor Documentation

10.173.2.1 JPEG8Codec()

```
gdcm::JPEG8Codec::JPEG8Codec ( )
```

10.173.2.2 ~JPEG8Codec()

```
gdcm::JPEG8Codec::~JPEG8Codec ( )  [override]
```

10.173.3 Member Function Documentation

10.173.3.1 DecodeByStreams()

```
bool gdcm::JPEG8Codec::DecodeByStreams (
            std::istream & is,
            std::ostream & os )  [override], [virtual]
```

Reimplemented from gdcm::ImageCodec.

10.173.3.2 EncodeBuffer()

```
bool gdcm::JPEG8Codec::EncodeBuffer (
            std::ostream & os,
            const char * data,
            size_t datalen )  [override], [protected], [virtual]
```

Reimplemented from gdcm::JPEGCodec.

10.173.3.3 GetHeaderInfo()

```
bool gdcm::JPEG8Codec::GetHeaderInfo (
            std::istream & is,
            TransferSyntax & ts )  [override], [virtual]
```

Reimplemented from gdcm::ImageCodec.

10.173.3.4 InternalCode()

```
bool gdcm::JPEG8Codec::InternalCode (
            const char * input,
            unsigned long len,
            std::ostream & os )  [override], [virtual]
```

Reimplemented from gdcm::Coder.

10.173.3.5 IsStateSuspension()

```
bool gdcm::JPEG8Codec::IsStateSuspension ( ) const  [override], [protected], [virtual]
```

Reimplemented from gdcm::JPEGCodec.

The documentation for this class was generated from the following file:

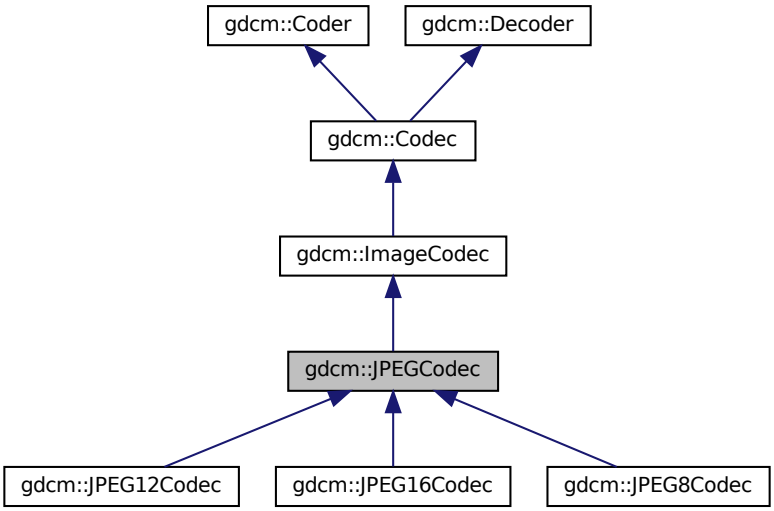
- gdcmJPEG8Codec.h

10.174 gdcm::JPEGCodec Class Reference

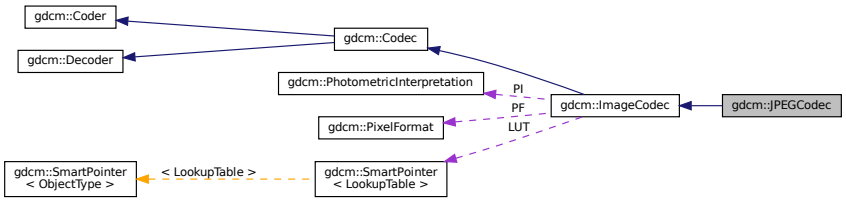
JPEG codec.

`#include <gdcmJPEGCodec.h>`

Inheritance diagram for gdcm::JPEGCodec:



Collaboration diagram for gdcm::JPEGCodec:



Public Member Functions

- JPEGCodec ()
- ~JPEGCodec () override
- bool CanCode (TransferSyntax const &ts) const override

  Return whether this coder support this transfer syntax (can code it)
- bool CanDecode (TransferSyntax const &ts) const override

Return whether this decoder support this transfer syntax (can decode it)

- ImageCodec * Clone () const override
- bool Code (DataElement const &in, DataElement &out) override

  Compress into JPEG.
- void ComputeOffsetTable (bool b)

  Compute the offset table:
- bool Decode (DataElement const &is, DataElement &os) override

  Decode.
- virtual bool EncodeBuffer (std::ostream &out, const char *inbuffer, size_t inlen)
- bool GetHeaderInfo (std::istream &is, TransferSyntax &ts) override
- bool GetLossless () const
- double GetQuality () const
- void SetLossless (bool l)
- void SetPixelFormat (PixelFormat const &pf) override
- void SetQuality (double q)

Protected Member Functions

- bool AppendFrameEncode (std::ostream &out, const char *data, size_t datalen) override
- bool AppendRowEncode (std::ostream &out, const char *data, size_t datalen) override
- bool DecodeByStreams (std::istream &is, std::ostream &os) override
- bool DecodeExtent (char *buffer, unsigned int xmin, unsigned int xmax, unsigned int ymin, unsigned int ymax, unsigned int zmin, unsigned int zmax, std::istream &is)
- bool IsFrameEncoder () override
- bool IsRowEncoder () override
- virtual bool IsStateSuspension () const
- bool IsValid (PhotometricInterpretation const &pi) override
- void SetBitSample (int bit)
- bool StartEncode (std::ostream &) override
- bool StopEncode (std::ostream &) override

Protected Attributes

- int BitSample
- int Quality

Friends

- class ImageRegionReader

Additional Inherited Members

10.174.1 Detailed Description

JPEG codec.

Class to do JPEG (8bits, 12bits, 16bits lossy & lossless). It redispatch in between the different codec implementation: JPEG8Codec, JPEG12Codec & JPEG16Codec It also support inconsistency in between DICOM header and JPEG compressed stream ImageCodec implementation for the JPEG case

Note

Things you should know if you ever want to dive into DICOM/JPEG world (among other):

- http://groups.google.com/group/comp.protocols.dicom/browse_thread/thread/625e46919f208
- http://groups.google.com/group/comp.protocols.dicom/browse_thread/thread/75fdfccc65a62
- http://groups.google.com/group/comp.protocols.dicom/browse_thread/thread/2d525ef6a2f09
- http://groups.google.com/group/comp.protocols.dicom/browse_thread/thread/6b93af410f8c9

Examples

CompressLossyJPEG.cs, FileChangeTSLossy.cs, and GetJPEGSamplePrecision.cxx.

10.174.2 Constructor & Destructor Documentation

10.174.2.1 JPEGCodec()

```
gdcm::JPEGCodec::JPEGCodec ( )
```

10.174.2.2 ~JPEGCodec()

```
gdcm::JPEGCodec::~JPEGCodec ( )  [override]
```

10.174.3 Member Function Documentation

10.174.3.1 AppendFrameEncode()

```
bool gdcm::JPEGCodec::AppendFrameEncode (
            std::ostream & out,
            const char * data,
            size_t datalen )  [override], [protected], [virtual]
```

Reimplemented from gdcm::ImageCodec.

10.174.3.2 AppendRowEncode()

```
bool gdcm::JPEGCodec::AppendRowEncode (
            std::ostream & out,
            const char * data,
            size_t datalen )  [override], [protected], [virtual]
```

Reimplemented from gdcm::ImageCodec.

10.174.3.3 CanCode()

```
bool gdcm::JPEGCodec::CanCode (
            TransferSyntax const &  ) const  [override], [virtual]
```

Return whether this coder support this transfer syntax (can code it)

Reimplemented from gdcm::ImageCodec.

Examples

CompressLossyJPEG.cs.

10.174.3.4 CanDecode()

```
bool gdcm::JPEGCodec::CanDecode (
            TransferSyntax const &  ) const  [override], [virtual]
```

Return whether this decoder support this transfer syntax (can decode it)

Reimplemented from gdcm::ImageCodec.

10.174.3.5 Clone()

```
ImageCodec * gdcm::JPEGCodec::Clone ( ) const  [override], [virtual]
```

Implements gdcm::ImageCodec.

10.174.3.6 Code()

```
bool gdcm::JPEGCodec::Code (
            DataElement const & in,
            DataElement & out )  [override], [virtual]
```

Compress into JPEG.

Reimplemented from gdcm::Coder.

10.174.3.7 ComputeOffsetTable()

```
void gdcm::JPEGCodec::ComputeOffsetTable (
            bool b )
```

Compute the offset table:

10.174.3.8 Decode()

```
bool gdcm::JPEGCodec::Decode (
            DataElement const & ,
            DataElement &  )  [override], [virtual]
```

Decode.

Reimplemented from gdcm::ImageCodec.

10.174.3.9 DecodeByStreams()

```
bool gdcm::JPEGCodec::DecodeByStreams (
            std::istream & is,
            std::ostream & os )  [override], [protected], [virtual]
```

Reimplemented from gdcm::ImageCodec.

10.174.3.10 DecodeExtent()

```
bool gdcm::JPEGCodec::DecodeExtent (
            char * buffer,
            unsigned int xmin,
            unsigned int xmax,
            unsigned int ymin,
            unsigned int ymax,
            unsigned int zmin,
            unsigned int zmax,
            std::istream & is )  [protected]
```

10.174.3.11 EncodeBuffer()

```
virtual bool gdcm::JPEGCodec::EncodeBuffer (
            std::ostream & out,
            const char * inbuffer,
            size_t inlen )  [virtual]
```

Reimplemented in gdcm::JPEG12Codec, gdcm::JPEG16Codec, and gdcm::JPEG8Codec.

10.174.3.12 GetHeaderInfo()

```
bool gdcm::JPEGCodec::GetHeaderInfo (
            std::istream & is,
            TransferSyntax & ts )  [override], [virtual]
```

Reimplemented from gdcm::ImageCodec.

Examples

GetJPEGSamplePrecision.cxx.

10.174.3.13 GetLossless()

```
bool gdcm::JPEGCodec::GetLossless ( ) const
```

10.174.3.14 GetQuality()

```
double gdcm::JPEGCodec::GetQuality ( ) const
```

10.174.3.15 IsFrameEncoder()

```
bool gdcm::JPEGCodec::IsFrameEncoder ( )  [override], [protected], [virtual]
```

Reimplemented from gdcm::ImageCodec.

10.174.3.16 IsRowEncoder()

```
bool gdcm::JPEGCodec::IsRowEncoder ( )  [override], [protected], [virtual]
```

Reimplemented from gdcm::ImageCodec.

10.174.3.17 IsStateSuspension()

```
virtual bool gdcm::JPEGCodec::IsStateSuspension ( ) const  [protected], [virtual]
```

Reimplemented in gdcm::JPEG12Codec, gdcm::JPEG16Codec, and gdcm::JPEG8Codec.

10.174.3.18 IsValid()

```
bool gdcm::JPEGCodec::IsValid (
            PhotometricInterpretation const & pi )  [override], [protected], [virtual]
```

Reimplemented from gdcm::ImageCodec.

10.174.3.19 SetBitSample()

```
void gdcm::JPEGCodec::SetBitSample (
            int bit )  [protected]
```

10.174.3.20 SetLossless()

```
void gdcm::JPEGCodec::SetLossless (
            bool l )
```

Examples

CompressLossyJPEG.cs, and FileChangeTSLossy.cs.

10.174.3.21 SetPixelFormat()

```
void gdcm::JPEGCodec::SetPixelFormat (
            PixelFormat const & pf )  [override], [virtual]
```

Reimplemented from gdcm::ImageCodec.

Examples

GetJPEGSamplePrecision.cxx.

10.174.3.22 SetQuality()

```
void gdcm::JPEGCodec::SetQuality (
            double q )
```

Examples

CompressLossyJPEG.cs, and FileChangeTSLossy.cs.

10.174.3.23 StartEncode()

```
bool gdcm::JPEGCodec::StartEncode (
            std::ostream &  )  [override], [protected], [virtual]
```

Reimplemented from gdcm::ImageCodec.

10.174.3.24 StopEncode()

```
bool gdcm::JPEGCodec::StopEncode (
            std::ostream &  )  [override], [protected], [virtual]
```

Reimplemented from gdcm::ImageCodec.

10.174.4 Friends And Related Function Documentation

10.174.4.1 ImageRegionReader

```
friend class ImageRegionReader  [friend]
```

10.174.5 Member Data Documentation

10.174.5.1 BitSample

```
int gdcm::JPEGCodec::BitSample  [protected]
```

10.174.5.2 Quality

```
int gdcm::JPEGCodec::Quality  [protected]
```

The documentation for this class was generated from the following file:

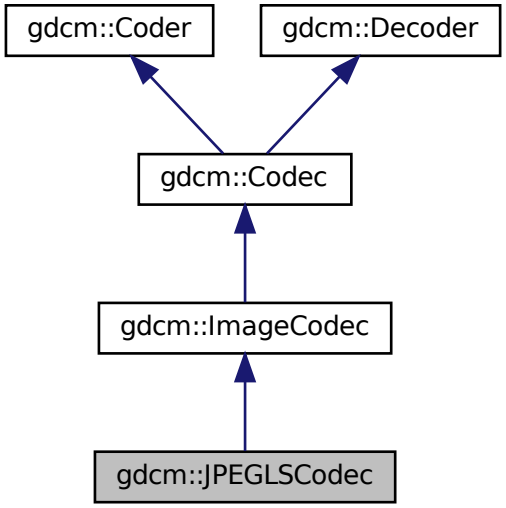
- gdcmJPEGCodec.h

10.175 gdcm::JPEGLSCodec Class Reference

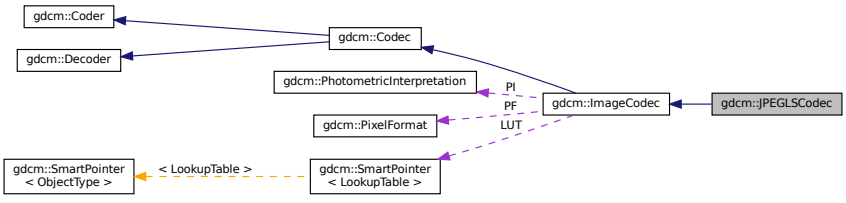
JPEG-LS.

`#include <gdcmJPEGLSCodec.h>`

Inheritance diagram for gdcm::JPEGLSCodec:



Collaboration diagram for gdcm::JPEGLSCodec:



Public Member Functions

- JPEGLSCodec ()
- ~JPEGLSCodec () override
- bool CanCode (TransferSyntax const &ts) const override

Return whether this coder support this transfer syntax (can code it)

- bool CanDecode (TransferSyntax const &ts) const override

  Return whether this decoder support this transfer syntax (can decode it)
- ImageCodec * Clone () const override
- bool Code (DataElement const &in, DataElement &out) override

  Code.
- bool Decode (DataElement const &in, char *outBuffer, size_t inBufferLength, uint32_t inXMin, uint32_t inXMax, uint32_t inYMin, uint32_t inYMax, uint32_t inZMin, uint32_t inZMax)
- bool Decode (DataElement const &is, DataElement &os) override

  Decode.
- unsigned long GetBufferLength () const
- bool GetHeaderInfo (std::istream &is, TransferSyntax &ts) override
- bool GetLossless () const
- void SetBufferLength (unsigned long l)
- void SetLossless (bool l)
- void SetLossyError (int error)

  [0-3] generally

Protected Member Functions

- bool AppendFrameEncode (std::ostream &out, const char *data, size_t datalen) override
- bool AppendRowEncode (std::ostream &out, const char *data, size_t datalen) override
- bool DecodeExtent (char *buffer, unsigned int xmin, unsigned int xmax, unsigned int ymin, unsigned int ymax, unsigned int zmin, unsigned int zmax, std::istream &is)
- bool IsFrameEncoder () override
- bool IsRowEncoder () override
- bool StartEncode (std::ostream &) override
- bool StopEncode (std::ostream &) override

Friends

- class ImageRegionReader

Additional Inherited Members

10.175.1 Detailed Description

JPEG-LS.

Note

codec that implement the JPEG-LS compression this is an implementation of ImageCodec for JPEG-LS

It uses the CharLS JPEG-LS implementation `https://github.com/team-charls/charls`

10.175.2 Constructor & Destructor Documentation

10.175.2.1 JPEGLSCodec()

```
gdcm::JPEGLSCodec::JPEGLSCodec ( )
```

10.175.2.2 ~JPEGLSCodec()

```
gdcm::JPEGLSCodec::~JPEGLSCodec ( )  [override]
```

10.175.3 Member Function Documentation

10.175.3.1 AppendFrameEncode()

```
bool gdcm::JPEGLSCodec::AppendFrameEncode (
            std::ostream & out,
            const char * data,
            size_t datalen )  [override], [protected], [virtual]
```

Reimplemented from gdcm::ImageCodec.

10.175.3.2 AppendRowEncode()

```
bool gdcm::JPEGLSCodec::AppendRowEncode (
            std::ostream & out,
            const char * data,
            size_t datalen )  [override], [protected], [virtual]
```

Reimplemented from gdcm::ImageCodec.

10.175.3.3 CanCode()

```
bool gdcm::JPEGLSCodec::CanCode (
            TransferSyntax const &  ) const  [override], [virtual]
```

Return whether this coder support this transfer syntax (can code it)

Reimplemented from gdcm::ImageCodec.

10.175.3.4 CanDecode()

```
bool gdcm::JPEGLSCodec::CanDecode (
            TransferSyntax const &  ) const  [override], [virtual]
```

Return whether this decoder support this transfer syntax (can decode it)

Reimplemented from gdcm::ImageCodec.

10.175.3.5 Clone()

```
ImageCodec * gdcm::JPEGLSCodec::Clone ( ) const  [override], [virtual]
```

Implements gdcm::ImageCodec.

10.175.3.6 Code()

```
bool gdcm::JPEGLSCodec::Code (
            DataElement const & in_,
            DataElement & out_ )  [override], [virtual]
```

Code.

Reimplemented from gdcm::Coder.

10.175.3.7 Decode() [1/2]

```
bool gdcm::JPEGLSCodec::Decode (
            DataElement const & in,
            char * outBuffer,
            size_t inBufferLength,
            uint32_t inXMin,
            uint32_t inXMax,
            uint32_t inYMin,
            uint32_t inYMax,
            uint32_t inZMin,
            uint32_t inZMax )
```

10.175.3.8 Decode() [2/2]

```
bool gdcm::JPEGLSCodec::Decode (
            DataElement const & ,
            DataElement &  )  [override], [virtual]
```

Decode.

Reimplemented from gdcm::ImageCodec.

10.175.3.9 DecodeExtent()

```
bool gdcm::JPEGLSCodec::DecodeExtent (
            char * buffer,
            unsigned int xmin,
            unsigned int xmax,
            unsigned int ymin,
            unsigned int ymax,
            unsigned int zmin,
            unsigned int zmax,
            std::istream & is )  [protected]
```

10.175.3.10 GetBufferLength()

```
unsigned long gdcm::JPEGLSCodec::GetBufferLength ( ) const  [inline]
```

10.175.3.11 GetHeaderInfo()

```
bool gdcm::JPEGLSCodec::GetHeaderInfo (
            std::istream & is,
            TransferSyntax & ts )  [override], [virtual]
```

Reimplemented from gdcm::ImageCodec.

10.175.3.12 GetLossless()

```
bool gdcm::JPEGLSCodec::GetLossless ( ) const
```

10.175.3.13 IsFrameEncoder()

```
bool gdcm::JPEGLSCodec::IsFrameEncoder ( )  [override], [protected], [virtual]
```

Reimplemented from gdcm::ImageCodec.

10.175.3.14 IsRowEncoder()

```
bool gdcm::JPEGLSCodec::IsRowEncoder ( )  [override], [protected], [virtual]
```

Reimplemented from gdcm::ImageCodec.

10.175.3.15 SetBufferLength()

```
void gdcm::JPEGLSCodec::SetBufferLength (
            unsigned long l )  [inline]
```

10.175.3.16 SetLossless()

```
void gdcm::JPEGLSCodec::SetLossless (
            bool l )
```

10.175.3.17 SetLossyError()

```
void gdcm::JPEGLSCodec::SetLossyError (
            int error )
```

[0-3] generally

10.175.3.18 StartEncode()

```
bool gdcm::JPEGLSCodec::StartEncode (
            std::ostream &  )  [override], [protected], [virtual]
```

Reimplemented from gdcm::ImageCodec.

10.175.3.19 StopEncode()

```
bool gdcm::JPEGLSCodec::StopEncode (
            std::ostream &  )  [override], [protected], [virtual]
```

Reimplemented from gdcm::ImageCodec.

10.175.4 Friends And Related Function Documentation

10.175.4.1 ImageRegionReader

```
friend class ImageRegionReader  [friend]
```

The documentation for this class was generated from the following file:

- gdcmJPEGLSCodec.h

10.176 gdcm::JSON Class Reference

```
#include <gdcmJSON.h>
```

Public Member Functions

- JSON ()
- ~JSON ()
- bool Code (DataSet const &in, std::ostream &os)
- bool Decode (std::istream &is, DataSet &out)
- bool GetPrettyPrint () const
- void PrettyPrintOff ()
- void PrettyPrintOn ()
- void SetPrettyPrint (bool onoff)

10.176.1 Detailed Description

Examples

QIDO-RS.cxx.

10.176.2 Constructor & Destructor Documentation

10.176.2.1 JSON()

```
gdcm::JSON::JSON ( )
```

10.176.2.2 ~JSON()

```
gdcm::JSON::~JSON ( )
```

10.176.3 Member Function Documentation

10.176.3.1 Code()

```
bool gdcm::JSON::Code (
            DataSet const & in,
            std::ostream & os )
```

Examples

QIDO-RS.cxx.

10.176.3.2 Decode()

```
bool gdcm::JSON::Decode (
            std::istream & is,
            DataSet & out )
```

Examples

QIDO-RS.cxx.

10.176.3.3 GetPrettyPrint()

```
bool gdcm::JSON::GetPrettyPrint ( ) const
```

10.176.3.4 PrettyPrintOff()

```
void gdcm::JSON::PrettyPrintOff ( )
```

10.176.3.5 PrettyPrintOn()

```
void gdcm::JSON::PrettyPrintOn ( )
```

Examples

QIDO-RS.cxx.

10.176.3.6 SetPrettyPrint()

```
void gdcm::JSON::SetPrettyPrint (
            bool onoff )
```

The documentation for this class was generated from the following file:

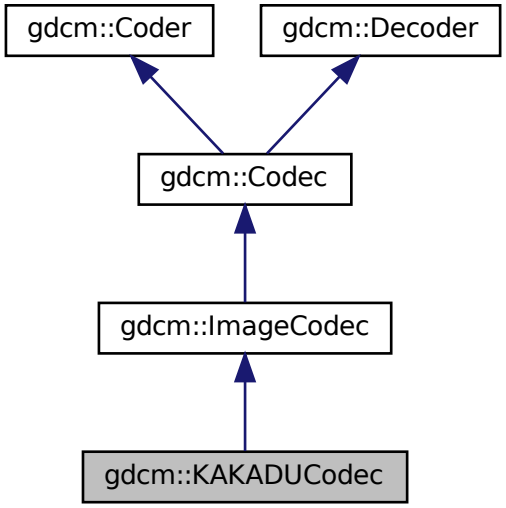
- gdcmJSON.h

10.177 gdcm::KAKADUCodec Class Reference

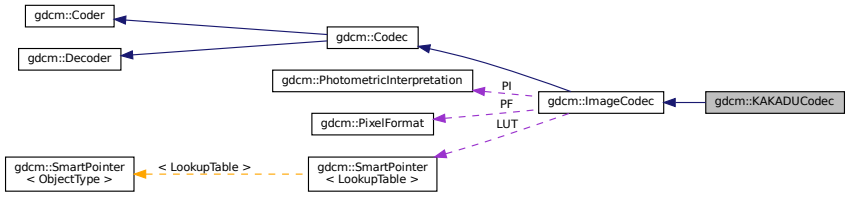
KAKADUCodec.

`#include <gdcmKAKADUCodec.h>`

Inheritance diagram for gdcm::KAKADUCodec:



Collaboration diagram for gdcm::KAKADUCodec:



Public Member Functions

- KAKADUCodec ()
- ~KAKADUCodec () override
- bool CanCode (TransferSyntax const &ts) const override

Return whether this coder support this transfer syntax (can code it)

- bool CanDecode (TransferSyntax const &ts) const override

  Return whether this decoder support this transfer syntax (can decode it)
- ImageCodec * Clone () const override
- bool Code (DataElement const &in, DataElement &out) override

  Code.
- bool Decode (DataElement const &is, DataElement &os) override

  Decode.

Additional Inherited Members

10.177.1 Detailed Description

KAKADUCodec.

10.177.2 Constructor & Destructor Documentation

10.177.2.1 KAKADUCodec()

```
gdcm::KAKADUCodec::KAKADUCodec ( )
```

10.177.2.2 ~KAKADUCodec()

```
gdcm::KAKADUCodec::~KAKADUCodec ( )  [override]
```

10.177.3 Member Function Documentation

10.177.3.1 CanCode()

```
bool gdcm::KAKADUCodec::CanCode (
            TransferSyntax const &  ) const  [override], [virtual]
```

Return whether this coder support this transfer syntax (can code it)

Reimplemented from gdcm::ImageCodec.

10.177.3.2 CanDecode()

```
bool gdcm::KAKADUCodec::CanDecode (
            TransferSyntax const &  ) const  [override], [virtual]
```

Return whether this decoder support this transfer syntax (can decode it)

Reimplemented from gdcm::ImageCodec.

10.177.3.3 Clone()

```
ImageCodec * gdcm::KAKADUCodec::Clone ( ) const  [override], [virtual]
```

Implements gdcm::ImageCodec.

10.177.3.4 Code()

```
bool gdcm::KAKADUCodec::Code (
            DataElement const & in_,
            DataElement & out_ )  [override], [virtual]
```

Code.

Reimplemented from gdcm::Coder.

10.177.3.5 Decode()

```
bool gdcm::KAKADUCodec::Decode (
            DataElement const & ,
            DataElement &  )  [override], [virtual]
```

Decode.

Reimplemented from gdcm::ImageCodec.

The documentation for this class was generated from the following file:

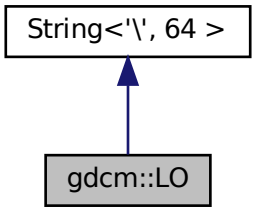
- gdcmKAKADUCodec.h

10.178 gdcm::LO Class Reference

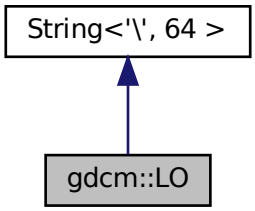
LO.

`#include <gdcmLO.h>`

Inheritance diagram for gdcm::LO:



Collaboration diagram for gdcm::LO:



Public Types

- typedef Superclass::const_iterator const_iterator
- typedef Superclass::const_reference const_reference
- typedef Superclass::const_reverse_iterator const_reverse_iterator
- typedef Superclass::difference_type difference_type
- typedef Superclass::iterator iterator
- typedef Superclass::pointer pointer
- typedef Superclass::reference reference
- typedef Superclass::reverse_iterator reverse_iterator
- typedef Superclass::size_type size_type
- typedef String<'\\', 64 > Superclass
- typedef Superclass::value_type value_type

Public Member Functions

- LO ()
- LO (const Superclass &s, size_type pos=0, size_type n=npos)
- LO (const value_type *s)
- LO (const value_type *s, size_type n)
- bool IsValid () const

10.178.1 Detailed Description

LO.

Note

TODO

10.178.2 Member Typedef Documentation

10.178.2.1 const_iterator

```
typedef Superclass::const_iterator gdcm::LO::const_iterator
```

10.178.2.2 const_reference

```
typedef Superclass::const_reference gdcm::LO::const_reference
```

10.178.2.3 const_reverse_iterator

```
typedef Superclass::const_reverse_iterator gdcm::LO::const_reverse_iterator
```

10.178.2.4 difference_type

```
typedef Superclass::difference_type gdcm::LO::difference_type
```

10.178.2.5 iterator

```
typedef Superclass::iterator gdcm::LO::iterator
```

10.178.2.6 pointer

```
typedef Superclass::pointer gdcm::LO::pointer
```

10.178.2.7 reference

```
typedef Superclass::reference gdcm::LO::reference
```

10.178.2.8 reverse_iterator

```
typedef Superclass::reverse_iterator gdcm::LO::reverse_iterator
```

10.178.2.9 size_type

```
typedef Superclass::size_type gdcm::LO::size_type
```

10.178.2.10 Superclass

```
typedef String<'\\',64> gdcm::LO::Superclass
```

10.178.2.11 value_type

```
typedef Superclass::value_type gdcm::LO::value_type
```

10.178.3 Constructor & Destructor Documentation

10.178.3.1 LO() [1/4]

```
gdcm::LO::LO ( )  [inline]
```

10.178.3.2 LO() [2/4]

```
gdcm::LO::LO (
            const value_type * s )  [inline]
```

10.178.3.3 LO() [3/4]

```
gdcm::LO::LO (
            const value_type * s,
            size_type n )  [inline]
```

10.178.3.4 LO() [4/4]

```
gdcm::LO::LO (
            const Superclass & s,
            size_type pos = 0,
            size_type n = npos )  [inline]
```

10.178.4 Member Function Documentation

10.178.4.1 IsValid()

```
bool gdcm::LO::IsValid ( ) const  [inline]
```

References gdcm::String< TDelimiter, TMaxLength, TPadChar >::IsValid().

The documentation for this class was generated from the following file:

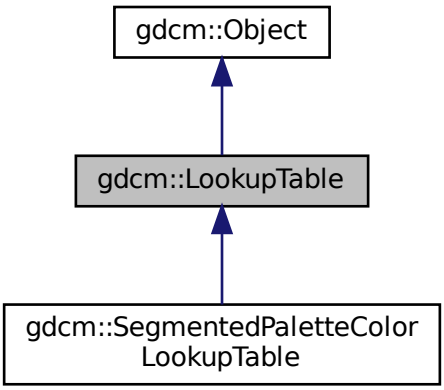
- gdcmLO.h

10.179 gdcm::LookupTable Class Reference

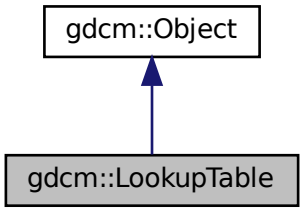
LookupTable class.

`#include <gdcmLookupTable.h>`

Inheritance diagram for gdcm::LookupTable:



Collaboration diagram for gdcm::LookupTable:



Public Types

- enum LookupTableType {
  RED = 0 ,
  GREEN ,
  BLUE ,
  GRAY ,
  UNKNOWN }

Public Member Functions

- LookupTable ()
- LookupTable (LookupTable const &lut)
- ~LookupTable () override
- void Allocate (unsigned short bitsample=8)

  Allocate the LUT.
- void Clear ()

  Clear the LUT.
- bool Decode (char *outputbuffer, size_t outlen, const char *inputbuffer, size_t inlen) const
- void Decode (std::istream &is, std::ostream &os) const

  Decode the LUT.
- bool Decode8 (char *outputbuffer, size_t outlen, const char *inputbuffer, size_t inlen) const

  Decode into RGB 8 bits space.
- unsigned short GetBitSample () const

  return the bit sample
- bool GetBufferAsRGBA (unsigned char *rgba) const

  return the LUT as RGBA buffer
- void GetLUT (LookupTableType type, unsigned char *array, unsigned int &length) const
- void GetLUTDescriptor (LookupTableType type, unsigned short &length, unsigned short &subscript, unsigned short &bitsize) const
- unsigned int GetLUTLength (LookupTableType type) const
- const unsigned char * GetPointer () const

  return a raw pointer to the LUT
- void InitializeBlueLUT (unsigned short length, unsigned short subscript, unsigned short bitsize)
- bool Initialized () const

  return whether the LUT has been initialized
- void InitializeGreenLUT (unsigned short length, unsigned short subscript, unsigned short bitsize)
- void InitializeLUT (LookupTableType type, unsigned short length, unsigned short subscript, unsigned short bitsize)

  Generic interface:
- void InitializeRedLUT (unsigned short length, unsigned short subscript, unsigned short bitsize)

  RED / GREEN / BLUE specific:
- bool IsRGB8 () const

  Return whether 16 bits LUT is in RGB 8 bits space.
- void Print (std::ostream &) const override
- void SetBlueLUT (const unsigned char *blue, unsigned int length)
- void SetGreenLUT (const unsigned char *green, unsigned int length)
- virtual void SetLUT (LookupTableType type, const unsigned char *array, unsigned int length)
- void SetRedLUT (const unsigned char *red, unsigned int length)
- bool WriteBufferAsRGBA (const unsigned char *rgba)

  Write the LUT as RGBA.

Protected Attributes

- unsigned short BitSample
- bool IncompleteLUT:1
- LookupTableInternal * Internal

Additional Inherited Members

10.179.1 Detailed Description

LookupTable class.

Examples

ExtractImageRegionWithLUT.cs, and PrintLUT.cxx.

10.179.2 Member Enumeration Documentation

10.179.2.1 LookupTableType

```
enum gdcm::LookupTable::LookupTableType
```

Enumerator

RED	
GREEN	
BLUE	
GRAY	
UNKNOWN	

10.179.3 Constructor & Destructor Documentation

10.179.3.1 LookupTable() [1/2]

```
gdcm::LookupTable::LookupTable ( )
```

10.179.3.2 ~LookupTable()

```
gdcm::LookupTable::~LookupTable ( )  [override]
```

10.179.3.3 LookupTable() [2/2]

```
gdcm::LookupTable::LookupTable (
            LookupTable const & lut )  [inline]
```

10.179.4 Member Function Documentation

10.179.4.1 Allocate()

```
void gdcm::LookupTable::Allocate (
            unsigned short bitsample = 8 )
```

Allocate the LUT.

10.179.4.2 Clear()

```
void gdcm::LookupTable::Clear ( )
```

Clear the LUT.

10.179.4.3 Decode() [1/2]

```
bool gdcm::LookupTable::Decode (
            char * outputbuffer,
            size_t outlen,
            const char * inputbuffer,
            size_t inlen ) const
```

Decode the LUT outputbuffer will contains the RGB decoded PALETTE COLOR input image of size inlen the outputbuffer should be at least 3 times the size of inlen

10.179.4.4 Decode() [2/2]

```
void gdcm::LookupTable::Decode (
            std::istream & is,
            std::ostream & os ) const
```

Decode the LUT.

Examples

ExtractImageRegionWithLUT.cs.

10.179.4.5 Decode8()

```
bool gdcm::LookupTable::Decode8 (
            char * outputbuffer,
            size_t outlen,
            const char * inputbuffer,
            size_t inlen ) const
```

Decode into RGB 8 bits space.

10.179.4.6 GetBitSample()

```
unsigned short gdcm::LookupTable::GetBitSample ( ) const  [inline]
```

return the bit sample

10.179.4.7 GetBufferAsRGBA()

```
bool gdcm::LookupTable::GetBufferAsRGBA (
            unsigned char * rgba ) const
```

return the LUT as RGBA buffer

10.179.4.8 GetLUT()

```
void gdcm::LookupTable::GetLUT (
            LookupTableType type,
            unsigned char * array,
            unsigned int & length ) const
```

10.179.4.9 GetLUTDescriptor()

```
void gdcm::LookupTable::GetLUTDescriptor (
            LookupTableType type,
            unsigned short & length,
            unsigned short & subscript,
            unsigned short & bitsize ) const
```

10.179.4.10 GetLUTLength()

```
unsigned int gdcm::LookupTable::GetLUTLength (
             LookupTableType type ) const
```

10.179.4.11 GetPointer()

```
const unsigned char * gdcm::LookupTable::GetPointer ( ) const
```

return a raw pointer to the LUT

10.179.4.12 InitializeBlueLUT()

```
void gdcm::LookupTable::InitializeBlueLUT (
             unsigned short length,
             unsigned short subscript,
             unsigned short bitsize )
```

10.179.4.13 Initialized()

```
bool gdcm::LookupTable::Initialized ( ) const
```

return whether the LUT has been initialized

10.179.4.14 InitializeGreenLUT()

```
void gdcm::LookupTable::InitializeGreenLUT (
             unsigned short length,
             unsigned short subscript,
             unsigned short bitsize )
```

10.179.4.15 InitializeLUT()

```
void gdcm::LookupTable::InitializeLUT (
             LookupTableType type,
             unsigned short length,
             unsigned short subscript,
             unsigned short bitsize )
```

Generic interface:

10.179.4.16 InitializeRedLUT()

```
void gdcm::LookupTable::InitializeRedLUT (
             unsigned short length,
             unsigned short subscript,
             unsigned short bitsize )
```

RED / GREEN / BLUE specific:

10.179.4.17 IsRGB8()

```
bool gdcm::LookupTable::IsRGB8 ( ) const
```

Return whether 16 bits LUT is in RGB 8 bits space.

10.179.4.18 Print()

```
void gdcm::LookupTable::Print (
             std::ostream &  ) const  [override], [virtual]
```

Reimplemented from gdcm::Object.

Reimplemented in gdcm::SegmentedPaletteColorLookupTable.

Examples

PrintLUT.cxx.

10.179.4.19 SetBlueLUT()

```
void gdcm::LookupTable::SetBlueLUT (
            const unsigned char * blue,
            unsigned int length )
```

10.179.4.20 SetGreenLUT()

```
void gdcm::LookupTable::SetGreenLUT (
            const unsigned char * green,
            unsigned int length )
```

10.179.4.21 SetLUT()

```
virtual void gdcm::LookupTable::SetLUT (
            LookupTableType type,
            const unsigned char * array,
            unsigned int length )  [virtual]
```

Reimplemented in gdcm::SegmentedPaletteColorLookupTable.

10.179.4.22 SetRedLUT()

```
void gdcm::LookupTable::SetRedLUT (
            const unsigned char * red,
            unsigned int length )
```

10.179.4.23 WriteBufferAsRGBA()

```
bool gdcm::LookupTable::WriteBufferAsRGBA (
            const unsigned char * rgba )
```

Write the LUT as RGBA.

10.179.5 Member Data Documentation

10.179.5.1 BitSample

```
unsigned short gdcm::LookupTable::BitSample  [protected]
```

10.179.5.2 IncompleteLUT

```
bool gdcm::LookupTable::IncompleteLUT  [protected]
```

10.179.5.3 Internal

```
LookupTableInternal* gdcm::LookupTable::Internal  [protected]
```

The documentation for this class was generated from the following file:

- gdcmLookupTable.h

10.180 gdcm::Scanner2::ltstr Struct Reference

```
#include <gdcmScanner2.h>
```

Public Member Functions

- bool operator() (const char *s1, const char *s2) const

10.180.1 Member Function Documentation

10.180.1.1 operator()()

```
bool gdcm::Scanner2::ltstr::operator() (
            const char * s1,
            const char * s2 ) const  [inline]
```

The documentation for this struct was generated from the following file:

- gdcmScanner2.h

10.181 gdcm::Scanner::ltstr Struct Reference

`#include <gdcmScanner.h>`

Public Member Functions

- bool operator() (const char *s1, const char *s2) const

10.181.1 Member Function Documentation

10.181.1.1 operator()()

```
bool gdcm::Scanner::ltstr::operator() (
            const char * s1,
            const char * s2 ) const  [inline]
```

The documentation for this struct was generated from the following file:

- gdcmScanner.h

10.182 gdcm::StrictScanner2::ltstr Struct Reference

`#include <gdcmStrictScanner2.h>`

Public Member Functions

- bool operator() (const char *s1, const char *s2) const

10.182.1 Member Function Documentation

10.182.1.1 operator()()

```
bool gdcm::StrictScanner2::ltstr::operator() (
            const char * s1,
            const char * s2 ) const  [inline]
```

The documentation for this struct was generated from the following file:

- gdcmStrictScanner2.h

10.183 gdcm::StrictScanner::ltstr Struct Reference

`#include <gdcmStrictScanner.h>`

Public Member Functions

- bool operator() (const char *s1, const char *s2) const

10.183.1 Member Function Documentation

10.183.1.1 operator()()

```
bool gdcm::StrictScanner::ltstr::operator() (
            const char * s1,
            const char * s2 ) const  [inline]
```

The documentation for this struct was generated from the following file:

- gdcmStrictScanner.h

10.184 gdcm::Macro Class Reference

Class for representing a Macro.

`#include <gdcmMacro.h>`

Public Types

- typedef std::vector< std::string > ArrayIncludeMacrosType
- typedef std::map< Tag, MacroEntry > MapModuleEntry

Public Member Functions

- Macro ()=default
- void AddMacroEntry (const Tag &tag, const MacroEntry &module)

  Will add a ModuleEntry directly at root-level. See Macro for nested-included level.
- void Clear ()
- bool FindMacroEntry (const Tag &tag) const
- const MacroEntry & GetMacroEntry (const Tag &tag) const
- const char * GetName () const
- void SetName (const char *name)
- bool Verify (const DataSet &ds, Usage const &usage) const

Friends

- std::ostream & operator<< (std::ostream &_os, const Macro &_val)

10.184.1 Detailed Description

Class for representing a Macro.

Note

Attribute Macro: a set of Attributes that are described in a single table that is referenced by multiple Module or other tables.

See also

Module

10.184.2 Member Typedef Documentation

10.184.2.1 ArrayIncludeMacrosType

```
typedef std::vector<std::string> gdcm::Macro::ArrayIncludeMacrosType
```

10.184.2.2 MapModuleEntry

```
typedef std::map<Tag, MacroEntry> gdcm::Macro::MapModuleEntry
```

10.184.3 Constructor & Destructor Documentation

10.184.3.1 Macro()

```
gdcm::Macro::Macro ( )  [default]
```

10.184.4 Member Function Documentation

10.184.4.1 AddMacroEntry()

```
void gdcm::Macro::AddMacroEntry (
            const Tag & tag,
            const MacroEntry & module )  [inline]
```

Will add a ModuleEntry directly at root-level. See Macro for nested-included level.

10.184.4.2 Clear()

```
void gdcm::Macro::Clear ( )  [inline]
```

10.184.4.3 FindMacroEntry()

```
bool gdcm::Macro::FindMacroEntry (
            const Tag & tag ) const
```

Find or Get a ModuleEntry. ModuleEntry are either search are root-level or within nested-macro included in module.

10.184.4.4 GetMacroEntry()

```
const MacroEntry & gdcm::Macro::GetMacroEntry (
            const Tag & tag ) const
```

10.184.4.5 GetName()

```
const char * gdcm::Macro::GetName ( ) const  [inline]
```

10.184.4.6 SetName()

```
void gdcm::Macro::SetName (
            const char * name )  [inline]
```

10.184.4.7 Verify()

```
bool gdcm::Macro::Verify (
            const DataSet & ds,
            Usage const & usage ) const
```

10.184.5 Friends And Related Function Documentation

10.184.5.1 operator<<

```
std::ostream & operator<< (
            std::ostream & _os,
            const Macro & _val )  [friend]
```

The documentation for this class was generated from the following file:

- gdcmMacro.h

10.185 gdcm::Macros Class Reference

Class for representing a Modules.

```
#include <gdcmMacros.h>
```

Public Types

- typedef std::map< std::string, Macro > ModuleMapType

Public Member Functions

- Macros ()=default
- void AddMacro (const char *ref, const Macro &module)
- void Clear ()
- const Macro & GetMacro (const char *name) const
- bool IsEmpty () const

Friends

- std::ostream & operator<< (std::ostream &_os, const Macros &_val)

10.185.1 Detailed Description

Class for representing a Modules.

Note

bla

See also

Module

Examples

TraverseModules.cxx.

10.185.2 Member Typedef Documentation

10.185.2.1 ModuleMapType

```
typedef std::map<std::string, Macro> gdcm::Macros::ModuleMapType
```

10.185.3 Constructor & Destructor Documentation

10.185.3.1 Macros()

```
gdcm::Macros::Macros ( )  [default]
```

10.185.4 Member Function Documentation

10.185.4.1 AddMacro()

```
void gdcm::Macros::AddMacro (
            const char * ref,
            const Macro & module )  [inline]
```

10.185.4.2 Clear()

```
void gdcm::Macros::Clear ( )  [inline]
```

10.185.4.3 GetMacro()

```
const Macro & gdcm::Macros::GetMacro (
            const char * name ) const  [inline]
```

10.185.4.4 IsEmpty()

```
bool gdcm::Macros::IsEmpty ( ) const  [inline]
```

10.185.5 Friends And Related Function Documentation

10.185.5.1 operator<<

```
std::ostream & operator<< (
            std::ostream & _os,
            const Macros & _val )  [friend]
```

The documentation for this class was generated from the following file:

- gdcmMacros.h

10.186 gdcm::network::MaximumLengthSub Class Reference

MaximumLengthSub.

```
#include <gdcmMaximumLengthSub.h>
```

Public Member Functions

- MaximumLengthSub ()
- uint32_t GetMaximumLength () const
- void Print (std::ostream &os) const
- std::istream & Read (std::istream &is)
- void SetMaximumLength (uint32_t maximumlength)
- size_t Size () const
- const std::ostream & Write (std::ostream &os) const

10.186.1 Detailed Description

MaximumLengthSub.

Annex D Table D.1-1 MAXIMUM LENGTH SUB-ITEM FIELDS (A-ASSOCIATE-RQ)

or

Table D.1-2 Maximum length sub-item fields (A-ASSOCIATE-AC)

10.186.2 Constructor & Destructor Documentation

10.186.2.1 MaximumLengthSub()

```
gdcm::network::MaximumLengthSub::MaximumLengthSub ( )
```

10.186.3 Member Function Documentation

10.186.3.1 GetMaximumLength()

```
uint32_t gdcm::network::MaximumLengthSub::GetMaximumLength ( ) const  [inline]
```

10.186.3.2 Print()

```
void gdcm::network::MaximumLengthSub::Print (
            std::ostream & os ) const
```

10.186.3.3 Read()

```
std::istream & gdcm::network::MaximumLengthSub::Read (
             std::istream & is )
```

10.186.3.4 SetMaximumLength()

```
void gdcm::network::MaximumLengthSub::SetMaximumLength (
             uint32_t maximumlength )
```

10.186.3.5 Size()

```
size_t gdcm::network::MaximumLengthSub::Size ( ) const
```

10.186.3.6 Write()

```
const std::ostream & gdcm::network::MaximumLengthSub::Write (
             std::ostream & os ) const
```

The documentation for this class was generated from the following file:

- gdcmMaximumLengthSub.h

10.187 gdcm::MD5 Class Reference

Class for MD5.

```
#include <gdcmMD5.h>
```

Static Public Member Functions

- static bool Compute (const char *buffer, size_t buf_len, char digest_str[33])
- static bool ComputeFile (const char *filename, char digest_str[33])

  Compute md5 from a file `filename`

10.187.1 Detailed Description

Class for MD5.

Warning

this class is able to pick from two implementations:

1. a lightweight md5 implementation (when GDCM_BUILD_TESTING is turned ON)
2. the one from OpenSSL (when GDCM_USE_SYSTEM_OPENSSL is turned ON)

In all other cases it will return an error

10.187.2 Member Function Documentation

10.187.2.1 Compute()

```
static bool gdcm::MD5::Compute (
             const char * buffer,
             size_t buf_len,
             char digest_str[33] )  [static]
```

10.187.2.2 ComputeFile()

```
static bool gdcm::MD5::ComputeFile (
             const char * filename,
             char digest_str[33] )  [static]
```

Compute md5 from a file `filename`

The documentation for this class was generated from the following file:

- gdcmMD5.h

10.188 gdcm::MediaStorage Class Reference

MediaStorage.

```
#include <gdcmMediaStorage.h>
```

Public Types

- enum MSType {
  MediaStorageDirectoryStorage = 0 ,
  ComputedRadiographyImageStorage ,
  DigitalXRayImageStorageForPresentation ,
  DigitalXRayImageStorageForProcessing ,
  DigitalMammographyImageStorageForPresentation ,
  DigitalMammographyImageStorageForProcessing ,
  DigitalIntraoralXrayImageStorageForPresentation ,
  DigitalIntraoralXRayImageStorageForProcessing ,
  CTImageStorage ,
  EnhancedCTImageStorage ,
  UltrasoundImageStorageRetired ,
  UltrasoundImageStorage ,
  UltrasoundMultiFrameImageStorageRetired ,
  UltrasoundMultiFrameImageStorage ,
  MRImageStorage ,
  EnhancedMRImageStorage ,
  MRSpectroscopyStorage ,
  NuclearMedicineImageStorageRetired ,
  SecondaryCaptureImageStorage ,
  MultiframeSingleBitSecondaryCaptureImageStorage ,
  MultiframeGrayscaleByteSecondaryCaptureImageStorage ,
  MultiframeGrayscaleWordSecondaryCaptureImageStorage ,
  MultiframeTrueColorSecondaryCaptureImageStorage ,
  StandaloneOverlayStorage ,
  StandaloneCurveStorage ,
  LeadECGWaveformStorage ,
  GeneralECGWaveformStorage ,
  AmbulatoryECGWaveformStorage ,
  HemodynamicWaveformStorage ,
  CardiacElectrophysiologyWaveformStorage ,
  BasicVoiceAudioWaveformStorage ,
  StandaloneModalityLUTStorage ,
  StandaloneVOILUTStorage ,
  GrayscaleSoftcopyPresentationStateStorageSOPClass ,
  XRayAngiographicImageStorage ,
  XRayRadiofluoroscopingImageStorage ,
  XRayAngiographicBiPlaneImageStorageRetired ,
  NuclearMedicineImageStorage ,
  RawDataStorage ,
  SpacialRegistrationStorage ,
  SpacialFiducialsStorage ,
  PETImageStorage ,
  RTImageStorage ,
  RTDoseStorage ,
  RTStructureSetStorage ,
  RTPlanStorage ,
  CSANonImageStorage ,
  Philips3D ,
  EnhancedSR ,
  BasicTextSR ,
  HardcopyGrayscaleImageStorage ,

ComprehensiveSR ,
DetachedStudyManagementSOPClass ,
EncapsulatedPDFStorage ,
EncapsulatedCDAStorage ,
StudyComponentManagementSOPClass ,
DetachedVisitManagementSOPClass ,
DetachedPatientManagementSOPClass ,
VideoEndoscopicImageStorage ,
GeneralElectricMagneticResonanceImageStorage ,
GEPrivate3DModelStorage ,
ToshibaPrivateDataStorage ,
MammographyCADSR ,
KeyObjectSelectionDocument ,
HangingProtocolStorage ,
ModalityPerformedProcedureStepSOPClass ,
PhilipsPrivateMRSyntheticImageStorage ,
VLPhotographicImageStorage ,
SegmentationStorage ,
RTIonPlanStorage ,
XRay3DAngiographicImageStorage ,
EnhancedXAImageStorage ,
RTIonBeamsTreatmentRecordStorage ,
SurfaceSegmentationStorage ,
VLWholeSlideMicroscopyImageStorage ,
RTTreatmentSummaryRecordStorage ,
EnhancedUSVolumeStorage ,
XRayRadiationDoseSR ,
VLEndoscopicImageStorage ,
BreastTomosynthesisImageStorage ,
FujiPrivateCRImageStorage ,
OphthalmicPhotography8BitImageStorage ,
OphthalmicTomographyImageStorage ,
VLMicroscopicImageStorage ,
EnhancedPETImageStorage ,
VideoPhotographicImageStorage ,
XRay3DCraniofacialImageStorage ,
IVOCTForPresentation ,
IVOCTForProcessing ,
LegacyConvertedEnhancedCTImageStorage ,
LegacyConvertedEnhancedMRImageStorage ,
LegacyConvertedEnhancedPETImageStorage ,
BreastProjectionXRayImageStorageForPresentation ,
BreastProjectionXRayImageStorageForProcessing ,
HardcopyColorImageStorage ,
EnhancedMRColorImageStorage ,
FujiPrivateMammoCRImageStorage ,
OphthalmicPhotography16BitImageStorage ,
VideoMicroscopicImageStorage ,
MS_END }

- enum ObjectType {
NoObject = 0 ,
Video ,
Waveform ,
Audio ,

PDF ,
URI ,
Segmentation ,
ObjectEnd }

Public Member Functions

- MediaStorage (MSType type=MS_END)
- const char * GetModality () const
- unsigned int GetModalityDimension () const
- const char * GetString () const

  Return the Media String of the object.
- void GuessFromModality (const char *modality, unsigned int dimension=2)
- bool IsUndefined () const
- operator MSType () const
- bool SetFromDataSet (DataSet const &ds)
- bool SetFromFile (File const &file)
- bool SetFromHeader (FileMetaInformation const &fmi)
- bool SetFromModality (DataSet const &ds)

Static Public Member Functions

- static const char * GetMSString (MSType ts)

  Return the Media String associated. Will return NULL for MS_END.
- static MSType GetMSType (const char *str)
- static unsigned int GetNumberOfModality ()
- static unsigned int GetNumberOfMSString ()
- static unsigned int GetNumberOfMSType ()
- static bool IsImage (MSType ts)

Protected Member Functions

- void SetFromSourceImageSequence (DataSet const &ds)

Friends

- std::ostream & operator<< (std::ostream &os, const MediaStorage &ms)

10.188.1 Detailed Description

MediaStorage.

Note

FIXME There should not be any notion of Image and/or PDF at that point Only the codec can answer yes I support this Media Storage or not... For instance an ImageCodec will answer yes to most of them while a PDFCodec will answer only for the Encapsulated PDF

See also

UIDs

Examples

CreateJPIPDataSet.cxx, EncapsulateFileInRawData.cxx, Extracting_All_Resolution.cxx, Fake_Image_Using_Stream_Image_Writer GenAllVR.cxx, GenFakeIdentifyFile.cxx, GenerateStandardSOPClasses.cxx, GetSubSequenceData.cxx, MpegVideoInfo.cs, ReadAndDumpDICOMDIR.cxx, ReadAndDumpDICOMDIR2.cxx, StreamImageReaderTest.cxx, TemplateEmptyImage.cxx, TestReader.cxx, gdcmrtionplan.cxx, gdcmrtplan.cxx, and iU22tomultisc.cxx.

10.188.2 Member Enumeration Documentation

10.188.2.1 MSType

`enum gdcm::MediaStorage::MSType`

Enumerator

MediaStorageDirectoryStorage	
ComputedRadiographyImageStorage	
DigitalXRayImageStorageForPresentation	
DigitalXRayImageStorageForProcessing	
DigitalMammographyImageStorageForPresentation	
DigitalMammographyImageStorageForProcessing	
DigitalIntraoralXrayImageStorageForPresentation	
DigitalIntraoralXRayImageStorageForProcessing	
CTImageStorage	
EnhancedCTImageStorage	
UltrasoundImageStorageRetired	
UltrasoundImageStorage	
UltrasoundMultiFrameImageStorageRetired	
UltrasoundMultiFrameImageStorage	
MRImageStorage	

Enumerator

EnhancedMRImageStorage	
MRSpectroscopyStorage	
NuclearMedicineImageStorageRetired	
SecondaryCaptureImageStorage	
MultiframeSingleBitSecondaryCaptureImageStorage	
MultiframeGrayscaleByteSecondaryCaptureImageStorage	
MultiframeGrayscaleWordSecondaryCaptureImageStorage	
MultiframeTrueColorSecondaryCaptureImageStorage	
StandaloneOverlayStorage	
StandaloneCurveStorage	
LeadECGWaveformStorage	
GeneralECGWaveformStorage	
AmbulatoryECGWaveformStorage	
HemodynamicWaveformStorage	
CardiacElectrophysiologyWaveformStorage	
BasicVoiceAudioWaveformStorage	
StandaloneModalityLUTStorage	
StandaloneVOILUTStorage	
GrayscaleSoftcopyPresentationStateStorageSOPClass	
XRayAngiographicImageStorage	
XRayRadiofluoroscopingImageStorage	
XRayAngiographicBiPlaneImageStorageRetired	
NuclearMedicineImageStorage	
RawDataStorage	
SpacialRegistrationStorage	
SpacialFiducialsStorage	
PETImageStorage	
RTImageStorage	
RTDoseStorage	
RTStructureSetStorage	
RTPlanStorage	
CSANonImageStorage	
Philips3D	
EnhancedSR	
BasicTextSR	
HardcopyGrayscaleImageStorage	
ComprehensiveSR	
DetachedStudyManagementSOPClass	
EncapsulatedPDFStorage	
EncapsulatedCDAStorage	
StudyComponentManagementSOPClass	
DetachedVisitManagementSOPClass	
DetachedPatientManagementSOPClass	

Enumerator

VideoEndoscopicImageStorage	
GeneralElectricMagneticResonanceImageStorage	
GEPrivate3DModelStorage	
ToshibaPrivateDataStorage	
MammographyCADSR	
KeyObjectSelectionDocument	
HangingProtocolStorage	
ModalityPerformedProcedureStepSOPClass	
PhilipsPrivateMRSyntheticImageStorage	
VLPhotographicImageStorage	
SegmentationStorage	
RTIonPlanStorage	
XRay3DAngiographicImageStorage	
EnhancedXAImageStorage	
RTIonBeamsTreatmentRecordStorage	
SurfaceSegmentationStorage	
VLWholeSlideMicroscopyImageStorage	
RTTreatmentSummaryRecordStorage	
EnhancedUSVolumeStorage	
XRayRadiationDoseSR	
VLEndoscopicImageStorage	
BreastTomosynthesisImageStorage	
FujiPrivateCRImageStorage	
OphthalmicPhotography8BitImageStorage	
OphthalmicTomographyImageStorage	
VLMicroscopicImageStorage	
EnhancedPETImageStorage	
VideoPhotographicImageStorage	
XRay3DCraniofacialImageStorage	
IVOCTForPresentation	
IVOCTForProcessing	
LegacyConvertedEnhancedCTImageStorage	
LegacyConvertedEnhancedMRImageStorage	
LegacyConvertedEnhancedPETImageStorage	
BreastProjectionXRayImageStorageForPresentation	
BreastProjectionXRayImageStorageForProcessing	
HardcopyColorImageStorage	
EnhancedMRColorImageStorage	
FujiPrivateMammoCRImageStorage	
OphthalmicPhotography16BitImageStorage	
VideoMicroscopicImageStorage	
MS_END	

Examples

GenerateStandardSOPClasses.cxx, and MpegVideoInfo.cs.

10.188.2.2 ObjectType

```
enum gdcm::MediaStorage::ObjectType
```

Enumerator

NoObject	
Video	
Waveform	
Audio	
PDF	
URI	
Segmentation	
ObjectEnd	

10.188.3 Constructor & Destructor Documentation

10.188.3.1 MediaStorage()

```
gdcm::MediaStorage::MediaStorage (
            MSType type = MS_END )  [inline]
```

10.188.4 Member Function Documentation

10.188.4.1 GetModality()

```
const char * gdcm::MediaStorage::GetModality ( ) const
```

10.188.4.2 GetModalityDimension()

```
unsigned int gdcm::MediaStorage::GetModalityDimension ( ) const
```

10.188.4.3 GetMSString()

```
static const char * gdcm::MediaStorage::GetMSString (
            MSType ts )  [static]
```

Return the Media String associated. Will return NULL for MS_END.

Examples

GenerateStandardSOPClasses.cxx.

10.188.4.4 GetMSType()

```
static MSType gdcm::MediaStorage::GetMSType (
            const char * str )  [static]
```

Examples

MetaImageMD5Activiz.cs, and TestReader.cxx.

10.188.4.5 GetNumberOfModality()

```
static unsigned int gdcm::MediaStorage::GetNumberOfModality ( )  [static]
```

10.188.4.6 GetNumberOfMSString()

```
static unsigned int gdcm::MediaStorage::GetNumberOfMSString ( )  [static]
```

10.188.4.7 GetNumberOfMSType()

```
static unsigned int gdcm::MediaStorage::GetNumberOfMSType ( )  [static]
```

10.188.4.8 GetString()

```
const char * gdcm::MediaStorage::GetString ( ) const
```

Return the Media String of the object.

Examples

CreateJPIPDataSet.cxx, EncapsulateFileInRawData.cxx, Fake_Image_Using_Stream_Image_Writer.cxx, GetSubSequenceData.cxx, MpegVideoInfo.cs, StreamImageReaderTest.cxx, TemplateEmptyImage.cxx, and iU22tomultisc.cxx.

10.188.4.9 GuessFromModality()

```
void gdcm::MediaStorage::GuessFromModality (
            const char * modality,
            unsigned int dimension = 2 )
```

10.188.4.10 IsImage()

```
static bool gdcm::MediaStorage::IsImage (
            MSType ts )  [static]
```

Returns whether DICOM has a Pixel Data element (7fe0,0010)

Warning

MRSpectroscopyStorage could be image but are not

Examples

MetaImageMD5Activiz.cs.

10.188.4.11 IsUndefined()

```
bool gdcm::MediaStorage::IsUndefined ( ) const  [inline]
```

Examples

TestReader.cxx.

10.188.4.12 operator MSType()

```
gdcm::MediaStorage::operator MSType ( ) const  [inline]
```

10.188.4.13 SetFromDataSet()

```
bool gdcm::MediaStorage::SetFromDataSet (
            DataSet const & ds )
```

Advanced user only (functions should be protected level...) Those function are lower level than SetFromFile

10.188.4.14 SetFromFile()

```
bool gdcm::MediaStorage::SetFromFile (
            File const & file )
```

Attempt to set the MediaStorage from a file: WARNING: When no MediaStorage & Modality are found BUT a PixelData element is found then MediaStorage is set to the default SecondaryCaptureImageStorage (return value is false in this case)

Examples

ReadAndDumpDICOMDIR.cxx, ReadAndDumpDICOMDIR2.cxx, TestReader.cxx, gdcmrtionplan.cxx, and gdcmrtplan.cxx.

10.188.4.15 SetFromHeader()

```
bool gdcm::MediaStorage::SetFromHeader (
            FileMetaInformation const & fmi )
```

10.188.4.16 SetFromModality()

```
bool gdcm::MediaStorage::SetFromModality (
            DataSet const & ds )
```

10.188.4.17 SetFromSourceImageSequence()

```
void gdcm::MediaStorage::SetFromSourceImageSequence (
            DataSet const & ds )  [protected]
```

10.188.5 Friends And Related Function Documentation

10.188.5.1 operator<<

```
std::ostream & operator<< (
            std::ostream & os,
            const MediaStorage & ms )  [friend]
```

The documentation for this class was generated from the following file:

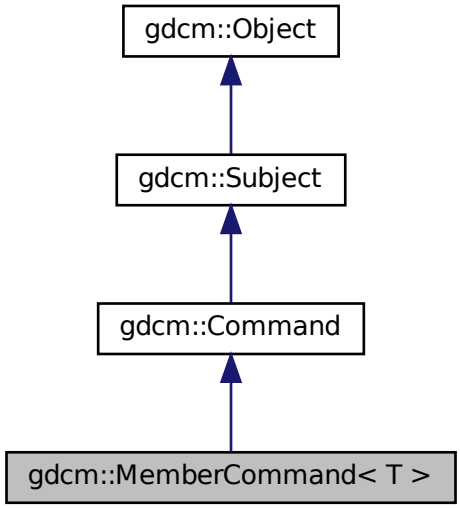
- gdcmMediaStorage.h

10.189 gdcm::MemberCommand< T > Class Template Reference

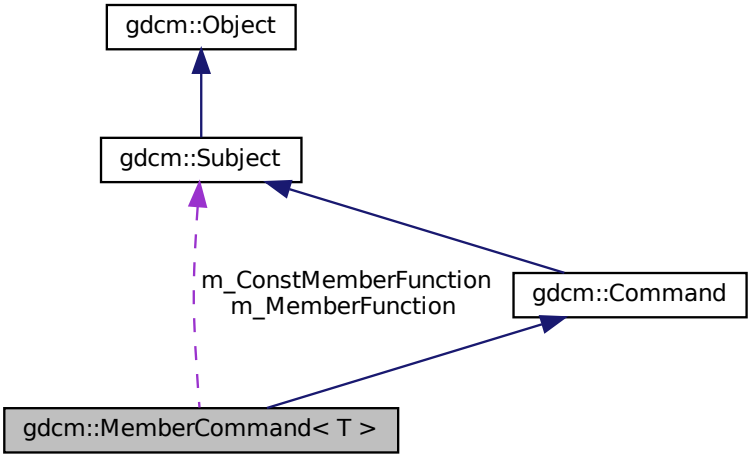
Command subclass that calls a pointer to a member function.

```
#include <gdcmCommand.h>
```

Inheritance diagram for gdcm::MemberCommand< T >:



Collaboration diagram for gdcm::MemberCommand< T >:



Public Types

- typedef MemberCommand Self

- typedef void(T::* TConstMemberFunctionPointer) (const Subject *, const Event &)
- typedef void(T::* TMemberFunctionPointer) (Subject *, const Event &)

Public Member Functions

- MemberCommand (const Self &)=delete
- void Execute (const Subject *caller, const Event &event) override
- void Execute (Subject *caller, const Event &event) override
- void operator= (const Self &)=delete
- void SetCallbackFunction (T *object, TConstMemberFunctionPointer memberFunction)
- void SetCallbackFunction (T *object, TMemberFunctionPointer memberFunction)

Static Public Member Functions

- static SmartPointer< MemberCommand > New ()

Protected Member Functions

- MemberCommand ()
- ~MemberCommand () override=default

Protected Attributes

- TConstMemberFunctionPointer m_ConstMemberFunction
- TMemberFunctionPointer m_MemberFunction
- T * m_This

10.189.1 Detailed Description

template<class T>
class gdcm::MemberCommand< T >

Command subclass that calls a pointer to a member function.

MemberCommand calls a pointer to a member function with the same arguments as Execute on Command.

10.189.2 Member Typedef Documentation

10.189.2.1 Self

```
template<class T >
typedef MemberCommand gdcm::MemberCommand< T >::Self
```

Standard class typedefs.

10.189.2.2 TConstMemberFunctionPointer

```
template<class T >
typedef void(T::* gdcm::MemberCommand< T >::TConstMemberFunctionPointer) (const Subject *, const
Event &)
```

10.189.2.3 TMemberFunctionPointer

```
template<class T >
typedef void(T::* gdcm::MemberCommand< T >::TMemberFunctionPointer) (Subject *, const Event &)
```

pointer to a member function that takes a Subject∗ and the event

10.189.3 Constructor & Destructor Documentation

10.189.3.1 MemberCommand() [1/2]

```
template<class T >
gdcm::MemberCommand< T >::MemberCommand (
            const Self &  )  [delete]
```

10.189.3.2 MemberCommand() [2/2]

```
template<class T >
gdcm::MemberCommand< T >::MemberCommand ( )  [inline], [protected]
```

Referenced by gdcm::MemberCommand< T >::New().

10.189.3.3 ~MemberCommand()

```
template<class T >
gdcm::MemberCommand< T >::~MemberCommand ( )  [override], [protected], [default]
```

10.189.4 Member Function Documentation

10.189.4.1 Execute() [1/2]

```
template<class T >
void gdcm::MemberCommand< T >::Execute (
            const Subject * caller,
            const Event & event )  [inline], [override], [virtual]
```

Invoke the member function with a const object.

Implements gdcm::Command.

References gdcm::MemberCommand< T >::m_ConstMemberFunction.

10.189.4.2 Execute() [2/2]

```
template<class T >
void gdcm::MemberCommand< T >::Execute (
            Subject * caller,
            const Event & event )  [inline], [override], [virtual]
```

Invoke the member function.

Implements gdcm::Command.

References gdcm::MemberCommand< T >::m_MemberFunction.

10.189.4.3 New()

```
template<class T >
static SmartPointer< MemberCommand > gdcm::MemberCommand< T >::New ( )  [inline], [static]
```

Method for creation through the object factory.

References gdcm::MemberCommand< T >::MemberCommand().

10.189.4.4 operator=()

```
template<class T >
void gdcm::MemberCommand< T >::operator= (
             const Self &  )  [delete]
```

10.189.4.5 SetCallbackFunction() [1/2]

```
template<class T >
void gdcm::MemberCommand< T >::SetCallbackFunction (
             T * object,
             TConstMemberFunctionPointer memberFunction )  [inline]
```

References gdcm::MemberCommand< T >::m_ConstMemberFunction, and gdcm::MemberCommand< T >::m_This.

10.189.4.6 SetCallbackFunction() [2/2]

```
template<class T >
void gdcm::MemberCommand< T >::SetCallbackFunction (
             T * object,
             TMemberFunctionPointer memberFunction )  [inline]
```

Run-time type information (and related methods). Set the callback function along with the object that it will be invoked on.

References gdcm::MemberCommand< T >::m_MemberFunction, and gdcm::MemberCommand< T >::m_This.

10.189.5 Member Data Documentation

10.189.5.1 m_ConstMemberFunction

```
template<class T >
TConstMemberFunctionPointer gdcm::MemberCommand< T >::m_ConstMemberFunction  [protected]
```

Referenced by gdcm::MemberCommand< T >::Execute(), and gdcm::MemberCommand< T >::SetCallbackFunction().

10.189.5.2 m_MemberFunction

```
template<class T >
TMemberFunctionPointer gdcm::MemberCommand< T >::m_MemberFunction  [protected]
```

Referenced by gdcm::MemberCommand< T >::Execute(), and gdcm::MemberCommand< T >::SetCallbackFunction().

10.189.5.3 m_This

```
template<class T >
T* gdcm::MemberCommand< T >::m_This  [protected]
```

Referenced by gdcm::MemberCommand< T >::SetCallbackFunction().

The documentation for this class was generated from the following file:

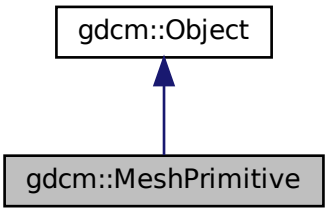
- gdcmCommand.h

10.190 gdcm::MeshPrimitive Class Reference

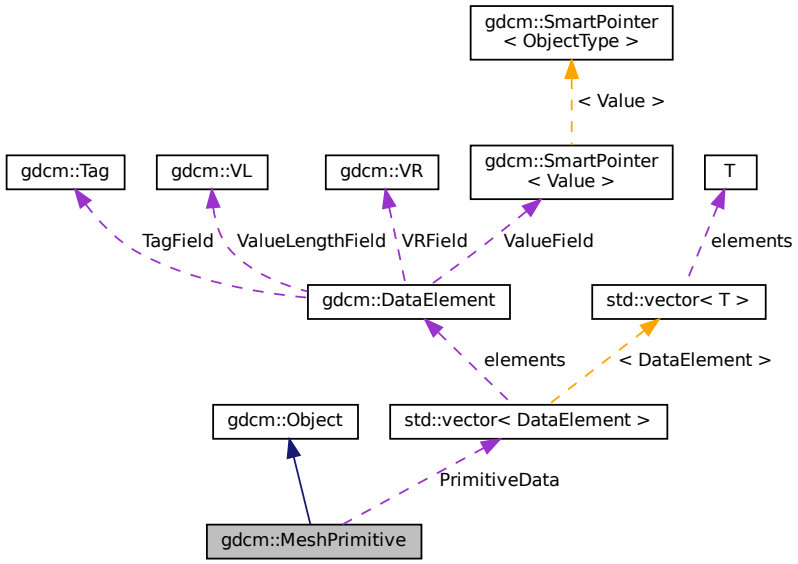
This class defines surface mesh primitives.

```
#include <gdcmMeshPrimitive.h>
```

Inheritance diagram for gdcm::MeshPrimitive:



Collaboration diagram for gdcm::MeshPrimitive:



Public Types

- enum MPType {
  VERTEX = 0 ,
  EDGE ,
  TRIANGLE ,
  TRIANGLE_STRIP ,
  TRIANGLE_FAN ,
  LINE ,
  FACET ,
  MPType_END }

  This enumeration defines primitive types.
- typedef std::vector< DataElement > PrimitivesData

Public Member Functions

- MeshPrimitive ()
- ~MeshPrimitive () override
- void AddPrimitiveData (DataElement const &de)
- unsigned int GetNumberOfPrimitivesData () const
- DataElement & GetPrimitiveData ()
- const DataElement & GetPrimitiveData () const
- DataElement & GetPrimitiveData (const unsigned int idx)
- const DataElement & GetPrimitiveData (const unsigned int idx) const
- PrimitivesData & GetPrimitivesData ()

- const PrimitivesData & GetPrimitivesData () const
- MPType GetPrimitiveType () const
- void SetPrimitiveData (const unsigned int idx, DataElement const &de)
- void SetPrimitiveData (DataElement const &de)
- void SetPrimitivesData (PrimitivesData const &DEs)
- void SetPrimitiveType (const MPType type)

Static Public Member Functions

- static MPType GetMPType (const char *type)
- static const char * GetMPTypeString (const MPType type)

Protected Attributes

- PrimitivesData PrimitiveData
- MPType PrimitiveType

Additional Inherited Members

10.190.1 Detailed Description

This class defines surface mesh primitives.

It is designed from surface mesh primitives macro.

See also

PS 3.3 C.27.4

10.190.2 Member Typedef Documentation

10.190.2.1 PrimitivesData

```
typedef std::vector< DataElement > gdcm::MeshPrimitive::PrimitivesData
```

10.190.3 Member Enumeration Documentation

10.190.3.1 MPType

```
enum gdcm::MeshPrimitive::MPType
```

This enumeration defines primitive types.

See also

PS 3.3 C.27.4.1

Enumerator

VERTEX	
EDGE	
TRIANGLE	
TRIANGLE_STRIP	
TRIANGLE_FAN	
LINE	
FACET	
MPType_END	

10.190.4 Constructor & Destructor Documentation

10.190.4.1 MeshPrimitive()

```
gdcm::MeshPrimitive::MeshPrimitive ( )
```

10.190.4.2 ~MeshPrimitive()

```
gdcm::MeshPrimitive::~MeshPrimitive ( )  [override]
```

10.190.5 Member Function Documentation

10.190.5.1 AddPrimitiveData()

```
void gdcm::MeshPrimitive::AddPrimitiveData (
            DataElement const & de )
```

10.190.5.2 GetMPType()

```
static MPType gdcm::MeshPrimitive::GetMPType (
            const char * type )  [static]
```

10.190.5.3 GetMPTypeString()

```
static const char * gdcm::MeshPrimitive::GetMPTypeString (
            const MPType type )  [static]
```

10.190.5.4 GetNumberOfPrimitivesData()

```
unsigned int gdcm::MeshPrimitive::GetNumberOfPrimitivesData ( ) const
```

10.190.5.5 GetPrimitiveData() [1/4]

```
DataElement & gdcm::MeshPrimitive::GetPrimitiveData ( )
```

10.190.5.6 GetPrimitiveData() [2/4]

```
const DataElement & gdcm::MeshPrimitive::GetPrimitiveData ( ) const
```

10.190.5.7 GetPrimitiveData() [3/4]

```
DataElement & gdcm::MeshPrimitive::GetPrimitiveData (
            const unsigned int idx )
```

10.190.5.8 GetPrimitiveData() [4/4]

```
const DataElement & gdcm::MeshPrimitive::GetPrimitiveData (
            const unsigned int idx ) const
```

10.190.5.9 GetPrimitivesData() [1/2]

```
PrimitivesData & gdcm::MeshPrimitive::GetPrimitivesData ( )
```

10.190.5.10 GetPrimitivesData() [2/2]

```
const PrimitivesData & gdcm::MeshPrimitive::GetPrimitivesData ( ) const
```

10.190.5.11 GetPrimitiveType()

```
MPType gdcm::MeshPrimitive::GetPrimitiveType ( ) const
```

10.190.5.12 SetPrimitiveData() [1/2]

```
void gdcm::MeshPrimitive::SetPrimitiveData (
            const unsigned int idx,
            DataElement const & de )
```

10.190.5.13 SetPrimitiveData() [2/2]

```
void gdcm::MeshPrimitive::SetPrimitiveData (
            DataElement const & de )
```

10.190.5.14 SetPrimitivesData()

```
void gdcm::MeshPrimitive::SetPrimitivesData (
            PrimitivesData const & DEs )
```

10.190.5.15 SetPrimitiveType()

```
void gdcm::MeshPrimitive::SetPrimitiveType (
            const MPType type )
```

10.190.6 Member Data Documentation

10.190.6.1 PrimitiveData

```
PrimitivesData gdcm::MeshPrimitive::PrimitiveData  [protected]
```

10.190.6.2 PrimitiveType

```
MPType gdcm::MeshPrimitive::PrimitiveType  [protected]
```

The documentation for this class was generated from the following file:

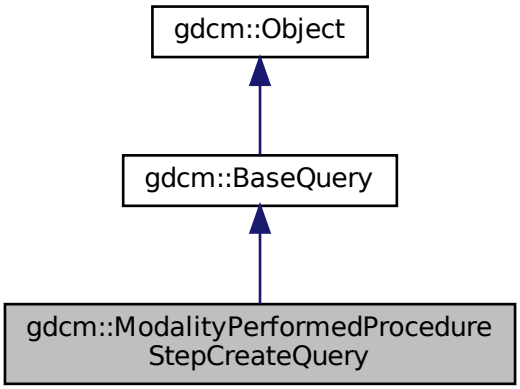
- gdcmMeshPrimitive.h

10.191 gdcm::ModalityPerformedProcedureStepCreateQuery Class Reference

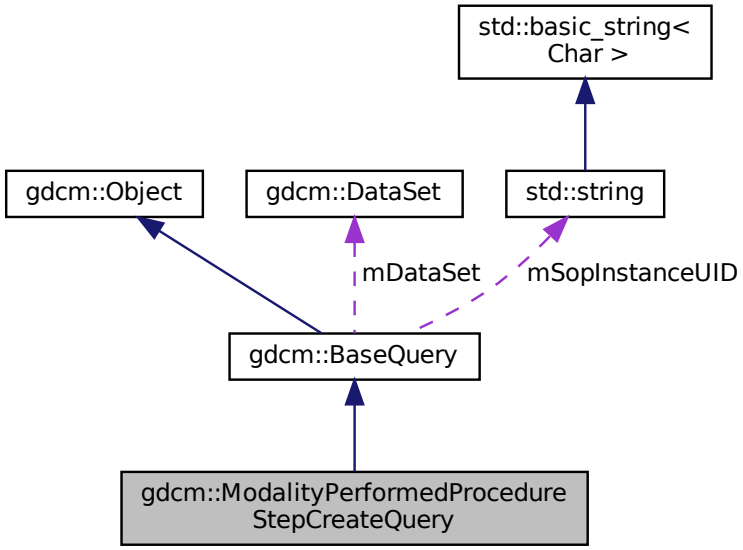
ModalityPerformedProcedureStepCreateQuery.

```
#include <gdcmModalityPerformedProcedureStepCreateQuery.h>
```

Inheritance diagram for gdcm::ModalityPerformedProcedureStepCreateQuery:



Collaboration diagram for gdcm::ModalityPerformedProcedureStepCreateQuery:



Public Member Functions

- ModalityPerformedProcedureStepCreateQuery (const std::string &iSopInstanceUID)
- UIDs::TSName GetAbstractSyntaxUID () const override
- gdcm::DataSet GetRequiredDataSet () const
- bool ValidateQuery (bool inStrict=true) const override

Friends

- class QueryFactory

Additional Inherited Members

10.191.1 Detailed Description

ModalityPerformedProcedureStepCreateQuery.

contains: the class which will produce a dataset for n-create for Modality Performed Procedure Step sop class

10.191.2 Constructor & Destructor Documentation

10.191.2.1 ModalityPerformedProcedureStepCreateQuery()

```
gdcm::ModalityPerformedProcedureStepCreateQuery::ModalityPerformedProcedureStepCreateQuery (
            const std::string & iSopInstanceUID )
```

10.191.3 Member Function Documentation

10.191.3.1 GetAbstractSyntaxUID()

```
UIDs::TSName gdcm::ModalityPerformedProcedureStepCreateQuery::GetAbstractSyntaxUID ( ) const  [override],
[virtual]
```

Implements gdcm::BaseQuery.

10.191.3.2 GetRequiredDataSet()

```
gdcm::DataSet gdcm::ModalityPerformedProcedureStepCreateQuery::GetRequiredDataSet ( ) const
```

10.191.3.3 ValidateQuery()

```
bool gdcm::ModalityPerformedProcedureStepCreateQuery::ValidateQuery (
            bool inStrict = true ) const  [override], [virtual]
```

Implements gdcm::BaseQuery.

10.191.4 Friends And Related Function Documentation

10.191.4.1 QueryFactory

```
friend class QueryFactory  [friend]
```

The documentation for this class was generated from the following file:

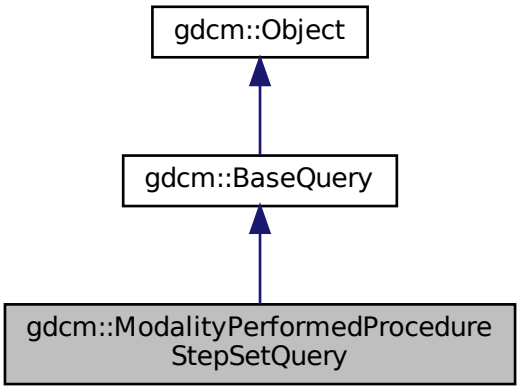
- gdcmModalityPerformedProcedureStepCreateQuery.h

10.192 gdcm::ModalityPerformedProcedureStepSetQuery Class Reference

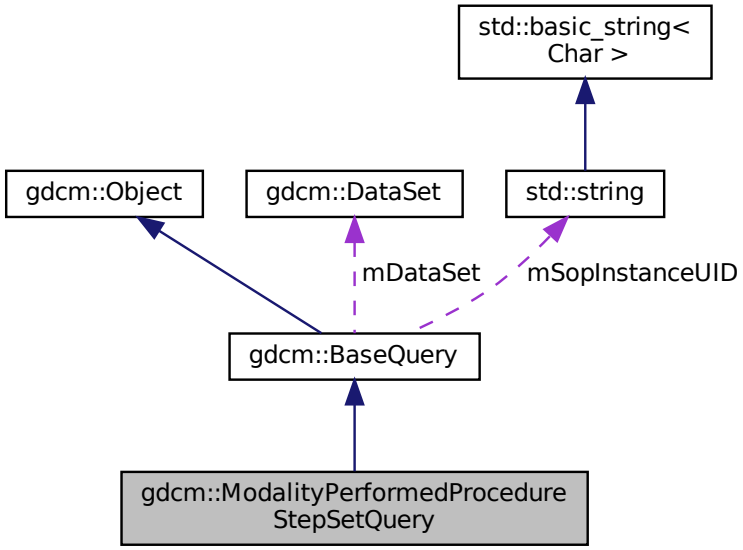
ModalityPerformedProcedureStepSetQuery.

```
#include <gdcmModalityPerformedProcedureStepSetQuery.h>
```

Inheritance diagram for gdcm::ModalityPerformedProcedureStepSetQuery:



Collaboration diagram for gdcm::ModalityPerformedProcedureStepSetQuery:



Public Member Functions

- ModalityPerformedProcedureStepSetQuery (const std::string &iSopInstanceUID)
- UIDs::TSName GetAbstractSyntaxUID () const override
- gdcm::DataSet GetRequiredDataSet () const
- bool ValidateQuery (bool inStrict=true) const override

Friends

- class QueryFactory

Additional Inherited Members

10.192.1 Detailed Description

ModalityPerformedProcedureStepSetQuery.

contains: the class which will produce a dataset for n-set for Modality Performed Procedure Step sop class

10.192.2 Constructor & Destructor Documentation

10.192.2.1 ModalityPerformedProcedureStepSetQuery()

```
gdcm::ModalityPerformedProcedureStepSetQuery::ModalityPerformedProcedureStepSetQuery (
            const std::string & iSopInstanceUID )
```

10.192.3 Member Function Documentation

10.192.3.1 GetAbstractSyntaxUID()

```
UIDs::TSName gdcm::ModalityPerformedProcedureStepSetQuery::GetAbstractSyntaxUID ( ) const  [override],
[virtual]
```

Implements gdcm::BaseQuery.

10.192.3.2 GetRequiredDataSet()

```
gdcm::DataSet gdcm::ModalityPerformedProcedureStepSetQuery::GetRequiredDataSet ( ) const
```

10.192.3.3 ValidateQuery()

```
bool gdcm::ModalityPerformedProcedureStepSetQuery::ValidateQuery (
            bool inStrict = true ) const  [override], [virtual]
```

Implements gdcm::BaseQuery.

10.192.4 Friends And Related Function Documentation

10.192.4.1 QueryFactory

```
friend class QueryFactory  [friend]
```

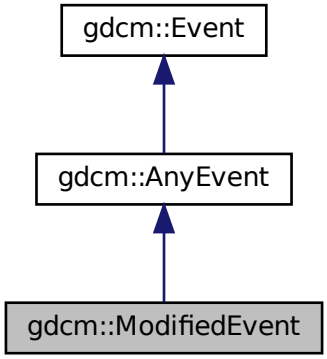
The documentation for this class was generated from the following file:

- gdcmModalityPerformedProcedureStepSetQuery.h

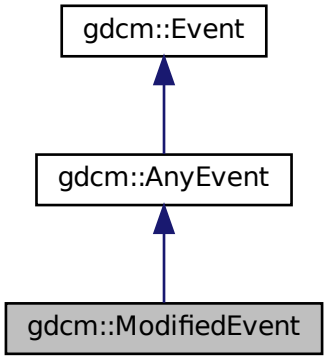
10.193 gdcm::ModifiedEvent Class Reference

```
#include <gdcmEvent.h>
```

Inheritance diagram for gdcm::ModifiedEvent:



Collaboration diagram for gdcm::ModifiedEvent:



Additional Inherited Members

The documentation for this class was generated from the following file:

- gdcmEvent.h

10.194 gdcm::Module Class Reference

Class for representing a Module.

`#include <gdcmModule.h>`

Public Types

- typedef std::vector< std::string > ArrayIncludeMacrosType
- typedef std::map< Tag, ModuleEntry > MapModuleEntry

Public Member Functions

- Module ()=default
- void AddMacro (const char *include)
- void AddModuleEntry (const Tag &tag, const ModuleEntry &module)

  Will add a ModuleEntry directly at root-level. See Macro for nested-included level.
- void Clear ()
- bool FindModuleEntryInMacros (Macros const &macros, const Tag &tag) const
- const ModuleEntry & GetModuleEntryInMacros (Macros const &macros, const Tag &tag) const
- const char * GetName () const
- void SetName (const char *name)
- bool Verify (const DataSet &ds, Usage const &usage) const

Friends

- std::ostream & operator<< (std::ostream &_os, const Module &_val)

10.194.1 Detailed Description

Class for representing a Module.

Note

Module: A set of Attributes within an Information Entity or Normalized IOD which are logically related to each other.

See also

Macro

Examples

TraverseModules.cxx.

10.194.2 Member Typedef Documentation

10.194.2.1 ArrayIncludeMacrosType

```
typedef std::vector<std::string> gdcm::Module::ArrayIncludeMacrosType
```

10.194.2.2 MapModuleEntry

```
typedef std::map<Tag, ModuleEntry> gdcm::Module::MapModuleEntry
```

10.194.3 Constructor & Destructor Documentation

10.194.3.1 Module()

```
gdcm::Module::Module ( )  [default]
```

10.194.4 Member Function Documentation

10.194.4.1 AddMacro()

```
void gdcm::Module::AddMacro (
            const char * include )  [inline]
```

10.194.4.2 AddModuleEntry()

```
void gdcm::Module::AddModuleEntry (
            const Tag & tag,
            const ModuleEntry & module )  [inline]
```

Will add a ModuleEntry directly at root-level. See Macro for nested-included level.

10.194.4.3 Clear()

```
void gdcm::Module::Clear ( )  [inline]
```

10.194.4.4 FindModuleEntryInMacros()

```
bool gdcm::Module::FindModuleEntryInMacros (
            Macros const & macros,
            const Tag & tag ) const
```

Find or Get a ModuleEntry. ModuleEntry are either search are root-level or within nested-macro included in module.

Examples

TraverseModules.cxx.

10.194.4.5 GetModuleEntryInMacros()

```
const ModuleEntry & gdcm::Module::GetModuleEntryInMacros (
            Macros const & macros,
            const Tag & tag ) const
```

Examples

TraverseModules.cxx.

10.194.4.6 GetName()

```
const char * gdcm::Module::GetName ( ) const  [inline]
```

10.194.4.7 SetName()

```
void gdcm::Module::SetName (
            const char * name )  [inline]
```

10.194.4.8 Verify()

```
bool gdcm::Module::Verify (
             const DataSet & ds,
             Usage const & usage ) const
```

10.194.5 Friends And Related Function Documentation

10.194.5.1 operator<<

```
std::ostream & operator<< (
             std::ostream & _os,
             const Module & _val )  [friend]
```

The documentation for this class was generated from the following file:

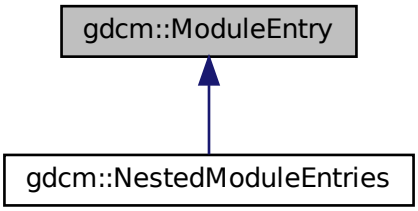
- gdcmModule.h

10.195 gdcm::ModuleEntry Class Reference

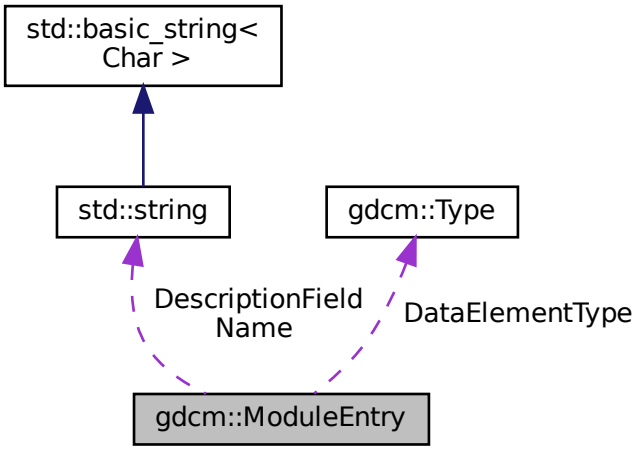
Class for representing a ModuleEntry.

`#include <gdcmModuleEntry.h>`

Inheritance diagram for gdcm::ModuleEntry:



Collaboration diagram for gdcm::ModuleEntry:



Public Types

- typedef std::string Description

Public Member Functions

- ModuleEntry (const char ∗name="", const char ∗type="3", const char ∗description="")
- virtual ∼ModuleEntry ()=default
- const Description & GetDescription () const
- const char ∗ GetName () const
- const Type & GetType () const
- void SetDescription (const char ∗d)
- void SetName (const char ∗name)
- void SetType (const Type &type)

Protected Attributes

- Type DataElementType
- Description DescriptionField
- std::string Name

Friends

- std::ostream & operator<< (std::ostream &_os, const ModuleEntry &_val)

10.195.1 Detailed Description

Class for representing a ModuleEntry.

Note

bla

See also

DictEntry

Examples

TraverseModules.cxx.

10.195.2 Member Typedef Documentation

10.195.2.1 Description

```
typedef std::string gdcm::ModuleEntry::Description
```

10.195.3 Constructor & Destructor Documentation

10.195.3.1 ModuleEntry()

```
gdcm::ModuleEntry::ModuleEntry (
             const char * name = "",
             const char * type = "3",
             const char * description = "" )  [inline]
```

References gdcm::Type::GetTypeType().

10.195.3.2 ~ModuleEntry()

```
virtual gdcm::ModuleEntry::~ModuleEntry ( )  [virtual], [default]
```

10.195.4 Member Function Documentation

10.195.4.1 GetDescription()

```
const Description & gdcm::ModuleEntry::GetDescription ( ) const  [inline]
```

10.195.4.2 GetName()

```
const char * gdcm::ModuleEntry::GetName ( ) const  [inline]
```

10.195.4.3 GetType()

```
const Type & gdcm::ModuleEntry::GetType ( ) const  [inline]
```

Examples

TraverseModules.cxx.

10.195.4.4 SetDescription()

```
void gdcm::ModuleEntry::SetDescription (
            const char * d )  [inline]
```

10.195.4.5 SetName()

```
void gdcm::ModuleEntry::SetName (
            const char * name )  [inline]
```

10.195.4.6 SetType()

```
void gdcm::ModuleEntry::SetType (
            const Type & type )  [inline]
```

10.195.5 Friends And Related Function Documentation

10.195.5.1 operator<<

```
std::ostream & operator<< (
            std::ostream & _os,
            const ModuleEntry & _val )  [friend]
```

10.195.6 Member Data Documentation

10.195.6.1 DataElementType

```
Type gdcm::ModuleEntry::DataElementType  [protected]
```

10.195.6.2 DescriptionField

```
Description gdcm::ModuleEntry::DescriptionField  [protected]
```

10.195.6.3 Name

```
std::string gdcm::ModuleEntry::Name  [protected]
```

The documentation for this class was generated from the following file:

- gdcmModuleEntry.h

10.196 gdcm::Modules Class Reference

Class for representing a Modules.

```
#include <gdcmModules.h>
```

Public Types

- typedef std::map< std::string, Module > ModuleMapType

Public Member Functions

- Modules ()=default
- void AddModule (const char *ref, const Module &module)
- void Clear ()
- const Module & GetModule (const char *name) const
- bool IsEmpty () const

Friends

- std::ostream & operator<< (std::ostream &_os, const Modules &_val)

10.196.1 Detailed Description

Class for representing a Modules.

Note

bla

See also

Module

Examples

TraverseModules.cxx.

10.196.2 Member Typedef Documentation

10.196.2.1 ModuleMapType

```
typedef std::map<std::string, Module> gdcm::Modules::ModuleMapType
```

10.196.3 Constructor & Destructor Documentation

10.196.3.1 Modules()

```
gdcm::Modules::Modules ( )  [default]
```

10.196.4 Member Function Documentation

10.196.4.1 AddModule()

```
void gdcm::Modules::AddModule (
            const char * ref,
            const Module & module )  [inline]
```

10.196.4.2 Clear()

```
void gdcm::Modules::Clear ( )  [inline]
```

10.196.4.3 GetModule()

```
const Module & gdcm::Modules::GetModule (
            const char * name ) const  [inline]
```

Examples

TraverseModules.cxx.

10.196.4.4 IsEmpty()

```
bool gdcm::Modules::IsEmpty ( ) const  [inline]
```

10.196.5 Friends And Related Function Documentation

10.196.5.1 operator<<

```
std::ostream & operator<< (
             std::ostream & _os,
             const Modules & _val )  [friend]
```

The documentation for this class was generated from the following file:

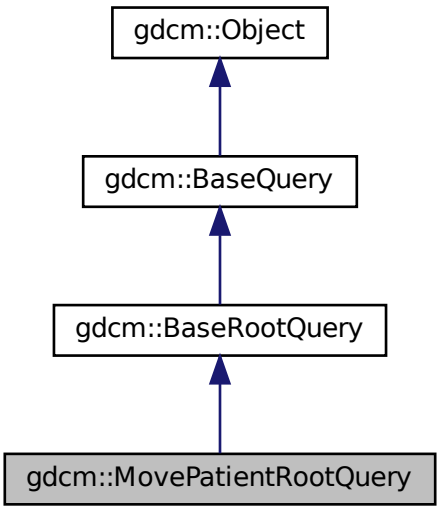
- gdcmModules.h

10.197 gdcm::MovePatientRootQuery Class Reference

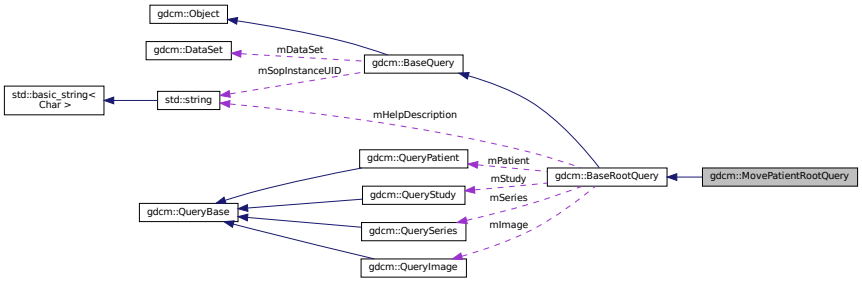
MovePatientRootQuery.

`#include <gdcmMovePatientRootQuery.h>`

Inheritance diagram for gdcm::MovePatientRootQuery:



Collaboration diagram for gdcm::MovePatientRootQuery:



Public Member Functions

- MovePatientRootQuery ()
- UIDs::TSName GetAbstractSyntaxUID () const override
- std::vector< Tag > GetTagListByLevel (const EQueryLevel &inQueryLevel) override
- void InitializeDataSet (const EQueryLevel &inQueryLevel) override
- bool ValidateQuery (bool inStrict=true) const override

Friends

- class QueryFactory

Additional Inherited Members

10.197.1 Detailed Description

MovePatientRootQuery.

contains: the class which will produce a dataset for c-move with patient root

10.197.2 Constructor & Destructor Documentation

10.197.2.1 MovePatientRootQuery()

```
gdcm::MovePatientRootQuery::MovePatientRootQuery ( )
```

10.197.3 Member Function Documentation

10.197.3.1 GetAbstractSyntaxUID()

```
UIDs::TSName gdcm::MovePatientRootQuery::GetAbstractSyntaxUID ( ) const  [override], [virtual]
```

Implements gdcm::BaseQuery.

10.197.3.2 GetTagListByLevel()

```
std::vector< Tag > gdcm::MovePatientRootQuery::GetTagListByLevel (
            const EQueryLevel & inQueryLevel )  [override], [virtual]
```

this function will return all tags at a given query level, so that they maybe selected for searching. The boolean forFind is true if the query is a find query, or false for a move query.

Implements gdcm::BaseRootQuery.

10.197.3.3 InitializeDataSet()

```
void gdcm::MovePatientRootQuery::InitializeDataSet (
            const EQueryLevel & inQueryLevel )  [override], [virtual]
```

this function sets tag 8,52 to the appropriate value based on query level also fills in the right unique tags, as per the standard's requirements should allow for connection with dcmtk

Implements gdcm::BaseRootQuery.

10.197.3.4 ValidateQuery()

```
bool gdcm::MovePatientRootQuery::ValidateQuery (
            bool inStrict = true ) const  [override], [virtual]
```

have to be able to ensure that 0x8,0x52 is set (which will be true if InitializeDataSet is called...) that the level is appropriate (ie, not setting PATIENT for a study query that the tags in the query match the right level (either required, unique, optional) by default, this function checks to see if the query is for finding, which is more permissive than for moving. For moving, only the unique tags are allowed. 10 Jan 2011: adding in the 'strict' mode. according to the standard (at least, how I've read it), only tags for a particular level should be allowed in a particular query (ie, just series level tags in a series level query). However, it seems that dcm4chee doesn't share that interpretation. So, if 'inStrict' is false, then tags from the current level and all higher levels are now considered valid. So, if you're doing a non-strict series-level query, tags from the patient and study level can be passed along as well.

Implements gdcm::BaseRootQuery.

10.197.4 Friends And Related Function Documentation

10.197.4.1 QueryFactory

```
friend class QueryFactory  [friend]
```

The documentation for this class was generated from the following file:

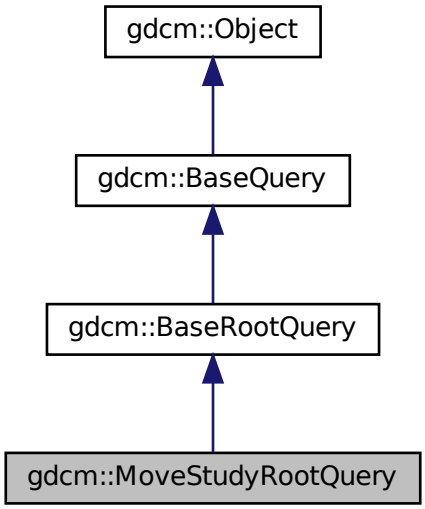
- gdcmMovePatientRootQuery.h

10.198 gdcm::MoveStudyRootQuery Class Reference

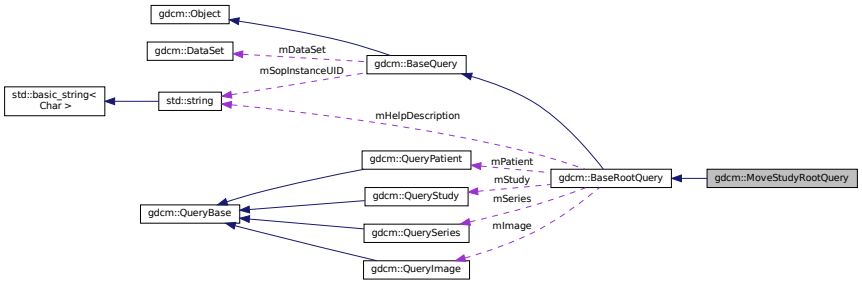
MoveStudyRootQuery.

```
#include <gdcmMoveStudyRootQuery.h>
```

Inheritance diagram for gdcm::MoveStudyRootQuery:



Collaboration diagram for gdcm::MoveStudyRootQuery:



Public Member Functions

- MoveStudyRootQuery ()
- UIDs::TSName GetAbstractSyntaxUID () const override
- std::vector< Tag > GetTagListByLevel (const EQueryLevel &inQueryLevel) override
- void InitializeDataSet (const EQueryLevel &inQueryLevel) override
- bool ValidateQuery (bool inStrict=true) const override

Friends

- class QueryFactory

Additional Inherited Members

10.198.1 Detailed Description

MoveStudyRootQuery.

contains: the class which will produce a dataset for C-MOVE with study root

10.198.2 Constructor & Destructor Documentation

10.198.2.1 MoveStudyRootQuery()

```
gdcm::MoveStudyRootQuery::MoveStudyRootQuery ( )
```

10.198.3 Member Function Documentation

10.198.3.1 GetAbstractSyntaxUID()

```
UIDs::TSName gdcm::MoveStudyRootQuery::GetAbstractSyntaxUID ( ) const  [override], [virtual]
```

Implements gdcm::BaseQuery.

10.198.3.2 GetTagListByLevel()

```
std::vector< Tag > gdcm::MoveStudyRootQuery::GetTagListByLevel (
            const EQueryLevel & inQueryLevel )  [override], [virtual]
```

this function will return all tags at a given query level, so that they maybe selected for searching. The boolean forFind is true if the query is a find query, or false for a move query.

Implements gdcm::BaseRootQuery.

10.198.3.3 InitializeDataSet()

```
void gdcm::MoveStudyRootQuery::InitializeDataSet (
            const EQueryLevel & inQueryLevel )  [override], [virtual]
```

this function sets tag 8,52 to the appropriate value based on query level also fills in the right unique tags, as per the standard's requirements should allow for connection with dcmtk

Implements gdcm::BaseRootQuery.

10.198.3.4 ValidateQuery()

```
bool gdcm::MoveStudyRootQuery::ValidateQuery (
            bool inStrict = true ) const  [override], [virtual]
```

have to be able to ensure that 0x8,0x52 is set (which will be true if InitializeDataSet is called...) that the level is appropriate (ie, not setting PATIENT for a study query that the tags in the query match the right level (either required, unique, optional) by default, this function checks to see if the query is for finding, which is more permissive than for moving. For moving, only the unique tags are allowed. 10 Jan 2011: adding in the 'strict' mode. according to the standard (at least, how I've read it), only tags for a particular level should be allowed in a particular query (ie, just series level tags in a series level query). However, it seems that dcm4chee doesn't share that interpretation. So, if 'inStrict' is false, then tags from the current level and all higher levels are now considered valid. So, if you're doing a non-strict series-level query, tags from the patient and study level can be passed along as well.

Implements gdcm::BaseRootQuery.

10.198.4 Friends And Related Function Documentation

10.198.4.1 QueryFactory

```
friend class QueryFactory  [friend]
```

The documentation for this class was generated from the following file:

- gdcmMoveStudyRootQuery.h

10.199 gdcm::MrProtocol Class Reference

Class for MrProtocol.

```
#include <gdcmMrProtocol.h>
```

Classes

- struct Slice
- struct SliceArray
- struct Vector3

Public Member Functions

- MrProtocol ()
- ~MrProtocol ()
- bool FindMrProtocolByName (const char *name) const
- const char * GetMrProtocolByName (const char *name) const
- bool GetSliceArray (MrProtocol::SliceArray &sa) const
- int GetVersion () const
- bool Load (const ByteValue *bv, const char *str, int version)
- void Print (std::ostream &os) const

Friends

- std::ostream & operator<< (std::ostream &_os, const MrProtocol &d)

10.199.1 Detailed Description

Class for MrProtocol.

Examples

MrProtocol.cxx.

10.199.2 Constructor & Destructor Documentation

10.199.2.1 MrProtocol()

```
gdcm::MrProtocol::MrProtocol ( )
```

10.199.2.2 ~MrProtocol()

```
gdcm::MrProtocol::~MrProtocol ( )
```

10.199.3 Member Function Documentation

10.199.3.1 FindMrProtocolByName()

```
bool gdcm::MrProtocol::FindMrProtocolByName (
            const char * name ) const
```

10.199.3.2 GetMrProtocolByName()

```
const char * gdcm::MrProtocol::GetMrProtocolByName (
            const char * name ) const
```

10.199.3.3 GetSliceArray()

```
bool gdcm::MrProtocol::GetSliceArray (
            MrProtocol::SliceArray & sa ) const
```

10.199.3.4 GetVersion()

```
int gdcm::MrProtocol::GetVersion ( ) const
```

10.199.3.5 Load()

```
bool gdcm::MrProtocol::Load (
            const ByteValue * bv,
            const char * str,
            int version )
```

10.199.3.6 Print()

```
void gdcm::MrProtocol::Print (
            std::ostream & os ) const
```

10.199.4 Friends And Related Function Documentation

10.199.4.1 operator<<

```
std::ostream & operator<< (
            std::ostream & _os,
            const MrProtocol & d )  [friend]
```

The documentation for this class was generated from the following file:

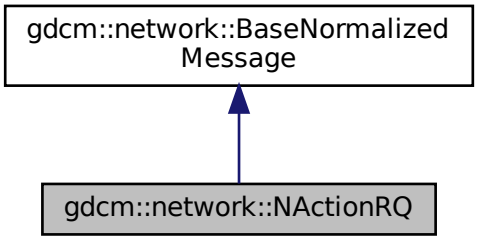
- gdcmMrProtocol.h

10.200 gdcm::network::NActionRQ Class Reference

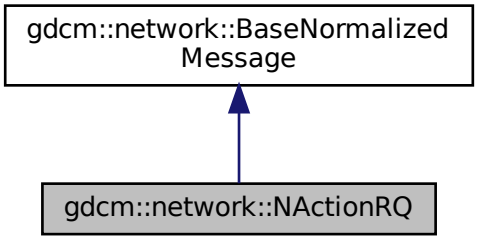
NActionRQ.

`#include <gdcmNActionMessages.h>`

Inheritance diagram for gdcm::network::NActionRQ:



Collaboration diagram for gdcm::network::NActionRQ:



Public Member Functions

- std::vector< PresentationDataValue > ConstructPDV (const ULConnection &inConnection, const BaseQuery *inQuery) override

10.200.1 Detailed Description

NActionRQ.

this file defines the messages for the NAction action

10.200.2 Member Function Documentation

10.200.2.1 ConstructPDV()

```
std::vector< PresentationDataValue > gdcm::network::NActionRQ::ConstructPDV (
            const ULConnection & inConnection,
            const BaseQuery * inQuery )  [override], [virtual]
```

Implements gdcm::network::BaseNormalizedMessage.

The documentation for this class was generated from the following file:

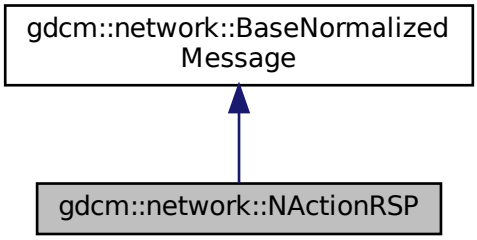
- gdcmNActionMessages.h

10.201 gdcm::network::NActionRSP Class Reference

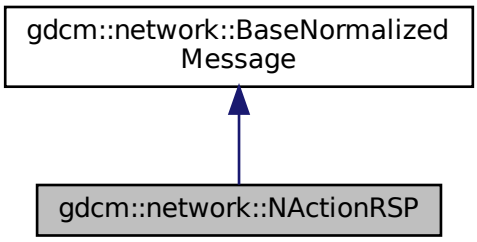
NActionRSP this file defines the messages for the NAction action.

```
#include <gdcmNActionMessages.h>
```

Inheritance diagram for gdcm::network::NActionRSP:



Collaboration diagram for gdcm::network::NActionRSP:



Public Member Functions

- std::vector< PresentationDataValue > ConstructPDVByDataSet (const DataSet *inDataSet)

10.201.1 Detailed Description

NActionRSP this file defines the messages for the NAction action.

10.201.2 Member Function Documentation

10.201.2.1 ConstructPDVByDataSet()

```
std::vector< PresentationDataValue > gdcm::network::NActionRSP::ConstructPDVByDataSet (
            const DataSet * inDataSet )
```

The documentation for this class was generated from the following file:

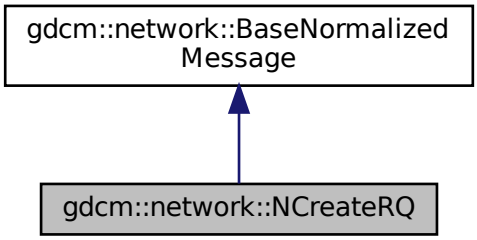
- gdcmNActionMessages.h

10.202 gdcm::network::NCreateRQ Class Reference

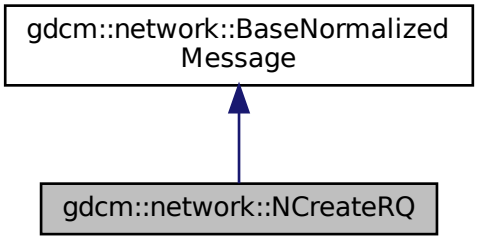
NCreateRQ.

`#include <gdcmNCreateMessages.h>`

Inheritance diagram for gdcm::network::NCreateRQ:



Collaboration diagram for gdcm::network::NCreateRQ:



Public Member Functions

- std::vector< PresentationDataValue > ConstructPDV (const ULConnection &inConnection, const BaseQuery *inQuery) override

10.202.1 Detailed Description

NCreateRQ.

this file defines the messages for the ncreate action

10.202.2 Member Function Documentation

10.202.2.1 ConstructPDV()

```
std::vector< PresentationDataValue > gdcm::network::NCreateRQ::ConstructPDV (
            const ULConnection & inConnection,
            const BaseQuery * inQuery )  [override], [virtual]
```

Implements gdcm::network::BaseNormalizedMessage.

The documentation for this class was generated from the following file:

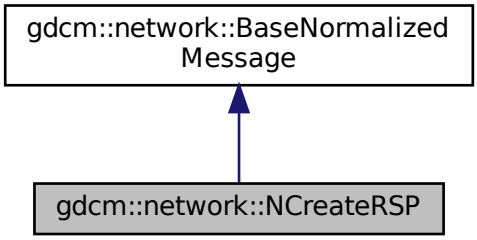
- gdcmNCreateMessages.h

10.203 gdcm::network::NCreateRSP Class Reference

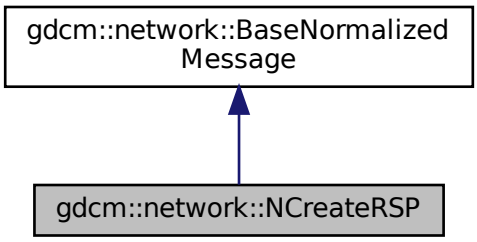
NCreateRSP this file defines the messages for the ncreate action.

```
#include <gdcmNCreateMessages.h>
```

Inheritance diagram for gdcm::network::NCreateRSP:



Collaboration diagram for gdcm::network::NCreateRSP:



Public Member Functions

- std::vector< PresentationDataValue > ConstructPDVByDataSet (const DataSet *inDataSet)

10.203.1 Detailed Description

NCreateRSP this file defines the messages for the ncreate action.

10.203.2 Member Function Documentation

10.203.2.1 ConstructPDVByDataSet()

```
std::vector< PresentationDataValue > gdcm::network::NCreateRSP::ConstructPDVByDataSet (
            const DataSet * inDataSet )
```

The documentation for this class was generated from the following file:

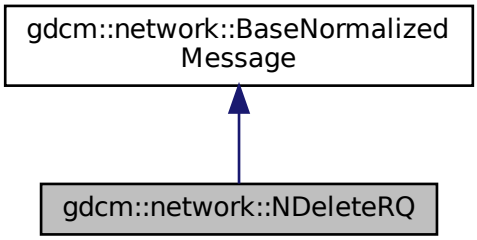
- gdcmNCreateMessages.h

10.204 gdcm::network::NDeleteRQ Class Reference

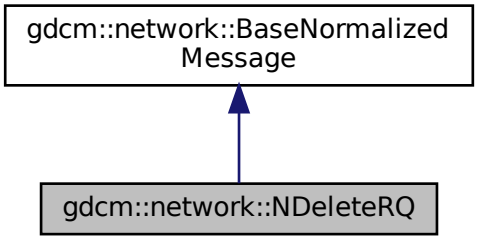
NDeleteRQ.

`#include <gdcmNDeleteMessages.h>`

Inheritance diagram for gdcm::network::NDeleteRQ:



Collaboration diagram for gdcm::network::NDeleteRQ:



Public Member Functions

- std::vector< PresentationDataValue > ConstructPDV (const ULConnection &inConnection, const BaseQuery *inQuery) override

10.204.1 Detailed Description

NDeleteRQ.

this file defines the messages for the ndelete action

10.204.2 Member Function Documentation

10.204.2.1 ConstructPDV()

```
std::vector< PresentationDataValue > gdcm::network::NDeleteRQ::ConstructPDV (
            const ULConnection & inConnection,
            const BaseQuery * inQuery )  [override], [virtual]
```

Implements gdcm::network::BaseNormalizedMessage.

The documentation for this class was generated from the following file:

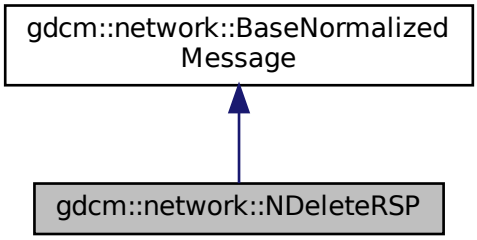
- gdcmNDeleteMessages.h

10.205 gdcm::network::NDeleteRSP Class Reference

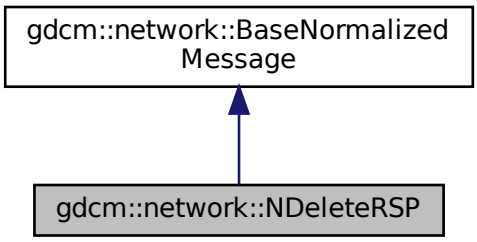
NDeleteRSP this file defines the messages for the ndelete action.

```
#include <gdcmNDeleteMessages.h>
```

Inheritance diagram for gdcm::network::NDeleteRSP:



Collaboration diagram for gdcm::network::NDeleteRSP:



Public Member Functions

- std::vector< PresentationDataValue > ConstructPDVByDataSet (const DataSet *inDataSet)

10.205.1 Detailed Description

NDeleteRSP this file defines the messages for the ndelete action.

10.205.2 Member Function Documentation

10.205.2.1 ConstructPDVByDataSet()

```
std::vector< PresentationDataValue > gdcm::network::NDeleteRSP::ConstructPDVByDataSet (
            const DataSet * inDataSet )
```

The documentation for this class was generated from the following file:

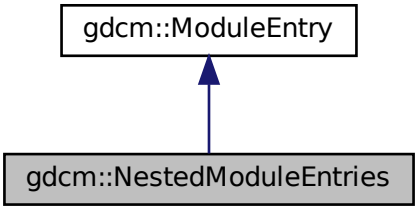
- gdcmNDeleteMessages.h

10.206 gdcm::NestedModuleEntries Class Reference

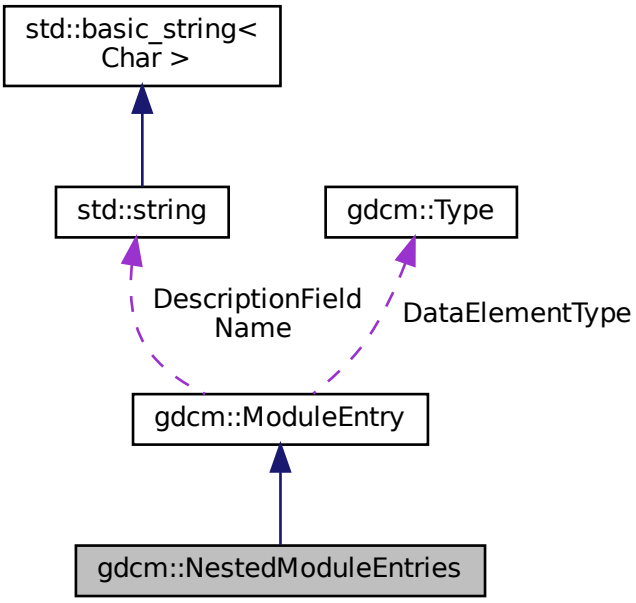
Class for representing a NestedModuleEntries.

`#include <gdcmNestedModuleEntries.h>`

Inheritance diagram for gdcm::NestedModuleEntries:



Collaboration diagram for gdcm::NestedModuleEntries:



Public Types

- typedef std::vector< ModuleEntry >::size_type SizeType

Public Member Functions

- NestedModuleEntries (const char *name="", const char *type="3", const char *description="")
- void AddModuleEntry (const ModuleEntry &me)
- ModuleEntry & GetModuleEntry (SizeType idx)
- const ModuleEntry & GetModuleEntry (SizeType idx) const
- SizeType GetNumberOfModuleEntries ()

Friends

- std::ostream & operator<< (std::ostream &_os, const NestedModuleEntries &_val)

Additional Inherited Members

10.206.1 Detailed Description

Class for representing a NestedModuleEntries.

Note

bla

See also

ModuleEntry

10.206.2 Member Typedef Documentation

10.206.2.1 SizeType

```
typedef std::vector<ModuleEntry>::size_type gdcm::NestedModuleEntries::SizeType
```

10.206.3 Constructor & Destructor Documentation

10.206.3.1 NestedModuleEntries()

```
gdcm::NestedModuleEntries::NestedModuleEntries (
            const char * name = "",
            const char * type = "3",
            const char * description = "" )  [inline]
```

10.206.4 Member Function Documentation

10.206.4.1 AddModuleEntry()

```
void gdcm::NestedModuleEntries::AddModuleEntry (
            const ModuleEntry & me )  [inline]
```

10.206.4.2 GetModuleEntry() [1/2]

```
ModuleEntry & gdcm::NestedModuleEntries::GetModuleEntry (
            SizeType idx )  [inline]
```

10.206.4.3 GetModuleEntry() [2/2]

```
const ModuleEntry & gdcm::NestedModuleEntries::GetModuleEntry (
            SizeType idx ) const  [inline]
```

10.206.4.4 GetNumberOfModuleEntries()

```
SizeType gdcm::NestedModuleEntries::GetNumberOfModuleEntries ( )  [inline]
```

10.206.5 Friends And Related Function Documentation

10.206.5.1 operator<<

```
std::ostream & operator<< (
            std::ostream & _os,
            const NestedModuleEntries & _val )  [friend]
```

The documentation for this class was generated from the following file:

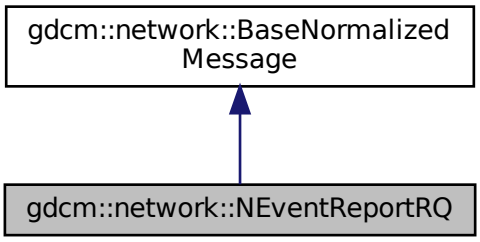
- gdcmNestedModuleEntries.h

10.207 gdcm::network::NEventReportRQ Class Reference

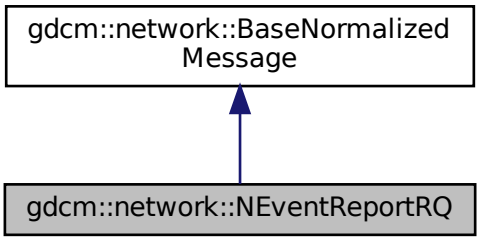
NEventReportRQ.

```
#include <gdcmNEventReportMessages.h>
```

Inheritance diagram for gdcm::network::NEventReportRQ:



Collaboration diagram for gdcm::network::NEventReportRQ:



Public Member Functions

- std::vector< PresentationDataValue > ConstructPDV (const ULConnection &inConnection, const BaseQuery *inQuery) override

10.207.1 Detailed Description

NEventReportRQ.

this file defines the messages for the neventreport action

10.207.2 Member Function Documentation

10.207.2.1 ConstructPDV()

```
std::vector< PresentationDataValue > gdcm::network::NEventReportRQ::ConstructPDV (
            const ULConnection & inConnection,
            const BaseQuery * inQuery )  [override], [virtual]
```

Implements gdcm::network::BaseNormalizedMessage.

The documentation for this class was generated from the following file:

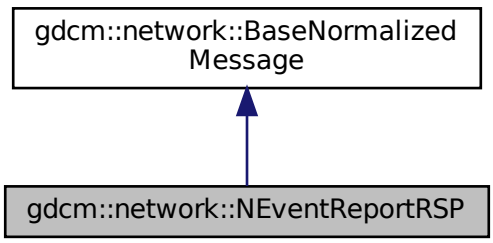
- gdcmNEventReportMessages.h

10.208 gdcm::network::NEventReportRSP Class Reference

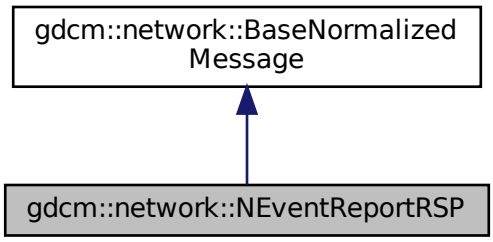
NEventReportRSP this file defines the messages for the neventreport action.

```
#include <gdcmNEventReportMessages.h>
```

Inheritance diagram for gdcm::network::NEventReportRSP:



Collaboration diagram for gdcm::network::NEventReportRSP:



Public Member Functions

- std::vector< PresentationDataValue > ConstructPDVByDataSet (const DataSet *inDataSet)

10.208.1 Detailed Description

NEventReportRSP this file defines the messages for the neventreport action.

10.208.2 Member Function Documentation

10.208.2.1 ConstructPDVByDataSet()

```
std::vector< PresentationDataValue > gdcm::network::NEventReportRSP::ConstructPDVByDataSet (
            const DataSet * inDataSet )
```

The documentation for this class was generated from the following file:

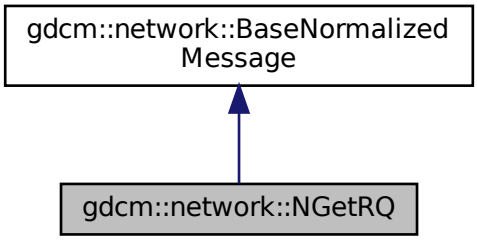
- gdcmNEventReportMessages.h

10.209 gdcm::network::NGetRQ Class Reference

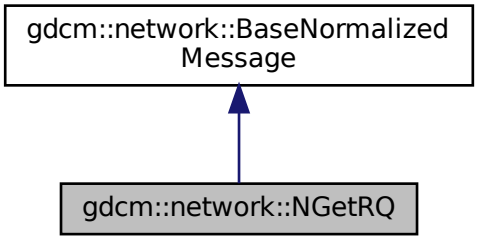
NGetRQ.

`#include <gdcmNGetMessages.h>`

Inheritance diagram for gdcm::network::NGetRQ:



Collaboration diagram for gdcm::network::NGetRQ:



Public Member Functions

- std::vector< PresentationDataValue > ConstructPDV (const ULConnection &inConnection, const BaseQuery *inQuery) override

10.209.1 Detailed Description

NGetRQ.

this file defines the messages for the nget action

10.209.2 Member Function Documentation

10.209.2.1 ConstructPDV()

```
std::vector< PresentationDataValue > gdcm::network::NGetRQ::ConstructPDV (
            const ULConnection & inConnection,
            const BaseQuery * inQuery )  [override], [virtual]
```

Implements gdcm::network::BaseNormalizedMessage.

The documentation for this class was generated from the following file:

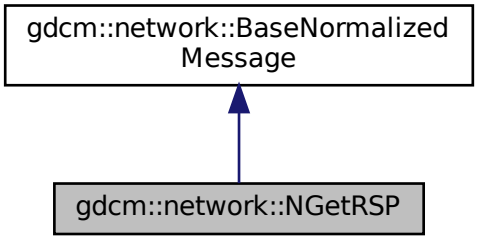
- gdcmNGetMessages.h

10.210 gdcm::network::NGetRSP Class Reference

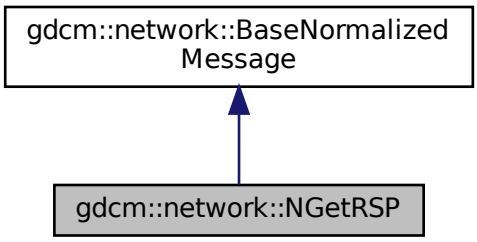
NGetRSP this file defines the messages for the nget action.

```
#include <gdcmNGetMessages.h>
```

Inheritance diagram for gdcm::network::NGetRSP:



Collaboration diagram for gdcm::network::NGetRSP:



Public Member Functions

- std::vector< PresentationDataValue > ConstructPDVByDataSet (const DataSet *inDataSet)

10.210.1 Detailed Description

NGetRSP this file defines the messages for the nget action.

10.210.2 Member Function Documentation

10.210.2.1 ConstructPDVByDataSet()

```
std::vector< PresentationDataValue > gdcm::network::NGetRSP::ConstructPDVByDataSet (
            const DataSet * inDataSet )
```

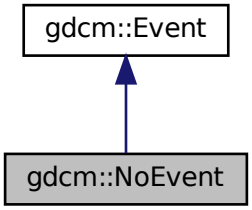
The documentation for this class was generated from the following file:

- gdcmNGetMessages.h

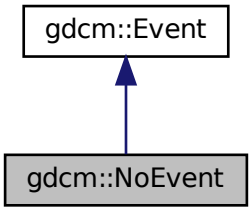
10.211 gdcm::NoEvent Class Reference

`#include <gdcmEvent.h>`

Inheritance diagram for gdcm::NoEvent:



Collaboration diagram for gdcm::NoEvent:



Additional Inherited Members

10.211.1 Detailed Description

Define some common GDCM events

The documentation for this class was generated from the following file:

- gdcmEvent.h

10.212 gdcm::network::NormalizedMessageFactory Class Reference

`#include <gdcmNormalizedMessageFactory.h>`

Static Public Member Functions

- static std::vector< PresentationDataValue > ConstructNAction (const ULConnection &inConnection, const BaseQuery *inQuery)
- static std::vector< PresentationDataValue > ConstructNCreate (const ULConnection &inConnection, const BaseQuery *inQuery)
- static std::vector< PresentationDataValue > ConstructNDelete (const ULConnection &inConnection, const BaseQuery *inQuery)
- static std::vector< PresentationDataValue > ConstructNEventReport (const ULConnection &inConnection, const BaseQuery *inQuery)
- static std::vector< PresentationDataValue > ConstructNGet (const ULConnection &inConnection, const BaseQuery *inQuery)
- static std::vector< PresentationDataValue > ConstructNSet (const ULConnection &inConnection, const BaseQuery *inQuery)

10.212.1 Member Function Documentation

10.212.1.1 ConstructNAction()

```
static std::vector< PresentationDataValue > gdcm::network::NormalizedMessageFactory::Construct↩
NAction (
            const ULConnection & inConnection,
            const BaseQuery * inQuery )  [static]
```

10.212.1.2 ConstructNCreate()

```
static std::vector< PresentationDataValue > gdcm::network::NormalizedMessageFactory::Construct↩
NCreate (
            const ULConnection & inConnection,
            const BaseQuery * inQuery )  [static]
```

10.212.1.3 ConstructNDelete()

```
static std::vector< PresentationDataValue > gdcm::network::NormalizedMessageFactory::Construct↩
NDelete (
              const ULConnection & inConnection,
              const BaseQuery * inQuery )  [static]
```

10.212.1.4 ConstructNEventReport()

```
static std::vector< PresentationDataValue > gdcm::network::NormalizedMessageFactory::Construct↩
NEventReport (
              const ULConnection & inConnection,
              const BaseQuery * inQuery )  [static]
```

10.212.1.5 ConstructNGet()

```
static std::vector< PresentationDataValue > gdcm::network::NormalizedMessageFactory::Construct↩
NGet (
              const ULConnection & inConnection,
              const BaseQuery * inQuery )  [static]
```

10.212.1.6 ConstructNSet()

```
static std::vector< PresentationDataValue > gdcm::network::NormalizedMessageFactory::Construct↩
NSet (
              const ULConnection & inConnection,
              const BaseQuery * inQuery )  [static]
```

The documentation for this class was generated from the following file:

- gdcmNormalizedMessageFactory.h

10.213 gdcm::NormalizedNetworkFunctions Class Reference

Normalized Network Functions.

```
#include <gdcmNormalizedNetworkFunctions.h>
```

Static Public Member Functions

- static BaseQuery * ConstructQuery (const std::string &sopInstanceUID, const DataSet &queryds, ENQueryType queryType=eCreateMMPS)
- static bool NAction (const char *remote, uint16_t portno, const BaseQuery *query, std::vector< DataSet > &retDataSets, const char *aetitle, const char *call)
- static bool NCreate (const char *remote, uint16_t portno, BaseQuery *query, std::vector< DataSet > &retDataSets, const char *aetitle, const char *call)
- static bool NDelete (const char *remote, uint16_t portno, const BaseQuery *query, std::vector< DataSet > &retDataSets, const char *aetitle, const char *call)
- static bool NEventReport (const char *remote, uint16_t portno, const BaseQuery *query, std::vector< DataSet > &retDataSets, const char *aetitle, const char *call)
- static bool NGet (const char *remote, uint16_t portno, const BaseQuery *query, std::vector< DataSet > &retDataSets, const char *aetitle, const char *call)
- static bool NSet (const char *remote, uint16_t portno, const BaseQuery *query, std::vector< DataSet > &retDataSets, const char *aetitle, const char *call)

10.213.1 Detailed Description

Normalized Network Functions.

These functions provide a generic API to the DICOM functions implemented in GDCM. Advanced users can use this code as a template for building their own versions of these functions (for instance, to provide progress bars or some other way of handling returned query information), but for most users, these functions should be sufficient to interface with a PACS to a local machine. Note that these functions are not contained within a static class or some other class-style interface, because multiple connections can be instantiated in the same program. The DICOM standard is much more function oriented rather than class oriented in this instance, so the design of this API reflects that functional approach. These functions implements the following SCU operations:

- N-EVENT-REPORT
- N-GET
- N-SET
- N-ACTION
- N-CREATE
- N-DELETE

10.213.2 Member Function Documentation

10.213.2.1 ConstructQuery()

```
static BaseQuery * gdcm::NormalizedNetworkFunctions::ConstructQuery (
            const std::string & sopInstanceUID,
            const DataSet & queryds,
            ENQueryType queryType = eCreateMMPS )  [static]
```

10.213.2.2 NAction()

```
static bool gdcm::NormalizedNetworkFunctions::NAction (
            const char * remote,
            uint16_t portno,
            const BaseQuery * query,
            std::vector< DataSet > & retDataSets,
            const char * aetitle,
            const char * call )  [static]
```

10.213.2.3 NCreate()

```
static bool gdcm::NormalizedNetworkFunctions::NCreate (
            const char * remote,
            uint16_t portno,
            BaseQuery * query,
            std::vector< DataSet > & retDataSets,
            const char * aetitle,
            const char * call )  [static]
```

10.213.2.4 NDelete()

```
static bool gdcm::NormalizedNetworkFunctions::NDelete (
            const char * remote,
            uint16_t portno,
            const BaseQuery * query,
            std::vector< DataSet > & retDataSets,
            const char * aetitle,
            const char * call )  [static]
```

10.213.2.5 NEventReport()

```
static bool gdcm::NormalizedNetworkFunctions::NEventReport (
            const char * remote,
            uint16_t portno,
            const BaseQuery * query,
            std::vector< DataSet > & retDataSets,
            const char * aetitle,
            const char * call )  [static]
```

10.213.2.6 NGet()

```
static bool gdcm::NormalizedNetworkFunctions::NGet (
            const char * remote,
            uint16_t portno,
            const BaseQuery * query,
            std::vector< DataSet > & retDataSets,
            const char * aetitle,
            const char * call )  [static]
```

10.213.2.7 NSet()

```
static bool gdcm::NormalizedNetworkFunctions::NSet (
            const char * remote,
            uint16_t portno,
            const BaseQuery * query,
            std::vector< DataSet > & retDataSets,
            const char * aetitle,
            const char * call )  [static]
```

The documentation for this class was generated from the following file:

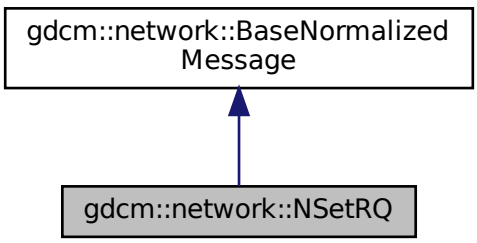
- gdcmNormalizedNetworkFunctions.h

10.214 gdcm::network::NSetRQ Class Reference

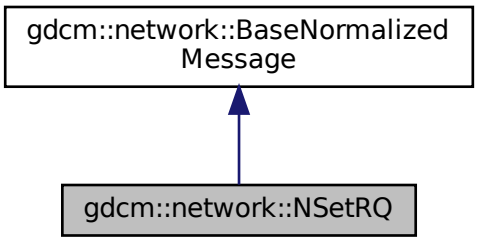
NSetRQ.

`#include <gdcmNSetMessages.h>`

Inheritance diagram for gdcm::network::NSetRQ:



Collaboration diagram for gdcm::network::NSetRQ:



Public Member Functions

- std::vector< PresentationDataValue > ConstructPDV (const ULConnection &inConnection, const BaseQuery *inQuery) override

10.214.1 Detailed Description

NSetRQ.

this file defines the messages for the nset action

10.214.2 Member Function Documentation

10.214.2.1 ConstructPDV()

```
std::vector< PresentationDataValue > gdcm::network::NSetRQ::ConstructPDV (
            const ULConnection & inConnection,
            const BaseQuery * inQuery )  [override], [virtual]
```

Implements gdcm::network::BaseNormalizedMessage.

The documentation for this class was generated from the following file:

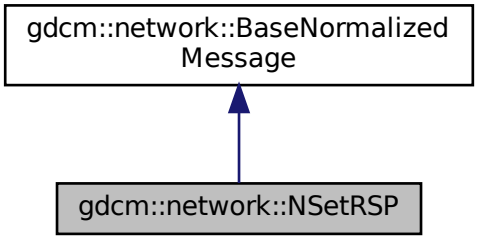
- gdcmNSetMessages.h

10.215 gdcm::network::NSetRSP Class Reference

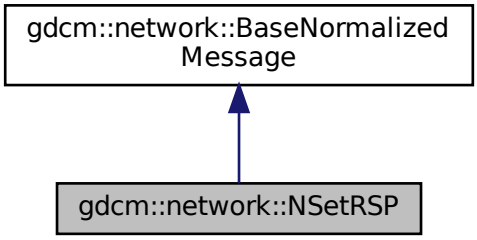
NSetRSP this file defines the messages for the nset action.

`#include <gdcmNSetMessages.h>`

Inheritance diagram for gdcm::network::NSetRSP:



Collaboration diagram for gdcm::network::NSetRSP:



Public Member Functions

- std::vector< PresentationDataValue > ConstructPDVByDataSet (const DataSet *inDataSet)

10.215.1 Detailed Description

NSetRSP this file defines the messages for the nset action.

10.215.2 Member Function Documentation

10.215.2.1 ConstructPDVByDataSet()

```
std::vector< PresentationDataValue > gdcm::network::NSetRSP::ConstructPDVByDataSet (
            const DataSet * inDataSet )
```

The documentation for this class was generated from the following file:

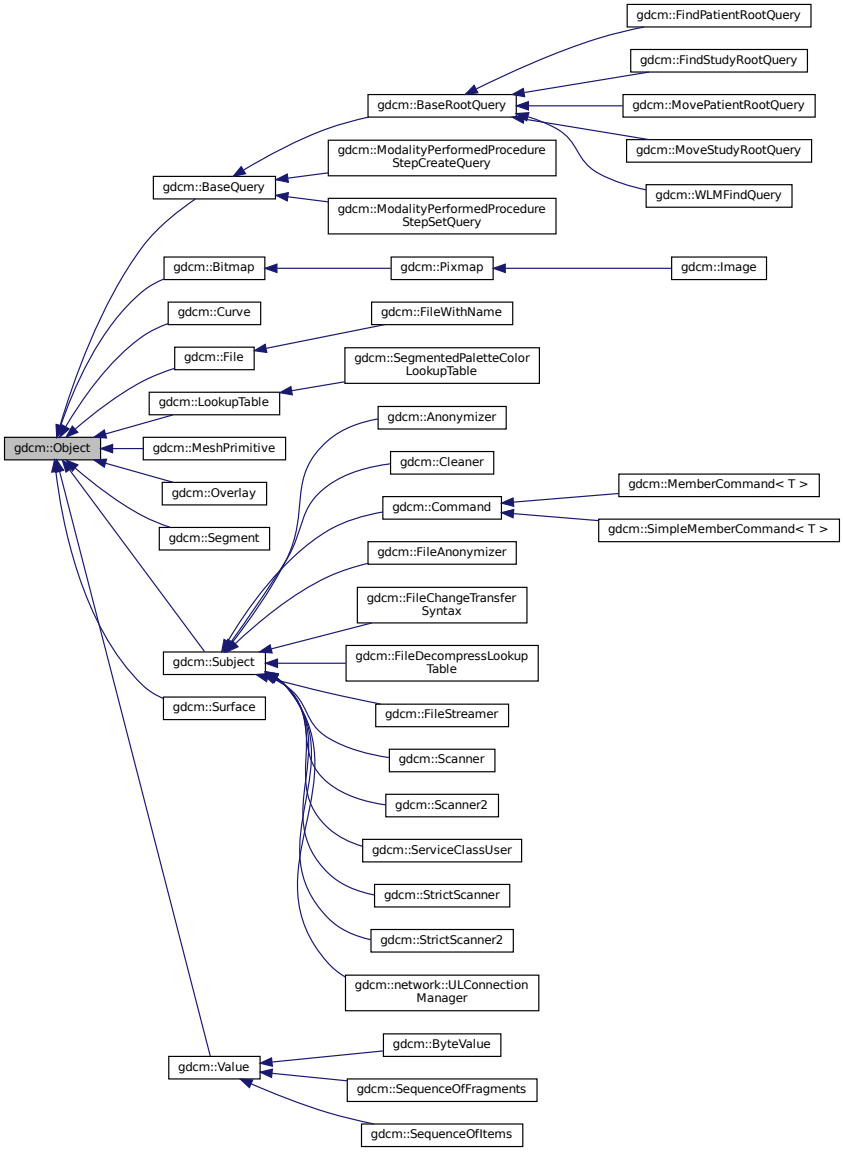
- gdcmNSetMessages.h

10.216 gdcm::Object Class Reference

Object.

```
#include <gdcmObject.h>
```

Inheritance diagram for gdcm::Object:



Public Member Functions

- Object ()
- Object (const Object &)

  Special requirement for copy/cstor, assignment operator.
- virtual ~Object ()
- void operator= (const Object &)
- virtual void Print (std::ostream &) const

Protected Member Functions

- void Register ()
- void UnRegister ()

Friends

- std::ostream & operator<< (std::ostream &os, const Object &obj)
- template<class ObjectType >
  class SmartPointer

10.216.1 Detailed Description

Object.

Note

main superclass for object that want to use SmartPointer invasive ref counting system

See also

SmartPointer

10.216.2 Constructor & Destructor Documentation

10.216.2.1 Object() [1/2]

```
gdcm::Object::Object ( )  [inline]
```

10.216.2.2 ~Object()

```
virtual gdcm::Object::~Object ( )  [inline], [virtual]
```

10.216.2.3 Object() [2/2]

```
gdcm::Object::Object (
            const Object &  )  [inline]
```

Special requirement for copy/cstor, assignment operator.

10.216.3 Member Function Documentation

10.216.3.1 operator=()

```
void gdcm::Object::operator= (
            const Object &  )  [inline]
```

10.216.3.2 Print()

```
virtual void gdcm::Object::Print (
            std::ostream &  ) const  [inline], [virtual]
```

Reimplemented in gdcm::Bitmap, gdcm::Curve, gdcm::LookupTable, gdcm::Overlay, gdcm::Pixmap, gdcm::SegmentedPaletteColorLooku
gdcm::ByteValue, gdcm::SequenceOfFragments, gdcm::SequenceOfItems, gdcm::Image, gdcm::Scanner, gdcm::Scanner2,
gdcm::StrictScanner, gdcm::StrictScanner2, and gdcm::BaseQuery.

Examples

ReadAndDumpDICOMDIR.cxx.

10.216.3.3 Register()

```
void gdcm::Object::Register ( )  [inline], [protected]
```

10.216.3.4 UnRegister()

```
void gdcm::Object::UnRegister ( )  [inline], [protected]
```

10.216.4 Friends And Related Function Documentation

10.216.4.1 operator<<

```
std::ostream & operator<< (
             std::ostream & os,
             const Object & obj )  [friend]
```

10.216.4.2 SmartPointer

```
template<class ObjectType >
friend class SmartPointer  [friend]
```

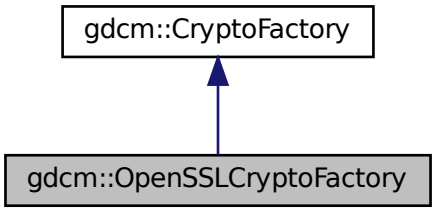
The documentation for this class was generated from the following file:

- gdcmObject.h

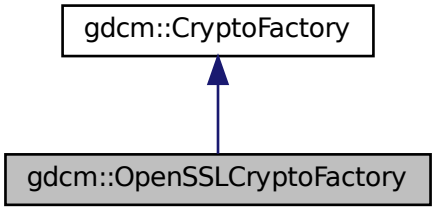
10.217 gdcm::OpenSSLCryptoFactory Class Reference

`#include <gdcmOpenSSLCryptoFactory.h>`

Inheritance diagram for gdcm::OpenSSLCryptoFactory:



Collaboration diagram for gdcm::OpenSSLCryptoFactory:



Public Member Functions

- OpenSSLCryptoFactory (CryptoLib id)
- CryptographicMessageSyntax * CreateCMSProvider ()

Protected Member Functions

- void InitOpenSSL ()

Additional Inherited Members

10.217.1 Constructor & Destructor Documentation

10.217.1.1 OpenSSLCryptoFactory()

```
gdcm::OpenSSLCryptoFactory::OpenSSLCryptoFactory (
            CryptoLib id )  [inline]
```

References gdcmDebugMacro.

10.217.2 Member Function Documentation

10.217.2.1 CreateCMSProvider()

```
CryptographicMessageSyntax * gdcm::OpenSSLCryptoFactory::CreateCMSProvider ( )  [inline], [virtual]
```

Implements gdcm::CryptoFactory.

10.217.2.2 InitOpenSSL()

```
void gdcm::OpenSSLCryptoFactory::InitOpenSSL ( )  [protected]
```

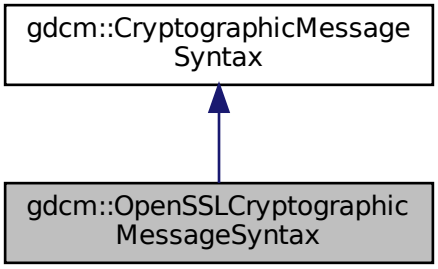
The documentation for this class was generated from the following file:

- gdcmOpenSSLCryptoFactory.h

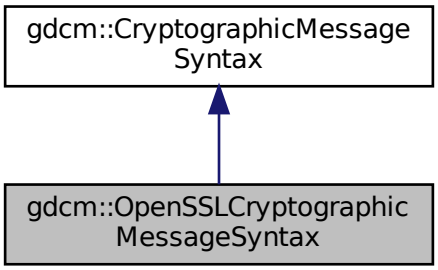
10.218 gdcm::OpenSSLCryptographicMessageSyntax Class Reference

`#include <gdcmOpenSSLCryptographicMessageSyntax.h>`

Inheritance diagram for gdcm::OpenSSLCryptographicMessageSyntax:



Collaboration diagram for gdcm::OpenSSLCryptographicMessageSyntax:



Public Member Functions

- OpenSSLCryptographicMessageSyntax ()
- ~OpenSSLCryptographicMessageSyntax ()
- bool Decrypt (char *output, size_t &outlen, const char *array, size_t len) const

  decrypt content from a PKCS#7 envelopedData structure
- bool Encrypt (char *output, size_t &outlen, const char *array, size_t len) const

  create a CMS envelopedData structure

- CipherTypes GetCipherType () const
- bool ParseCertificateFile (const char *filename)
- bool ParseKeyFile (const char *filename)
- void SetCipherType (CipherTypes type)
- bool SetPassword (const char *pass, size_t passLen)

Additional Inherited Members

10.218.1 Constructor & Destructor Documentation

10.218.1.1 OpenSSLCryptographicMessageSyntax()

```
gdcm::OpenSSLCryptographicMessageSyntax::OpenSSLCryptographicMessageSyntax ( )
```

10.218.1.2 ~OpenSSLCryptographicMessageSyntax()

```
gdcm::OpenSSLCryptographicMessageSyntax::~OpenSSLCryptographicMessageSyntax ( )
```

10.218.2 Member Function Documentation

10.218.2.1 Decrypt()

```
bool gdcm::OpenSSLCryptographicMessageSyntax::Decrypt (
            char * output,
            size_t & outlen,
            const char * array,
            size_t len ) const  [virtual]
```

decrypt content from a PKCS#7 envelopedData structure

Implements gdcm::CryptographicMessageSyntax.

10.218.2.2 Encrypt()

```
bool gdcm::OpenSSLCryptographicMessageSyntax::Encrypt (
            char * output,
            size_t & outlen,
            const char * array,
            size_t len ) const  [virtual]
```

create a CMS envelopedData structure

Implements gdcm::CryptographicMessageSyntax.

10.218.2.3 GetCipherType()

```
CipherTypes gdcm::OpenSSLCryptographicMessageSyntax::GetCipherType ( ) const  [virtual]
```

Implements gdcm::CryptographicMessageSyntax.

10.218.2.4 ParseCertificateFile()

```
bool gdcm::OpenSSLCryptographicMessageSyntax::ParseCertificateFile (
            const char * filename )  [virtual]
```

Implements gdcm::CryptographicMessageSyntax.

10.218.2.5 ParseKeyFile()

```
bool gdcm::OpenSSLCryptographicMessageSyntax::ParseKeyFile (
            const char * filename )  [virtual]
```

Implements gdcm::CryptographicMessageSyntax.

10.218.2.6 SetCipherType()

```
void gdcm::OpenSSLCryptographicMessageSyntax::SetCipherType (
            CipherTypes type )  [virtual]
```

Set Cipher Type. Default is: AES256_CIPHER

Implements gdcm::CryptographicMessageSyntax.

10.218.2.7 SetPassword()

```
bool gdcm::OpenSSLCryptographicMessageSyntax::SetPassword (
            const char * pass,
            size_t passLen )  [virtual]
```

Implements gdcm::CryptographicMessageSyntax.

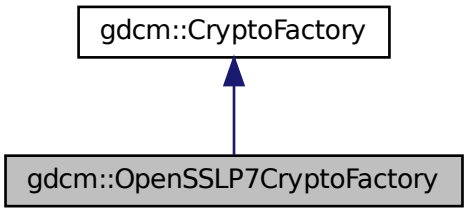
The documentation for this class was generated from the following file:

- gdcmOpenSSLCryptographicMessageSyntax.h

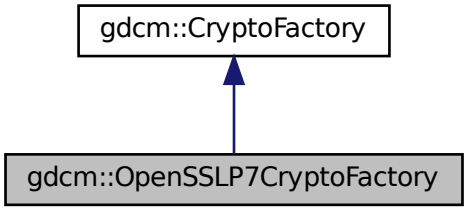
10.219 gdcm::OpenSSLP7CryptoFactory Class Reference

```
#include <gdcmOpenSSLP7CryptoFactory.h>
```

Inheritance diagram for gdcm::OpenSSLP7CryptoFactory:



Collaboration diagram for gdcm::OpenSSLP7CryptoFactory:



Public Member Functions

- OpenSSLP7CryptoFactory (CryptoLib id)
- CryptographicMessageSyntax * CreateCMSProvider ()

Additional Inherited Members

10.219.1 Constructor & Destructor Documentation

10.219.1.1 OpenSSLP7CryptoFactory()

```
gdcm::OpenSSLP7CryptoFactory::OpenSSLP7CryptoFactory (
            CryptoLib id )  [inline]
```

References gdcmDebugMacro.

10.219.2 Member Function Documentation

10.219.2.1 CreateCMSProvider()

```
CryptographicMessageSyntax * gdcm::OpenSSLP7CryptoFactory::CreateCMSProvider ( )  [inline], [virtual]
```

Implements gdcm::CryptoFactory.

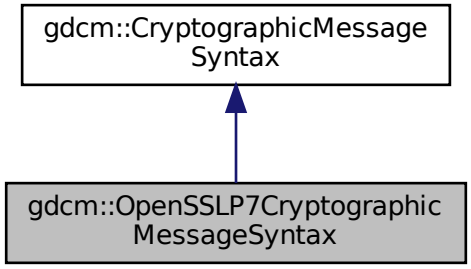
The documentation for this class was generated from the following file:

- gdcmOpenSSLP7CryptoFactory.h

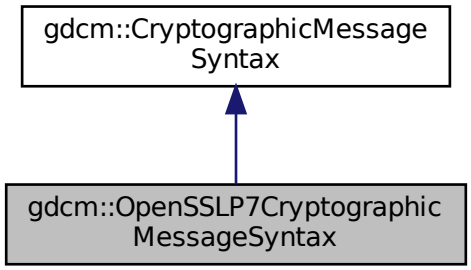
10.220 gdcm::OpenSSLP7CryptographicMessageSyntax Class Reference

`#include <gdcmOpenSSLP7CryptographicMessageSyntax.h>`

Inheritance diagram for gdcm::OpenSSLP7CryptographicMessageSyntax:



Collaboration diagram for gdcm::OpenSSLP7CryptographicMessageSyntax:



Public Member Functions

- OpenSSLP7CryptographicMessageSyntax ()
- ~OpenSSLP7CryptographicMessageSyntax ()
- bool Decrypt (char *output, size_t &outlen, const char *array, size_t len) const

  decrypt content from a PKCS#7 envelopedData structure
- bool Encrypt (char *output, size_t &outlen, const char *array, size_t len) const

  create a PKCS#7 envelopedData structure

- CipherTypes GetCipherType () const
- bool ParseCertificateFile (const char *filename)
- bool ParseKeyFile (const char *filename)
- void SetCipherType (CipherTypes type)
- bool SetPassword (const char *, size_t)

Additional Inherited Members

10.220.1 Detailed Description

Class for CryptographicMessageSyntax encryption. This is just a simple wrapper around openssl PKCS7_encrypt functionalities

See online documentation `http://www.openssl.org/docs/crypto/PKCS7_encrypt.html`

10.220.2 Constructor & Destructor Documentation

10.220.2.1 OpenSSLP7CryptographicMessageSyntax()

```
gdcm::OpenSSLP7CryptographicMessageSyntax::OpenSSLP7CryptographicMessageSyntax ( )
```

10.220.2.2 ~OpenSSLP7CryptographicMessageSyntax()

```
gdcm::OpenSSLP7CryptographicMessageSyntax::~OpenSSLP7CryptographicMessageSyntax ( )
```

10.220.3 Member Function Documentation

10.220.3.1 Decrypt()

```
bool gdcm::OpenSSLP7CryptographicMessageSyntax::Decrypt (
            char * output,
            size_t & outlen,
            const char * array,
            size_t len ) const  [virtual]
```

decrypt content from a PKCS#7 envelopedData structure

Implements gdcm::CryptographicMessageSyntax.

10.220.3.2 Encrypt()

```
bool gdcm::OpenSSLP7CryptographicMessageSyntax::Encrypt (
            char * output,
            size_t & outlen,
            const char * array,
            size_t len ) const  [virtual]
```

create a PKCS#7 envelopedData structure

Implements gdcm::CryptographicMessageSyntax.

10.220.3.3 GetCipherType()

```
CipherTypes gdcm::OpenSSLP7CryptographicMessageSyntax::GetCipherType ( ) const  [virtual]
```

Implements gdcm::CryptographicMessageSyntax.

10.220.3.4 ParseCertificateFile()

```
bool gdcm::OpenSSLP7CryptographicMessageSyntax::ParseCertificateFile (
            const char * filename )  [virtual]
```

Implements gdcm::CryptographicMessageSyntax.

10.220.3.5 ParseKeyFile()

```
bool gdcm::OpenSSLP7CryptographicMessageSyntax::ParseKeyFile (
            const char * filename )  [virtual]
```

Implements gdcm::CryptographicMessageSyntax.

10.220.3.6 SetCipherType()

```
void gdcm::OpenSSLP7CryptographicMessageSyntax::SetCipherType (
            CipherTypes type )  [virtual]
```

Set Cipher Type. Default is: AES256_CIPHER

Implements gdcm::CryptographicMessageSyntax.

10.220.3.7 SetPassword()

```
bool gdcm::OpenSSLP7CryptographicMessageSyntax::SetPassword (
            const char * ,
            size_t  )  [inline], [virtual]
```

Implements gdcm::CryptographicMessageSyntax.

References gdcmWarningMacro.

The documentation for this class was generated from the following file:

- gdcmOpenSSLP7CryptographicMessageSyntax.h

10.221 gdcm::Orientation Class Reference

class to handle Orientation

```
#include <gdcmOrientation.h>
```

Public Types

- enum OrientationType {
  UNKNOWN ,
  AXIAL ,
  CORONAL ,
  SAGITTAL ,
  OBLIQUE }

Public Member Functions

- Orientation ()
- ~Orientation ()
- void Print (std::ostream &) const

  Print.

Static Public Member Functions

- static const char * GetLabel (OrientationType type)

  Return the label of an Orientation.
- static double GetObliquityThresholdCosineValue ()
- static OrientationType GetType (const double dircos[6])
- static void SetObliquityThresholdCosineValue (double val)

  ObliquityThresholdCosineValue stuff.

Static Protected Member Functions

- static char GetMajorAxisFromPatientRelativeDirectionCosine (double x, double y, double z)

Friends

- std::ostream & operator<< (std::ostream &_os, const Orientation &o)

10.221.1 Detailed Description

class to handle Orientation

10.221.2 Member Enumeration Documentation

10.221.2.1 OrientationType

```
enum gdcm::Orientation::OrientationType
```

Enumerator

UNKNOWN	
AXIAL	
CORONAL	
SAGITTAL	
OBLIQUE	

10.221.3 Constructor & Destructor Documentation

10.221.3.1 Orientation()

```
gdcm::Orientation::Orientation ( )
```

10.221.3.2 ~Orientation()

```
gdcm::Orientation::~Orientation ( )
```

10.221.4 Member Function Documentation

10.221.4.1 GetLabel()

```
static const char * gdcm::Orientation::GetLabel (
            OrientationType type )  [static]
```

Return the label of an Orientation.

Examples

FixOrientation.cxx.

10.221.4.2 GetMajorAxisFromPatientRelativeDirectionCosine()

```
static char gdcm::Orientation::GetMajorAxisFromPatientRelativeDirectionCosine (
            double x,
            double y,
            double z )  [static], [protected]
```

10.221.4.3 GetObliquityThresholdCosineValue()

```
static double gdcm::Orientation::GetObliquityThresholdCosineValue ( )  [static]
```

10.221.4.4 GetType()

```
static OrientationType gdcm::Orientation::GetType (
            const double dircos[6] )  [static]
```

Return the type of orientation from a direction cosines Input is an array of 6 double

Examples

FixOrientation.cxx.

10.221.4.5 Print()

```
void gdcm::Orientation::Print (
            std::ostream &  ) const
```

Print.

10.221.4.6 SetObliquityThresholdCosineValue()

```
static void gdcm::Orientation::SetObliquityThresholdCosineValue (
            double val )  [static]
```

ObliquityThresholdCosineValue stuff.

10.221.5 Friends And Related Function Documentation

10.221.5.1 operator<<

```
std::ostream & operator<< (
            std::ostream & _os,
            const Orientation & o )  [friend]
```

The documentation for this class was generated from the following file:

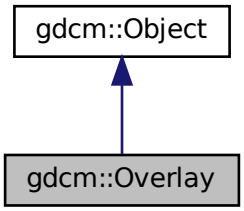
- gdcmOrientation.h

10.222 gdcm::Overlay Class Reference

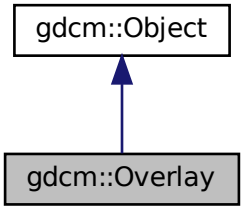
Overlay class.

```
#include <gdcmOverlay.h>
```

Inheritance diagram for gdcm::Overlay:



Collaboration diagram for gdcm::Overlay:



Public Types

- enum OverlayType {
  Invalid = 0 ,
  Graphics = 1 ,
  ROI = 2 }

Public Member Functions

- Overlay ()
- Overlay (Overlay const &ov)
- ~Overlay () override
- void Decompress (std::ostream &os) const

  Decode the internal OverlayData (packed bits) into unpacked representation.
- unsigned short GetBitPosition () const

  return bit position

- unsigned short GetBitsAllocated () const

  return bits allocated
- unsigned short GetColumns () const

  get columns
- const char * GetDescription () const

  get description
- unsigned short GetGroup () const

  Get Group number.
- const signed short * GetOrigin () const

  get origin
- const ByteValue & GetOverlayData () const
- unsigned short GetRows () const

  get rows
- const char * GetType () const

  get type
- OverlayType GetTypeAsEnum () const
- bool GetUnpackBuffer (char *buffer, size_t len) const
- size_t GetUnpackBufferLength () const
- bool GrabOverlayFromPixelData (DataSet const &ds)
- bool IsEmpty () const

  Return whether or not the Overlay is empty:
- bool IsInPixelData () const

  return if the Overlay is stored in the pixel data or not
- void IsInPixelData (bool b)

  Set whether or no the OverlayData is in the Pixel Data:
- bool IsZero () const

  return true if all bits are set to 0
- Overlay & operator= (Overlay const &ov)
- void Print (std::ostream &) const override

  Print.
- void SetBitPosition (unsigned short bitposition)

  set bit position
- void SetBitsAllocated (unsigned short bitsallocated)

  set bits allocated
- void SetColumns (unsigned short columns)

  set columns
- void SetDescription (const char *description)

  set description
- void SetFrameOrigin (unsigned short frameorigin)

  set frame origin
- void SetGroup (unsigned short group)

  Set Group number.
- void SetNumberOfFrames (unsigned int numberofframes)

  set number of frames
- void SetOrigin (const signed short origin[2])

  set origin
- void SetOverlay (const char *array, size_t length)

set overlay from byte array + length

- void SetRows (unsigned short rows)

  set rows

- void SetType (const char *type)

  set type

- void Update (const DataElement &de)

  Update overlay from data element de:

Static Public Member Functions

- static const char * GetOverlayTypeAsString (OverlayType ot)
- static OverlayType GetOverlayTypeFromString (const char *)

Additional Inherited Members

10.222.1 Detailed Description

Overlay class.

Note

see AreOverlaysInPixelData

Todo Is there actually any way to recognize an overlay ? On images with multiple overlay I do not see any way to differentiate them (other than the group tag).

Example:

10.222.2 Member Enumeration Documentation

10.222.2.1 OverlayType

`enum gdcm::Overlay::OverlayType`

Enumerator

Invalid	
Graphics	
ROI	

10.222.3 Constructor & Destructor Documentation

10.222.3.1 Overlay() [1/2]

```
gdcm::Overlay::Overlay ( )
```

10.222.3.2 ~Overlay()

```
gdcm::Overlay::~Overlay ( )  [override]
```

10.222.3.3 Overlay() [2/2]

```
gdcm::Overlay::Overlay (
            Overlay const & ov )
```

10.222.4 Member Function Documentation

10.222.4.1 Decompress()

```
void gdcm::Overlay::Decompress (
            std::ostream & os ) const
```

Decode the internal OverlayData (packed bits) into unpacked representation.

10.222.4.2 GetBitPosition()

```
unsigned short gdcm::Overlay::GetBitPosition ( ) const
```

return bit position

10.222.4.3 GetBitsAllocated()

```
unsigned short gdcm::Overlay::GetBitsAllocated ( ) const
```

return bits allocated

10.222.4.4 GetColumns()

```
unsigned short gdcm::Overlay::GetColumns ( ) const
```

get columns

10.222.4.5 GetDescription()

```
const char * gdcm::Overlay::GetDescription ( ) const
```

get description

10.222.4.6 GetGroup()

```
unsigned short gdcm::Overlay::GetGroup ( ) const
```

Get Group number.

10.222.4.7 GetOrigin()

```
const signed short * gdcm::Overlay::GetOrigin ( ) const
```

get origin

10.222.4.8 GetOverlayData()

```
const ByteValue & gdcm::Overlay::GetOverlayData ( ) const
```

Return the Overlay Data as ByteValue: Not thread safe

10.222.4.9 GetOverlayTypeAsString()

```
static const char * gdcm::Overlay::GetOverlayTypeAsString (
            OverlayType ot )  [static]
```

10.222.4.10 GetOverlayTypeFromString()

```
static OverlayType gdcm::Overlay::GetOverlayTypeFromString (
            const char *  )  [static]
```

10.222.4.11 GetRows()

```
unsigned short gdcm::Overlay::GetRows ( ) const
```

get rows

10.222.4.12 GetType()

```
const char * gdcm::Overlay::GetType ( ) const
```

get type

10.222.4.13 GetTypeAsEnum()

```
OverlayType gdcm::Overlay::GetTypeAsEnum ( ) const
```

10.222.4.14 GetUnpackBuffer()

```
bool gdcm::Overlay::GetUnpackBuffer (
            char * buffer,
            size_t len ) const
```

Retrieve the unpack buffer for Overlay. This is an error if the size if below GetUnpackBufferLength()

10.222.4.15 GetUnpackBufferLength()

```
size_t gdcm::Overlay::GetUnpackBufferLength ( ) const
```

Retrieve the size of the buffer needed to hold the Overlay as specified by Col & Row parameters

10.222.4.16 GrabOverlayFromPixelData()

```
bool gdcm::Overlay::GrabOverlayFromPixelData (
            DataSet const & ds )
```

10.222.4.17 IsEmpty()

```
bool gdcm::Overlay::IsEmpty ( ) const
```

Return whether or not the Overlay is empty:

10.222.4.18 IsInPixelData() [1/2]

```
bool gdcm::Overlay::IsInPixelData ( ) const
```

return if the Overlay is stored in the pixel data or not

10.222.4.19 IsInPixelData() [2/2]

```
void gdcm::Overlay::IsInPixelData (
            bool b )
```

Set whether or no the OverlayData is in the Pixel Data:

10.222.4.20 IsZero()

```
bool gdcm::Overlay::IsZero ( ) const
```

return true if all bits are set to 0

10.222.4.21 operator=()

```
Overlay & gdcm::Overlay::operator= (
            Overlay const & ov )
```

10.222.4.22 Print()

```
void gdcm::Overlay::Print (
            std::ostream &  ) const  [override], [virtual]
```

Print.

Reimplemented from gdcm::Object.

10.222.4.23 SetBitPosition()

```
void gdcm::Overlay::SetBitPosition (
            unsigned short bitposition )
```

set bit position

10.222.4.24 SetBitsAllocated()

```
void gdcm::Overlay::SetBitsAllocated (
            unsigned short bitsallocated )
```

set bits allocated

10.222.4.25 SetColumns()

```
void gdcm::Overlay::SetColumns (
            unsigned short columns )
```

set columns

10.222.4.26 SetDescription()

```
void gdcm::Overlay::SetDescription (
            const char * description )
```

set description

10.222.4.27 SetFrameOrigin()

```
void gdcm::Overlay::SetFrameOrigin (
            unsigned short frameorigin )
```

set frame origin

10.222.4.28 SetGroup()

```
void gdcm::Overlay::SetGroup (
            unsigned short group )
```

Set Group number.

10.222.4.29 SetNumberOfFrames()

```
void gdcm::Overlay::SetNumberOfFrames (
            unsigned int numberofframes )
```

set number of frames

10.222.4.30 SetOrigin()

```
void gdcm::Overlay::SetOrigin (
            const signed short origin[2] )
```

set origin

10.222.4.31 SetOverlay()

```
void gdcm::Overlay::SetOverlay (
            const char * array,
            size_t length )
```

set overlay from byte array + length

10.222.4.32 SetRows()

```
void gdcm::Overlay::SetRows (
            unsigned short rows )
```

set rows

10.222.4.33 SetType()

```
void gdcm::Overlay::SetType (
            const char * type )
```

set type

10.222.4.34 Update()

```
void gdcm::Overlay::Update (
            const DataElement & de )
```

Update overlay from data element de:

The documentation for this class was generated from the following file:

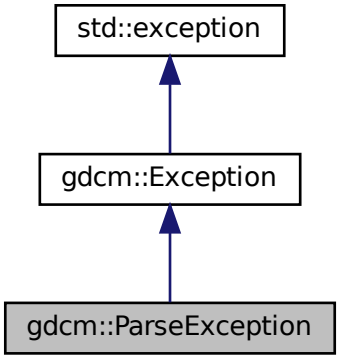
- gdcmOverlay.h

10.223 gdcm::ParseException Class Reference

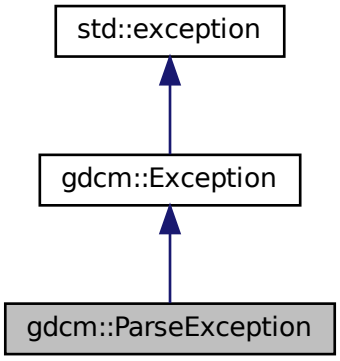
ParseException Standard exception handling object.

`#include <gdcmParseException.h>`

Inheritance diagram for gdcm::ParseException:



Collaboration diagram for gdcm::ParseException:



Public Member Functions

- ParseException ()=default
- ParseException (const ParseException &orig)
- ~ParseException () override throw ()
- const DataElement & GetLastElement () const
- ParseException & operator= (const ParseException &orig)
- void SetLastElement (DataElement &de)

10.223.1 Detailed Description

ParseException Standard exception handling object.

10.223.2 Constructor & Destructor Documentation

10.223.2.1 ParseException() [1/2]

```
gdcm::ParseException::ParseException ( )  [default]
```

10.223.2.2 ~ParseException()

```
gdcm::ParseException::~ParseException ( ) throw ( )   [inline], [override]
```

10.223.2.3 ParseException() [2/2]

```
gdcm::ParseException::ParseException (
            const ParseException & orig )  [inline]
```

10.223.3 Member Function Documentation

10.223.3.1 GetLastElement()

```
const DataElement & gdcm::ParseException::GetLastElement ( ) const  [inline]
```

10.223.3.2 operator=()

```
ParseException & gdcm::ParseException::operator= (
            const ParseException & orig )  [inline]
```

Assignment operator.

10.223.3.3 SetLastElement()

```
void gdcm::ParseException::SetLastElement (
            DataElement & de )  [inline]
```

Equivalence operator.

Referenced by gdcm::BasicOffsetTable::Read(), gdcm::Fragment::ReadBacktrack(), and gdcm::Fragment::ReadValue().

The documentation for this class was generated from the following file:

- gdcmParseException.h

10.224 gdcm::Parser Class Reference

Parser ala XML_Parser from expat (SAX)

```
#include <gdcmParser.h>
```

Public Types

- typedef void(* EndElementHandler) (void *userData, const Tag &name)
- enum ErrorType {
  NoError ,
  NoMemoryError ,
  SyntaxError ,
  NoElementsError ,
  TagMismatchError ,
  DuplicateAttributeError ,
  JunkAfterDocElementError ,
  UndefinedEntityError ,
  UnexpectedStateError }
- typedef void(* StartElementHandler) (void *userData, const Tag &tag, const char *atts[])

Public Member Functions

- Parser ()
- ~Parser ()
- unsigned long GetCurrentByteIndex () const
- ErrorType GetErrorCode () const
- void * GetUserData () const
- bool Parse (const char *s, int len, bool isFinal)
- void SetElementHandler (StartElementHandler start, EndElementHandler end)
- void SetUserData (void *userData)

Static Public Member Functions

- static const char * GetErrorString (ErrorType const &err)

Protected Member Functions

- char * GetBuffer (int len)
- bool ParseBuffer (int len, bool isFinal)
- ErrorType Process ()

10.224.1 Detailed Description

Parser ala XML_Parser from expat (SAX)

Detailed description here

Note

Simple API for DICOM

10.224.2 Member Typedef Documentation

10.224.2.1 EndElementHandler

```
typedef void(* gdcm::Parser::EndElementHandler) (void *userData, const Tag &name)
```

10.224.2.2 StartElementHandler

```
typedef void(* gdcm::Parser::StartElementHandler) (void *userData, const Tag &tag, const char
*atts[])
```

10.224.3 Member Enumeration Documentation

10.224.3.1 ErrorType

enum gdcm::Parser::ErrorType

Enumerator

NoError	
NoMemoryError	
SyntaxError	
NoElementsError	
TagMismatchError	
DuplicateAttributeError	
JunkAfterDocElementError	
UndefinedEntityError	
UnexpectedStateError	

10.224.4 Constructor & Destructor Documentation

10.224.4.1 Parser()

```
gdcm::Parser::Parser ( )  [inline]
```

10.224.4.2 ~Parser()

```
gdcm::Parser::~Parser ( )  [inline]
```

10.224.5 Member Function Documentation

10.224.5.1 GetBuffer()

```
char * gdcm::Parser::GetBuffer (
            int len )  [protected]
```

10.224.5.2 GetCurrentByteIndex()

```
unsigned long gdcm::Parser::GetCurrentByteIndex ( ) const
```

10.224.5.3 GetErrorCode()

```
ErrorType gdcm::Parser::GetErrorCode ( ) const
```

10.224.5.4 GetErrorString()

```
static const char * gdcm::Parser::GetErrorString (
            ErrorType const & err )  [static]
```

10.224.5.5 GetUserData()

```
void * gdcm::Parser::GetUserData ( ) const
```

10.224.5.6 Parse()

```
bool gdcm::Parser::Parse (
            const char * s,
            int len,
            bool isFinal )
```

10.224.5.7 ParseBuffer()

```
bool gdcm::Parser::ParseBuffer (
            int len,
            bool isFinal )  [protected]
```

10.224.5.8 Process()

```
ErrorType gdcm::Parser::Process ( )  [protected]
```

10.224.5.9 SetElementHandler()

```
void gdcm::Parser::SetElementHandler (
            StartElementHandler start,
            EndElementHandler end )
```

10.224.5.10 SetUserData()

```
void gdcm::Parser::SetUserData (
            void * userData )
```

The documentation for this class was generated from the following file:

- gdcmParser.h

10.225 gdcm::Patient Class Reference

See PS 3.3 - 2007 DICOM MODEL OF THE REAL-WORLD, p 54.

`#include <gdcmPatient.h>`

Public Member Functions

- Patient ()=default

10.225.1 Detailed Description

See PS 3.3 - 2007 DICOM MODEL OF THE REAL-WORLD, p 54.

10.225.2 Constructor & Destructor Documentation

10.225.2.1 Patient()

```
gdcm::Patient::Patient ( )  [default]
```

The documentation for this class was generated from the following file:

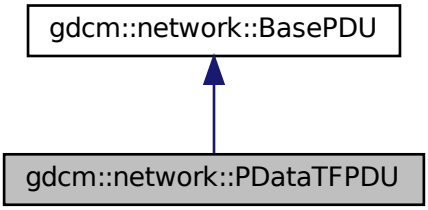
- gdcmPatient.h

10.226 gdcm::network::PDataTFPDU Class Reference

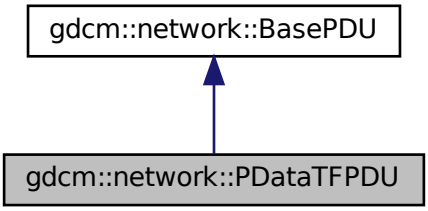
PDataTFPDU.

`#include <gdcmPDataTFPDU.h>`

Inheritance diagram for gdcm::network::PDataTFPDU:



Collaboration diagram for gdcm::network::PDataTFPDU:



Public Types

- typedef std::vector< PresentationDataValue >::size_type SizeType

Public Member Functions

- PDataTFPDU ()
- void AddPresentationDataValue (PresentationDataValue const &pdv)
- SizeType GetNumberOfPresentationDataValues () const
- PresentationDataValue const & GetPresentationDataValue (SizeType i) const
- bool IsLastFragment () const override
- void Print (std::ostream &os) const override
- std::istream & Read (std::istream &is) override
- size_t Size () const override
- const std::ostream & Write (std::ostream &os) const override

Protected Member Functions

- std::istream & ReadInto (std::istream &is, std::ostream &os)

10.226.1 Detailed Description

PDataTFPDU.

Table 9-22 P-DATA-TF PDU FIELDS

10.226.2 Member Typedef Documentation

10.226.2.1 SizeType

```
typedef std::vector<PresentationDataValue>::size_type gdcm::network::PDataTFPDU::SizeType
```

10.226.3 Constructor & Destructor Documentation

10.226.3.1 PDataTFPDU()

```
gdcm::network::PDataTFPDU::PDataTFPDU ( )
```

10.226.4 Member Function Documentation

10.226.4.1 AddPresentationDataValue()

```
void gdcm::network::PDataTFPDU::AddPresentationDataValue (
            PresentationDataValue const & pdv )  [inline]
```

10.226.4.2 GetNumberOfPresentationDataValues()

```
SizeType gdcm::network::PDataTFPDU::GetNumberOfPresentationDataValues ( ) const  [inline]
```

10.226.4.3 GetPresentationDataValue()

```
PresentationDataValue const & gdcm::network::PDataTFPDU::GetPresentationDataValue (
            SizeType i ) const  [inline]
```

10.226.4.4 IsLastFragment()

```
bool gdcm::network::PDataTFPDU::IsLastFragment ( ) const  [override], [virtual]
```

Implements gdcm::network::BasePDU.

10.226.4.5 Print()

```
void gdcm::network::PDataTFPDU::Print (
            std::ostream & os ) const  [override], [virtual]
```

Implements gdcm::network::BasePDU.

10.226.4.6 Read()

```
std::istream & gdcm::network::PDataTFPDU::Read (
            std::istream & is )  [override], [virtual]
```

Implements gdcm::network::BasePDU.

10.226.4.7 ReadInto()

```
std::istream & gdcm::network::PDataTFPDU::ReadInto (
            std::istream & is,
            std::ostream & os )  [protected]
```

10.226.4.8 Size()

```
size_t gdcm::network::PDataTFPDU::Size ( ) const  [override], [virtual]
```

Implements gdcm::network::BasePDU.

10.226.4.9 Write()

```
const std::ostream & gdcm::network::PDataTFPDU::Write (
            std::ostream & os ) const  [override], [virtual]
```

Implements gdcm::network::BasePDU.

The documentation for this class was generated from the following file:

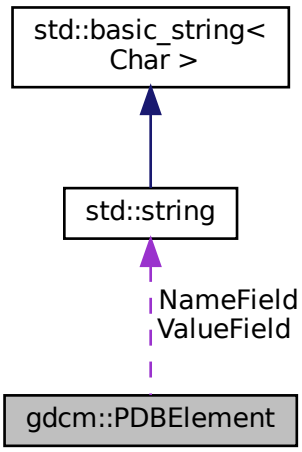
- gdcmPDataTFPDU.h

10.227 gdcm::PDBElement Class Reference

Class to represent a PDB Element.

```
#include <gdcmPDBElement.h>
```

Collaboration diagram for gdcm::PDBElement:



Public Member Functions

- PDBElement ()=default
- const char * GetName () const

  Set/Get Name.
- const char * GetValue () const

  Set/Get Value.
- bool operator== (const PDBElement &de) const
- void SetName (const char *name)
- void SetValue (const char *value)

Protected Attributes

- std::string NameField
- std::string ValueField

Friends

- std::ostream & operator<< (std::ostream &os, const PDBElement &val)

10.227.1 Detailed Description

Class to represent a PDB Element.

See also

PDBHeader

10.227.2 Constructor & Destructor Documentation

10.227.2.1 PDBElement()

```
gdcm::PDBElement::PDBElement ( )  [default]
```

10.227.3 Member Function Documentation

10.227.3.1 GetName()

```
const char * gdcm::PDBElement::GetName ( ) const  [inline]
```

Set/Get Name.

10.227.3.2 GetValue()

```
const char * gdcm::PDBElement::GetValue ( ) const  [inline]
```

Set/Get Value.

10.227.3.3 operator==()

```
bool gdcm::PDBElement::operator== (
            const PDBElement & de ) const  [inline]
```

References NameField, and ValueField.

10.227.3.4 SetName()

```
void gdcm::PDBElement::SetName (
            const char * name )  [inline]
```

10.227.3.5 SetValue()

```
void gdcm::PDBElement::SetValue (
            const char * value )  [inline]
```

10.227.4 Friends And Related Function Documentation

10.227.4.1 operator<<

```
std::ostream & operator<< (
            std::ostream & os,
            const PDBElement & val )  [friend]
```

10.227.5 Member Data Documentation

10.227.5.1 NameField

```
std::string gdcm::PDBElement::NameField  [protected]
```

Referenced by operator==().

10.227.5.2 ValueField

```
std::string gdcm::PDBElement::ValueField  [protected]
```

Referenced by operator==().

The documentation for this class was generated from the following file:

- gdcmPDBElement.h

10.228 gdcm::PDBHeader Class Reference

Class for PDBHeader.

```
#include <gdcmPDBHeader.h>
```

Public Member Functions

- PDBHeader ()=default
- ~PDBHeader ()=default
- bool FindPDBElementByName (const char *name)

  Return true if the PDB element matching name is found or not.
- const PDBElement & GetPDBElementByName (const char *name)
- bool LoadFromDataElement (DataElement const &de)

  Load the PDB Header from a DataElement of a DataSet.
- void Print (std::ostream &os) const

  Print.

Static Public Member Functions

- static const PrivateTag & GetPDBInfoTag ()

  Return the Private Tag where the PDB header is stored within a DICOM DataSet.

Protected Member Functions

- const PDBElement & GetPDBEEnd () const

Friends

- std::ostream & operator<< (std::ostream &_os, const PDBHeader &d)

10.228.1 Detailed Description

Class for PDBHeader.

GEMS MR Image have an Attribute (0025,1b,GEMS_SERS_01) which store the Acquisition parameter of the MR Image. It is compressed and can therefore not be used as is. This class de-encapsulated the Protocol Data Block and allow users to query element by name.

Warning

Everything you do with this code is at your own risk, since decoding process was not written from specification documents.

: the API of this class might change.

: SEDESC is not always pure ASCII it can contains latin1

See also

CSAHeader

10.228.2 Constructor & Destructor Documentation

10.228.2.1 PDBHeader()

```
gdcm::PDBHeader::PDBHeader ( )  [default]
```

10.228.2.2 ~PDBHeader()

```
gdcm::PDBHeader::~PDBHeader ( )  [default]
```

10.228.3 Member Function Documentation

10.228.3.1 FindPDBElementByName()

```
bool gdcm::PDBHeader::FindPDBElementByName (
            const char * name )
```

Return true if the PDB element matching name is found or not.

10.228.3.2 GetPDBEEnd()

```
const PDBElement & gdcm::PDBHeader::GetPDBEEnd ( ) const  [protected]
```

10.228.3.3 GetPDBElementByName()

```
const PDBElement & gdcm::PDBHeader::GetPDBElementByName (
            const char * name )
```

Lookup in the PDB header if a PDB element match the name 'name':

Warning

Case Sensitive

10.228.3.4 GetPDBInfoTag()

```
static const PrivateTag & gdcm::PDBHeader::GetPDBInfoTag ( )  [static]
```

Return the Private Tag where the PDB header is stored within a DICOM DataSet.

10.228.3.5 LoadFromDataElement()

```
bool gdcm::PDBHeader::LoadFromDataElement (
            DataElement const & de )
```

Load the PDB Header from a DataElement of a DataSet.

10.228.3.6 Print()

```
void gdcm::PDBHeader::Print (
            std::ostream & os ) const
```

Print.

10.228.4 Friends And Related Function Documentation

10.228.4.1 operator<<

```
std::ostream & operator<< (
            std::ostream & _os,
            const PDBHeader & d )  [friend]
```

The documentation for this class was generated from the following file:

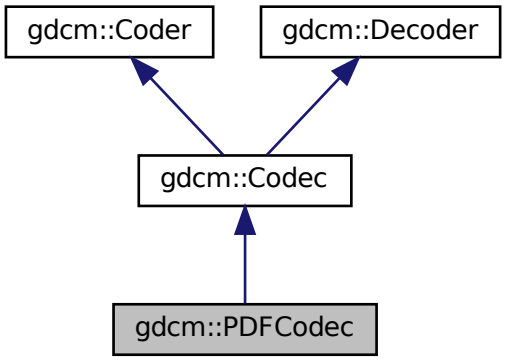
- gdcmPDBHeader.h

10.229 gdcm::PDFCodec Class Reference

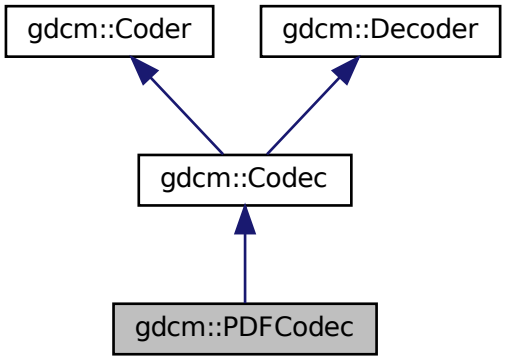
PDFCodec class.

```
#include <gdcmPDFCodec.h>
```

Inheritance diagram for gdcm::PDFCodec:



Collaboration diagram for gdcm::PDFCodec:



Public Member Functions

- PDFCodec ()
- ~PDFCodec () override
- bool CanCode (TransferSyntax const &) const override

  Return whether this coder support this transfer syntax (can code it)
- bool CanDecode (TransferSyntax const &) const override

  Return whether this decoder support this transfer syntax (can decode it)
- bool Decode (DataElement const &is, DataElement &os) override

  Decode.

Additional Inherited Members

10.229.1 Detailed Description

PDFCodec class.

10.229.2 Constructor & Destructor Documentation

10.229.2.1 PDFCodec()

```
gdcm::PDFCodec::PDFCodec ( )
```

10.229.2.2 ~PDFCodec()

```
gdcm::PDFCodec::~PDFCodec ( )  [override]
```

10.229.3 Member Function Documentation

10.229.3.1 CanCode()

```
bool gdcm::PDFCodec::CanCode (
            TransferSyntax const &  ) const  [inline], [override], [virtual]
```

Return whether this coder support this transfer syntax (can code it)

Implements gdcm::Coder.

10.229.3.2 CanDecode()

```
bool gdcm::PDFCodec::CanDecode (
            TransferSyntax const &  ) const  [inline], [override], [virtual]
```

Return whether this decoder support this transfer syntax (can decode it)

Implements gdcm::Decoder.

10.229.3.3 Decode()

```
bool gdcm::PDFCodec::Decode (
             DataElement const & ,
             DataElement &  )  [override], [virtual]
```

Decode.

Reimplemented from gdcm::Decoder.

The documentation for this class was generated from the following file:

- gdcmPDFCodec.h

10.230 gdcm::network::PDUFactory Class Reference

PDUFactory basically, given an initial byte, construct the.

```
#include <gdcmPDUFactory.h>
```

Static Public Member Functions

- static BasePDU * ConstructAbortPDU ()
- static BasePDU * ConstructPDU (uint8_t itemtype)
- static BasePDU * ConstructReleasePDU ()
- static std::vector< BasePDU * > CreateCEchoPDU (const ULConnection &inConnection)
- static std::vector< BasePDU * > CreateCFindPDU (const ULConnection &inConnection, const BaseRootQuery *inRootQuery)
- static std::vector< BasePDU * > CreateCMovePDU (const ULConnection &inConnection, const BaseRootQuery *inRootQuery)
- static std::vector< BasePDU * > CreateCStoreRQPDU (const ULConnection &inConnection, const File &file, bool writeDataSet=true)
- static std::vector< BasePDU * > CreateCStoreRSPPDU (const DataSet *inDataSet, const BasePDU *inPC)
- static std::vector< BasePDU * > CreateNActionPDU (const ULConnection &inConnection, const BaseQuery *inQuery)
- static std::vector< BasePDU * > CreateNCreatePDU (const ULConnection &inConnection, const BaseQuery *inQuery)
- static std::vector< BasePDU * > CreateNDeletePDU (const ULConnection &inConnection, const BaseQuery *inQuery)
- static std::vector< BasePDU * > CreateNEventReportPDU (const ULConnection &inConnection, const BaseQuery *inQuery)
- static std::vector< BasePDU * > CreateNGetPDU (const ULConnection &inConnection, const BaseQuery *in↩ Query)
- static std::vector< BasePDU * > CreateNSetPDU (const ULConnection &inConnection, const BaseQuery *in↩ Query)
- static EEventID DetermineEventByPDU (const BasePDU *inPDU)
- static std::vector< PresentationDataValue > GetPDVs (const std::vector< BasePDU * > &inDataPDUs)

10.230.1 Detailed Description

PDUFactory basically, given an initial byte, construct the.

appropriate PDU. This way, the event loop doesn't have to know about all the different PDU types.

10.230.2 Member Function Documentation

10.230.2.1 ConstructAbortPDU()

```
static BasePDU * gdcm::network::PDUFactory::ConstructAbortPDU ( )  [static]
```

10.230.2.2 ConstructPDU()

```
static BasePDU * gdcm::network::PDUFactory::ConstructPDU (
            uint8_t itemtype )  [static]
```

10.230.2.3 ConstructReleasePDU()

```
static BasePDU * gdcm::network::PDUFactory::ConstructReleasePDU ( )  [static]
```

10.230.2.4 CreateCEchoPDU()

```
static std::vector< BasePDU * > gdcm::network::PDUFactory::CreateCEchoPDU (
            const ULConnection & inConnection )  [static]
```

10.230.2.5 CreateCFindPDU()

```
static std::vector< BasePDU * > gdcm::network::PDUFactory::CreateCFindPDU (
            const ULConnection & inConnection,
            const BaseRootQuery * inRootQuery )  [static]
```

10.230.2.6 CreateCMovePDU()

```
static std::vector< BasePDU * > gdcm::network::PDUFactory::CreateCMovePDU (
            const ULConnection & inConnection,
            const BaseRootQuery * inRootQuery )  [static]
```

10.230.2.7 CreateCStoreRQPDU()

```
static std::vector< BasePDU * > gdcm::network::PDUFactory::CreateCStoreRQPDU (
            const ULConnection & inConnection,
            const File & file,
            bool writeDataSet = true )  [static]
```

10.230.2.8 CreateCStoreRSPPDU()

```
static std::vector< BasePDU * > gdcm::network::PDUFactory::CreateCStoreRSPPDU (
            const DataSet * inDataSet,
            const BasePDU * inPC )  [static]
```

10.230.2.9 CreateNActionPDU()

```
static std::vector< BasePDU * > gdcm::network::PDUFactory::CreateNActionPDU (
            const ULConnection & inConnection,
            const BaseQuery * inQuery )  [static]
```

10.230.2.10 CreateNCreatePDU()

```
static std::vector< BasePDU * > gdcm::network::PDUFactory::CreateNCreatePDU (
            const ULConnection & inConnection,
            const BaseQuery * inQuery )  [static]
```

10.230.2.11 CreateNDeletePDU()

```
static std::vector< BasePDU * > gdcm::network::PDUFactory::CreateNDeletePDU (
            const ULConnection & inConnection,
            const BaseQuery * inQuery )  [static]
```

10.230.2.12 CreateNEventReportPDU()

```
static std::vector< BasePDU * > gdcm::network::PDUFactory::CreateNEventReportPDU (
            const ULConnection & inConnection,
            const BaseQuery * inQuery )  [static]
```

10.230.2.13 CreateNGetPDU()

```
static std::vector< BasePDU * > gdcm::network::PDUFactory::CreateNGetPDU (
            const ULConnection & inConnection,
            const BaseQuery * inQuery )  [static]
```

10.230.2.14 CreateNSetPDU()

```
static std::vector< BasePDU * > gdcm::network::PDUFactory::CreateNSetPDU (
            const ULConnection & inConnection,
            const BaseQuery * inQuery )  [static]
```

10.230.2.15 DetermineEventByPDU()

```
static EEventID gdcm::network::PDUFactory::DetermineEventByPDU (
            const BasePDU * inPDU )  [static]
```

10.230.2.16 GetPDVs()

```
static std::vector< PresentationDataValue > gdcm::network::PDUFactory::GetPDVs (
            const std::vector< BasePDU * > & inDataPDUs )  [static]
```

The documentation for this class was generated from the following file:

- gdcmPDUFactory.h

10.231 gdcm::PersonName Class Reference

PersonName class.

```
#include <gdcmPersonName.h>
```

Public Member Functions

- unsigned int GetMaxLength () const
- unsigned int GetNumberOfComponents () const
- void Print (std::ostream &os) const
- void SetBlob (const std::vector< char > &v)
- void SetComponents (const char *comp1="", const char *comp2="", const char *comp3="", const char *comp4="", const char *comp5="")
- void SetComponents (const char *components[ ])

Public Attributes

- char Component [MaxNumberOfComponents][MaxLength+1]

Static Public Attributes

- static const unsigned int MaxLength = 64
- static const unsigned int MaxNumberOfComponents = 5
- static const char Padding = ' '
- static const char Separator = '^'

10.231.1 Detailed Description

PersonName class.

10.231.2 Member Function Documentation

10.231.2.1 GetMaxLength()

```
unsigned int gdcm::PersonName::GetMaxLength ( ) const  [inline]
```

10.231.2.2 GetNumberOfComponents()

```
unsigned int gdcm::PersonName::GetNumberOfComponents ( ) const  [inline]
```

10.231.2.3 Print()

```
void gdcm::PersonName::Print (
            std::ostream & os ) const  [inline]
```

10.231.2.4 SetBlob()

```
void gdcm::PersonName::SetBlob (
            const std::vector< char > & v )  [inline]
```

10.231.2.5 SetComponents() [1/2]

```
void gdcm::PersonName::SetComponents (
            const char * comp1 = "",
            const char * comp2 = "",
            const char * comp3 = "",
            const char * comp4 = "",
            const char * comp5 = "" )  [inline]
```

10.231.2.6 SetComponents() [2/2]

```
void gdcm::PersonName::SetComponents (
            const char * components[] )  [inline]
```

10.231.3 Member Data Documentation

10.231.3.1 Component

```
char gdcm::PersonName::Component[MaxNumberOfComponents][MaxLength+1]
```

10.231.3.2 MaxLength

```
const unsigned int gdcm::PersonName::MaxLength = 64  [static]
```

10.231.3.3 MaxNumberOfComponents

```
const unsigned int gdcm::PersonName::MaxNumberOfComponents = 5  [static]
```

10.231.3.4 Padding

```
const char gdcm::PersonName::Padding = ' '  [static]
```

10.231.3.5 Separator

```
const char gdcm::PersonName::Separator = '^'  [static]
```

The documentation for this class was generated from the following file:

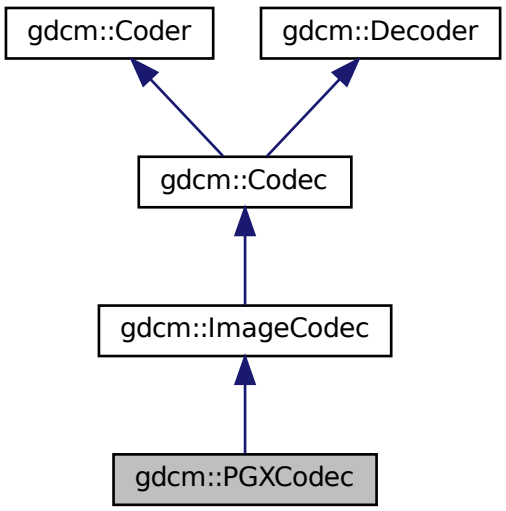
- gdcmPersonName.h

10.232 gdcm::PGXCodec Class Reference

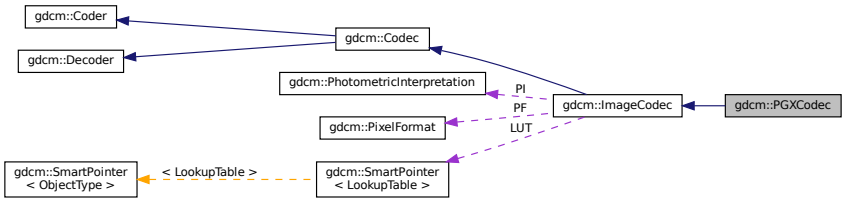
Class to do PGX.

```
#include <gdcmPGXCodec.h>
```

Inheritance diagram for gdcm::PGXCodec:



Collaboration diagram for gdcm::PGXCodec:



Public Member Functions

- PGXCodec ()
- ~PGXCodec () override
- bool CanCode (TransferSyntax const &ts) const override

  Return whether this coder support this transfer syntax (can code it)
- bool CanDecode (TransferSyntax const &ts) const override

  Return whether this decoder support this transfer syntax (can decode it)
- ImageCodec * Clone () const override
- bool GetHeaderInfo (std::istream &is, TransferSyntax &ts) override
- bool Read (const char *filename, DataElement &out) const
- bool Write (const char *filename, const DataElement &out) const

Additional Inherited Members

10.232.1 Detailed Description

Class to do PGX.

See PGX as used in JPEG 2000 implementation and reference images

10.232.2 Constructor & Destructor Documentation

10.232.2.1 PGXCodec()

```
gdcm::PGXCodec::PGXCodec ( )
```

10.232.2.2 ~PGXCodec()

```
gdcm::PGXCodec::~PGXCodec ( )  [override]
```

10.232.3 Member Function Documentation

10.232.3.1 CanCode()

```
bool gdcm::PGXCodec::CanCode (
            TransferSyntax const &  ) const  [override], [virtual]
```

Return whether this coder support this transfer syntax (can code it)

Reimplemented from gdcm::ImageCodec.

10.232.3.2 CanDecode()

```
bool gdcm::PGXCodec::CanDecode (
            TransferSyntax const &  ) const  [override], [virtual]
```

Return whether this decoder support this transfer syntax (can decode it)

Reimplemented from gdcm::ImageCodec.

10.232.3.3 Clone()

```
ImageCodec * gdcm::PGXCodec::Clone ( ) const  [override], [virtual]
```

Implements gdcm::ImageCodec.

10.232.3.4 GetHeaderInfo()

```
bool gdcm::PGXCodec::GetHeaderInfo (
            std::istream & is,
            TransferSyntax & ts )  [override], [virtual]
```

Reimplemented from gdcm::ImageCodec.

10.232.3.5 Read()

```
bool gdcm::PGXCodec::Read (
            const char * filename,
            DataElement & out ) const
```

10.232.3.6 Write()

```
bool gdcm::PGXCodec::Write (
            const char * filename,
            const DataElement & out ) const
```

The documentation for this class was generated from the following file:

- gdcmPGXCodec.h

10.233 gdcm::PhotometricInterpretation Class Reference

Class to represent an PhotometricInterpretation.

```
#include <gdcmPhotometricInterpretation.h>
```

Public Types

- enum PIType {
  UNKNOWN = 0 ,
  MONOCHROME1 ,
  MONOCHROME2 ,
  PALETTE_COLOR ,
  RGB ,
  HSV ,
  ARGB ,
  CMYK ,
  YBR_FULL ,
  YBR_FULL_422 ,
  YBR_PARTIAL_422 ,
  YBR_PARTIAL_420 ,
  YBR_ICT ,
  YBR_RCT ,
  PI_END }

Public Member Functions

- PhotometricInterpretation (PIType pi=UNKNOWN)
- unsigned short GetSamplesPerPixel () const

  return the value for Sample Per Pixel associated with a particular Photometric Interpretation
- const char * GetString () const
- PIType GetType () const
- bool IsLossless () const
- bool IsLossy () const
- bool IsSameColorSpace (PhotometricInterpretation const &pi) const
- operator PIType () const

Static Public Member Functions

- static const char * GetPIString (PIType pi)
- static PIType GetPIType (const char *pi)
- static bool IsRetired (PIType pi)

Friends

- std::ostream & operator<< (std::ostream &os, const PhotometricInterpretation &pi)

10.233.1 Detailed Description

Class to represent an PhotometricInterpretation.

Examples

CreateARGBImage.cxx, CreateCMYKImage.cxx, DecompressImage.cs, DecompressImageMultiframe.cs, DecompressJPEGFile.cs, ExtractImageRegion.cs, FileChangeTS.cs, FileChangeTSLossy.cs, HelloVizWorld.cxx, MpegVideoInfo.cs, csa2img.cxx, and iU22tomultisc.cxx.

10.233.2 Member Enumeration Documentation

Enumerator

10.233.2.1 PIType

```
enum gdcm::PhotometricInterpretation::PIType
```

Enumerator

UNKNOWN	
MONOCHROME1	
MONOCHROME2	
PALETTE_COLOR	
RGB	
HSV	
ARGB	
CMYK	
YBR_FULL	
YBR_FULL_422	
YBR_PARTIAL_422	
YBR_PARTIAL_420	
YBR_ICT	
YBR_RCT	
PI_END	

Examples

DecompressImageMultiframe.cs, DecompressJPEGFile.cs, FileChangeTS.cs, FileChangeTSLossy.cs, and MpegVideoInfo.cs.

10.233.3 Constructor & Destructor Documentation

10.233.3.1 PhotometricInterpretation()

```
gdcm::PhotometricInterpretation::PhotometricInterpretation (
            PIType pi = UNKNOWN )  [inline]
```

10.233.4 Member Function Documentation

10.233.4.1 GetPIString()

```
static const char * gdcm::PhotometricInterpretation::GetPIString (
            PIType pi )  [static]
```

10.233.4.2 GetPIType()

```
static PIType gdcm::PhotometricInterpretation::GetPIType (
            const char * pi )  [static]
```

10.233.4.3 GetSamplesPerPixel()

```
unsigned short gdcm::PhotometricInterpretation::GetSamplesPerPixel ( ) const
```

return the value for Sample Per Pixel associated with a particular Photometric Interpretation

10.233.4.4 GetString()

```
const char * gdcm::PhotometricInterpretation::GetString ( ) const
```

10.233.4.5 GetType()

```
PIType gdcm::PhotometricInterpretation::GetType ( ) const  [inline]
```

10.233.4.6 IsLossless()

```
bool gdcm::PhotometricInterpretation::IsLossless ( ) const
```

10.233.4.7 IsLossy()

```
bool gdcm::PhotometricInterpretation::IsLossy ( ) const
```

10.233.4.8 IsRetired()

```
static bool gdcm::PhotometricInterpretation::IsRetired (
            PIType pi )  [static]
```

10.233.4.9 IsSameColorSpace()

```
bool gdcm::PhotometricInterpretation::IsSameColorSpace (
            PhotometricInterpretation const & pi ) const
```

10.233.4.10 operator PIType()

```
gdcm::PhotometricInterpretation::operator PIType ( ) const  [inline]
```

10.233.5 Friends And Related Function Documentation

10.233.5.1 operator<<

```
std::ostream & operator<< (
            std::ostream & os,
            const PhotometricInterpretation & pi )  [friend]
```

The documentation for this class was generated from the following file:

- gdcmPhotometricInterpretation.h

10.234 gdcm::PixelFormat Class Reference

PixelFormat.

```
#include <gdcmPixelFormat.h>
```

Public Types

- enum ScalarType {
  UINT8 ,
  INT8 ,
  UINT12 ,
  INT12 ,
  UINT16 ,
  INT16 ,
  UINT32 ,
  INT32 ,
  UINT64 ,
  INT64 ,
  FLOAT16 ,
  FLOAT32 ,
  FLOAT64 ,
  SINGLEBIT ,
  UNKNOWN }

Public Member Functions

- PixelFormat ()
- PixelFormat (ScalarType st)
- PixelFormat (unsigned short samplesperpixel, unsigned short bitsallocated=8, unsigned short bitsstored=8, unsigned short highbit=7, unsigned short pixelrepresentation=0)
- unsigned short GetBitsAllocated () const

  BitsAllocated see Tag (0028,0100) US Bits Allocated.
- unsigned short GetBitsStored () const

  BitsStored see Tag (0028,0101) US Bits Stored.
- unsigned short GetHighBit () const

  HighBit see Tag (0028,0102) US High Bit.
- int64_t GetMax () const

  return the max possible of the pixel
- int64_t GetMin () const

  return the min possible of the pixel
- unsigned short GetPixelRepresentation () const

  PixelRepresentation: 0 or 1, see Tag (0028,0103) US Pixel Representation.
- uint8_t GetPixelSize () const
- unsigned short GetSamplesPerPixel () const
- ScalarType GetScalarType () const

  ScalarType does not take into account the sample per pixel.
- const char * GetScalarTypeAsString () const
- bool IsCompatible (const TransferSyntax &ts) const
- bool IsValid () const

  return IsValid
- operator ScalarType () const
- bool operator!= (const PixelFormat &pf) const
- bool operator!= (ScalarType st) const
- bool operator== (const PixelFormat &pf) const
- bool operator== (ScalarType st) const

- void Print (std::ostream &os) const

  Print.
- void SetBitsAllocated (unsigned short ba)
- void SetBitsStored (unsigned short bs)
- void SetHighBit (unsigned short hb)
- void SetPixelRepresentation (unsigned short pr)
- void SetSamplesPerPixel (unsigned short spp)
- void SetScalarType (ScalarType st)

Protected Member Functions

- bool Validate ()

  When image with 24/24/23 was read, need to validate.

Friends

- class Bitmap
- std::ostream & operator<< (std::ostream &_os, const PixelFormat &pf)

10.234.1 Detailed Description

PixelFormat.

By default the Pixel Type will be instantiated with the following parameters:

- SamplesPerPixel : 1
- BitsAllocated : 8
- BitsStored : 8
- HighBit : 7
- PixelRepresentation : 0

Fundamentally PixelFormat is very close to what DICOM allows. It will be very hard to extend this class for the upcoming DICOM standard where Floating 32 and 64bits will be allowed.

It is also very hard for this class to fully support 64bits integer type (see GetMin / GetMax signature restricted to 64bits signed).

Examples

CreateARGBImage.cxx, CreateCMYKImage.cxx, DecompressImage.cs, DecompressImageMultiframe.cs, DecompressJPEGFile.cs, ExtractImageRegion.cs, ExtractImageRegionWithLUT.cs, ExtractOneFrame.cs, FixJAIBugJPEGLS.cxx, GetArray.cs, GetJPEGSamplePrecision.cxx, MpegVideoInfo.cs, RescaleImage.cs, TemplateEmptyImage.cxx, csa2img.cxx, iU22tomultisc.cxx, and threadgdcm.cxx.

10.234.2 Member Enumeration Documentation

10.234.2.1 ScalarType

`enum gdcm::PixelFormat::ScalarType`

Enumerator

UINT8	
INT8	
UINT12	
INT12	
UINT16	
INT16	
UINT32	
INT32	
UINT64	
INT64	
FLOAT16	
FLOAT32	
FLOAT64	
SINGLEBIT	
UNKNOWN	

Examples

GetArray.cs.

10.234.3 Constructor & Destructor Documentation

10.234.3.1 PixelFormat() [1/3]

```
gdcm::PixelFormat::PixelFormat ( )  [inline]
```

10.234.3.2 PixelFormat() [2/3]

```
gdcm::PixelFormat::PixelFormat (
            unsigned short samplesperpixel,
            unsigned short bitsallocated = 8,
            unsigned short bitsstored = 8,
            unsigned short highbit = 7,
            unsigned short pixelrepresentation = 0 )  [inline], [explicit]
```

10.234.3.3 PixelFormat() [3/3]

```
gdcm::PixelFormat::PixelFormat (
            ScalarType st )
```

10.234.4 Member Function Documentation

10.234.4.1 GetBitsAllocated()

```
unsigned short gdcm::PixelFormat::GetBitsAllocated ( ) const  [inline]
```

BitsAllocated see Tag (0028,0100) US Bits Allocated.

Examples

GetJPEGSamplePrecision.cxx.

10.234.4.2 GetBitsStored()

```
unsigned short gdcm::PixelFormat::GetBitsStored ( ) const  [inline]
```

BitsStored see Tag (0028,0101) US Bits Stored.

Examples

GetJPEGSamplePrecision.cxx.

10.234.4.3 GetHighBit()

```
unsigned short gdcm::PixelFormat::GetHighBit ( ) const  [inline]
```

HighBit see Tag (0028,0102) US High Bit.

10.234.4.4 GetMax()

```
int64_t gdcm::PixelFormat::GetMax ( ) const
```

return the max possible of the pixel

10.234.4.5 GetMin()

```
int64_t gdcm::PixelFormat::GetMin ( ) const
```

return the min possible of the pixel

10.234.4.6 GetPixelRepresentation()

```
unsigned short gdcm::PixelFormat::GetPixelRepresentation ( ) const  [inline]
```

PixelRepresentation: 0 or 1, see Tag (0028,0103) US Pixel Representation.

10.234.4.7 GetPixelSize()

```
uint8_t gdcm::PixelFormat::GetPixelSize ( ) const
```

return the size of the pixel This is the number of words it would take to store one pixel

Warning

the return value takes into account the SamplesPerPixel

in the rare case when BitsAllocated == 12, the function assume word padding and value returned will be identical as if BitsAllocated == 16

Examples

ExtractImageRegion.cs, ExtractImageRegionWithLUT.cs, ExtractOneFrame.cs, and threadgdcm.cxx.

10.234.4.8 GetSamplesPerPixel()

```
unsigned short gdcm::PixelFormat::GetSamplesPerPixel ( ) const
```

Samples Per Pixel see (0028,0002) US Samples Per Pixel DICOM - only allows 1, 3 and 4 as valid value. Other value are undefined behavior.

Examples

threadgdcm.cxx.

10.234.4.9 GetScalarType()

```
ScalarType gdcm::PixelFormat::GetScalarType ( ) const
```

ScalarType does not take into account the sample per pixel.

Examples

GetArray.cs.

10.234.4.10 GetScalarTypeAsString()

```
const char * gdcm::PixelFormat::GetScalarTypeAsString ( ) const
```

Examples

GetArray.cs.

10.234.4.11 IsCompatible()

```
bool gdcm::PixelFormat::IsCompatible (
            const TransferSyntax & ts ) const
```

10.234.4.12 IsValid()

```
bool gdcm::PixelFormat::IsValid ( ) const
```

return IsValid

10.234.4.13 operator ScalarType()

```
gdcm::PixelFormat::operator ScalarType ( ) const  [inline]
```

10.234.4.14 operator"!=() [1/2]

```
bool gdcm::PixelFormat::operator!= (
            const PixelFormat & pf ) const  [inline]
```

10.234.4.15 operator"!=() [2/2]

```
bool gdcm::PixelFormat::operator!= (
            ScalarType st ) const  [inline]
```

10.234.4.16 operator==() [1/2]

```
bool gdcm::PixelFormat::operator== (
            const PixelFormat & pf ) const  [inline]
```

10.234.4.17 operator==() [2/2]

```
bool gdcm::PixelFormat::operator== (
            ScalarType st ) const  [inline]
```

10.234.4.18 Print()

```
void gdcm::PixelFormat::Print (
            std::ostream & os ) const
```

Print.

10.234.4.19 SetBitsAllocated()

```
void gdcm::PixelFormat::SetBitsAllocated (
            unsigned short ba )  [inline]
```

10.234.4.20 SetBitsStored()

```
void gdcm::PixelFormat::SetBitsStored (
            unsigned short bs )  [inline]
```

10.234.4.21 SetHighBit()

```
void gdcm::PixelFormat::SetHighBit (
            unsigned short hb )  [inline]
```

10.234.4.22 SetPixelRepresentation()

```
void gdcm::PixelFormat::SetPixelRepresentation (
            unsigned short pr )  [inline]
```

Examples

TemplateEmptyImage.cxx.

10.234.4.23 SetSamplesPerPixel()

```
void gdcm::PixelFormat::SetSamplesPerPixel (
            unsigned short spp )  [inline]
```

Examples

CreateARGBImage.cxx, CreateCMYKImage.cxx, and GenFakeImage.cxx.

References gdcmAssertMacro.

10.234.4.24 SetScalarType()

```
void gdcm::PixelFormat::SetScalarType (
            ScalarType st )
```

Set PixelFormat based only on the ScalarType

Warning

: You need to call SetScalarType *before* SetSamplesPerPixel

10.234.4.25 Validate()

```
bool gdcm::PixelFormat::Validate ( )  [protected]
```

When image with 24/24/23 was read, need to validate.

Referenced by gdcm::Bitmap::SetPixelFormat().

10.234.5 Friends And Related Function Documentation

10.234.5.1 Bitmap

```
friend class Bitmap  [friend]
```

10.234.5.2 operator<<

```
std::ostream & operator<< (
             std::ostream & _os,
             const PixelFormat & pf )  [friend]
```

The documentation for this class was generated from the following file:

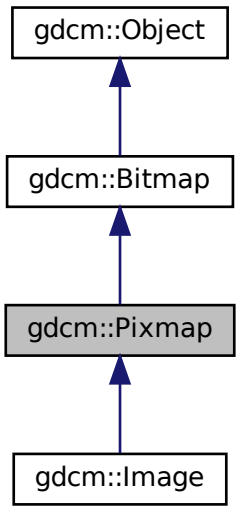
- gdcmPixelFormat.h

10.235 gdcm::Pixmap Class Reference

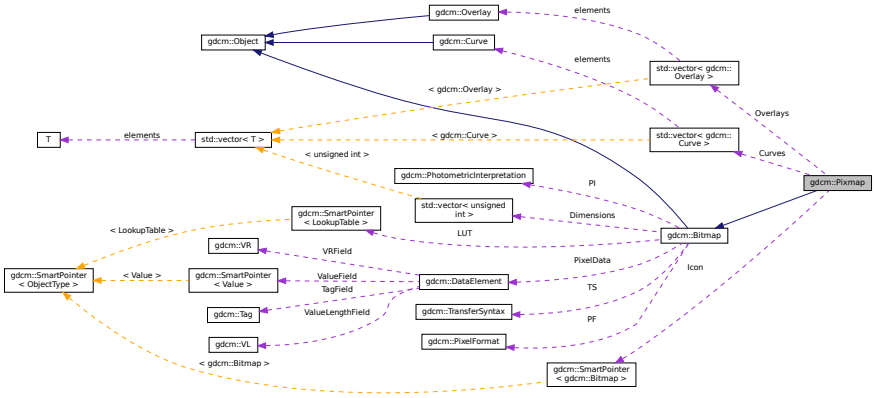
Pixmap class.

```
#include <gdcmPixmap.h>
```

Inheritance diagram for gdcm::Pixmap:



Collaboration diagram for gdcm::Pixmap:



Public Member Functions

- Pixmap ()
- ~Pixmap () override
- bool AreOverlaysInPixelData () const override

  returns if Overlays are stored in the unused bit of the pixel data:
- Curve & GetCurve (size_t i=0)

  Curve: group 50xx.
- const Curve & GetCurve (size_t i=0) const
- IconImage & GetIconImage ()
- const IconImage & GetIconImage () const

  Set/Get Icon Image.
- size_t GetNumberOfCurves () const
- size_t GetNumberOfOverlays () const
- Overlay & GetOverlay (size_t i=0)

  Overlay: group 60xx.
- const Overlay & GetOverlay (size_t i=0) const
- void Print (std::ostream &) const override
- void RemoveOverlay (size_t i)
- void SetIconImage (IconImage const &ii)
- void SetNumberOfCurves (size_t n)
- void SetNumberOfOverlays (size_t n)
- bool UnusedBitsPresentInPixelData () const override

  returns if there are unused bits in the pixel data

Protected Attributes

- std::vector< Curve > Curves
- SmartPointer< IconImage > Icon
- std::vector< Overlay > Overlays

Additional Inherited Members

10.235.1 Detailed Description

Pixmap class.

A bitmap based image. Used as parent for both IconImage and the main Pixel Data Image It does not contains any World Space information (IPP, IOP)

See also

PixmapReader

Examples

FileChangeTS.cs, FileChangeTSLossy.cs, and StandardizeFiles.cs.

10.235.2 Constructor & Destructor Documentation

10.235.2.1 Pixmap()

```
gdcm::Pixmap::Pixmap ( )
```

10.235.2.2 ~Pixmap()

```
gdcm::Pixmap::~Pixmap ( )  [override]
```

10.235.3 Member Function Documentation

10.235.3.1 AreOverlaysInPixelData()

```
bool gdcm::Pixmap::AreOverlaysInPixelData ( ) const  [override], [virtual]
```

returns if Overlays are stored in the unused bit of the pixel data:

Reimplemented from gdcm::Bitmap.

10.235.3.2 GetCurve() [1/2]

```
Curve & gdcm::Pixmap::GetCurve (
            size_t i = 0 )  [inline]
```

Curve: group 50xx.

10.235.3.3 GetCurve() [2/2]

```
const Curve & gdcm::Pixmap::GetCurve (
            size_t i = 0 ) const  [inline]
```

10.235.3.4 GetIconImage() [1/2]

```
IconImage & gdcm::Pixmap::GetIconImage ( )  [inline]
```

10.235.3.5 GetIconImage() [2/2]

```
const IconImage & gdcm::Pixmap::GetIconImage ( ) const  [inline]
```

Set/Get Icon Image.

10.235.3.6 GetNumberOfCurves()

```
size_t gdcm::Pixmap::GetNumberOfCurves ( ) const  [inline]
```

10.235.3.7 GetNumberOfOverlays()

```
size_t gdcm::Pixmap::GetNumberOfOverlays ( ) const  [inline]
```

10.235.3.8 GetOverlay() [1/2]

```
Overlay & gdcm::Pixmap::GetOverlay (
            size_t i = 0 )  [inline]
```

Overlay: group 60xx.

10.235.3.9 GetOverlay() [2/2]

```
const Overlay & gdcm::Pixmap::GetOverlay (
            size_t i = 0 ) const  [inline]
```

10.235.3.10 Print()

```
void gdcm::Pixmap::Print (
            std::ostream &  ) const  [override], [virtual]
```

Reimplemented from gdcm::Bitmap.

10.235.3.11 RemoveOverlay()

```
void gdcm::Pixmap::RemoveOverlay (
            size_t i )  [inline]
```

10.235.3.12 SetIconImage()

```
void gdcm::Pixmap::SetIconImage (
            IconImage const & ii )  [inline]
```

10.235.3.13 SetNumberOfCurves()

```
void gdcm::Pixmap::SetNumberOfCurves (
            size_t n )  [inline]
```

10.235.3.14 SetNumberOfOverlays()

```
void gdcm::Pixmap::SetNumberOfOverlays (
            size_t n )  [inline]
```

10.235.3.15 UnusedBitsPresentInPixelData()

```
bool gdcm::Pixmap::UnusedBitsPresentInPixelData ( ) const  [override], [virtual]
```

returns if there are unused bits in the pixel data

Reimplemented from gdcm::Bitmap.

10.235.4 Member Data Documentation

10.235.4.1 Curves

```
std::vector<Curve> gdcm::Pixmap::Curves  [protected]
```

10.235.4.2 Icon

```
SmartPointer<IconImage> gdcm::Pixmap::Icon  [protected]
```

10.235.4.3 Overlays

```
std::vector<Overlay> gdcm::Pixmap::Overlays  [protected]
```

The documentation for this class was generated from the following file:

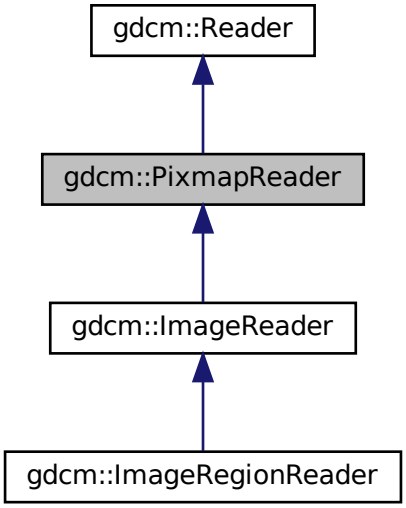
- gdcmPixmap.h

10.236 gdcm::PixmapReader Class Reference

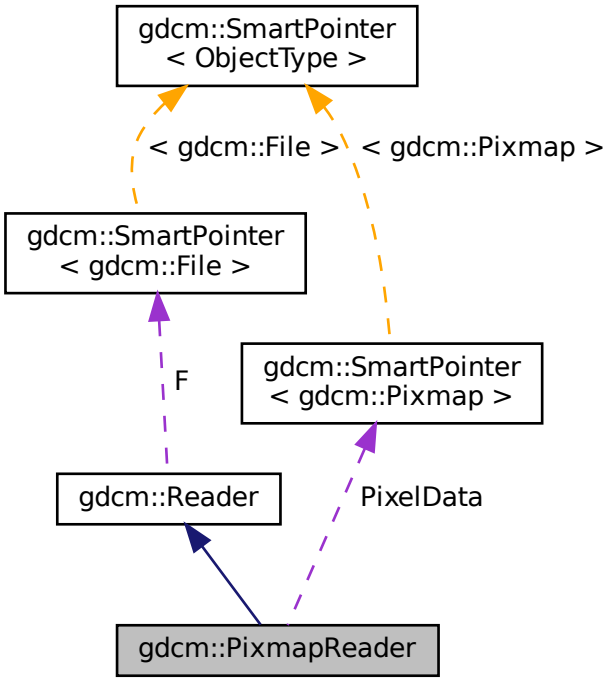
PixmapReader.

`#include <gdcmPixmapReader.h>`

Inheritance diagram for gdcm::PixmapReader:



Collaboration diagram for gdcm::PixmapReader:



Public Member Functions

- PixmapReader ()
- ~PixmapReader () override
- Pixmap & GetPixmap ()
- const Pixmap & GetPixmap () const

  Return the read image (need to call Read() first)
- bool Read () override

Protected Member Functions

- virtual bool ReadACRNEMAImage ()
- virtual bool ReadImage (MediaStorage const &ms)
- bool ReadImageInternal (MediaStorage const &ms, bool handlepixeldata=true)

Protected Attributes

- SmartPointer< Pixmap > PixelData

10.236.1 Detailed Description

PixmapReader.

Note

its role is to convert the DICOM DataSet into a Pixmap representation By default it is also loading the lookup table and overlay when found as they impact the rendering or the image

See PS 3.3-2008, Table C.7-11b IMAGE PIXEL MACRO ATTRIBUTES for the list of attribute that belong to what gdcm calls a 'Pixmap'

Warning

the API ReadUpToTag and ReadSelectedTag

See also

Pixmap

Examples

StandardizeFiles.cs.

10.236.2 Constructor & Destructor Documentation

10.236.2.1 PixmapReader()

```
gdcm::PixmapReader::PixmapReader ( )
```

10.236.2.2 ~PixmapReader()

```
gdcm::PixmapReader::~PixmapReader ( )  [override]
```

10.236.3 Member Function Documentation

10.236.3.1 GetPixmap() [1/2]

```
Pixmap & gdcm::PixmapReader::GetPixmap ( )
```

10.236.3.2 GetPixmap() [2/2]

```
const Pixmap & gdcm::PixmapReader::GetPixmap ( ) const
```

Return the read image (need to call Read() first)

Examples

StandardizeFiles.cs.

10.236.3.3 Read()

```
bool gdcm::PixmapReader::Read ( )  [override], [virtual]
```

Read the DICOM image. There are two reason for failure:

1. The input filename is not DICOM
2. The input DICOM file does not contains an Pixmap.

Reimplemented from gdcm::Reader.

Examples

StandardizeFiles.cs.

10.236.3.4 ReadACRNEMAImage()

```
virtual bool gdcm::PixmapReader::ReadACRNEMAImage ( )  [protected], [virtual]
```

Reimplemented in gdcm::ImageReader.

10.236.3.5 ReadImage()

```
virtual bool gdcm::PixmapReader::ReadImage (
            MediaStorage const & ms )  [protected], [virtual]
```

Reimplemented in gdcm::ImageReader.

10.236.3.6 ReadImageInternal()

```
bool gdcm::PixmapReader::ReadImageInternal (
            MediaStorage const & ms,
            bool handlepixeldata = true )  [protected]
```

10.236.4 Member Data Documentation

10.236.4.1 PixelData

```
SmartPointer<Pixmap> gdcm::PixmapReader::PixelData  [protected]
```

The documentation for this class was generated from the following file:

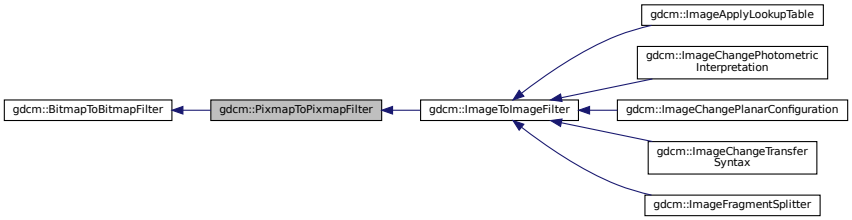
- gdcmPixmapReader.h

10.237 gdcm::PixmapToPixmapFilter Class Reference

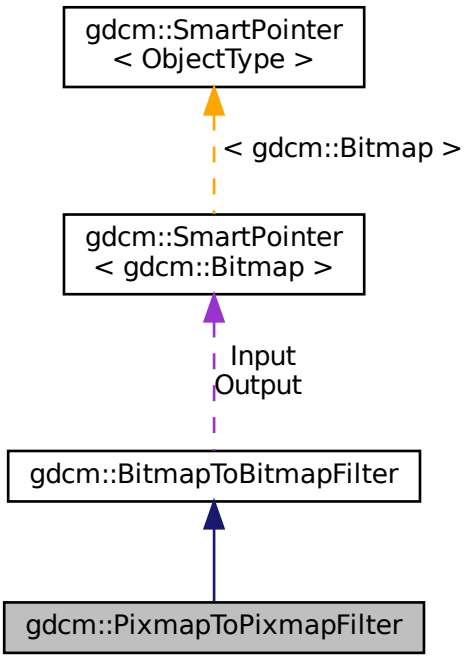
PixmapToPixmapFilter class.

```
#include <gdcmPixmapToPixmapFilter.h>
```

Inheritance diagram for gdcm::PixmapToPixmapFilter:



Collaboration diagram for gdcm::PixmapToPixmapFilter:



Public Member Functions

- PixmapToPixmapFilter ()
- ~PixmapToPixmapFilter ()=default
- Pixmap & GetInput ()
- const Pixmap & GetOutput () const

  Get Output image.
- const Pixmap & GetOutputAsPixmap () const

Additional Inherited Members

10.237.1 Detailed Description

PixmapToPixmapFilter class.

Super class for all filter taking an image and producing an output image

Examples

StandardizeFiles.cs.

10.237.2 Constructor & Destructor Documentation

10.237.2.1 PixmapToPixmapFilter()

```
gdcm::PixmapToPixmapFilter::PixmapToPixmapFilter ( )
```

10.237.2.2 ~PixmapToPixmapFilter()

```
gdcm::PixmapToPixmapFilter::~PixmapToPixmapFilter ( )  [default]
```

10.237.3 Member Function Documentation

10.237.3.1 GetInput()

```
Pixmap & gdcm::PixmapToPixmapFilter::GetInput ( )
```

10.237.3.2 GetOutput()

```
const Pixmap & gdcm::PixmapToPixmapFilter::GetOutput ( ) const
```

Get Output image.

10.237.3.3 GetOutputAsPixmap()

```
const Pixmap & gdcm::PixmapToPixmapFilter::GetOutputAsPixmap ( ) const
```

The documentation for this class was generated from the following file:

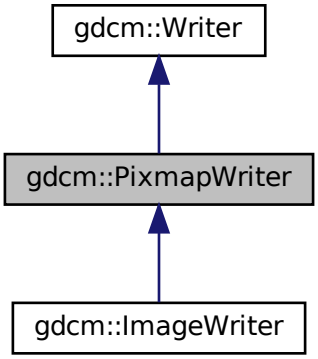
- gdcmPixmapToPixmapFilter.h

10.238 gdcm::PixmapWriter Class Reference

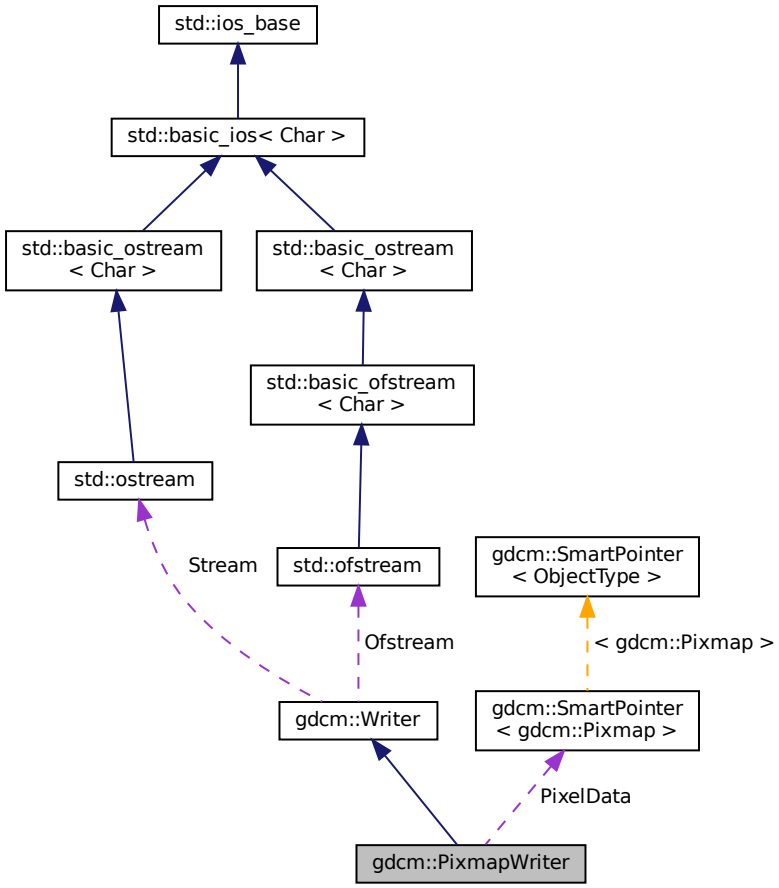
PixmapWriter.

`#include <gdcmPixmapWriter.h>`

Inheritance diagram for gdcm::PixmapWriter:



Collaboration diagram for gdcm::PixmapWriter:



Public Member Functions

- PixmapWriter ()
- ~PixmapWriter () override
- virtual Pixmap & GetImage ()
- virtual const Pixmap & GetImage () const
- Pixmap & GetPixmap ()
- const Pixmap & GetPixmap () const
- virtual void SetImage (Pixmap const &img)
- void SetPixmap (Pixmap const &img)
- bool Write () override

  Write.

Protected Member Functions

- void DoIconImage (DataSet &ds, Pixmap const &image)
- bool PrepareWrite (MediaStorage const &refms)

Protected Attributes

- SmartPointer< Pixmap > PixelData

10.238.1 Detailed Description

PixmapWriter.

This class will takes two inputs:

1. The DICOM DataSet
2. The Image input It will override any info from the Image over the DataSet.

For instance when one read in a lossy compressed image and write out as unencapsulated (ie implicitly lossless) then some attribute are definitely needed to mark this dataset as Lossy (typically 0028,2114)

Examples

StandardizeFiles.cs.

10.238.2 Constructor & Destructor Documentation

10.238.2.1 PixmapWriter()

```
gdcm::PixmapWriter::PixmapWriter ( )
```

10.238.2.2 ~PixmapWriter()

```
gdcm::PixmapWriter::~PixmapWriter ( )  [override]
```

10.238.3 Member Function Documentation

10.238.3.1 DoIconImage()

```
void gdcm::PixmapWriter::DoIconImage (
            DataSet & ds,
            Pixmap const & image ) [protected]
```

10.238.3.2 GetImage() [1/2]

```
virtual Pixmap & gdcm::PixmapWriter::GetImage ( ) [inline], [virtual]
```

Reimplemented in gdcm::ImageWriter.

10.238.3.3 GetImage() [2/2]

```
virtual const Pixmap & gdcm::PixmapWriter::GetImage ( ) const [inline], [virtual]
```

Set/Get Pixmap to be written It will overwrite anything Pixmap infos found in DataSet (see parent class to see how to pass dataset)

Reimplemented in gdcm::ImageWriter.

10.238.3.4 GetPixmap() [1/2]

```
Pixmap & gdcm::PixmapWriter::GetPixmap ( ) [inline]
```

10.238.3.5 GetPixmap() [2/2]

```
const Pixmap & gdcm::PixmapWriter::GetPixmap ( ) const [inline]
```

10.238.3.6 PrepareWrite()

```
bool gdcm::PixmapWriter::PrepareWrite (
            MediaStorage const & refms ) [protected]
```

10.238.3.7 SetImage()

```
virtual void gdcm::PixmapWriter::SetImage (
            Pixmap const & img )  [virtual]
```

Examples

BasicImageAnonymizer.cs, CompressImage.cxx, CompressLossyJPEG.cs, DecompressImage.cs, GenFakeImage.cxx, GetSubSequenceData.cxx, HelloVizWorld.cxx, MergeTwoFiles.cxx, MpegVideoInfo.cs, and TemplateEmptyImage.cxx.

10.238.3.8 SetPixmap()

```
void gdcm::PixmapWriter::SetPixmap (
            Pixmap const & img )
```

Examples

StandardizeFiles.cs.

10.238.3.9 Write()

```
bool gdcm::PixmapWriter::Write ( )  [override], [virtual]
```

Write.

Reimplemented from gdcm::Writer.

Examples

StandardizeFiles.cs.

10.238.4 Member Data Documentation

10.238.4.1 PixelData

```
SmartPointer<Pixmap> gdcm::PixmapWriter::PixelData  [protected]
```

The documentation for this class was generated from the following file:

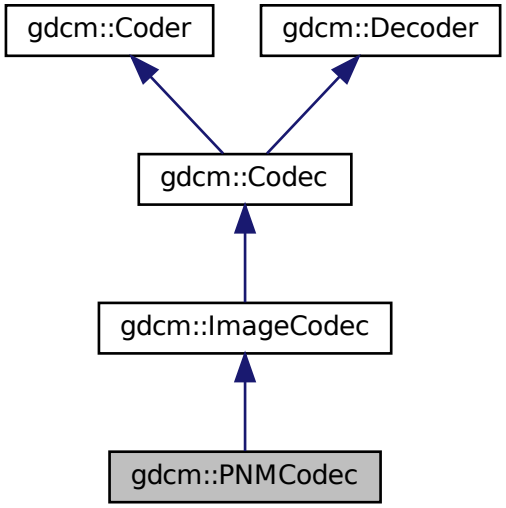
- gdcmPixmapWriter.h

10.239 gdcm::PNMCodec Class Reference

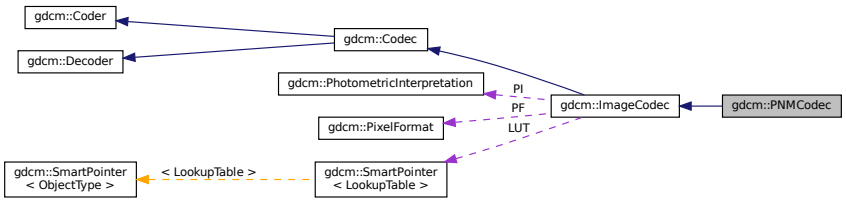
Class to do PNM.

`#include <gdcmPNMCodec.h>`

Inheritance diagram for gdcm::PNMCodec:



Collaboration diagram for gdcm::PNMCodec:



Public Member Functions

- PNMCodec ()
- ~PNMCodec () override
- bool CanCode (TransferSyntax const &ts) const override

Return whether this coder support this transfer syntax (can code it)

- bool CanDecode (TransferSyntax const &ts) const override

  Return whether this decoder support this transfer syntax (can decode it)
- ImageCodec * Clone () const override
- unsigned long GetBufferLength () const
- bool GetHeaderInfo (std::istream &is, TransferSyntax &ts) override
- bool Read (const char *filename, DataElement &out) const
- void SetBufferLength (unsigned long l)
- bool Write (const char *filename, const DataElement &out) const

Additional Inherited Members

10.239.1 Detailed Description

Class to do PNM.

PNM is the Portable anymap file format. The main web page can be found at: `http://netpbm.sourceforge.net/`

Note

Only support P5 & P6 PNM file (binary grayscale and binary rgb)

Examples

ExtractIconFromFile.cxx.

10.239.2 Constructor & Destructor Documentation

10.239.2.1 PNMCodec()

```
gdcm::PNMCodec::PNMCodec ( )
```

10.239.2.2 ~PNMCodec()

```
gdcm::PNMCodec::~PNMCodec ( )  [override]
```

10.239.3 Member Function Documentation

10.239.3.1 CanCode()

```
bool gdcm::PNMCodec::CanCode (
            TransferSyntax const &  ) const  [override], [virtual]
```

Return whether this coder support this transfer syntax (can code it)

Reimplemented from gdcm::ImageCodec.

10.239.3.2 CanDecode()

```
bool gdcm::PNMCodec::CanDecode (
            TransferSyntax const &  ) const  [override], [virtual]
```

Return whether this decoder support this transfer syntax (can decode it)

Reimplemented from gdcm::ImageCodec.

10.239.3.3 Clone()

```
ImageCodec * gdcm::PNMCodec::Clone ( ) const  [override], [virtual]
```

Implements gdcm::ImageCodec.

10.239.3.4 GetBufferLength()

```
unsigned long gdcm::PNMCodec::GetBufferLength ( ) const  [inline]
```

10.239.3.5 GetHeaderInfo()

```
bool gdcm::PNMCodec::GetHeaderInfo (
            std::istream & is,
            TransferSyntax & ts )  [override], [virtual]
```

Reimplemented from gdcm::ImageCodec.

10.239.3.6 Read()

```
bool gdcm::PNMCodec::Read (
             const char * filename,
             DataElement & out ) const
```

10.239.3.7 SetBufferLength()

```
void gdcm::PNMCodec::SetBufferLength (
             unsigned long l )  [inline]
```

10.239.3.8 Write()

```
bool gdcm::PNMCodec::Write (
             const char * filename,
             const DataElement & out ) const
```

Examples

ExtractIconFromFile.cxx.

The documentation for this class was generated from the following file:

- gdcmPNMCodec.h

10.240 gdcm::Preamble Class Reference

DICOM Preamble (Part 10)

```
#include <gdcmPreamble.h>
```

Public Member Functions

- Preamble ()
- Preamble (Preamble const &)
- ~Preamble ()
- void Clear ()

  Clear.
- void Create ()
- const char * GetInternal () const

  Get internal pointer to preamble.
- VL GetLength () const

  Return size of Preamble.
- bool IsEmpty () const

  Check if Preamble is empty.
- Preamble & operator= (Preamble const &)
- void Print (std::ostream &os) const

  Print Preamble.
- std::istream & Read (std::istream &is)

  Read Preamble.
- void Remove ()
- void Valid ()

  Set Preamble to the default one.
- std::ostream const & Write (std::ostream &os) const

  Write Preamble.

Protected Member Functions

- bool IsValid () const

Friends

- std::ostream & operator<< (std::ostream &_os, const Preamble &_val)

10.240.1 Detailed Description

DICOM Preamble (Part 10)

10.240.2 Constructor & Destructor Documentation

10.240.2.1 Preamble() [1/2]

```
gdcm::Preamble::Preamble ( )
```

10.240.2.2 ~Preamble()

```
gdcm::Preamble::~Preamble ( )
```

10.240.2.3 Preamble() [2/2]

```
gdcm::Preamble::Preamble (
            Preamble const &  )  [inline]
```

10.240.3 Member Function Documentation

10.240.3.1 Clear()

```
void gdcm::Preamble::Clear ( )
```

Clear.

10.240.3.2 Create()

```
void gdcm::Preamble::Create ( )
```

10.240.3.3 GetInternal()

```
const char * gdcm::Preamble::GetInternal ( ) const  [inline]
```

Get internal pointer to preamble.

10.240.3.4 GetLength()

```
VL gdcm::Preamble::GetLength ( ) const  [inline]
```

Return size of Preamble.

10.240.3.5 IsEmpty()

```
bool gdcm::Preamble::IsEmpty ( ) const  [inline]
```

Check if Preamble is empty.

10.240.3.6 IsValid()

```
bool gdcm::Preamble::IsValid ( ) const  [inline], [protected]
```

10.240.3.7 operator=()

```
Preamble & gdcm::Preamble::operator= (
            Preamble const &  )  [inline]
```

10.240.3.8 Print()

```
void gdcm::Preamble::Print (
            std::ostream & os ) const
```

Print Preamble.

10.240.3.9 Read()

```
std::istream & gdcm::Preamble::Read (
            std::istream & is )
```

Read Preamble.

10.240.3.10 Remove()

```
void gdcm::Preamble::Remove ( )
```

10.240.3.11 Valid()

```
void gdcm::Preamble::Valid ( )
```

Set Preamble to the default one.

10.240.3.12 Write()

```
std::ostream const & gdcm::Preamble::Write (
            std::ostream & os ) const
```

Write Preamble.

10.240.4 Friends And Related Function Documentation

10.240.4.1 operator<<

```
std::ostream & operator<< (
            std::ostream & _os,
            const Preamble & _val )  [friend]
```

The documentation for this class was generated from the following file:

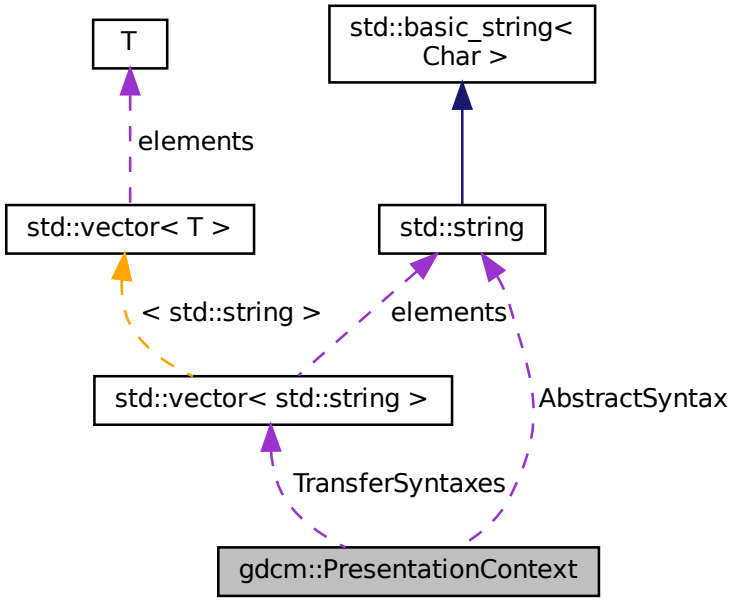
- gdcmPreamble.h

10.241 gdcm::PresentationContext Class Reference

PresentationContext.

`#include <gdcmPresentationContext.h>`

Collaboration diagram for gdcm::PresentationContext:



Public Types

- typedef TransferSyntaxArrayType::size_type SizeType
- typedef std::vector< std::string > TransferSyntaxArrayType

Public Member Functions

- PresentationContext ()
- PresentationContext (UIDs::TSName asname, UIDs::TSName tsname=UIDs::ImplicitVRLittleEndianDefaultTransferSyntaxforDICOM
- void AddTransferSyntax (const char *tsstr)
- const char * GetAbstractSyntax () const
- SizeType GetNumberOfTransferSyntaxes () const
- uint8_t GetPresentationContextID () const
- const char * GetTransferSyntax (SizeType i) const
- bool operator== (const PresentationContext &pc) const
- void Print (std::ostream &os) const
- void SetAbstractSyntax (const char *absyn)
- void SetPresentationContextID (uint8_t id)

Protected Attributes

- std::string AbstractSyntax
- uint8_t ID
- std::vector< std::string > TransferSyntaxes

10.241.1 Detailed Description

PresentationContext.

See also

PresentationContextAC PresentationContextRQ

10.241.2 Member Typedef Documentation

10.241.2.1 SizeType

```
typedef TransferSyntaxArrayType::size_type gdcm::PresentationContext::SizeType
```

10.241.2.2 TransferSyntaxArrayType

```
typedef std::vector<std::string> gdcm::PresentationContext::TransferSyntaxArrayType
```

10.241.3 Constructor & Destructor Documentation

10.241.3.1 PresentationContext() [1/2]

```
gdcm::PresentationContext::PresentationContext ( )
```

10.241.3.2 PresentationContext() [2/2]

```
gdcm::PresentationContext::PresentationContext (
            UIDs::TSName asname,
            UIDs::TSName tsname = UIDs::ImplicitVRLittleEndianDefaultTransferSyntaxforDICOM )
```

Initialize Presentation Context with AbstractSyntax set to asname and with a single TransferSyntax set to tsname (default to Implicit VR LittleEndian when not specified).

10.241.4 Member Function Documentation

10.241.4.1 AddTransferSyntax()

```
void gdcm::PresentationContext::AddTransferSyntax (
            const char * tsstr )
```

10.241.4.2 GetAbstractSyntax()

```
const char * gdcm::PresentationContext::GetAbstractSyntax ( ) const  [inline]
```

10.241.4.3 GetNumberOfTransferSyntaxes()

```
SizeType gdcm::PresentationContext::GetNumberOfTransferSyntaxes ( ) const  [inline]
```

10.241.4.4 GetPresentationContextID()

```
uint8_t gdcm::PresentationContext::GetPresentationContextID ( ) const
```

10.241.4.5 GetTransferSyntax()

```
const char * gdcm::PresentationContext::GetTransferSyntax (
            SizeType i ) const  [inline]
```

10.241.4.6 operator==()

```
bool gdcm::PresentationContext::operator== (
            const PresentationContext & pc ) const  [inline]
```

References AbstractSyntax, and TransferSyntaxes.

10.241.4.7 Print()

```
void gdcm::PresentationContext::Print (
            std::ostream & os ) const
```

10.241.4.8 SetAbstractSyntax()

```
void gdcm::PresentationContext::SetAbstractSyntax (
            const char * absyn )  [inline]
```

10.241.4.9 SetPresentationContextID()

```
void gdcm::PresentationContext::SetPresentationContextID (
            uint8_t id )
```

10.241.5 Member Data Documentation

10.241.5.1 AbstractSyntax

```
std::string gdcm::PresentationContext::AbstractSyntax  [protected]
```

Referenced by operator==().

10.241.5.2 ID

```
uint8_t gdcm::PresentationContext::ID  [protected]
```

10.241.5.3 TransferSyntaxes

```
std::vector<std::string> gdcm::PresentationContext::TransferSyntaxes  [protected]
```

Referenced by operator==().

The documentation for this class was generated from the following file:

- gdcmPresentationContext.h

10.242 gdcm::network::PresentationContextAC Class Reference

PresentationContextAC.

```
#include <gdcmPresentationContextAC.h>
```

Public Member Functions

- PresentationContextAC ()
- uint8_t GetPresentationContextID () const
- uint8_t GetReason () const
- TransferSyntaxSub const & GetTransferSyntax () const
- void Print (std::ostream &os) const
- std::istream & Read (std::istream &is)
- void SetPresentationContextID (uint8_t id)
- void SetReason (uint8_t r)
- void SetTransferSyntax (TransferSyntaxSub const &ts)
- size_t Size () const
- const std::ostream & Write (std::ostream &os) const

10.242.1 Detailed Description

PresentationContextAC.

Table 9-18 PRESENTATION CONTEXT ITEM FIELDS

See also

PresentationContext

10.242.2 Constructor & Destructor Documentation

10.242.2.1 PresentationContextAC()

```
gdcm::network::PresentationContextAC::PresentationContextAC ( )
```

10.242.3 Member Function Documentation

10.242.3.1 GetPresentationContextID()

```
uint8_t gdcm::network::PresentationContextAC::GetPresentationContextID ( ) const  [inline]
```

10.242.3.2 GetReason()

```
uint8_t gdcm::network::PresentationContextAC::GetReason ( ) const  [inline]
```

10.242.3.3 GetTransferSyntax()

```
TransferSyntaxSub const & gdcm::network::PresentationContextAC::GetTransferSyntax ( ) const  [inline]
```

10.242.3.4 Print()

```
void gdcm::network::PresentationContextAC::Print (
            std::ostream & os ) const
```

10.242.3.5 Read()

```
std::istream & gdcm::network::PresentationContextAC::Read (
            std::istream & is )
```

10.242.3.6 SetPresentationContextID()

```
void gdcm::network::PresentationContextAC::SetPresentationContextID (
            uint8_t id )
```

10.242.3.7 SetReason()

```
void gdcm::network::PresentationContextAC::SetReason (
            uint8_t r )  [inline]
```

10.242.3.8 SetTransferSyntax()

```
void gdcm::network::PresentationContextAC::SetTransferSyntax (
            TransferSyntaxSub const & ts )
```

10.242.3.9 Size()

```
size_t gdcm::network::PresentationContextAC::Size ( ) const
```

10.242.3.10 Write()

```
const std::ostream & gdcm::network::PresentationContextAC::Write (
            std::ostream & os ) const
```

The documentation for this class was generated from the following file:

- gdcmPresentationContextAC.h

10.243 gdcm::PresentationContextGenerator Class Reference

PresentationContextGenerator.

```
#include <gdcmPresentationContextGenerator.h>
```

Public Types

- typedef std::vector< PresentationContext > PresentationContextArrayType
- typedef PresentationContextArrayType::size_type SizeType

Public Member Functions

- PresentationContextGenerator ()
- bool AddFromFile (const File &file)
- bool GenerateFromFilenames (const Directory::FilenamesType &files)
- bool GenerateFromUID (UIDs::TSName asname)

  Generate the PresentationContext array from a UID (eg. VerificationSOPClass)
- PresentationContextArrayType const & GetPresentationContexts ()
- void SetDefaultTransferSyntax (const TransferSyntax &ts)

  Not implemented for now. GDCM internally uses Implicit Little Endian.
- void SetMergeModeToAbstractSyntax ()
- void SetMergeModeToTransferSyntax ()

Protected Member Functions

- bool AddPresentationContext (const char *absyn, const char *ts)
- const char * GetDefaultTransferSyntax () const

10.243.1 Detailed Description

PresentationContextGenerator.

This class is responsible for generating the proper PresentationContext that will be used in subsequent operation during a DICOM Query/Retrieve association. The step of the association is very sensible as special care need to be taken to explicitly define what instance are going to be send and how they are encoded.

For example a PresentationContext will express that negotiation requires that CT Image Storage are send using JPEG Lossless, while US Image Storage are sent using RLE Transfer Syntax.

Two very different API are exposed one which will always default to little endian transfer syntax see GenerateFromUID() This API is used for C-ECHO, C-FIND and C-MOVE (SCU). Another API: GenerateFromFilenames() is used for C-STORE (SCU) as it will loop over all filenames argument to detect the actual encoding. and therefore find the proper encoding to be used.

Two modes are available. The default mode (SetMergeModeToAbstractSyntax) append PresentationContext (one AbstractSyntax and one TransferSyntax), as long a they are different. Eg MR Image Storage/JPEG2000 and MR Image Storage/JPEGLossless would be considered different. the other mode SetMergeModeToTransferSyntax merge any new TransferSyntax to the already existing PresentationContext in order to re-use the same AbstractSyntax.

See also

PresentationContext

Examples

CStoreQtProgress.cxx.

10.243.2 Member Typedef Documentation

10.243.2.1 PresentationContextArrayType

```
typedef std::vector<PresentationContext> gdcm::PresentationContextGenerator::PresentationContextArrayType
```

10.243.2.2 SizeType

```
typedef PresentationContextArrayType::size_type gdcm::PresentationContextGenerator::SizeType
```

10.243.3 Constructor & Destructor Documentation

10.243.3.1 PresentationContextGenerator()

```
gdcm::PresentationContextGenerator::PresentationContextGenerator ( )
```

10.243.4 Member Function Documentation

10.243.4.1 AddFromFile()

```
bool gdcm::PresentationContextGenerator::AddFromFile (
            const File & file )
```

Add a single PresentationContext from a single File. Call multiple times when dealing with multiple files.

10.243.4.2 AddPresentationContext()

```
bool gdcm::PresentationContextGenerator::AddPresentationContext (
            const char * absyn,
            const char * ts )  [protected]
```

10.243.4.3 GenerateFromFilenames()

```
bool gdcm::PresentationContextGenerator::GenerateFromFilenames (
            const Directory::FilenamesType & files )
```

Generate the PresentationContext array from a File-Set. File specified needs to be valid DICOM files. Used for C-STORE operations

Examples

CStoreQtProgress.cxx.

10.243.4.4 GenerateFromUID()

```
bool gdcm::PresentationContextGenerator::GenerateFromUID (
            UIDs::TSName asname )
```

Generate the PresentationContext array from a UID (eg. VerificationSOPClass)

10.243.4.5 GetDefaultTransferSyntax()

```
const char * gdcm::PresentationContextGenerator::GetDefaultTransferSyntax ( ) const  [protected]
```

10.243.4.6 GetPresentationContexts()

```
PresentationContextArrayType const & gdcm::PresentationContextGenerator::GetPresentationContexts (
) [inline]
```

Examples

CStoreQtProgress.cxx.

10.243.4.7 SetDefaultTransferSyntax()

```
void gdcm::PresentationContextGenerator::SetDefaultTransferSyntax (
            const TransferSyntax & ts )
```

Not implemented for now. GDCM internally uses Implicit Little Endian.

10.243.4.8 SetMergeModeToAbstractSyntax()

```
void gdcm::PresentationContextGenerator::SetMergeModeToAbstractSyntax ( )
```

10.243.4.9 SetMergeModeToTransferSyntax()

```
void gdcm::PresentationContextGenerator::SetMergeModeToTransferSyntax ( )
```

The documentation for this class was generated from the following file:

- gdcmPresentationContextGenerator.h

10.244 gdcm::network::PresentationContextRQ Class Reference

PresentationContextRQ.

```
#include <gdcmPresentationContextRQ.h>
```

Public Types

- typedef std::vector< TransferSyntaxSub >::size_type SizeType

Public Member Functions

- PresentationContextRQ ()
- PresentationContextRQ (const PresentationContext &pc)
- PresentationContextRQ (UIDs::TSName asname, UIDs::TSName tsname=UIDs::ImplicitVRLittleEndianDefaultTransferSyntaxforDI
- void AddTransferSyntax (TransferSyntaxSub const &ts)
- AbstractSyntax & GetAbstractSyntax ()
- AbstractSyntax const & GetAbstractSyntax () const
- SizeType GetNumberOfTransferSyntaxes () const
- uint8_t GetPresentationContextID () const
- TransferSyntaxSub & GetTransferSyntax (SizeType i)
- TransferSyntaxSub const & GetTransferSyntax (SizeType i) const
- std::vector< TransferSyntaxSub > const & GetTransferSyntaxes () const
- bool operator== (const PresentationContextRQ &pc) const
- void Print (std::ostream &os) const
- std::istream & Read (std::istream &is)
- void SetAbstractSyntax (AbstractSyntax const &absyn)
- void SetPresentationContextID (uint8_t id)
- size_t Size () const
- const std::ostream & Write (std::ostream &os) const

10.244.1 Detailed Description

PresentationContextRQ.

Table 9-13 PRESENTATION CONTEXT ITEM FIELDS

See also

PresentationContextAC

10.244.2 Member Typedef Documentation

10.244.2.1 SizeType

```
typedef std::vector<TransferSyntaxSub>::size_type gdcm::network::PresentationContextRQ::SizeType
```

10.244.3 Constructor & Destructor Documentation

10.244.3.1 PresentationContextRQ() [1/3]

```
gdcm::network::PresentationContextRQ::PresentationContextRQ ( )
```

10.244.3.2 PresentationContextRQ() [2/3]

```
gdcm::network::PresentationContextRQ::PresentationContextRQ (
            UIDs::TSName asname,
            UIDs::TSName tsname = UIDs::ImplicitVRLittleEndianDefaultTransferSyntaxforDICOM )
```

Initialize Presentation Context with AbstractSyntax set to asname and with a single TransferSyntax set to tsname (dfault to Implicit VR LittleEndian when not specified).

10.244.3.3 PresentationContextRQ() [3/3]

```
gdcm::network::PresentationContextRQ::PresentationContextRQ (
            const PresentationContext & pc )
```

10.244.4 Member Function Documentation

10.244.4.1 AddTransferSyntax()

```
void gdcm::network::PresentationContextRQ::AddTransferSyntax (
            TransferSyntaxSub const & ts )
```

10.244.4.2 GetAbstractSyntax() [1/2]

```
AbstractSyntax & gdcm::network::PresentationContextRQ::GetAbstractSyntax ( )  [inline]
```

10.244.4.3 GetAbstractSyntax() [2/2]

```
AbstractSyntax const & gdcm::network::PresentationContextRQ::GetAbstractSyntax ( ) const  [inline]
```

10.244.4.4 GetNumberOfTransferSyntaxes()

```
SizeType gdcm::network::PresentationContextRQ::GetNumberOfTransferSyntaxes ( ) const  [inline]
```

10.244.4.5 GetPresentationContextID()

```
uint8_t gdcm::network::PresentationContextRQ::GetPresentationContextID ( ) const
```

10.244.4.6 GetTransferSyntax() [1/2]

```
TransferSyntaxSub & gdcm::network::PresentationContextRQ::GetTransferSyntax (
            SizeType i )  [inline]
```

10.244.4.7 GetTransferSyntax() [2/2]

```
TransferSyntaxSub const & gdcm::network::PresentationContextRQ::GetTransferSyntax (
            SizeType i ) const  [inline]
```

10.244.4.8 GetTransferSyntaxes()

```
std::vector< TransferSyntaxSub > const & gdcm::network::PresentationContextRQ::GetTransfer↩
Syntaxes ( ) const  [inline]
```

10.244.4.9 operator==()

```
bool gdcm::network::PresentationContextRQ::operator== (
            const PresentationContextRQ & pc ) const  [inline]
```

10.244.4.10 Print()

```
void gdcm::network::PresentationContextRQ::Print (
            std::ostream & os ) const
```

10.244.4.11 Read()

```
std::istream & gdcm::network::PresentationContextRQ::Read (
            std::istream & is )
```

10.244.4.12 SetAbstractSyntax()

```
void gdcm::network::PresentationContextRQ::SetAbstractSyntax (
            AbstractSyntax const & absyn )
```

10.244.4.13 SetPresentationContextID()

```
void gdcm::network::PresentationContextRQ::SetPresentationContextID (
            uint8_t id )
```

10.244.4.14 Size()

```
size_t gdcm::network::PresentationContextRQ::Size ( ) const
```

10.244.4.15 Write()

```
const std::ostream & gdcm::network::PresentationContextRQ::Write (
            std::ostream & os ) const
```

The documentation for this class was generated from the following file:

- gdcmPresentationContextRQ.h

10.245 gdcm::network::PresentationDataValue Class Reference

PresentationDataValue.

`#include <gdcmPresentationDataValue.h>`

Public Member Functions

- PresentationDataValue ()
- const std::string & GetBlob () const
- bool GetIsCommand () const
- bool GetIsLastFragment () const
- uint8_t GetMessageHeader () const
- uint8_t GetPresentationContextID () const
- void Print (std::ostream &os) const
- std::istream & Read (std::istream &is)
- std::istream & ReadInto (std::istream &is, std::ostream &os)
- void SetBlob (const std::string &partialblob)
- void SetCommand (bool inCommand)
- void SetDataSet (const DataSet &ds)
- void SetLastFragment (bool inLast)
- void SetMessageHeader (uint8_t messageheader)
- void SetPresentationContextID (uint8_t id)
- size_t Size () const
- const std::ostream & Write (std::ostream &os) const

Static Public Member Functions

- static DataSet ConcatenatePDVBlobs (const std::vector< PresentationDataValue > &inPDVs)
- static DataSet ConcatenatePDVBlobsAsExplicit (const std::vector< PresentationDataValue > &inPDVs)

10.245.1 Detailed Description

PresentationDataValue.

Table 9-23 PRESENTATION-DATA-VALUE ITEM FIELDS

10.245.2 Constructor & Destructor Documentation

10.245.2.1 PresentationDataValue()

```
gdcm::network::PresentationDataValue::PresentationDataValue ( )
```

10.245.3 Member Function Documentation

10.245.3.1 ConcatenatePDVBlobs()

```
static DataSet gdcm::network::PresentationDataValue::ConcatenatePDVBlobs (
            const std::vector< PresentationDataValue > & inPDVs )  [static]
```

Warning

DataSet will be read as Implicit Little Endian TS

10.245.3.2 ConcatenatePDVBlobsAsExplicit()

```
static DataSet gdcm::network::PresentationDataValue::ConcatenatePDVBlobsAsExplicit (
            const std::vector< PresentationDataValue > & inPDVs )  [static]
```

10.245.3.3 GetBlob()

```
const std::string & gdcm::network::PresentationDataValue::GetBlob ( ) const
```

10.245.3.4 GetIsCommand()

```
bool gdcm::network::PresentationDataValue::GetIsCommand ( ) const
```

10.245.3.5 GetIsLastFragment()

```
bool gdcm::network::PresentationDataValue::GetIsLastFragment ( ) const
```

10.245.3.6 GetMessageHeader()

```
uint8_t gdcm::network::PresentationDataValue::GetMessageHeader ( ) const  [inline]
```

10.245.3.7 GetPresentationContextID()

```
uint8_t gdcm::network::PresentationDataValue::GetPresentationContextID ( ) const  [inline]
```

10.245.3.8 Print()

```
void gdcm::network::PresentationDataValue::Print (
             std::ostream & os ) const
```

10.245.3.9 Read()

```
std::istream & gdcm::network::PresentationDataValue::Read (
             std::istream & is )
```

10.245.3.10 ReadInto()

```
std::istream & gdcm::network::PresentationDataValue::ReadInto (
            std::istream & is,
            std::ostream & os )
```

10.245.3.11 SetBlob()

```
void gdcm::network::PresentationDataValue::SetBlob (
            const std::string & partialblob )
```

10.245.3.12 SetCommand()

```
void gdcm::network::PresentationDataValue::SetCommand (
            bool inCommand )
```

10.245.3.13 SetDataSet()

```
void gdcm::network::PresentationDataValue::SetDataSet (
            const DataSet & ds )
```

Set DataSet. Write DataSet in implicit.

Warning

size of dataset should be below maxpdusize

10.245.3.14 SetLastFragment()

```
void gdcm::network::PresentationDataValue::SetLastFragment (
            bool inLast )
```

10.245.3.15 SetMessageHeader()

```
void gdcm::network::PresentationDataValue::SetMessageHeader (
            uint8_t messageheader )  [inline]
```

10.245.3.16 SetPresentationContextID()

```
void gdcm::network::PresentationDataValue::SetPresentationContextID (
            uint8_t id )  [inline]
```

10.245.3.17 Size()

```
size_t gdcm::network::PresentationDataValue::Size ( ) const
```

10.245.3.18 Write()

```
const std::ostream & gdcm::network::PresentationDataValue::Write (
            std::ostream & os ) const
```

The documentation for this class was generated from the following file:

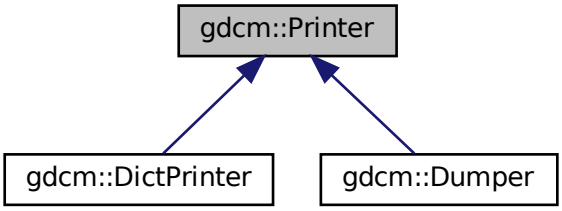
- gdcmPresentationDataValue.h

10.246 gdcm::Printer Class Reference

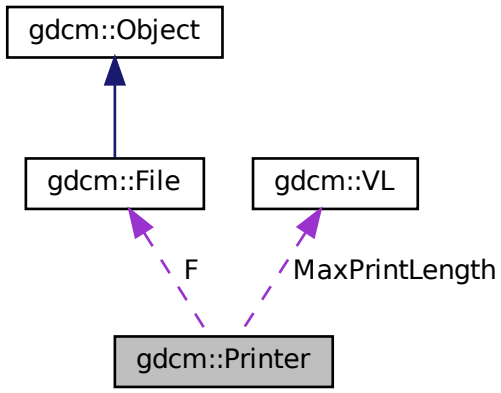
Printer class.

`#include <gdcmPrinter.h>`

Inheritance diagram for gdcm::Printer:



Collaboration diagram for gdcm::Printer:



Public Types

- enum PrintStyles {
  VERBOSE_STYLE = 0 ,
  CONDENSED_STYLE ,
  XML ,
  CXX }

Public Member Functions

- Printer ()
- ~Printer ()
- PrintStyles GetPrintStyle () const

  Get PrintStyle value.
- void Print (std::ostream &os)

  Print.
- void PrintDataSet (const DataSet &ds, std::ostream &os, const std::string &s="")

  Print an individual dataset.
- void SetColor (bool c)

  Set color mode or not.
- void SetFile (File const &f)

  Set file.
- void SetStyle (PrintStyles ps)

  Set PrintStyle value.

Protected Member Functions

- VR PrintDataElement (std::ostringstream &os, const Dicts &dicts, const DataSet &ds, const DataElement &de, std::ostream &out, std::string const &indent)
- void PrintSQ (const SequenceOfItems *sqi, std::ostream &os, std::string const &indent)

Protected Attributes

- const File * F
- VL MaxPrintLength
- PrintStyles PrintStyle

10.246.1 Detailed Description

Printer class.

Examples

DumpSiemensBase64.cxx, DumpToshibaDTI.cxx, and DumpToshibaDTI2.cxx.

10.246.2 Member Enumeration Documentation

10.246.2.1 PrintStyles

```
enum gdcm::Printer::PrintStyles
```

Enumerator

VERBOSE_STYLE	
CONDENSED_STYLE	
XML	
CXX	

10.246.3 Constructor & Destructor Documentation

10.246.3.1 Printer()

```
gdcm::Printer::Printer ( )
```

10.246.3.2 ~Printer()

```
gdcm::Printer::~Printer ( )
```

10.246.4 Member Function Documentation

10.246.4.1 GetPrintStyle()

```
PrintStyles gdcm::Printer::GetPrintStyle ( ) const  [inline]
```

Get PrintStyle value.

10.246.4.2 Print()

```
void gdcm::Printer::Print (
            std::ostream & os )
```

Print.

Examples

DumpSiemensBase64.cxx.

10.246.4.3 PrintDataElement()

```
VR gdcm::Printer::PrintDataElement (
            std::ostringstream & os,
            const Dicts & dicts,
            const DataSet & ds,
            const DataElement & de,
            std::ostream & out,
            std::string const & indent )  [protected]
```

10.246.4.4 PrintDataSet()

```
void gdcm::Printer::PrintDataSet (
            const DataSet & ds,
            std::ostream & os,
            const std::string & s = "" )
```

Print an individual dataset.

10.246.4.5 PrintSQ()

```
void gdcm::Printer::PrintSQ (
            const SequenceOfItems * sqi,
            std::ostream & os,
            std::string const & indent )  [protected]
```

10.246.4.6 SetColor()

```
void gdcm::Printer::SetColor (
            bool c )
```

Set color mode or not.

10.246.4.7 SetFile()

```
void gdcm::Printer::SetFile (
            File const & f )  [inline]
```

Set file.

Examples

DumpSiemensBase64.cxx, DumpToshibaDTI.cxx, and DumpToshibaDTI2.cxx.

10.246.4.8 SetStyle()

```
void gdcm::Printer::SetStyle (
            PrintStyles ps )  [inline]
```

Set PrintStyle value.

10.246.5 Member Data Documentation

10.246.5.1 F

```
const File* gdcm::Printer::F  [protected]
```

10.246.5.2 MaxPrintLength

```
VL gdcm::Printer::MaxPrintLength  [protected]
```

10.246.5.3 PrintStyle

```
PrintStyles gdcm::Printer::PrintStyle  [protected]
```

The documentation for this class was generated from the following file:

- gdcmPrinter.h

10.247 gdcm::PrivateDict Class Reference

Private Dict.

```
#include <gdcmDict.h>
```

Public Member Functions

- PrivateDict ()=default
- ~PrivateDict ()=default
- void AddDictEntry (const PrivateTag &tag, const DictEntry &de)
- bool FindDictEntry (const PrivateTag &tag) const
- const DictEntry & GetDictEntry (const PrivateTag &tag) const
- bool IsEmpty () const
- void PrintXML () const
- bool RemoveDictEntry (const PrivateTag &tag)

Protected Member Functions

- void LoadDefault ()

Friends

- class Dicts
- std::ostream & operator<< (std::ostream &os, const PrivateDict &val)

10.247.1 Detailed Description

Private Dict.

10.247.2 Constructor & Destructor Documentation

10.247.2.1 PrivateDict()

```
gdcm::PrivateDict::PrivateDict ( )  [default]
```

10.247.2.2 ~PrivateDict()

```
gdcm::PrivateDict::~PrivateDict ( )  [default]
```

10.247.3 Member Function Documentation

10.247.3.1 AddDictEntry()

```
void gdcm::PrivateDict::AddDictEntry (
            const PrivateTag & tag,
            const DictEntry & de )  [inline]
```

References gdcm::DictEntry::GetVM(), gdcm::DictEntry::GetVR(), gdcm::DictEntry::SetVM(), gdcm::DictEntry::SetVR(), and gdcm::VR::UN.

10.247.3.2 FindDictEntry()

```
bool gdcm::PrivateDict::FindDictEntry (
            const PrivateTag & tag ) const  [inline]
```

10.247.3.3 GetDictEntry()

```
const DictEntry & gdcm::PrivateDict::GetDictEntry (
            const PrivateTag & tag ) const  [inline]
```

10.247.3.4 IsEmpty()

```
bool gdcm::PrivateDict::IsEmpty ( ) const  [inline]
```

10.247.3.5 LoadDefault()

```
void gdcm::PrivateDict::LoadDefault ( )  [protected]
```

10.247.3.6 PrintXML()

```
void gdcm::PrivateDict::PrintXML ( ) const  [inline]
```

References gdcm::Tag::GetElement(), gdcm::Tag::GetGroup(), gdcm::DictEntry::GetName(), gdcm::PrivateTag::GetOwner(), gdcm::DictEntry::GetVM(), and gdcm::DictEntry::GetVR().

10.247.3.7 RemoveDictEntry()

```
bool gdcm::PrivateDict::RemoveDictEntry (
            const PrivateTag & tag )  [inline]
```

Remove entry 'tag'. Return true on success (element was found and remove). return false if element was not found.

10.247.4 Friends And Related Function Documentation

10.247.4.1 Dicts

```
friend class Dicts  [friend]
```

10.247.4.2 operator<<

```
std::ostream & operator<< (
             std::ostream & os,
             const PrivateDict & val )  [friend]
```

The documentation for this class was generated from the following file:

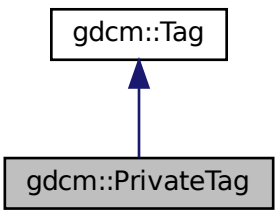
- gdcmDict.h

10.248 gdcm::PrivateTag Class Reference

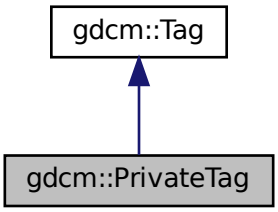
Class to represent a Private DICOM Data Element (Attribute) Tag (Group, Element, Owner)

```
#include <gdcmPrivateTag.h>
```

Inheritance diagram for gdcm::PrivateTag:



Collaboration diagram for gdcm::PrivateTag:



Public Member Functions

- PrivateTag (Tag const &t, const char *owner="")
- PrivateTag (uint16_t group=0, uint16_t element=0, const char *owner="")
- DataElement GetAsDataElement () const
- const char * GetOwner () const
- bool operator!= (const PrivateTag &_val) const
- bool operator!= (const Tag &_val) const
- bool operator< (const PrivateTag &_val) const
- PrivateTag & operator= (const PrivateTag &_val)
- bool operator== (const PrivateTag &_val) const
- bool operator== (const Tag &_val) const
- bool ReadFromCommaSeparatedString (const char *str)
- void SetOwner (const char *owner)

Friends

- std::ostream & operator<< (std::ostream &_os, const PrivateTag &_val)

10.248.1 Detailed Description

Class to represent a Private DICOM Data Element (Attribute) Tag (Group, Element, Owner)

Note

private tag have element value in: [0x10,0xff], for instance 0x0009,0x0000 is NOT a private tag

Examples

ChangePrivateTags.cxx, DumpADAC.cxx, DumpCSA.cs, DumpExamCard.cxx, DumpGEMSMovieGroup.cxx, DumpImageHeaderInfo.cxx, DumpPhilipsECHO.cxx, DumpSiemensBase64.cxx, DumpToshibaDTI.cxx, DumpToshibaDTI2.cxx, DumpVisusChange.cxx, ELSCINT1WaveToText.cxx, FileStreaming.cs, GetSubSequenceData.cxx, MrProtocol.cxx, PublicDict.cxx, ReadGEMSSDO.cxx, csa2img.cxx, iU22tomultisc.cxx, pmsct_rgb1.cxx, and rle2img.cxx.

10.248.2 Constructor & Destructor Documentation

10.248.2.1 PrivateTag() [1/2]

```
gdcm::PrivateTag::PrivateTag (
            uint16_t group = 0,
            uint16_t element = 0,
            const char * owner = "" )  [inline]
```

10.248.2.2 PrivateTag() [2/2]

```
gdcm::PrivateTag::PrivateTag (
            Tag const & t,
            const char * owner = "" )  [inline]
```

References gdcm::Tag::GetElement().

10.248.3 Member Function Documentation

10.248.3.1 GetAsDataElement()

```
DataElement gdcm::PrivateTag::GetAsDataElement ( ) const
```

10.248.3.2 GetOwner()

```
const char * gdcm::PrivateTag::GetOwner ( ) const  [inline]
```

Examples

PublicDict.cxx.

Referenced by gdcm::PrivateDict::PrintXML().

10.248.3.3 operator"!=() [1/2]

```
bool gdcm::PrivateTag::operator!= (
            const PrivateTag & _val ) const  [inline]
```

References gdcm::Tag::GetElementTag().

10.248.3.4 operator"!=() [2/2]

```
bool gdcm::PrivateTag::operator!= (
            const Tag & _val ) const  [inline]
```

References gdcm::Tag::GetElementTag().

10.248.3.5 operator<()

```
bool gdcm::PrivateTag::operator< (
            const PrivateTag & _val ) const
```

10.248.3.6 operator=()

```
PrivateTag & gdcm::PrivateTag::operator= (
            const PrivateTag & _val )  [inline]
```

References gdcm::Tag::GetElementTag().

10.248.3.7 operator==() [1/2]

```
bool gdcm::PrivateTag::operator== (
            const PrivateTag & _val ) const  [inline]
```

References gdcm::Tag::GetElementTag().

10.248.3.8 operator==() [2/2]

```
bool gdcm::PrivateTag::operator== (
            const Tag & _val ) const  [inline]
```

References gdcm::Tag::GetElementTag().

10.248.3.9 ReadFromCommaSeparatedString()

```
bool gdcm::PrivateTag::ReadFromCommaSeparatedString (
            const char * str )
```

Read PrivateTag from a string. Element number will be truncated to 8bits. Eg: "1234,5678,GDCM" is private tag: (1234,78,"GDCM")

10.248.3.10 SetOwner()

```
void gdcm::PrivateTag::SetOwner (
            const char * owner )  [inline]
```

10.248.4 Friends And Related Function Documentation

10.248.4.1 operator<<

```
std::ostream & operator<< (
            std::ostream & _os,
            const PrivateTag & _val )  [friend]
```

The documentation for this class was generated from the following file:

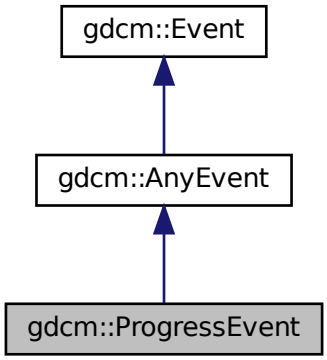
- gdcmPrivateTag.h

10.249 gdcm::ProgressEvent Class Reference

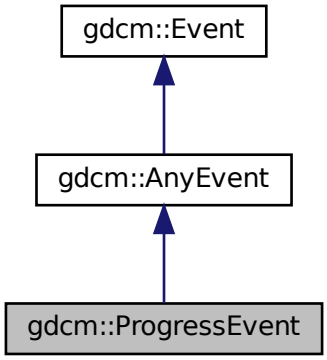
ProgressEvent.

`#include <gdcmProgressEvent.h>`

Inheritance diagram for gdcm::ProgressEvent:



Collaboration diagram for gdcm::ProgressEvent:



Public Types

- typedef ProgressEvent Self
- typedef AnyEvent Superclass

Public Member Functions

- ProgressEvent (const Self &s)
- ProgressEvent (double p=0)
- ~ProgressEvent () override=default
- bool CheckEvent (const ::gdcm::Event *e) const override
- const char * GetEventName () const override
- double GetProgress () const
- ::gdcm::Event * MakeObject () const override
- void operator= (const Self &)=delete
- void SetProgress (double p)

10.249.1 Detailed Description

ProgressEvent.

Special type of event triggered during

See also

AnyEvent

Examples

BasicAnonymizer.cs, and ClinicalTrialIdentificationWorkflow.cs.

10.249.2 Member Typedef Documentation

10.249.2.1 Self

```
typedef ProgressEvent gdcm::ProgressEvent::Self
```

10.249.2.2 Superclass

```
typedef AnyEvent gdcm::ProgressEvent::Superclass
```

10.249.3 Constructor & Destructor Documentation

10.249.3.1 ProgressEvent() [1/2]

```
gdcm::ProgressEvent::ProgressEvent (
            double p = 0 )  [inline]
```

10.249.3.2 ~ProgressEvent()

```
gdcm::ProgressEvent::~ProgressEvent ( )  [override], [default]
```

10.249.3.3 ProgressEvent() [2/2]

```
gdcm::ProgressEvent::ProgressEvent (
            const Self & s )  [inline]
```

10.249.4 Member Function Documentation

10.249.4.1 CheckEvent()

```
bool gdcm::ProgressEvent::CheckEvent (
            const ::gdcm::Event * e ) const  [inline], [override]
```

10.249.4.2 GetEventName()

```
const char * gdcm::ProgressEvent::GetEventName ( ) const  [inline], [override], [virtual]
```

Return the StringName associated with the event.

Implements gdcm::Event.

10.249.4.3 GetProgress()

```
double gdcm::ProgressEvent::GetProgress ( ) const  [inline]
```

Examples

BasicAnonymizer.cs, and ClinicalTrialIdentificationWorkflow.cs.

10.249.4.4 MakeObject()

```
::gdcm::Event * gdcm::ProgressEvent::MakeObject ( ) const  [inline], [override], [virtual]
```

Create an Event of this type This method work as a Factory for creating events of each particular type.

Implements gdcm::Event.

10.249.4.5 operator=()

```
void gdcm::ProgressEvent::operator= (
            const Self &  )  [delete]
```

10.249.4.6 SetProgress()

```
void gdcm::ProgressEvent::SetProgress (
            double p )  [inline]
```

The documentation for this class was generated from the following file:

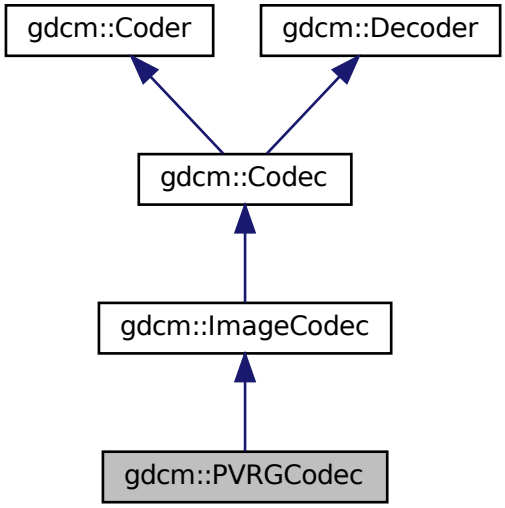
- gdcmProgressEvent.h

10.250 gdcm::PVRGCodec Class Reference

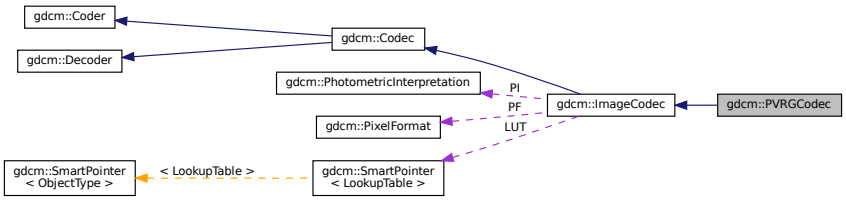
PVRGCodec.

`#include <gdcmPVRGCodec.h>`

Inheritance diagram for gdcm::PVRGCodec:



Collaboration diagram for gdcm::PVRGCodec:



Public Member Functions

- PVRGCodec ()
- ~PVRGCodec () override
- bool CanCode (TransferSyntax const &ts) const override

Return whether this coder support this transfer syntax (can code it)

- bool CanDecode (TransferSyntax const &ts) const override

  Return whether this decoder support this transfer syntax (can decode it)
- ImageCodec * Clone () const override
- bool Code (DataElement const &in, DataElement &out) override

  Code.
- bool Decode (DataElement const &is, DataElement &os) override

  Decode.
- void SetLossyFlag (bool l)

Additional Inherited Members

10.250.1 Detailed Description

PVRGCodec.

Note

pvrg is a broken implementation of the JPEG standard. It is known to have a bug in the 16bits lossless implementation of the standard.

In an ideal world, you should not need this codec at all. But to support some broken file such as:

PHILIPS_Gyroscan-12-Jpeg_Extended_Process_2_4.dcm

we have to...

10.250.2 Constructor & Destructor Documentation

10.250.2.1 PVRGCodec()

```
gdcm::PVRGCodec::PVRGCodec ( )
```

10.250.2.2 ~PVRGCodec()

```
gdcm::PVRGCodec::~PVRGCodec ( )  [override]
```

10.250.3 Member Function Documentation

10.250.3.1 CanCode()

```
bool gdcm::PVRGCodec::CanCode (
            TransferSyntax const &  ) const  [override], [virtual]
```

Return whether this coder support this transfer syntax (can code it)

Reimplemented from gdcm::ImageCodec.

10.250.3.2 CanDecode()

```
bool gdcm::PVRGCodec::CanDecode (
            TransferSyntax const &  ) const  [override], [virtual]
```

Return whether this decoder support this transfer syntax (can decode it)

Reimplemented from gdcm::ImageCodec.

10.250.3.3 Clone()

```
ImageCodec * gdcm::PVRGCodec::Clone ( ) const  [override], [virtual]
```

Implements gdcm::ImageCodec.

10.250.3.4 Code()

```
bool gdcm::PVRGCodec::Code (
            DataElement const & in_,
            DataElement & out_ )  [override], [virtual]
```

Code.

Reimplemented from gdcm::Coder.

10.250.3.5 Decode()

```
bool gdcm::PVRGCodec::Decode (
            DataElement const & ,
            DataElement &  )  [override], [virtual]
```

Decode.

Reimplemented from gdcm::ImageCodec.

10.250.3.6 SetLossyFlag()

```
void gdcm::PVRGCodec::SetLossyFlag (
            bool l )
```

The documentation for this class was generated from the following file:

- gdcmPVRGCodec.h

10.251 gdcm::PythonFilter Class Reference

PythonFilter PythonFilter is the class that make gdcm2.x looks more like gdcm1 and transform the binary blob contained in a DataElement into a string, typically this is a nice feature to have for wrapped language.

```
#include <gdcmPythonFilter.h>
```

Public Member Functions

- PythonFilter ()
- ~PythonFilter ()
- File & GetFile ()
- const File & GetFile () const
- void SetDicts (const Dicts &dicts)
- void SetFile (const File &f)
- PyObject * ToPyObject (const Tag &t) const
- void UseDictAlways (bool)

10.251.1 Detailed Description

PythonFilter PythonFilter is the class that make gdcm2.x looks more like gdcm1 and transform the binary blob contained in a DataElement into a string, typically this is a nice feature to have for wrapped language.

10.251.2 Constructor & Destructor Documentation

10.251.2.1 PythonFilter()

```
gdcm::PythonFilter::PythonFilter ( )
```

10.251.2.2 ~PythonFilter()

```
gdcm::PythonFilter::~PythonFilter ( )
```

10.251.3 Member Function Documentation

10.251.3.1 GetFile() [1/2]

```
File & gdcm::PythonFilter::GetFile ( )
```

10.251.3.2 GetFile() [2/2]

```
const File & gdcm::PythonFilter::GetFile ( ) const
```

10.251.3.3 SetDicts()

```
void gdcm::PythonFilter::SetDicts (
            const Dicts & dicts )
```

10.251.3.4 SetFile()

```
void gdcm::PythonFilter::SetFile (
            const File & f )
```

10.251.3.5 ToPyObject()

```
PyObject * gdcm::PythonFilter::ToPyObject (
            const Tag & t ) const
```

10.251.3.6 UseDictAlways()

```
void gdcm::PythonFilter::UseDictAlways (
            bool  )  [inline]
```

The documentation for this class was generated from the following file:

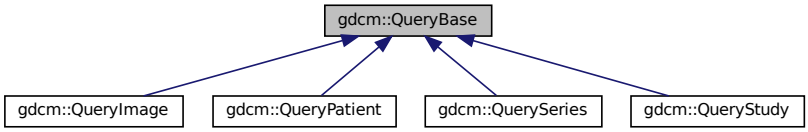
- gdcmPythonFilter.h

10.252 gdcm::QueryBase Class Reference

QueryBase.

```
#include <gdcmQueryBase.h>
```

Inheritance diagram for gdcm::QueryBase:



Public Member Functions

- virtual ~QueryBase ()=default
- std::vector< Tag > GetAllRequiredTags (const ERootType &inRootType) const
- std::vector< Tag > GetAllTags (const ERootType &inRootType) const
- virtual std::vector< Tag > GetHierachicalSearchTags (const ERootType &inRootType) const =0

  Return all Unique Key for a particular Query Root type (from the same level and above).
- virtual const char * GetName () const =0
- virtual std::vector< Tag > GetOptionalTags (const ERootType &inRootType) const =0
- virtual DataElement GetQueryLevel () const =0
- virtual std::vector< Tag > GetRequiredTags (const ERootType &inRootType) const =0
- virtual std::vector< Tag > GetUniqueTags (const ERootType &inRootType) const =0

10.252.1 Detailed Description

QueryBase.

contains: the base class for constructing a query dataset for a C-FIND and a C-MOVE

There are four levels of C-FIND and C-MOVE query:

- Patient
- Study
- Series
- Image

Each one has its own required and optional tags. This class provides an interface for getting those tags. This is an interface class.

See 3.4 C 6.1 and 3.4 C 6.2 for the patient and study root query types. These sections define the tags allowed by a particular query. The caller must pass in which root type they want, patient or study. A third root type, Modality Worklist Query, isn't yet supported.

This class (or rather it's derived classes) will be held in the RootQuery types. These query types actually make the dataset, and will use this dataset to list the required, unique, and optional tags for each type of query. This design is somewhat overly complicated, but is kept so that if we ever wanted to try to guess the query type from the given tags, we could do so.

10.252.2 Constructor & Destructor Documentation

10.252.2.1 ~QueryBase()

```
virtual gdcm::QueryBase::~QueryBase ( )  [virtual], [default]
```

10.252.3 Member Function Documentation

10.252.3.1 GetAllRequiredTags()

```
std::vector< Tag > gdcm::QueryBase::GetAllRequiredTags (
            const ERootType & inRootType ) const
```

In order to validate a query dataset we need to check that there exists at least one required (or unique) key

10.252.3.2 GetAllTags()

```
std::vector< Tag > gdcm::QueryBase::GetAllTags (
            const ERootType & inRootType ) const
```

In order to validate a query dataset, just check for the presence of a tag, not it's requirement level in the spec

10.252.3.3 GetHierachicalSearchTags()

```
virtual std::vector< Tag > gdcm::QueryBase::GetHierachicalSearchTags (
            const ERootType & inRootType ) const  [pure virtual]
```

Return all Unique Key for a particular Query Root type (from the same level and above).

Implemented in gdcm::QueryImage, gdcm::QueryPatient, gdcm::QuerySeries, and gdcm::QueryStudy.

10.252.3.4 GetName()

```
virtual const char * gdcm::QueryBase::GetName ( ) const  [pure virtual]
```

Implemented in gdcm::QueryImage, gdcm::QueryPatient, gdcm::QuerySeries, and gdcm::QueryStudy.

10.252.3.5 GetOptionalTags()

```
virtual std::vector< Tag > gdcm::QueryBase::GetOptionalTags (
            const ERootType & inRootType ) const  [pure virtual]
```

Implemented in gdcm::QueryImage, gdcm::QueryPatient, gdcm::QuerySeries, and gdcm::QueryStudy.

10.252.3.6 GetQueryLevel()

```
virtual DataElement gdcm::QueryBase::GetQueryLevel ( ) const  [pure virtual]
```

Implemented in gdcm::QueryImage, gdcm::QueryPatient, gdcm::QuerySeries, and gdcm::QueryStudy.

10.252.3.7 GetRequiredTags()

```
virtual std::vector< Tag > gdcm::QueryBase::GetRequiredTags (
            const ERootType & inRootType ) const  [pure virtual]
```

Implemented in gdcm::QueryImage, gdcm::QueryPatient, gdcm::QuerySeries, and gdcm::QueryStudy.

10.252.3.8 GetUniqueTags()

```
virtual std::vector< Tag > gdcm::QueryBase::GetUniqueTags (
            const ERootType & inRootType ) const  [pure virtual]
```

Implemented in gdcm::QueryImage, gdcm::QueryPatient, gdcm::QuerySeries, and gdcm::QueryStudy.

The documentation for this class was generated from the following file:

- gdcmQueryBase.h

10.253 gdcm::QueryFactory Class Reference

QueryFactory.h.

```
#include <gdcmQueryFactory.h>
```

Static Public Member Functions

- static ECharSet GetCharacterFromCurrentLocale ()
- static void ListCharSets (std::ostream &os)

  List all possible CharSet.
- static DataElement ProduceCharacterSetDataElement (const std::vector< ECharSet > &inCharSetType)
- static BaseQuery * ProduceQuery (const std::string &sopInstanceUID, ENQueryType inQueryType)
- static BaseRootQuery * ProduceQuery (ERootType inRootType, EQueryType inQueryType, EQueryLevel in↩ QueryLevel)

10.253.1 Detailed Description

QueryFactory.h.

Note

contains: a class to produce a query based off of user-entered information

Essentially, this class is used to construct a query based off of user input (typically from the command line; if in code directly, the query itself could just be instantiated)

In theory, could also be used as the interface to validate incoming datasets as belonging to a particular query style

10.253.2 Member Function Documentation

10.253.2.1 GetCharacterFromCurrentLocale()

```
static ECharSet gdcm::QueryFactory::GetCharacterFromCurrentLocale ( )  [static]
```

This function will return the corresponding ECharSet associated with the current locale of the running system (based on the value of locale()).

10.253.2.2 ListCharSets()

```
static void gdcm::QueryFactory::ListCharSets (
            std::ostream & os )  [static]
```

List all possible CharSet.

10.253.2.3 ProduceCharacterSetDataElement()

```
static DataElement gdcm::QueryFactory::ProduceCharacterSetDataElement (
            const std::vector< ECharSet > & inCharSetType )  [static]
```

This function will produce the appropriate dataelement given a list of charsets. The first charset will be used directly, while the second and subsequent will be prepended with "ISO2022 ". Redundant character sets are not permitted, so if they are encountered, they will just be skipped. if UTF8 or GB18030 is used, no subsequent character sets will be used if the vector passed in is empty, then the dataelement that's passed out will be empty and Latin1 is the presumed encoding

10.253.2.4 ProduceQuery() [1/2]

```
static BaseQuery * gdcm::QueryFactory::ProduceQuery (
            const std::string & sopInstanceUID,
            ENQueryType inQueryType )  [static]
```

10.253.2.5 ProduceQuery() [2/2]

```
static BaseRootQuery * gdcm::QueryFactory::ProduceQuery (
            ERootType inRootType,
            EQueryType inQueryType,
            EQueryLevel inQueryLevel )  [static]
```

this function will produce a query (basically, a wrapper to a dataset that can validate whether or not the query is a valid cfind/cmove query) and the level of the query (patient, study, series, image). If the user provides an invalid instantiation (ie, study root type, query level of patient), then the result is NULL.

The documentation for this class was generated from the following file:

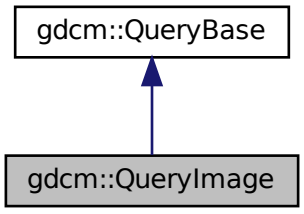
- gdcmQueryFactory.h

10.254 gdcm::QueryImage Class Reference

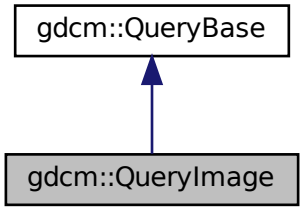
QueryImage.

`#include <gdcmQueryImage.h>`

Inheritance diagram for gdcm::QueryImage:



Collaboration diagram for gdcm::QueryImage:



Public Member Functions

- std::vector< Tag > GetHierachicalSearchTags (const ERootType &inRootType) const override

  Return all Unique Key for a particular Query Root type (from the same level and above).
- const char * GetName () const override
- std::vector< Tag > GetOptionalTags (const ERootType &inRootType) const override
- DataElement GetQueryLevel () const override
- std::vector< Tag > GetRequiredTags (const ERootType &inRootType) const override
- std::vector< Tag > GetUniqueTags (const ERootType &inRootType) const override

10.254.1 Detailed Description

QueryImage.

contains: class to construct an image-based query for C-FIND and C-MOVE

10.254.2 Member Function Documentation

10.254.2.1 GetHierachicalSearchTags()

```
std::vector< Tag > gdcm::QueryImage::GetHierachicalSearchTags (
            const ERootType & inRootType ) const  [override], [virtual]
```

Return all Unique Key for a particular Query Root type (from the same level and above).

Implements gdcm::QueryBase.

10.254.2.2 GetName()

```
const char * gdcm::QueryImage::GetName ( ) const  [override], [virtual]
```

Implements gdcm::QueryBase.

10.254.2.3 GetOptionalTags()

```
std::vector< Tag > gdcm::QueryImage::GetOptionalTags (
            const ERootType & inRootType ) const  [override], [virtual]
```

Implements gdcm::QueryBase.

10.254.2.4 GetQueryLevel()

```
DataElement gdcm::QueryImage::GetQueryLevel ( ) const  [override], [virtual]
```

Implements gdcm::QueryBase.

10.254.2.5 GetRequiredTags()

```
std::vector< Tag > gdcm::QueryImage::GetRequiredTags (
            const ERootType & inRootType ) const  [override], [virtual]
```

Implements gdcm::QueryBase.

10.254.2.6 GetUniqueTags()

```
std::vector< Tag > gdcm::QueryImage::GetUniqueTags (
            const ERootType & inRootType ) const  [override], [virtual]
```

Implements gdcm::QueryBase.

The documentation for this class was generated from the following file:

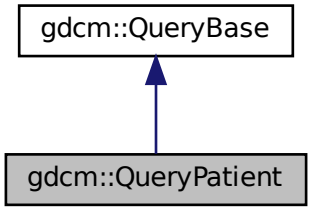
- gdcmQueryImage.h

10.255 gdcm::QueryPatient Class Reference

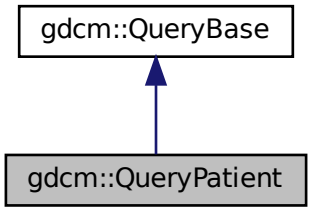
QueryPatient.

```
#include <gdcmQueryPatient.h>
```

Inheritance diagram for gdcm::QueryPatient:



Collaboration diagram for gdcm::QueryPatient:



Public Member Functions

- std::vector< Tag > GetHierachicalSearchTags (const ERootType &inRootType) const override
  Return all Unique Key for a particular Query Root type (from the same level and above).
- const char * GetName () const override
- std::vector< Tag > GetOptionalTags (const ERootType &inRootType) const override
- DataElement GetQueryLevel () const override
- std::vector< Tag > GetRequiredTags (const ERootType &inRootType) const override
- std::vector< Tag > GetUniqueTags (const ERootType &inRootType) const override

10.255.1 Detailed Description

QueryPatient.

contains: class to construct a patient-based query for c-find and c-move

10.255.2 Member Function Documentation

10.255.2.1 GetHierachicalSearchTags()

```
std::vector< Tag > gdcm::QueryPatient::GetHierachicalSearchTags (
            const ERootType & inRootType ) const  [override], [virtual]
```

Return all Unique Key for a particular Query Root type (from the same level and above).

Implements gdcm::QueryBase.

10.255.2.2 GetName()

```
const char * gdcm::QueryPatient::GetName ( ) const  [override], [virtual]
```

Implements gdcm::QueryBase.

10.255.2.3 GetOptionalTags()

```
std::vector< Tag > gdcm::QueryPatient::GetOptionalTags (
            const ERootType & inRootType ) const  [override], [virtual]
```

Implements gdcm::QueryBase.

10.255.2.4 GetQueryLevel()

```
DataElement gdcm::QueryPatient::GetQueryLevel ( ) const  [override], [virtual]
```

Implements gdcm::QueryBase.

10.255.2.5 GetRequiredTags()

```
std::vector< Tag > gdcm::QueryPatient::GetRequiredTags (
            const ERootType & inRootType ) const  [override], [virtual]
```

Implements gdcm::QueryBase.

10.255.2.6 GetUniqueTags()

```
std::vector< Tag > gdcm::QueryPatient::GetUniqueTags (
            const ERootType & inRootType ) const  [override], [virtual]
```

Implements gdcm::QueryBase.

The documentation for this class was generated from the following file:

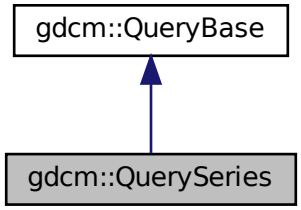
- gdcmQueryPatient.h

10.256 gdcm::QuerySeries Class Reference

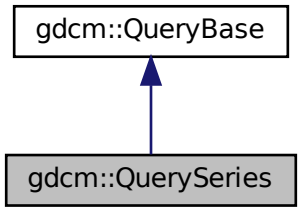
QuerySeries.

`#include <gdcmQuerySeries.h>`

Inheritance diagram for gdcm::QuerySeries:



Collaboration diagram for gdcm::QuerySeries:



Public Member Functions

- std::vector< Tag > GetHierachicalSearchTags (const ERootType &inRootType) const override
  Return all Unique Key for a particular Query Root type (from the same level and above).
- const char * GetName () const override
- std::vector< Tag > GetOptionalTags (const ERootType &inRootType) const override
- DataElement GetQueryLevel () const override
- std::vector< Tag > GetRequiredTags (const ERootType &inRootType) const override
- std::vector< Tag > GetUniqueTags (const ERootType &inRootType) const override

10.256.1 Detailed Description

QuerySeries.

contains: class to construct a series-based query for c-find and c-move

10.256.2 Member Function Documentation

10.256.2.1 GetHierachicalSearchTags()

```
std::vector< Tag > gdcm::QuerySeries::GetHierachicalSearchTags (
            const ERootType & inRootType ) const  [override], [virtual]
```

Return all Unique Key for a particular Query Root type (from the same level and above).

Implements gdcm::QueryBase.

10.256.2.2 GetName()

```
const char * gdcm::QuerySeries::GetName ( ) const  [override], [virtual]
```

Implements gdcm::QueryBase.

10.256.2.3 GetOptionalTags()

```
std::vector< Tag > gdcm::QuerySeries::GetOptionalTags (
            const ERootType & inRootType ) const  [override], [virtual]
```

Implements gdcm::QueryBase.

10.256.2.4 GetQueryLevel()

```
DataElement gdcm::QuerySeries::GetQueryLevel ( ) const  [override], [virtual]
```

Implements gdcm::QueryBase.

10.256.2.5 GetRequiredTags()

```
std::vector< Tag > gdcm::QuerySeries::GetRequiredTags (
            const ERootType & inRootType ) const  [override], [virtual]
```

Implements gdcm::QueryBase.

10.256.2.6 GetUniqueTags()

```
std::vector< Tag > gdcm::QuerySeries::GetUniqueTags (
            const ERootType & inRootType ) const  [override], [virtual]
```

Implements gdcm::QueryBase.

The documentation for this class was generated from the following file:

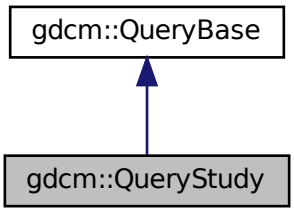
- gdcmQuerySeries.h

10.257 gdcm::QueryStudy Class Reference

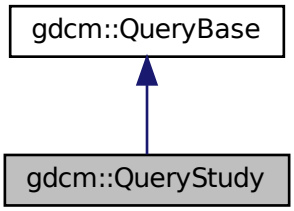
QueryStudy.h.

```
#include <gdcmQueryStudy.h>
```

Inheritance diagram for gdcm::QueryStudy:



Collaboration diagram for gdcm::QueryStudy:



Public Member Functions

- std::vector< Tag > GetHierachicalSearchTags (const ERootType &inRootType) const override
  Return all Unique Key for a particular Query Root type (from the same level and above).
- const char * GetName () const override
- std::vector< Tag > GetOptionalTags (const ERootType &inRootType) const override
- DataElement GetQueryLevel () const override
- std::vector< Tag > GetRequiredTags (const ERootType &inRootType) const override
- std::vector< Tag > GetUniqueTags (const ERootType &inRootType) const override

10.257.1 Detailed Description

QueryStudy.h.

contains: class to construct a study-based query for C-FIND and C-MOVE

10.257.2 Member Function Documentation

10.257.2.1 GetHierachicalSearchTags()

```
std::vector< Tag > gdcm::QueryStudy::GetHierachicalSearchTags (
            const ERootType & inRootType ) const  [override], [virtual]
```

Return all Unique Key for a particular Query Root type (from the same level and above).

Implements gdcm::QueryBase.

10.257.2.2 GetName()

```
const char * gdcm::QueryStudy::GetName ( ) const  [override], [virtual]
```

Implements gdcm::QueryBase.

10.257.2.3 GetOptionalTags()

```
std::vector< Tag > gdcm::QueryStudy::GetOptionalTags (
            const ERootType & inRootType ) const  [override], [virtual]
```

Implements gdcm::QueryBase.

10.257.2.4 GetQueryLevel()

```
DataElement gdcm::QueryStudy::GetQueryLevel ( ) const  [override], [virtual]
```

Implements gdcm::QueryBase.

10.257.2.5 GetRequiredTags()

```
std::vector< Tag > gdcm::QueryStudy::GetRequiredTags (
            const ERootType & inRootType ) const  [override], [virtual]
```

Implements gdcm::QueryBase.

10.257.2.6 GetUniqueTags()

```
std::vector< Tag > gdcm::QueryStudy::GetUniqueTags (
            const ERootType & inRootType ) const  [override], [virtual]
```

Implements gdcm::QueryBase.

The documentation for this class was generated from the following file:

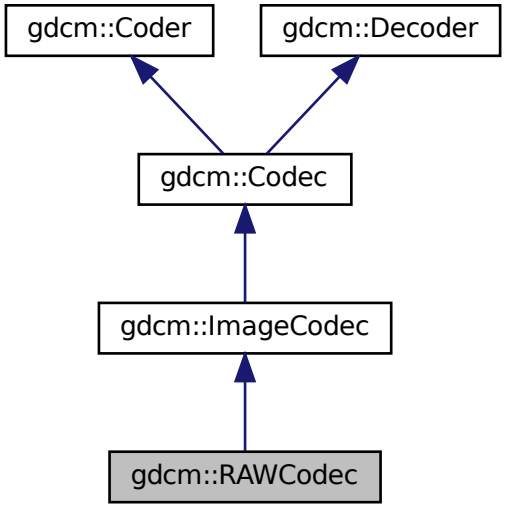
- gdcmQueryStudy.h

10.258 gdcm::RAWCodec Class Reference

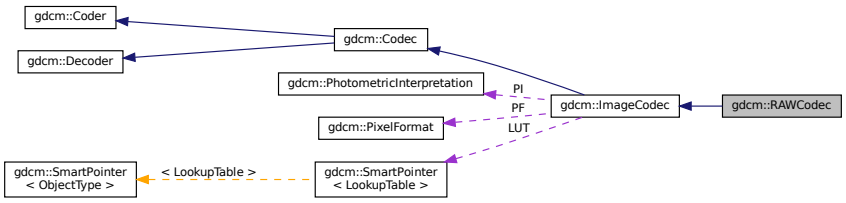
RAWCodec class.

`#include <gdcmRAWCodec.h>`

Inheritance diagram for gdcm::RAWCodec:



Collaboration diagram for gdcm::RAWCodec:



Public Member Functions

- RAWCodec ()
- ~RAWCodec () override
- bool CanCode (TransferSyntax const &ts) const override

Return whether this coder support this transfer syntax (can code it)

- bool CanDecode (TransferSyntax const &ts) const override

  Return whether this decoder support this transfer syntax (can decode it)
- ImageCodec * Clone () const override
- bool Code (DataElement const &in, DataElement &out) override

  Code.
- bool Decode (DataElement const &is, DataElement &os) override

  Decode.
- bool DecodeBytes (const char *inBytes, size_t inBufferLength, char *outBytes, size_t inOutBufferLength)
- bool GetHeaderInfo (std::istream &is, TransferSyntax &ts) override

Protected Member Functions

- bool DecodeByStreams (std::istream &is, std::ostream &os) override

Additional Inherited Members

10.258.1 Detailed Description

RAWCodec class.

10.258.2 Constructor & Destructor Documentation

10.258.2.1 RAWCodec()

```
gdcm::RAWCodec::RAWCodec ( )
```

10.258.2.2 ~RAWCodec()

```
gdcm::RAWCodec::~RAWCodec ( )  [override]
```

10.258.3 Member Function Documentation

10.258.3.1 CanCode()

```
bool gdcm::RAWCodec::CanCode (
            TransferSyntax const &  ) const  [override], [virtual]
```

Return whether this coder support this transfer syntax (can code it)

Reimplemented from gdcm::ImageCodec.

10.258.3.2 CanDecode()

```
bool gdcm::RAWCodec::CanDecode (
            TransferSyntax const &  ) const  [override], [virtual]
```

Return whether this decoder support this transfer syntax (can decode it)

Reimplemented from gdcm::ImageCodec.

10.258.3.3 Clone()

```
ImageCodec * gdcm::RAWCodec::Clone ( ) const  [override], [virtual]
```

Implements gdcm::ImageCodec.

10.258.3.4 Code()

```
bool gdcm::RAWCodec::Code (
            DataElement const & in_,
            DataElement & out_ )  [override], [virtual]
```

Code.

Reimplemented from gdcm::Coder.

10.258.3.5 Decode()

```
bool gdcm::RAWCodec::Decode (
            DataElement const & ,
            DataElement &  )  [override], [virtual]
```

Decode.

Reimplemented from gdcm::ImageCodec.

10.258.3.6 DecodeByStreams()

```
bool gdcm::RAWCodec::DecodeByStreams (
            std::istream & is,
            std::ostream & os )  [override], [protected], [virtual]
```

Reimplemented from gdcm::ImageCodec.

10.258.3.7 DecodeBytes()

```
bool gdcm::RAWCodec::DecodeBytes (
            const char * inBytes,
            size_t inBufferLength,
            char * outBytes,
            size_t inOutBufferLength )
```

Used by the ImageStreamReader– converts a read in buffer into one with the proper encodings.

10.258.3.8 GetHeaderInfo()

```
bool gdcm::RAWCodec::GetHeaderInfo (
            std::istream & is,
            TransferSyntax & ts )  [override], [virtual]
```

Reimplemented from gdcm::ImageCodec.

The documentation for this class was generated from the following file:

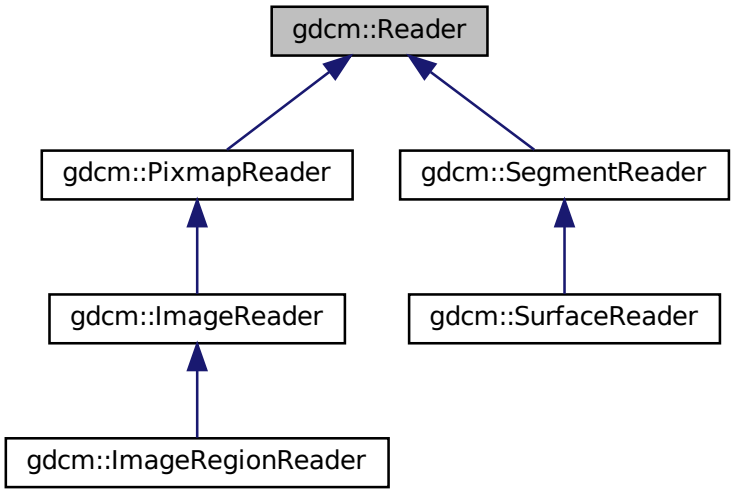
- gdcmRAWCodec.h

10.259 gdcm::Reader Class Reference

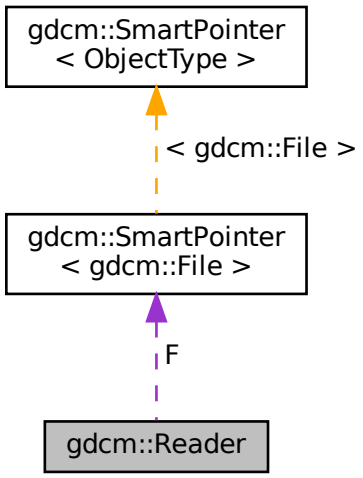
Reader ala DOM (Document Object Model)

```
#include <gdcmReader.h>
```

Inheritance diagram for gdcm::Reader:



Collaboration diagram for gdcm::Reader:



Public Member Functions

- Reader ()
- virtual ~Reader ()
- bool CanRead () const
- File & GetFile ()

  Set/Get File.
- const File & GetFile () const

  Set/Get File.
- size_t GetStreamCurrentPosition () const
- virtual bool Read ()

  Main function to read a file.
- bool ReadSelectedPrivateTags (std::set< PrivateTag > const &ptags, bool readvalues=true)

  Will only read the specified selected private tags.
- bool ReadSelectedTags (std::set< Tag > const &tags, bool readvalues=true)

  Will only read the specified selected tags.
- bool ReadUpToTag (const Tag &tag, std::set< Tag > const &skiptags=std::set< Tag >())
- void SetFile (File &file)

  Set/Get File.
- void SetFileName (const char *filename_native)
- void SetStream (std::istream &input_stream)

  Set the open-ed stream directly.

Protected Member Functions

- std::istream * GetStreamPtr () const
- bool ReadDataSet ()
- bool ReadMetaInformation ()
- bool ReadPreamble ()

Protected Attributes

- SmartPointer< File > F

Friends

- class StreamImageReader

10.259.1 Detailed Description

Reader ala DOM (Document Object Model)

This class is a non-validating reader, it will only performs well- formedness check only, and to some extent catch known error (non well-formed document).

Detailed description here

A DataSet DOES NOT contains group 0x0002 (see FileMetaInformation)

This is really a DataSet reader. This will not make sure the dataset conform to any IOD at all. This is a completely different step. The reasoning was that user could control the IOD there lib would handle and thus we would not be able to read a DataSet if the IOD was not found Instead we separate the reading from the validation.

Note

From GDCM1.x. Users will realize that one feature is missing from this DOM implementation. In GDCM 1.x user used to be able to control the size of the Value to be read. By default it was 0xfff. The main author of GDCM2 thought this was too dangerous and harmful and therefore this feature did not make it into GDCM2

Warning

GDCM will not produce warning for unorder (non-alphabetical order).

See also

Writer FileMetaInformation DataSet File

Examples

BasicAnonymizer.cs, ChangePrivateTags.cxx, ChangeSequenceUltrasound.cxx, ClinicalTrialAnnotate.cxx, ClinicalTrialIdentificationWorkflow.cs, CreateFakeRTDOSE.cxx, DeriveSeries.cxx, DiffFile.cxx, DumpADAC.cxx, DumpCSA.cs, DumpExamCard.cxx, DumpGEMSMovieGroup.cxx, DumpImageHeaderInfo.cxx, DumpPhilipsECHO.cxx, DumpSiemensBase64.cxx, DumpToshibaDTI.cxx, DumpToshibaDTI2.cxx, DumpVisusChange.cxx, DuplicatePCDE.cxx, ELSCINT1WaveToText.cxx, ExtractEncapsulatedFile.cs, ExtractEncryptedContent.cxx, FixBrokenJ2K.cxx, FixOrientation.cxx, GenLongSeqs.cxx, GenSeqs.cxx, GetSequenceUltrasound.cxx, GetSubSequenceData.cxx, HelloWorld.cxx, LargeVRDSExplicit.cxx, MakeTemplate.cxx, ManipulateFile.cs, NewSequence.cs, PatchFile.cxx, QIDO-RS.cxx, ReadAndDumpDICOMDIR.cxx, ReadAndDumpDICOMDIR2.cxx, ReadAndPrintAttributes.cxx, ReadExplicitLengthSQIVR.cxx, ReadGEMSSDO.cxx, ReadUTF8QtDir.cxx, ReformatFile.cs, SimplePrint.cs, SimplePrintPatientName.cs, TestReader.cxx, csa2img.cxx, gdcmrtionplan.cxx, gdcmrtplan.cxx, iU22tomultisc.cxx, pmsct_rgb1.cxx, and rle2img.cxx.

10.259.2 Constructor & Destructor Documentation

10.259.2.1 Reader()

```
gdcm::Reader::Reader ( )
```

10.259.2.2 ~Reader()

```
virtual gdcm::Reader::~Reader ( )  [virtual]
```

10.259.3 Member Function Documentation

10.259.3.1 CanRead()

```
bool gdcm::Reader::CanRead ( ) const
```

Test whether this is a DICOM file

Warning

need to call either SetFileName or SetStream first

Examples

ReadUTF8QtDir.cxx.

10.259.3.2 GetFile() [1/2]

```
File & gdcm::Reader::GetFile ( )  [inline]
```

Set/Get File.

10.259.3.3 GetFile() [2/2]

```
const File & gdcm::Reader::GetFile ( ) const  [inline]
```

Set/Get File.

Examples

BasicAnonymizer.cs, BasicImageAnonymizer.cs, ChangePrivateTags.cxx, ChangeSequenceUltrasound.cxx, ClinicalTrialAnnotate.cxx, ClinicalTrialIdentificationWorkflow.cs, CompressImage.cxx, CompressLossyJPEG.cs, CreateFakeRTDOSE.cxx, DecompressImage.cs, DeriveSeries.cxx, DiffFile.cxx, DumpADAC.cxx, DumpCSA.cs, DumpExamCard.cxx, DumpGEMSMovieGroup.cxx, DumpImageHeaderInfo.cxx, DumpPhilipsECHO.cxx, DumpSiemensBase64.cxx, DumpToshibaDTI.cxx, DumpToshibaDTI2.cxx, DumpVisusChange.cxx, DuplicatePCDE.cxx, ELSCINT1WaveToText.cxx, ExtractEncapsulatedFile.cs, ExtractEncryptedContent.cxx, ExtractIconFromFile.cxx, ExtractImageRegion.cs, ExtractImageRegionWithLUT.cs, FixBrokenJ2K.cxx, FixJAIBugJPEGLS.cxx, FixOrientation.cxx, GenLongSeqs.cxx, GenSeqs.cxx, GetJPEGSamplePrecision.cxx, GetSequenceUltrasound.cxx, GetSubSequenceData.cxx, HelloWorld.cxx, LargeVRDSExplicit.cxx, MakeTemplate.cxx, ManipulateFile.cs, MergeTwoFiles.cxx, MrProtocol.cxx, NewSequence.cs, PatchFile.cxx, QIDO-RS.cxx, ReadAndDumpDICOMDIR.cxx, ReadAndDumpDICOMDIR2.cxx, ReadAndPrintAttributes.cxx, ReadExplicitLengthSQIVR.cxx, ReadGEMSSDO.cxx, ReformatFile.cs, SimplePrint.cs, SimplePrintPatientName.cs, StandardizeFiles.cs, TestReader.cxx, csa2img.cxx, gdcmrtionplan.cxx, gdcmrtplan.cxx, iU22tomultisc.cxx, pmsct_rgb1.cxx, and rle2img.cxx.

10.259.3.4 GetStreamCurrentPosition()

```
size_t gdcm::Reader::GetStreamCurrentPosition ( ) const
```

For wrapped language. return type is compatible with System::FileSize return type Use native std::streampos / std↩ ::streamoff directly from the stream from C++

Examples

ExtractImageRegion.cs.

10.259.3.5 GetStreamPtr()

```
std::istream * gdcm::Reader::GetStreamPtr ( ) const  [inline], [protected]
```

10.259.3.6 Read()

```
virtual bool gdcm::Reader::Read ( )  [virtual]
```

Main function to read a file.

Reimplemented in gdcm::ImageReader, gdcm::ImageRegionReader, gdcm::PixmapReader, gdcm::SegmentReader, and gdcm::SurfaceReader.

Examples

BasicAnonymizer.cs, ChangePrivateTags.cxx, ChangeSequenceUltrasound.cxx, ClinicalTrialAnnotate.cxx, ClinicalTrialIdentificationWorkflow.cs, CreateFakeRTDOSE.cxx, DeriveSeries.cxx, DiffFile.cxx, DumpADAC.cxx, DumpCSA.cs, DumpExamCard.cxx, DumpGEMSMovieGroup.cxx, DumpImageHeaderInfo.cxx, DumpPhilipsECHO.cxx, DumpSiemensBase64.cxx, DumpToshibaDTI.cxx, DumpToshibaDTI2.cxx, DuplicatePCDE.cxx, ELSCINT1WaveToText.cxx, ExtractEncapsulatedFile.cs, ExtractEncryptedContent.cxx, FixBrokenJ2K.cxx, FixOrientation.cxx, GenLongSeqs.cxx, GenSeqs.cxx, GetSequenceUltrasound.cxx, GetSubSequenceData.cxx, HelloWorld.cxx, LargeVRDSExplicit.cxx, MakeTemplate.cxx, ManipulateFile.cs, NewSequence.cs, PatchFile.cxx, QIDO-RS.cxx, ReadAndDumpDICOMDIR.cxx, ReadAndDumpDICOMDIR2.cxx, ReadAndPrintAttributes.cxx, ReadExplicitLengthSQIVR.cxx, ReadGEMSSDO.cxx, ReformatFile.cs, SimplePrint.cs, SimplePrintPatientName.cs, TestReader.cxx, csa2img.cxx, gdcmrtionplan.cxx, gdcmrtplan.cxx, iU22tomultisc.cxx, pmsct_rgb1.cxx, and rle2img.cxx.

10.259.3.7 ReadDataSet()

```
bool gdcm::Reader::ReadDataSet ( )  [protected]
```

10.259.3.8 ReadMetaInformation()

```
bool gdcm::Reader::ReadMetaInformation ( )  [protected]
```

10.259.3.9 ReadPreamble()

```
bool gdcm::Reader::ReadPreamble ( )  [protected]
```

10.259.3.10 ReadSelectedPrivateTags()

```
bool gdcm::Reader::ReadSelectedPrivateTags (
            std::set< PrivateTag > const & ptags,
            bool readvalues = true )
```

Will only read the specified selected private tags.

10.259.3.11 ReadSelectedTags()

```
bool gdcm::Reader::ReadSelectedTags (
            std::set< Tag > const & tags,
            bool readvalues = true )
```

Will only read the specified selected tags.

10.259.3.12 ReadUpToTag()

```
bool gdcm::Reader::ReadUpToTag (
            const Tag & tag,
            std::set< Tag > const & skiptags = std::set< Tag >() )
```

Will read only up to Tag

Parameters

tag	and skipping any tag specified in
skiptags	

Examples

DumpVisusChange.cxx.

10.259.3.13 SetFile()

```
void gdcm::Reader::SetFile (
            File & file )  [inline]
```

Set/Get File.

10.259.3.14 SetFileName()

```
void gdcm::Reader::SetFileName (
            const char * filename_native )
```

Set the filename to open. This will create a std::ifstream internally See SetStream if you are dealing with different std::istream object

Examples

BasicAnonymizer.cs, BasicImageAnonymizer.cs, ChangePrivateTags.cxx, ChangeSequenceUltrasound.cxx, CheckBigEndianBug.cxx, ClinicalTrialAnnotate.cxx, ClinicalTrialIdentificationWorkflow.cs, CompressImage.cxx, CompressLossyJPEG.cs, ConvertToQImage.cxx, CreateFakeRTDOSE.cxx, DecompressImage.cs, DeriveSeries.cxx, DiffFile.cxx, DumpADAC.cxx, DumpCSA.cs, DumpExamCard.cxx, DumpGEMSMovieGroup.cxx, DumpImageHeaderInfo.cxx, DumpPhilipsECHO.cxx, DumpSiemensBase64.cxx, DumpToshibaDTI.cxx, DumpToshibaDTI2.cxx, DumpVisusChange.cxx, DuplicatePCDE.cxx, ELSCINT1WaveToText.cxx, ExtractEncapsulatedFile.cs, ExtractEncryptedContent.cxx, ExtractIconFromFile.cxx, ExtractImageRegion.cs, ExtractImageRegionWithLUT.cs, FixBrokenJ2K.cxx, FixJAIBugJPEGLS.cxx, FixOrientation.cxx, GenLongSeqs.cxx, GenSeqs.cxx, GetArray.cs, GetJPEGSamplePrecision.cxx, GetSequenceUltrasound.cxx, GetSubSequenceData.cxx, HelloVizWorld.cxx, HelloWorld.cxx, LargeVRDSExplicit.cxx, MakeTemplate.cxx, ManipulateFile.cs, MergeTwoFiles.cxx, MrProtocol.cxx, NewSequence.cs, PatchFile.cxx, PrintLUT.cxx, QIDO-RS.cxx, ReadAndDumpDICOMDIR.cxx, ReadAndDumpDICOMDIR2.cxx, ReadAndPrintAttributes.cxx, ReadExplicitLengthSQIVR.cxx, ReadGEMSSDO.cxx, ReadMultiTimesException.cxx, ReadUTF8QtDir.cxx, ReformatFile.cs, RescaleImage.cs, SimplePrint.cs, SimplePrintPatientName.cs, StandardizeFiles.cs, TemplateEmptyImage.cxx, TestReader.cxx, csa2img.cxx, gdcmrtionplan.cxx, gdcmrtplan.cxx, iU22tomultisc.cxx, pmsct_rgb1.cxx, rle2img.cxx, and threadgdcm.cxx.

10.259.3.15 SetStream()

```
void gdcm::Reader::SetStream (
            std::istream & input_stream )  [inline]
```

Set the open-ed stream directly.

Examples

ReadUTF8QtDir.cxx.

10.259.4 Friends And Related Function Documentation

10.259.4.1 StreamImageReader

```
friend class StreamImageReader  [friend]
```

10.259.5 Member Data Documentation

10.259.5.1 F

```
SmartPointer<File> gdcm::Reader::F  [protected]
```

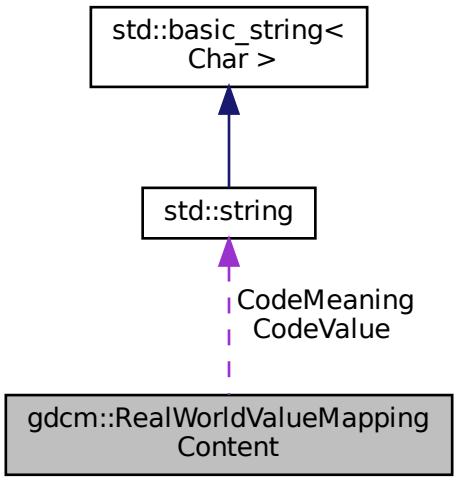
The documentation for this class was generated from the following file:

- gdcmReader.h

10.260 gdcm::RealWorldValueMappingContent Struct Reference

```
#include <gdcmImageHelper.h>
```

Collaboration diagram for gdcm::RealWorldValueMappingContent:



Public Attributes

- std::string CodeMeaning
- std::string CodeValue
- double RealWorldValueIntercept
- double RealWorldValueSlope

10.260.1 Member Data Documentation

10.260.1.1 CodeMeaning

```
std::string gdcm::RealWorldValueMappingContent::CodeMeaning
```

10.260.1.2 CodeValue

```
std::string gdcm::RealWorldValueMappingContent::CodeValue
```

10.260.1.3 RealWorldValueIntercept

```
double gdcm::RealWorldValueMappingContent::RealWorldValueIntercept
```

10.260.1.4 RealWorldValueSlope

```
double gdcm::RealWorldValueMappingContent::RealWorldValueSlope
```

The documentation for this struct was generated from the following file:

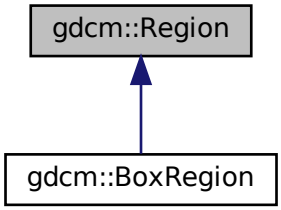
- gdcmImageHelper.h

10.261 gdcm::Region Class Reference

Class for manipulation region.

`#include <gdcmRegion.h>`

Inheritance diagram for gdcm::Region:



Public Member Functions

- Region ()
- virtual ∼Region ()
- virtual size_t Area () const =0

  compute the area
- virtual Region ∗ Clone () const =0
- virtual BoxRegion ComputeBoundingBox ()=0

  Return the Axis-Aligned minimum bounding box for all regions.
- virtual bool Empty () const =0

  return whether this domain is empty:
- virtual bool IsValid () const =0

  return whether this is valid domain
- virtual void Print (std::ostream &os=std::cout) const

  Print.

10.261.1 Detailed Description

Class for manipulation region.

10.261.2 Constructor & Destructor Documentation

10.261.2.1 Region()

```
gdcm::Region::Region ( )
```

10.261.2.2 ∼Region()

```
virtual gdcm::Region::~Region ( )  [virtual]
```

10.261.3 Member Function Documentation

10.261.3.1 Area()

```
virtual size_t gdcm::Region::Area ( ) const  [pure virtual]
```

compute the area

Implemented in gdcm::BoxRegion.

10.261.3.2 Clone()

```
virtual Region * gdcm::Region::Clone ( ) const  [pure virtual]
```

Implemented in gdcm::BoxRegion.

10.261.3.3 ComputeBoundingBox()

```
virtual BoxRegion gdcm::Region::ComputeBoundingBox ( )  [pure virtual]
```

Return the Axis-Aligned minimum bounding box for all regions.

Implemented in gdcm::BoxRegion.

10.261.3.4 Empty()

```
virtual bool gdcm::Region::Empty ( ) const  [pure virtual]
```

return whether this domain is empty:

Implemented in gdcm::BoxRegion.

10.261.3.5 IsValid()

```
virtual bool gdcm::Region::IsValid ( ) const  [pure virtual]
```

return whether this is valid domain

Implemented in gdcm::BoxRegion.

10.261.3.6 Print()

```
virtual void gdcm::Region::Print (
            std::ostream & os = std::cout ) const  [virtual]
```

Print.

Reimplemented in gdcm::BoxRegion.

Referenced by gdcm::operator<<().

The documentation for this class was generated from the following file:

- gdcmRegion.h

10.262 gdcm::Rescaler Class Reference

Rescale class.

```
#include <gdcmRescaler.h>
```

Public Member Functions

- Rescaler ()
- ~Rescaler ()=default
- PixelFormat::ScalarType ComputeInterceptSlopePixelType ()
- PixelFormat ComputePixelTypeFromMinMax ()
- double GetIntercept () const
- double GetSlope () const
- bool InverseRescale (char *out, const char *in, size_t n)

  Inverse transform.
- bool Rescale (char *out, const char *in, size_t n)

  Direct transform.
- void SetIntercept (double i)

  Set Intercept: used for both direct&inverse transformation.
- void SetMinMaxForPixelType (double min, double max)
- void SetPixelFormat (PixelFormat const &pf)

  Set Pixel Format of input data.
- void SetSlope (double s)

  Set Slope: user for both direct&inverse transformation.
- void SetTargetPixelType (PixelFormat const &targetst)
- void SetUseTargetPixelType (bool b)

  Override default behavior of Rescale.

Protected Member Functions

- template<typename TIn >
  void InverseRescaleFunctionIntoBestFit (char *out, const TIn *in, size_t n)
- template<typename TIn >
  void RescaleFunctionIntoBestFit (char *out, const TIn *in, size_t n)

10.262.1 Detailed Description

Rescale class.

This class is meant to apply the linear transform of Stored Pixel Value to Real World Value. This is mostly found in CT or PET dataset, where the value are stored using one type, but need to be converted to another scale using a linear transform. There are basically two cases: In CT: the linear transform is generally integer based. E.g. the Stored Pixel Type is unsigned short 12bits, but to get Hounsfield unit, one need to apply the linear transform:

$$RWV = 1. * SV - 1024$$

So the best scalar to store the Real World Value will be 16 bits signed type.

In PET: the linear transform is generally floating point based. Since the dynamic range can be quite high, the Rescale Slope / Rescale Intercept can be changing throughout the Series. So it is important to read all linear transform and deduce the best Pixel Type only at the end (when all the images to be read have been parsed).

Warning

Internally any time a floating point value is found either in the Rescale Slope or the Rescale Intercept it is assumed that the best matching output pixel type is FLOAT64 (in previous implementation it was FLOAT32). Because VR:DS is closer to a 64bits floating point type FLOAT64 is thus a best matching pixel type for the floating point transformation.

Example: Let say input is FLOAT64, and we want UINT16 as output, we would do:

```
Rescaler ir;
ir.SetIntercept( 0 );
ir.SetSlope( 5.6789 );
ir.SetPixelFormat( FLOAT64 );
ir.SetMinMaxForPixelType( ((PixelFormat)UINT16).GetMin(), ((PixelFormat)UINT16).GetMax() );
ir.InverseRescale(output,input,numberofbytes );
```

Note

handle floating point transformation back and forth to integer properly (no loss)

See also

Unpacker12Bits

Examples

RescaleImage.cs.

10.262.2 Constructor & Destructor Documentation

10.262.2.1 Rescaler()

```
gdcm::Rescaler::Rescaler ( )  [inline]
```

10.262.2.2 ~Rescaler()

```
gdcm::Rescaler::~Rescaler ( )  [default]
```

10.262.3 Member Function Documentation

10.262.3.1 ComputeInterceptSlopePixelType()

```
PixelFormat::ScalarType gdcm::Rescaler::ComputeInterceptSlopePixelType ( )
```

Compute the Pixel Format of the output data Used for direct transformation

Examples

RescaleImage.cs.

10.262.3.2 ComputePixelTypeFromMinMax()

```
PixelFormat gdcm::Rescaler::ComputePixelTypeFromMinMax ( )
```

Compute the Pixel Format of the output data Used for inverse transformation

10.262.3.3 GetIntercept()

```
double gdcm::Rescaler::GetIntercept ( ) const  [inline]
```

10.262.3.4 GetSlope()

```
double gdcm::Rescaler::GetSlope ( ) const  [inline]
```

10.262.3.5 InverseRescale()

```
bool gdcm::Rescaler::InverseRescale (
            char * out,
            const char * in,
            size_t n )
```

Inverse transform.

10.262.3.6 InverseRescaleFunctionIntoBestFit()

```
template<typename TIn >
void gdcm::Rescaler::InverseRescaleFunctionIntoBestFit (
            char * out,
            const TIn * in,
            size_t n )  [protected]
```

10.262.3.7 Rescale()

```
bool gdcm::Rescaler::Rescale (
            char * out,
            const char * in,
            size_t n )
```

Direct transform.

Examples

RescaleImage.cs.

10.262.3.8 RescaleFunctionIntoBestFit()

```
template<typename TIn >
void gdcm::Rescaler::RescaleFunctionIntoBestFit (
             char * out,
             const TIn * in,
             size_t n )  [protected]
```

10.262.3.9 SetIntercept()

```
void gdcm::Rescaler::SetIntercept (
             double i )  [inline]
```

Set Intercept: used for both direct&inverse transformation.

Examples

RescaleImage.cs.

10.262.3.10 SetMinMaxForPixelType()

```
void gdcm::Rescaler::SetMinMaxForPixelType (
             double min,
             double max )
```

Set target interval for output data. A best match will be computed (if possible) Used for inverse transformation

10.262.3.11 SetPixelFormat()

```
void gdcm::Rescaler::SetPixelFormat (
             PixelFormat const & pf )  [inline]
```

Set Pixel Format of input data.

Examples

RescaleImage.cs.

10.262.3.12 SetSlope()

```
void gdcm::Rescaler::SetSlope (
            double s )  [inline]
```

Set Slope: user for both direct&inverse transformation.

Examples

RescaleImage.cs.

10.262.3.13 SetTargetPixelType()

```
void gdcm::Rescaler::SetTargetPixelType (
            PixelFormat const & targetst )
```

By default (when UseTargetPixelType is false), a best matching Target Pixel Type is computed. However user can override this auto selection by switching UseTargetPixelType:true and also specifying the specifix Target Pixel Type

10.262.3.14 SetUseTargetPixelType()

```
void gdcm::Rescaler::SetUseTargetPixelType (
            bool b )
```

Override default behavior of Rescale.

The documentation for this class was generated from the following file:

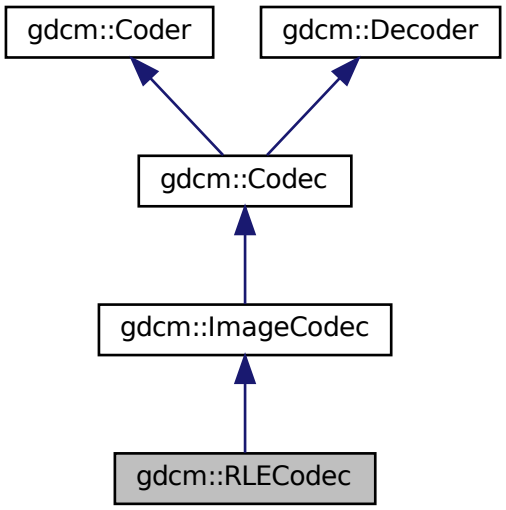
- gdcmRescaler.h

10.263 gdcm::RLECodec Class Reference

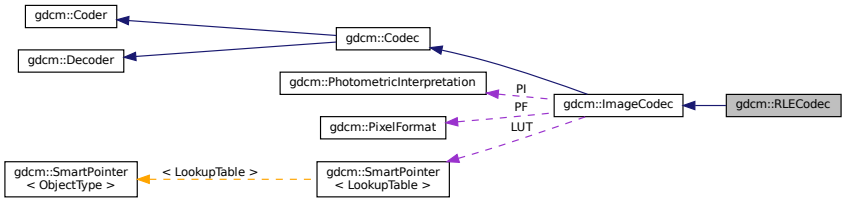
Class to do RLE.

```
#include <gdcmRLECodec.h>
```

Inheritance diagram for gdcm::RLECodec:



Collaboration diagram for gdcm::RLECodec:



Public Member Functions

- RLECodec ()
- ~RLECodec () override
- bool CanCode (TransferSyntax const &ts) const override

  Return whether this coder support this transfer syntax (can code it)
- bool CanDecode (TransferSyntax const &ts) const override

  Return whether this decoder support this transfer syntax (can decode it)
- ImageCodec * Clone () const override
- bool Code (DataElement const &in, DataElement &out) override

  Code.

- bool Decode (DataElement const &is, DataElement &os) override

  Decode.
- unsigned long GetBufferLength () const
- bool GetHeaderInfo (std::istream &is, TransferSyntax &ts) override
- void SetBufferLength (unsigned long l)
- void SetLength (unsigned long l)

Protected Member Functions

- bool AppendFrameEncode (std::ostream &out, const char *data, size_t datalen) override
- bool AppendRowEncode (std::ostream &out, const char *data, size_t datalen) override
- bool DecodeByStreams (std::istream &is, std::ostream &os) override
- bool DecodeExtent (char *buffer, unsigned int XMin, unsigned int XMax, unsigned int YMin, unsigned int YMax, unsigned int ZMin, unsigned int ZMax, std::istream &is)
- bool IsFrameEncoder () override
- bool IsRowEncoder () override
- bool StartEncode (std::ostream &) override
- bool StopEncode (std::ostream &) override

Friends

- class ImageRegionReader

Additional Inherited Members

10.263.1 Detailed Description

Class to do RLE.

Note

ANSI X3.9 A.4.2 RLE Compression Annex G defines a RLE Compression Transfer Syntax. This transfer Syntax is identified by the UID value "1.2.840.10008.1.2.5". If the object allows multi-frame images in the pixel data field, then each frame shall be encoded separately. Each frame shall be encoded in one and only one Fragment (see PS 3.5.8.2).

10.263.2 Constructor & Destructor Documentation

10.263.2.1 RLECodec()

```
gdcm::RLECodec::RLECodec ( )
```

10.263.2.2 ~RLECodec()

```
gdcm::RLECodec::~RLECodec ( )  [override]
```

10.263.3 Member Function Documentation

10.263.3.1 AppendFrameEncode()

```
bool gdcm::RLECodec::AppendFrameEncode (
            std::ostream & out,
            const char * data,
            size_t datalen )  [override], [protected], [virtual]
```

Reimplemented from gdcm::ImageCodec.

10.263.3.2 AppendRowEncode()

```
bool gdcm::RLECodec::AppendRowEncode (
            std::ostream & out,
            const char * data,
            size_t datalen )  [override], [protected], [virtual]
```

Reimplemented from gdcm::ImageCodec.

10.263.3.3 CanCode()

```
bool gdcm::RLECodec::CanCode (
            TransferSyntax const &  ) const  [override], [virtual]
```

Return whether this coder support this transfer syntax (can code it)

Reimplemented from gdcm::ImageCodec.

10.263.3.4 CanDecode()

```
bool gdcm::RLECodec::CanDecode (
            TransferSyntax const &  ) const  [override], [virtual]
```

Return whether this decoder support this transfer syntax (can decode it)

Reimplemented from gdcm::ImageCodec.

10.263.3.5 Clone()

```
ImageCodec * gdcm::RLECodec::Clone ( ) const  [override], [virtual]
```

Implements gdcm::ImageCodec.

10.263.3.6 Code()

```
bool gdcm::RLECodec::Code (
            DataElement const & in_,
            DataElement & out_ )  [override], [virtual]
```

Code.

Reimplemented from gdcm::Coder.

10.263.3.7 Decode()

```
bool gdcm::RLECodec::Decode (
            DataElement const & ,
            DataElement &  )  [override], [virtual]
```

Decode.

Reimplemented from gdcm::ImageCodec.

10.263.3.8 DecodeByStreams()

```
bool gdcm::RLECodec::DecodeByStreams (
            std::istream & is,
            std::ostream & os )  [override], [protected], [virtual]
```

Reimplemented from gdcm::ImageCodec.

10.263.3.9 DecodeExtent()

```
bool gdcm::RLECodec::DecodeExtent (
            char * buffer,
            unsigned int XMin,
            unsigned int XMax,
            unsigned int YMin,
            unsigned int YMax,
            unsigned int ZMin,
            unsigned int ZMax,
            std::istream & is )  [protected]
```

10.263.3.10 GetBufferLength()

```
unsigned long gdcm::RLECodec::GetBufferLength ( ) const  [inline]
```

10.263.3.11 GetHeaderInfo()

```
bool gdcm::RLECodec::GetHeaderInfo (
            std::istream & is,
            TransferSyntax & ts )  [override], [virtual]
```

Reimplemented from gdcm::ImageCodec.

10.263.3.12 IsFrameEncoder()

```
bool gdcm::RLECodec::IsFrameEncoder ( )  [override], [protected], [virtual]
```

Reimplemented from gdcm::ImageCodec.

10.263.3.13 IsRowEncoder()

```
bool gdcm::RLECodec::IsRowEncoder ( )  [override], [protected], [virtual]
```

Reimplemented from gdcm::ImageCodec.

10.263.3.14 SetBufferLength()

```
void gdcm::RLECodec::SetBufferLength (
            unsigned long l )  [inline]
```

10.263.3.15 SetLength()

```
void gdcm::RLECodec::SetLength (
            unsigned long l )  [inline]
```

10.263.3.16 StartEncode()

```
bool gdcm::RLECodec::StartEncode (
            std::ostream &  )  [override], [protected], [virtual]
```

Reimplemented from gdcm::ImageCodec.

10.263.3.17 StopEncode()

```
bool gdcm::RLECodec::StopEncode (
            std::ostream &  )  [override], [protected], [virtual]
```

Reimplemented from gdcm::ImageCodec.

10.263.4 Friends And Related Function Documentation

10.263.4.1 ImageRegionReader

```
friend class ImageRegionReader  [friend]
```

The documentation for this class was generated from the following file:

- gdcmRLECodec.h

10.264 gdcm::network::RoleSelectionSub Class Reference

RoleSelectionSub.

```
#include <gdcmRoleSelectionSub.h>
```

Public Member Functions

- RoleSelectionSub ()
- void Print (std::ostream &os) const
- std::istream & Read (std::istream &is)
- void SetTuple (const char *uid, uint8_t scurole, uint8_t scprole)
- size_t Size () const
- const std::ostream & Write (std::ostream &os) const

10.264.1 Detailed Description

RoleSelectionSub.

PS 3.7 Table D.3-9 SCP/SCU ROLE SELECTION SUB-ITEM FIELDS (A-ASSOCIATE-RQ)

10.264.2 Constructor & Destructor Documentation

10.264.2.1 RoleSelectionSub()

```
gdcm::network::RoleSelectionSub::RoleSelectionSub ( )
```

10.264.3 Member Function Documentation

10.264.3.1 Print()

```
void gdcm::network::RoleSelectionSub::Print (
             std::ostream & os ) const
```

10.264.3.2 Read()

```
std::istream & gdcm::network::RoleSelectionSub::Read (
             std::istream & is )
```

10.264.3.3 SetTuple()

```
void gdcm::network::RoleSelectionSub::SetTuple (
             const char * uid,
             uint8_t scurole,
             uint8_t scprole )
```

10.264.3.4 Size()

```
size_t gdcm::network::RoleSelectionSub::Size ( ) const
```

10.264.3.5 Write()

```
const std::ostream & gdcm::network::RoleSelectionSub::Write (
             std::ostream & os ) const
```

The documentation for this class was generated from the following file:

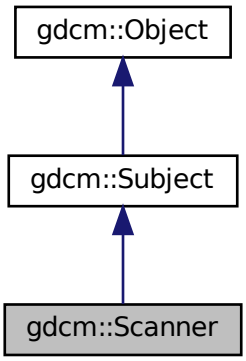
- gdcmRoleSelectionSub.h

10.265 gdcm::Scanner Class Reference

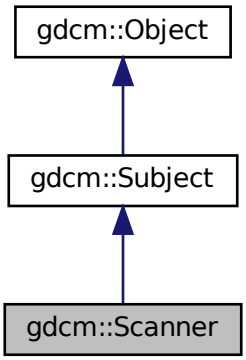
Scanner.

`#include <gdcmScanner.h>`

Inheritance diagram for gdcm::Scanner:



Collaboration diagram for gdcm::Scanner:



Classes

- struct ltstr

Public Types

- typedef MappingType::const_iterator ConstIterator
- typedef std::map< const char *, TagToValue, ltstr > MappingType
- typedef std::map< Tag, const char * > TagToValue
- typedef TagToValue::value_type TagToValueValueType
- typedef std::set< std::string > ValuesType

Public Member Functions

- Scanner ()
- ~Scanner () override
- void AddPrivateTag (PrivateTag const &t)
- void AddSkipTag (Tag const &t)

  Add a tag that will need to be skipped. Those are root level skip tags.
- void AddTag (Tag const &t)

  Add a tag that will need to be read. Those are root level tags.
- ConstIterator Begin () const
- void ClearSkipTags ()
- void ClearTags ()
- ConstIterator End () const
- Directory::FilenamesType GetAllFilenamesFromTagToValue (Tag const &t, const char *valueref) const
- const char * GetFilenameFromTagToValue (Tag const &t, const char *valueref) const
- Directory::FilenamesType const & GetFilenames () const
- Directory::FilenamesType GetKeys () const
- TagToValue const & GetMapping (const char *filename) const

  Get the std::map mapping filenames to value for file 'filename'.
- TagToValue const & GetMappingFromTagToValue (Tag const &t, const char *value) const

  See GetFilenameFromTagToValue(). This is simply GetFilenameFromTagToValue followed.
- MappingType const & GetMappings () const

  Mappings are the mapping from a particular tag to the map, mapping filename to value:
- Directory::FilenamesType GetOrderedValues (Tag const &t) const
- const char * GetValue (const char *filename, Tag const &t) const
- ValuesType const & GetValues () const

  Get all the values found (in lexicographic order)
- ValuesType GetValues (Tag const &t) const

  Get all the values found (in lexicographic order) associated with Tag 't'.
- bool IsKey (const char *filename) const
- void Print (std::ostream &os) const override

  Print result.
- void PrintTable (std::ostream &os) const
- bool Scan (Directory::FilenamesType const &filenames)

  Start the scan !

Static Public Member Functions

- static SmartPointer< Scanner > New ()

  for wrapped language: instantiate a reference counted object

Protected Member Functions

- void ProcessPublicTag (StringFilter &sf, const char *filename)

Friends

- std::ostream & operator<< (std::ostream &_os, const Scanner &s)

10.265.1 Detailed Description

Scanner.

This filter is meant for quickly browsing a FileSet (a set of files on disk). Special consideration are taken so as to read the minimum amount of information in each file in order to retrieve the user specified set of DICOM Attribute.

This filter is dealing with both VRASCII and VRBINARY element, thanks to the help of StringFilter

Warning

IMPORTANT In case of file where tags are not ordered (illegal as per DICOM specification), the output will be missing information

Note

implementation details. All values are stored in a std::set of std::string. Then the address of the cstring underlying the std::string is used in the std::map.

This class implement the Subject/Observer pattern trigger the following events:

- ProgressEvent
- StartEvent
- EndEvent

Examples

DiscriminateVolume.cxx, DumpToSQLITE3.cxx, SortImage.cxx, and VolumeSorter.cxx.

10.265.2 Member Typedef Documentation

10.265.2.1 ConstIterator

```
typedef MappingType::const_iterator gdcm::Scanner::ConstIterator
```

10.265.2.2 MappingType

```
typedef std::map<const char *,TagToValue, ltstr> gdcm::Scanner::MappingType
```

10.265.2.3 TagToValue

```
typedef std::map<Tag, const char*> gdcm::Scanner::TagToValue
```

struct to map a filename to a value Implementation note: all std::map in this class will be using const char * and not std::string since we are pointing to existing std::string (hold in a std::vector) this avoid an extra copy of the byte array. Tag are used as Tag class since sizeof(tag) <= sizeof(pointer)

10.265.2.4 TagToValueValueType

```
typedef TagToValue::value_type gdcm::Scanner::TagToValueValueType
```

10.265.2.5 ValuesType

```
typedef std::set< std::string > gdcm::Scanner::ValuesType
```

10.265.3 Constructor & Destructor Documentation

10.265.3.1 Scanner()

```
gdcm::Scanner::Scanner ( )  [inline]
```

10.265.3.2 ~Scanner()

```
gdcm::Scanner::~Scanner ( )  [override]
```

10.265.4 Member Function Documentation

10.265.4.1 AddPrivateTag()

```
void gdcm::Scanner::AddPrivateTag (
            PrivateTag const & t )
```

10.265.4.2 AddSkipTag()

```
void gdcm::Scanner::AddSkipTag (
            Tag const & t )
```

Add a tag that will need to be skipped. Those are root level skip tags.

10.265.4.3 AddTag()

```
void gdcm::Scanner::AddTag (
            Tag const & t )
```

Add a tag that will need to be read. Those are root level tags.

Examples

DiscriminateVolume.cxx, DumpToSQLITE3.cxx, SortImage.cxx, and VolumeSorter.cxx.

10.265.4.4 Begin()

```
ConstIterator gdcm::Scanner::Begin ( ) const  [inline]
```

10.265.4.5 ClearSkipTags()

```
void gdcm::Scanner::ClearSkipTags ( )
```

10.265.4.6 ClearTags()

```
void gdcm::Scanner::ClearTags ( )
```

10.265.4.7 End()

```
ConstIterator gdcm::Scanner::End ( ) const  [inline]
```

10.265.4.8 GetAllFilenamesFromTagToValue()

```
Directory::FilenamesType gdcm::Scanner::GetAllFilenamesFromTagToValue (
            Tag const & t,
            const char * valueref ) const
```

Will loop over all files and return a vector of std::strings of filenames where value match the reference value 'valueref'

10.265.4.9 GetFilenameFromTagToValue()

```
const char * gdcm::Scanner::GetFilenameFromTagToValue (
            Tag const & t,
            const char * valueref ) const
```

Will loop over all files and return the first file where value match the reference value 'valueref'

10.265.4.10 GetFilenames()

```
Directory::FilenamesType const & gdcm::Scanner::GetFilenames ( ) const  [inline]
```

10.265.4.11 GetKeys()

```
Directory::FilenamesType gdcm::Scanner::GetKeys ( ) const
```

Return the list of filename that are key in the internal map, which means those filename were properly parsed

Examples

VolumeSorter.cxx.

10.265.4.12 GetMapping()

```
TagToValue const & gdcm::Scanner::GetMapping (
            const char * filename ) const
```

Get the std::map mapping filenames to value for file 'filename'.

Examples

DumpToSQLITE3.cxx.

10.265.4.13 GetMappingFromTagToValue()

```
TagToValue const & gdcm::Scanner::GetMappingFromTagToValue (
            Tag const & t,
            const char * value ) const
```

See GetFilenameFromTagToValue(). This is simply GetFilenameFromTagToValue followed.

10.265.4.14 GetMappings()

```
MappingType const & gdcm::Scanner::GetMappings ( ) const  [inline]
```

Mappings are the mapping from a particular tag to the map, mapping filename to value:

10.265.4.15 GetOrderedValues()

```
Directory::FilenamesType gdcm::Scanner::GetOrderedValues (
            Tag const & t ) const
```

Get all the values found (in a vector) associated with Tag 't' This function is identical to GetValues, but is accessible from the wrapped layer (python, C#, java)

10.265.4.16 GetValue()

```
const char * gdcm::Scanner::GetValue (
            const char * filename,
            Tag const & t ) const
```

Retrieve the value found for tag: t associated with file: filename This is meant for a single short call. If multiple calls (multiple tags) should be done, prefer the GetMapping function, and then reuse the TagToValue hash table.

Warning

Tag 't' should have been added via AddTag() prior to the Scan() call !

10.265.4.17 GetValues() [1/2]

```
ValuesType const & gdcm::Scanner::GetValues ( ) const  [inline]
```

Get all the values found (in lexicographic order)

Examples

SortImage.cxx, and VolumeSorter.cxx.

10.265.4.18 GetValues() [2/2]

```
ValuesType gdcm::Scanner::GetValues (
            Tag const & t ) const
```

Get all the values found (in lexicographic order) associated with Tag 't'.

10.265.4.19 IsKey()

```
bool gdcm::Scanner::IsKey (
            const char * filename ) const
```

Check if filename is a key in the Mapping table. returns true only of file can be found, which means the file was indeed a DICOM file that could be processed

Examples

DumpToSQLITE3.cxx.

10.265.4.20 New()

```
static SmartPointer< Scanner > gdcm::Scanner::New ( )  [inline], [static]
```

for wrapped language: instantiate a reference counted object

10.265.4.21 Print()

```
void gdcm::Scanner::Print (
            std::ostream & os ) const  [override], [virtual]
```

Print result.

Reimplemented from gdcm::Object.

10.265.4.22 PrintTable()

```
void gdcm::Scanner::PrintTable (
            std::ostream & os ) const
```

10.265.4.23 ProcessPublicTag()

```
void gdcm::Scanner::ProcessPublicTag (
            StringFilter & sf,
            const char * filename )  [protected]
```

10.265.4.24 Scan()

```
bool gdcm::Scanner::Scan (
            Directory::FilenamesType const & filenames )
```

Start the scan !

Examples

DiscriminateVolume.cxx, DumpToSQLITE3.cxx, SortImage.cxx, and VolumeSorter.cxx.

10.265.5 Friends And Related Function Documentation

10.265.5.1 operator<<

```
std::ostream & operator<< (
             std::ostream & _os,
             const Scanner & s )  [friend]
```

The documentation for this class was generated from the following file:

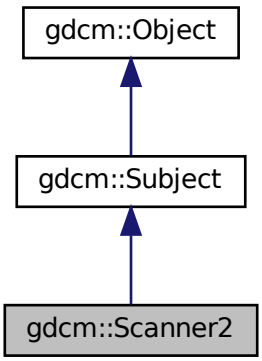
- gdcmScanner.h

10.266 gdcm::Scanner2 Class Reference

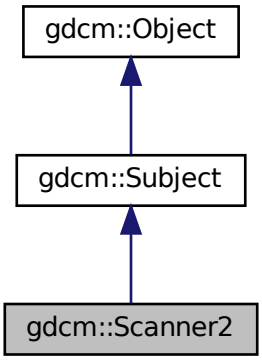
Scanner2.

```
#include <gdcmScanner2.h>
```

Inheritance diagram for gdcm::Scanner2:



Collaboration diagram for gdcm::Scanner2:



Classes

- struct ltstr

Public Types

- typedef PrivateMappingType::const_iterator PrivateConstIterator
- typedef std::map< const char *, PrivateTagToValue, ltstr > PrivateMappingType
- typedef std::map< PrivateTag, const char * > PrivateTagToValue
- typedef PrivateTagToValue::value_type PrivateTagToValueValueType
- typedef PublicMappingType::const_iterator PublicConstIterator
- typedef std::map< const char *, PublicTagToValue, ltstr > PublicMappingType
- typedef std::map< Tag, const char * > PublicTagToValue
- typedef PublicTagToValue::value_type PublicTagToValueValueType
- typedef std::set< std::string > ValuesType

Public Member Functions

- Scanner2 ()
- ~Scanner2 () override
- bool AddPrivateTag (PrivateTag const &pt)
- bool AddPublicTag (Tag const &t)

  Add a tag that will need to be read. Those are root level tags.
- bool AddSkipTag (Tag const &t)

  Add a tag that will need to be skipped. Those are root level skip tags.
- PublicConstIterator Begin () const
- void ClearPrivateTags ()

- void ClearPublicTags ()
- void ClearSkipTags ()
- PublicConstIterator End () const
- Directory::FilenamesType GetAllFilenamesFromPrivateTagToValue (PrivateTag const &pt, const char *valueref) const
- Directory::FilenamesType GetAllFilenamesFromPublicTagToValue (Tag const &t, const char *valueref) const
- const char * GetFilenameFromPrivateTagToValue (PrivateTag const &pt, const char *valueref) const
- const char * GetFilenameFromPublicTagToValue (Tag const &t, const char *valueref) const
- Directory::FilenamesType const & GetFilenames () const

  Return the list of filenames.
- Directory::FilenamesType GetKeys () const
- PrivateTagToValue const & GetMappingFromPrivateTagToValue (PrivateTag const &pt, const char *value) const
- PublicTagToValue const & GetMappingFromPublicTagToValue (Tag const &t, const char *value) const

  See GetFilenameFromTagToValue(). This is simply GetFilenameFromTagToValue followed.
- PrivateTagToValue const & GetPrivateMapping (const char *filename) const
- PrivateMappingType const & GetPrivateMappings () const
- Directory::FilenamesType GetPrivateOrderedValues (PrivateTag const &pt) const
- const char * GetPrivateValue (const char *filename, PrivateTag const &t) const
- ValuesType GetPrivateValues (PrivateTag const &pt) const

  Get all the values found (in lexicographic order) associated with PrivateTag 'pt'.
- PublicTagToValue const & GetPublicMapping (const char *filename) const

  Get the std::map mapping filenames to value for file 'filename'.
- PublicMappingType const & GetPublicMappings () const

  Mappings are the mapping from a particular tag to the map, mapping filename to value:
- Directory::FilenamesType GetPublicOrderedValues (Tag const &t) const
- const char * GetPublicValue (const char *filename, Tag const &t) const
- ValuesType GetPublicValues (Tag const &t) const

  Get all the values found (in lexicographic order) associated with Tag 't'.
- ValuesType const & GetValues () const

  Get all the values found (in lexicographic order)
- bool IsKey (const char *filename) const
- void Print (std::ostream &os) const override

  Print result.
- void PrintTable (std::ostream &os, bool header=false) const

  Print result as CSV table.
- PrivateConstIterator PrivateBegin () const
- PrivateConstIterator PrivateEnd () const
- bool Scan (Directory::FilenamesType const &filenames)

  Start the scan !

Static Public Member Functions

- static SmartPointer< Scanner2 > New ()

  for wrapped language: instantiate a reference counted object

Protected Member Functions

- void ProcessPrivateTag (StringFilter &sf, const char *filename)
- void ProcessPublicTag (StringFilter &sf, const char *filename)

Friends

- std::ostream & operator<< (std::ostream &_os, const Scanner2 &s)

10.266.1 Detailed Description

Scanner2.

This filter is meant for quickly browsing a FileSet (a set of files on disk). Special consideration are taken so as to read the minimum amount of information in each file in order to retrieve the user specified set of DICOM Attribute.

This filter is dealing with both VRASCII and VRBINARY element, thanks to the help of StringFilter

Warning

IMPORTANT In case of file where tags are not ordered (illegal as per DICOM specification), the output will be missing information

Note

implementation details. All values are stored in a std::set of std::string. Then the address of the cstring underlying the std::string is used in the std::map.

This class implement the Subject/Observer pattern trigger the following events:

- ProgressEvent
- StartEvent
- EndEvent

10.266.2 Member Typedef Documentation

10.266.2.1 PrivateConstIterator

```
typedef PrivateMappingType::const_iterator gdcm::Scanner2::PrivateConstIterator
```

10.266.2.2 PrivateMappingType

```
typedef std::map<const char *,PrivateTagToValue, ltstr> gdcm::Scanner2::PrivateMappingType
```

10.266.2.3 PrivateTagToValue

```
typedef std::map<PrivateTag, const char*> gdcm::Scanner2::PrivateTagToValue
```

10.266.2.4 PrivateTagToValueValueType

```
typedef PrivateTagToValue::value_type gdcm::Scanner2::PrivateTagToValueValueType
```

10.266.2.5 PublicConstIterator

```
typedef PublicMappingType::const_iterator gdcm::Scanner2::PublicConstIterator
```

10.266.2.6 PublicMappingType

```
typedef std::map<const char *,PublicTagToValue, ltstr> gdcm::Scanner2::PublicMappingType
```

10.266.2.7 PublicTagToValue

```
typedef std::map<Tag, const char*> gdcm::Scanner2::PublicTagToValue
```

struct to map a filename to a value Implementation note: all std::map in this class will be using const char * and not std::string since we are pointing to existing std::string (held in a std::vector) this avoid an extra copy of the byte array. Tag are used as Tag class since sizeof(tag) <= sizeof(pointer)

10.266.2.8 PublicTagToValueValueType

```
typedef PublicTagToValue::value_type gdcm::Scanner2::PublicTagToValueValueType
```

10.266.2.9 ValuesType

```
typedef std::set< std::string > gdcm::Scanner2::ValuesType
```

10.266.3 Constructor & Destructor Documentation

10.266.3.1 Scanner2()

```
gdcm::Scanner2::Scanner2 ( )  [inline]
```

10.266.3.2 ~Scanner2()

```
gdcm::Scanner2::~Scanner2 ( )  [override]
```

10.266.4 Member Function Documentation

10.266.4.1 AddPrivateTag()

```
bool gdcm::Scanner2::AddPrivateTag (
            PrivateTag const & pt )
```

10.266.4.2 AddPublicTag()

```
bool gdcm::Scanner2::AddPublicTag (
            Tag const & t )
```

Add a tag that will need to be read. Those are root level tags.

10.266.4.3 AddSkipTag()

```
bool gdcm::Scanner2::AddSkipTag (
            Tag const & t )
```

Add a tag that will need to be skipped. Those are root level skip tags.

10.266.4.4 Begin()

```
PublicConstIterator gdcm::Scanner2::Begin ( ) const  [inline]
```

10.266.4.5 ClearPrivateTags()

```
void gdcm::Scanner2::ClearPrivateTags ( )
```

10.266.4.6 ClearPublicTags()

```
void gdcm::Scanner2::ClearPublicTags ( )
```

10.266.4.7 ClearSkipTags()

```
void gdcm::Scanner2::ClearSkipTags ( )
```

10.266.4.8 End()

```
PublicConstIterator gdcm::Scanner2::End ( ) const  [inline]
```

10.266.4.9 GetAllFilenamesFromPrivateTagToValue()

```
Directory::FilenamesType gdcm::Scanner2::GetAllFilenamesFromPrivateTagToValue (
            PrivateTag const & pt,
            const char * valueref ) const
```

10.266.4.10 GetAllFilenamesFromPublicTagToValue()

```
Directory::FilenamesType gdcm::Scanner2::GetAllFilenamesFromPublicTagToValue (
            Tag const & t,
            const char * valueref ) const
```

Will loop over all files and return a vector of std::strings of filenames where value match the reference value 'valueref'

10.266.4.11 GetFilenameFromPrivateTagToValue()

```
const char * gdcm::Scanner2::GetFilenameFromPrivateTagToValue (
            PrivateTag const & pt,
            const char * valueref ) const
```

10.266.4.12 GetFilenameFromPublicTagToValue()

```
const char * gdcm::Scanner2::GetFilenameFromPublicTagToValue (
            Tag const & t,
            const char * valueref ) const
```

Will loop over all files and return the first file where value match the reference value 'valueref'

10.266.4.13 GetFilenames()

```
Directory::FilenamesType const & gdcm::Scanner2::GetFilenames ( ) const  [inline]
```

Return the list of filenames.

10.266.4.14 GetKeys()

```
Directory::FilenamesType gdcm::Scanner2::GetKeys ( ) const
```

Return the list of filename that are key in the internal map, which means those filename were properly parsed

10.266.4.15 GetMappingFromPrivateTagToValue()

```
PrivateTagToValue const & gdcm::Scanner2::GetMappingFromPrivateTagToValue (
            PrivateTag const & pt,
            const char * value ) const
```

10.266.4.16 GetMappingFromPublicTagToValue()

```
PublicTagToValue const & gdcm::Scanner2::GetMappingFromPublicTagToValue (
            Tag const & t,
            const char * value ) const
```

See GetFilenameFromTagToValue(). This is simply GetFilenameFromTagToValue followed.

10.266.4.17 GetPrivateMapping()

```
PrivateTagToValue const & gdcm::Scanner2::GetPrivateMapping (
            const char * filename ) const
```

10.266.4.18 GetPrivateMappings()

```
PrivateMappingType const & gdcm::Scanner2::GetPrivateMappings ( ) const  [inline]
```

10.266.4.19 GetPrivateOrderedValues()

```
Directory::FilenamesType gdcm::Scanner2::GetPrivateOrderedValues (
            PrivateTag const & pt ) const
```

10.266.4.20 GetPrivateValue()

```
const char * gdcm::Scanner2::GetPrivateValue (
            const char * filename,
            PrivateTag const & t ) const
```

10.266.4.21 GetPrivateValues()

```
ValuesType gdcm::Scanner2::GetPrivateValues (
            PrivateTag const & pt ) const
```

Get all the values found (in lexicographic order) associated with PrivateTag 'pt'.

10.266.4.22 GetPublicMapping()

```
PublicTagToValue const & gdcm::Scanner2::GetPublicMapping (
            const char * filename ) const
```

Get the std::map mapping filenames to value for file 'filename'.

10.266.4.23 GetPublicMappings()

```
PublicMappingType const & gdcm::Scanner2::GetPublicMappings ( ) const  [inline]
```

Mappings are the mapping from a particular tag to the map, mapping filename to value:

10.266.4.24 GetPublicOrderedValues()

```
Directory::FilenamesType gdcm::Scanner2::GetPublicOrderedValues (
            Tag const & t ) const
```

Get all the values found (in a vector) associated with Tag 't' This function is identical to GetValues, but is accessible from the wrapped layer (python, C#, java)

10.266.4.25 GetPublicValue()

```
const char * gdcm::Scanner2::GetPublicValue (
            const char * filename,
            Tag const & t ) const
```

Retrieve the value found for tag: t associated with file: filename This is meant for a single short call. If multiple calls (multiple tags) should be done, prefer the GetMapping function, and then reuse the TagToValue hash table.

Warning

Tag 't' should have been added via AddTag() prior to the Scan() call !

10.266.4.26 GetPublicValues()

```
ValuesType gdcm::Scanner2::GetPublicValues (
            Tag const & t ) const
```

Get all the values found (in lexicographic order) associated with Tag 't'.

10.266.4.27 GetValues()

```
ValuesType const & gdcm::Scanner2::GetValues ( ) const  [inline]
```

Get all the values found (in lexicographic order)

10.266.4.28 IsKey()

```
bool gdcm::Scanner2::IsKey (
            const char * filename ) const
```

Check if filename is a key in the Mapping table. returns true only of file can be found, which means the file was indeed a DICOM file that could be processed

10.266.4.29 New()

```
static SmartPointer< Scanner2 > gdcm::Scanner2::New ( )  [inline], [static]
```

for wrapped language: instantiate a reference counted object

10.266.4.30 Print()

```
void gdcm::Scanner2::Print (
            std::ostream & os ) const  [override], [virtual]
```

Print result.

Reimplemented from gdcm::Object.

10.266.4.31 PrintTable()

```
void gdcm::Scanner2::PrintTable (
            std::ostream & os,
            bool header = false ) const
```

Print result as CSV table.

10.266.4.32 PrivateBegin()

```
PrivateConstIterator gdcm::Scanner2::PrivateBegin ( ) const  [inline]
```

10.266.4.33 PrivateEnd()

```
PrivateConstIterator gdcm::Scanner2::PrivateEnd ( ) const  [inline]
```

10.266.4.34 ProcessPrivateTag()

```
void gdcm::Scanner2::ProcessPrivateTag (
             StringFilter & sf,
             const char * filename )  [protected]
```

10.266.4.35 ProcessPublicTag()

```
void gdcm::Scanner2::ProcessPublicTag (
             StringFilter & sf,
             const char * filename )  [protected]
```

10.266.4.36 Scan()

```
bool gdcm::Scanner2::Scan (
             Directory::FilenamesType const & filenames )
```

Start the scan !

10.266.5 Friends And Related Function Documentation

10.266.5.1 operator<<

```
std::ostream & operator<< (
             std::ostream & _os,
             const Scanner2 & s )  [friend]
```

The documentation for this class was generated from the following file:

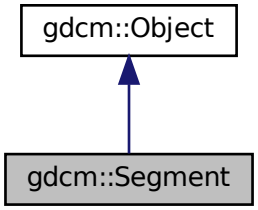
- gdcmScanner2.h

10.267 gdcm::Segment Class Reference

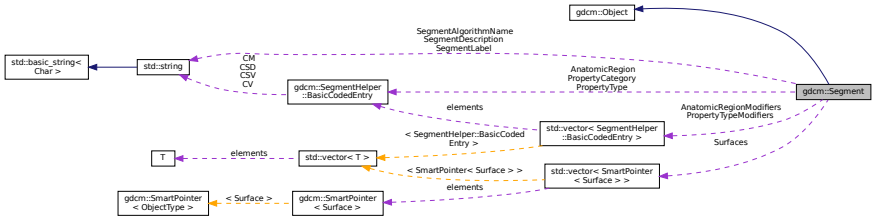
This class defines a segment.

`#include <gdcmSegment.h>`

Inheritance diagram for gdcm::Segment:



Collaboration diagram for gdcm::Segment:



Public Types

- enum ALGOType {
  AUTOMATIC = 0 ,
  SEMIAUTOMATIC ,
  MANUAL ,
  ALGOType_END }
- typedef std::vector< SegmentHelper::BasicCodedEntry > BasicCodedEntryVector
- typedef std::vector< SmartPointer< Surface > > SurfaceVector

Public Member Functions

- Segment ()
- ~Segment () override
- void AddSurface (SmartPointer< Surface > surface)
- SegmentHelper::BasicCodedEntry & GetAnatomicRegion ()
- SegmentHelper::BasicCodedEntry const & GetAnatomicRegion () const
- BasicCodedEntryVector & GetAnatomicRegionModifiers ()
- BasicCodedEntryVector const & GetAnatomicRegionModifiers () const
- SegmentHelper::BasicCodedEntry & GetPropertyCategory ()
- SegmentHelper::BasicCodedEntry const & GetPropertyCategory () const
- SegmentHelper::BasicCodedEntry & GetPropertyType ()
- SegmentHelper::BasicCodedEntry const & GetPropertyType () const
- BasicCodedEntryVector & GetPropertyTypeModifiers ()
- BasicCodedEntryVector const & GetPropertyTypeModifiers () const
- const char * GetSegmentAlgorithmName () const
- ALGOType GetSegmentAlgorithmType () const
- const char * GetSegmentDescription () const
- const char * GetSegmentLabel () const
- unsigned short GetSegmentNumber () const
- SmartPointer< Surface > GetSurface (const unsigned int idx=0) const
- unsigned long GetSurfaceCount ()
- SurfaceVector & GetSurfaces ()
- SurfaceVector const & GetSurfaces () const
- void SetAnatomicRegion (SegmentHelper::BasicCodedEntry const &BSE)
- void SetAnatomicRegionModifiers (BasicCodedEntryVector const &BSEV)
- void SetPropertyCategory (SegmentHelper::BasicCodedEntry const &BSE)
- void SetPropertyType (SegmentHelper::BasicCodedEntry const &BSE)
- void SetPropertyTypeModifiers (BasicCodedEntryVector const &BSEV)
- void SetSegmentAlgorithmName (const char *name)
- void SetSegmentAlgorithmType (ALGOType type)
- void SetSegmentAlgorithmType (const char *typeStr)
- void SetSegmentDescription (const char *description)
- void SetSegmentLabel (const char *label)
- void SetSegmentNumber (const unsigned short num)
- void SetSurfaceCount (const unsigned long nb)

Static Public Member Functions

- static ALGOType GetALGOType (const char *type)
- static const char * GetALGOTypeString (ALGOType type)

Protected Attributes

- SegmentHelper::BasicCodedEntry AnatomicRegion
- BasicCodedEntryVector AnatomicRegionModifiers
- SegmentHelper::BasicCodedEntry PropertyCategory
- SegmentHelper::BasicCodedEntry PropertyType
- BasicCodedEntryVector PropertyTypeModifiers
- std::string SegmentAlgorithmName
- ALGOType SegmentAlgorithmType
- std::string SegmentDescription
- std::string SegmentLabel
- unsigned short SegmentNumber
- unsigned long SurfaceCount
- SurfaceVector Surfaces

Additional Inherited Members

10.267.1 Detailed Description

This class defines a segment.

It mainly contains attributes of group 0x0062. In addition, it can be associated with surface.

See also

PS 3.3 C.8.20.2 and C.8.23

10.267.2 Member Typedef Documentation

10.267.2.1 BasicCodedEntryVector

```
typedef std::vector< SegmentHelper::BasicCodedEntry > gdcm::Segment::BasicCodedEntryVector
```

10.267.2.2 SurfaceVector

```
typedef std::vector< SmartPointer< Surface > > gdcm::Segment::SurfaceVector
```

10.267.3 Member Enumeration Documentation

10.267.3.1 ALGOType

```
enum gdcm::Segment::ALGOType
```

Enumerator

AUTOMATIC	
SEMIAUTOMATIC	
MANUAL	
ALGOType_END	

10.267.4 Constructor & Destructor Documentation

10.267.4.1 Segment()

```
gdcm::Segment::Segment ( )
```

10.267.4.2 ~Segment()

```
gdcm::Segment::~Segment ( )  [override]
```

10.267.5 Member Function Documentation

10.267.5.1 AddSurface()

```
void gdcm::Segment::AddSurface (
            SmartPointer< Surface > surface )
```

10.267.5.2 GetALGOType()

```
static ALGOType gdcm::Segment::GetALGOType (
            const char * type )  [static]
```

10.267.5.3 GetALGOTypeString()

```
static const char * gdcm::Segment::GetALGOTypeString (
            ALGOType type )  [static]
```

10.267.5.4 GetAnatomicRegion() [1/2]

```
SegmentHelper::BasicCodedEntry & gdcm::Segment::GetAnatomicRegion ( )
```

10.267.5.5 GetAnatomicRegion() [2/2]

```
SegmentHelper::BasicCodedEntry const & gdcm::Segment::GetAnatomicRegion ( ) const
```

10.267.5.6 GetAnatomicRegionModifiers() [1/2]

```
BasicCodedEntryVector & gdcm::Segment::GetAnatomicRegionModifiers ( )
```

10.267.5.7 GetAnatomicRegionModifiers() [2/2]

```
BasicCodedEntryVector const & gdcm::Segment::GetAnatomicRegionModifiers ( ) const
```

10.267.5.8 GetPropertyCategory() [1/2]

```
SegmentHelper::BasicCodedEntry & gdcm::Segment::GetPropertyCategory ( )
```

10.267.5.9 GetPropertyCategory() [2/2]

```
SegmentHelper::BasicCodedEntry const & gdcm::Segment::GetPropertyCategory ( ) const
```

10.267.5.10 GetPropertyType() [1/2]

```
SegmentHelper::BasicCodedEntry & gdcm::Segment::GetPropertyType ( )
```

10.267.5.11 GetPropertyType() [2/2]

```
SegmentHelper::BasicCodedEntry const & gdcm::Segment::GetPropertyType ( ) const
```

10.267.5.12 GetPropertyTypeModifiers() [1/2]

```
BasicCodedEntryVector & gdcm::Segment::GetPropertyTypeModifiers ( )
```

10.267.5.13 GetPropertyTypeModifiers() [2/2]

```
BasicCodedEntryVector const & gdcm::Segment::GetPropertyTypeModifiers ( ) const
```

10.267.5.14 GetSegmentAlgorithmName()

```
const char * gdcm::Segment::GetSegmentAlgorithmName ( ) const
```

10.267.5.15 GetSegmentAlgorithmType()

```
ALGOType gdcm::Segment::GetSegmentAlgorithmType ( ) const
```

10.267.5.16 GetSegmentDescription()

```
const char * gdcm::Segment::GetSegmentDescription ( ) const
```

10.267.5.17 GetSegmentLabel()

```
const char * gdcm::Segment::GetSegmentLabel ( ) const
```

10.267.5.18 GetSegmentNumber()

```
unsigned short gdcm::Segment::GetSegmentNumber ( ) const
```

10.267.5.19 GetSurface()

```
SmartPointer< Surface > gdcm::Segment::GetSurface (
            const unsigned int idx = 0 ) const
```

10.267.5.20 GetSurfaceCount()

```
unsigned long gdcm::Segment::GetSurfaceCount ( )
```

10.267.5.21 GetSurfaces() [1/2]

```
SurfaceVector & gdcm::Segment::GetSurfaces ( )
```

10.267.5.22 GetSurfaces() [2/2]

```
SurfaceVector const & gdcm::Segment::GetSurfaces ( ) const
```

10.267.5.23 SetAnatomicRegion()

```
void gdcm::Segment::SetAnatomicRegion (
            SegmentHelper::BasicCodedEntry const & BSE )
```

10.267.5.24 SetAnatomicRegionModifiers()

```
void gdcm::Segment::SetAnatomicRegionModifiers (
            BasicCodedEntryVector const & BSEV )
```

10.267.5.25 SetPropertyCategory()

```
void gdcm::Segment::SetPropertyCategory (
            SegmentHelper::BasicCodedEntry const & BSE )
```

10.267.5.26 SetPropertyType()

```
void gdcm::Segment::SetPropertyType (
            SegmentHelper::BasicCodedEntry const & BSE )
```

10.267.5.27 SetPropertyTypeModifiers()

```
void gdcm::Segment::SetPropertyTypeModifiers (
            BasicCodedEntryVector const & BSEV )
```

10.267.5.28 SetSegmentAlgorithmName()

```
void gdcm::Segment::SetSegmentAlgorithmName (
            const char * name )
```

10.267.5.29 SetSegmentAlgorithmType() [1/2]

```
void gdcm::Segment::SetSegmentAlgorithmType (
            ALGOType type )
```

10.267.5.30 SetSegmentAlgorithmType() [2/2]

```
void gdcm::Segment::SetSegmentAlgorithmType (
            const char * typeStr )
```

10.267.5.31 SetSegmentDescription()

```
void gdcm::Segment::SetSegmentDescription (
            const char * description )
```

10.267.5.32 SetSegmentLabel()

```
void gdcm::Segment::SetSegmentLabel (
            const char * label )
```

10.267.5.33 SetSegmentNumber()

```
void gdcm::Segment::SetSegmentNumber (
            const unsigned short num )
```

10.267.5.34 SetSurfaceCount()

```
void gdcm::Segment::SetSurfaceCount (
            const unsigned long nb )
```

10.267.6 Member Data Documentation

10.267.6.1 AnatomicRegion

```
SegmentHelper::BasicCodedEntry gdcm::Segment::AnatomicRegion  [protected]
```

10.267.6.2 AnatomicRegionModifiers

BasicCodedEntryVector gdcm::Segment::AnatomicRegionModifiers [protected]

10.267.6.3 PropertyCategory

SegmentHelper::BasicCodedEntry gdcm::Segment::PropertyCategory [protected]

10.267.6.4 PropertyType

SegmentHelper::BasicCodedEntry gdcm::Segment::PropertyType [protected]

10.267.6.5 PropertyTypeModifiers

BasicCodedEntryVector gdcm::Segment::PropertyTypeModifiers [protected]

10.267.6.6 SegmentAlgorithmName

std::string gdcm::Segment::SegmentAlgorithmName [protected]

10.267.6.7 SegmentAlgorithmType

ALGOType gdcm::Segment::SegmentAlgorithmType [protected]

10.267.6.8 SegmentDescription

std::string gdcm::Segment::SegmentDescription [protected]

10.267.6.9 SegmentLabel

```
std::string gdcm::Segment::SegmentLabel  [protected]
```

10.267.6.10 SegmentNumber

```
unsigned short gdcm::Segment::SegmentNumber  [protected]
```

10.267.6.11 SurfaceCount

```
unsigned long gdcm::Segment::SurfaceCount  [protected]
```

10.267.6.12 Surfaces

```
SurfaceVector gdcm::Segment::Surfaces  [protected]
```

The documentation for this class was generated from the following file:

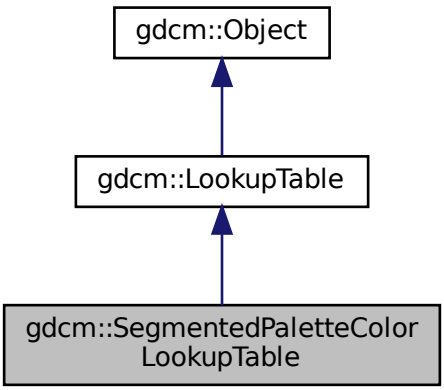
- gdcmSegment.h

10.268 gdcm::SegmentedPaletteColorLookupTable Class Reference

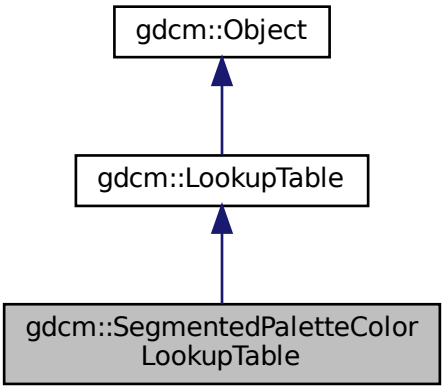
SegmentedPaletteColorLookupTable class.

`#include <gdcmSegmentedPaletteColorLookupTable.h>`

Inheritance diagram for gdcm::SegmentedPaletteColorLookupTable:



Collaboration diagram for gdcm::SegmentedPaletteColorLookupTable:



Public Member Functions

- SegmentedPaletteColorLookupTable ()
- ~SegmentedPaletteColorLookupTable () override
- void Print (std::ostream &) const override
- void SetLUT (LookupTableType type, const unsigned char *array, unsigned int length) override

  Initialize a SegmentedPaletteColorLookupTable.

Additional Inherited Members

10.268.1 Detailed Description

SegmentedPaletteColorLookupTable class.

10.268.2 Constructor & Destructor Documentation

10.268.2.1 SegmentedPaletteColorLookupTable()

```
gdcm::SegmentedPaletteColorLookupTable::SegmentedPaletteColorLookupTable ( )
```

10.268.2.2 ~SegmentedPaletteColorLookupTable()

```
gdcm::SegmentedPaletteColorLookupTable::~SegmentedPaletteColorLookupTable ( )  [override]
```

10.268.3 Member Function Documentation

10.268.3.1 Print()

```
void gdcm::SegmentedPaletteColorLookupTable::Print (
            std::ostream &  ) const  [inline], [override], [virtual]
```

Reimplemented from gdcm::LookupTable.

10.268.3.2 SetLUT()

```
void gdcm::SegmentedPaletteColorLookupTable::SetLUT (
            LookupTableType type,
            const unsigned char * array,
            unsigned int length )  [override], [virtual]
```

Initialize a SegmentedPaletteColorLookupTable.

Reimplemented from gdcm::LookupTable.

The documentation for this class was generated from the following file:

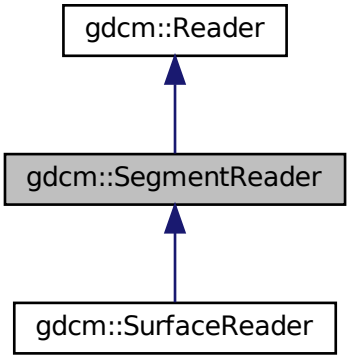
- gdcmSegmentedPaletteColorLookupTable.h

10.269 gdcm::SegmentReader Class Reference

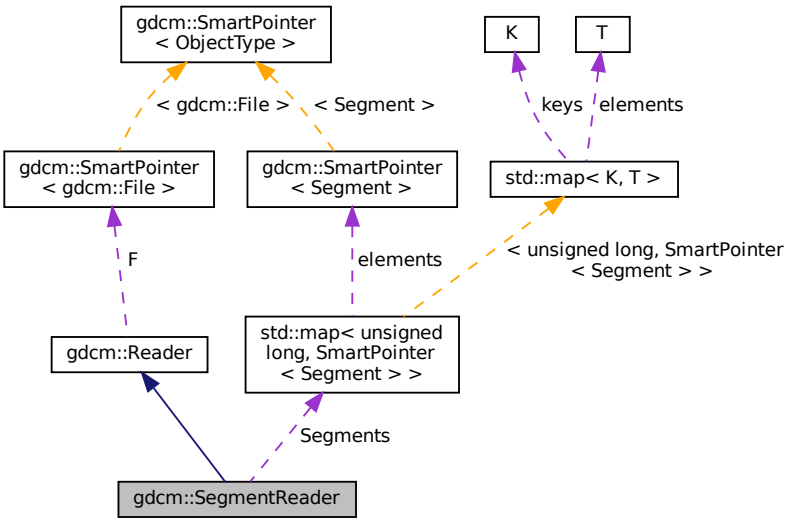
This class defines a segment reader.

`#include <gdcmSegmentReader.h>`

Inheritance diagram for gdcm::SegmentReader:



Collaboration diagram for gdcm::SegmentReader:



Public Types

- typedef std::vector< SmartPointer< Segment > > SegmentVector

Public Member Functions

- SegmentReader ()
- ~SegmentReader () override
- SegmentVector GetSegments ()
- const SegmentVector GetSegments () const
- bool Read () override

  Read.

Protected Types

- typedef std::map< unsigned long, SmartPointer< Segment > > SegmentMap

Protected Member Functions

- bool ReadSegment (const Item &segmentItem, const unsigned int idx)
- bool ReadSegments ()

Protected Attributes

- SegmentMap Segments

10.269.1 Detailed Description

This class defines a segment reader.

It reads attributes of group 0x0062.

See also

PS 3.3 C.8.20.2 and C.8.23

10.269.2 Member Typedef Documentation

10.269.2.1 SegmentMap

```
typedef std::map< unsigned long, SmartPointer< Segment > > gdcm::SegmentReader::SegmentMap  [protected]
```

10.269.2.2 SegmentVector

```
typedef std::vector< SmartPointer< Segment > > gdcm::SegmentReader::SegmentVector
```

10.269.3 Constructor & Destructor Documentation

10.269.3.1 SegmentReader()

```
gdcm::SegmentReader::SegmentReader ( )
```

10.269.3.2 ~SegmentReader()

```
gdcm::SegmentReader::~SegmentReader ( )  [override]
```

10.269.4 Member Function Documentation

10.269.4.1 GetSegments() [1/2]

```
SegmentVector gdcm::SegmentReader::GetSegments ( )
```

10.269.4.2 GetSegments() [2/2]

```
const SegmentVector gdcm::SegmentReader::GetSegments ( ) const
```

10.269.4.3 Read()

```
bool gdcm::SegmentReader::Read ( )  [override], [virtual]
```

Read.

Reimplemented from gdcm::Reader.

Reimplemented in gdcm::SurfaceReader.

10.269.4.4 ReadSegment()

```
bool gdcm::SegmentReader::ReadSegment (
            const Item & segmentItem,
            const unsigned int idx )  [protected]
```

10.269.4.5 ReadSegments()

```
bool gdcm::SegmentReader::ReadSegments ( )  [protected]
```

10.269.5 Member Data Documentation

10.269.5.1 Segments

```
SegmentMap gdcm::SegmentReader::Segments  [protected]
```

The documentation for this class was generated from the following file:

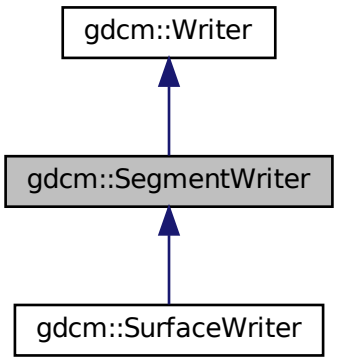
- gdcmSegmentReader.h

10.270 gdcm::SegmentWriter Class Reference

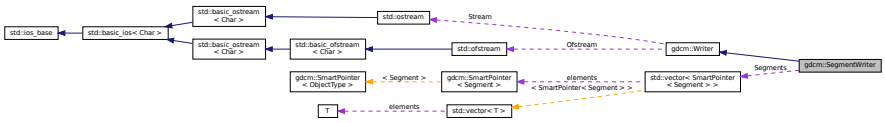
This class defines a segment writer.

`#include <gdcmSegmentWriter.h>`

Inheritance diagram for gdcm::SegmentWriter:



Collaboration diagram for gdcm::SegmentWriter:



Public Types

- typedef std::vector< SmartPointer< Segment > > SegmentVector

Public Member Functions

- SegmentWriter ()
- ~SegmentWriter () override
- void AddSegment (SmartPointer< Segment > segment)
- unsigned int GetNumberOfSegments () const
- SmartPointer< Segment > GetSegment (const unsigned int idx=0) const
- SegmentVector & GetSegments ()
- const SegmentVector & GetSegments () const
- void SetNumberOfSegments (const unsigned int size)
- void SetSegments (SegmentVector &segments)
- bool Write () override

  Write.

Protected Member Functions

- bool PrepareWrite ()

Protected Attributes

- SegmentVector Segments

10.270.1 Detailed Description

This class defines a segment writer.

It writes attributes of group 0x0062.

See also

PS 3.3 C.8.20.2 and C.8.23

10.270.2 Member Typedef Documentation

10.270.2.1 SegmentVector

```
typedef std::vector< SmartPointer< Segment > > gdcm::SegmentWriter::SegmentVector
```

10.270.3 Constructor & Destructor Documentation

10.270.3.1 SegmentWriter()

```
gdcm::SegmentWriter::SegmentWriter ( )
```

10.270.3.2 ~SegmentWriter()

```
gdcm::SegmentWriter::~SegmentWriter ( )  [override]
```

10.270.4 Member Function Documentation

10.270.4.1 AddSegment()

```
void gdcm::SegmentWriter::AddSegment (
            SmartPointer< Segment > segment )
```

10.270.4.2 GetNumberOfSegments()

```
unsigned int gdcm::SegmentWriter::GetNumberOfSegments ( ) const
```

10.270.4.3 GetSegment()

```
SmartPointer< Segment > gdcm::SegmentWriter::GetSegment (
            const unsigned int idx = 0 ) const
```

10.270.4.4 GetSegments() [1/2]

```
SegmentVector & gdcm::SegmentWriter::GetSegments ( )
```

10.270.4.5 GetSegments() [2/2]

```
const SegmentVector & gdcm::SegmentWriter::GetSegments ( ) const
```

10.270.4.6 PrepareWrite()

```
bool gdcm::SegmentWriter::PrepareWrite ( )  [protected]
```

10.270.4.7 SetNumberOfSegments()

```
void gdcm::SegmentWriter::SetNumberOfSegments (
            const unsigned int size )
```

10.270.4.8 SetSegments()

```
void gdcm::SegmentWriter::SetSegments (
            SegmentVector & segments )
```

10.270.4.9 Write()

```
bool gdcm::SegmentWriter::Write ( )  [override], [virtual]
```

Write.

Reimplemented from gdcm::Writer.

Reimplemented in gdcm::SurfaceWriter.

10.270.5 Member Data Documentation

10.270.5.1 Segments

```
SegmentVector gdcm::SegmentWriter::Segments  [protected]
```

The documentation for this class was generated from the following file:

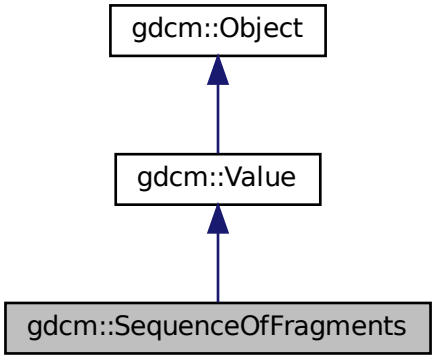
- gdcmSegmentWriter.h

10.271 gdcm::SequenceOfFragments Class Reference

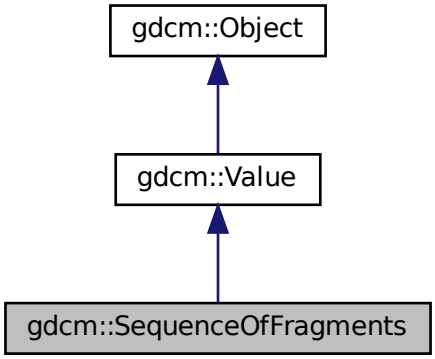
Class to represent a Sequence Of Fragments.

`#include <gdcmSequenceOfFragments.h>`

Inheritance diagram for gdcm::SequenceOfFragments:



Collaboration diagram for gdcm::SequenceOfFragments:



Public Types

- typedef FragmentVector::const_iterator ConstIterator
- typedef std::vector< Fragment > FragmentVector
- typedef FragmentVector::iterator Iterator
- typedef FragmentVector::size_type SizeType

Public Member Functions

- SequenceOfFragments ()

  constructor (UndefinedLength by default)
- void AddFragment (Fragment const &item)

  Appends a Fragment to the already added ones.
- Iterator Begin ()
- ConstIterator Begin () const
- void Clear () override

  Clear.
- unsigned long ComputeByteLength () const
- VL ComputeLength () const
- Iterator End ()
- ConstIterator End () const
- bool GetBuffer (char *buffer, unsigned long length) const
- bool GetFragBuffer (unsigned int fragNb, char *buffer, unsigned long &length) const
- const Fragment & GetFragment (SizeType num) const
- VL GetLength () const override

  Returns the SQ length, as read from disk.
- SizeType GetNumberOfFragments () const
- BasicOffsetTable & GetTable ()
- const BasicOffsetTable & GetTable () const
- bool operator== (const Value &val) const override
- void Print (std::ostream &os) const override
- template<typename TSwap >
  std::istream & Read (std::istream &is, bool readvalues=true)
- template<typename TSwap >
  std::istream & ReadPreValue (std::istream &is)
- template<typename TSwap >
  std::istream & ReadValue (std::istream &is, bool)
- void SetLength (VL length) override

  Sets the actual SQ length.
- template<typename TSwap >
  std::ostream const & Write (std::ostream &os) const
- bool WriteBuffer (std::ostream &os) const

Static Public Member Functions

- static SmartPointer< SequenceOfFragments > New ()

Additional Inherited Members

10.271.1 Detailed Description

Class to represent a Sequence Of Fragments.

Todo I do not enforce that Sequence of Fragments ends with a SQ end del

Examples

DecompressImageMultiframe.cs, DecompressJPEGFile.cs, FixBrokenJ2K.cxx, FixJAIBugJPEGLS.cxx, GetJPEGSamplePrecision.cxx, and MpegVideoInfo.cs.

10.271.2 Member Typedef Documentation

10.271.2.1 ConstIterator

```
typedef FragmentVector::const_iterator gdcm::SequenceOfFragments::ConstIterator
```

10.271.2.2 FragmentVector

```
typedef std::vector<Fragment> gdcm::SequenceOfFragments::FragmentVector
```

10.271.2.3 Iterator

```
typedef FragmentVector::iterator gdcm::SequenceOfFragments::Iterator
```

10.271.2.4 SizeType

```
typedef FragmentVector::size_type gdcm::SequenceOfFragments::SizeType
```

10.271.3 Constructor & Destructor Documentation

10.271.3.1 SequenceOfFragments()

```
gdcm::SequenceOfFragments::SequenceOfFragments ( )  [inline]
```

constructor (UndefinedLength by default)

10.271.4 Member Function Documentation

10.271.4.1 AddFragment()

```
void gdcm::SequenceOfFragments::AddFragment (
            Fragment const & item )
```

Appends a Fragment to the already added ones.

10.271.4.2 Begin() [1/2]

```
Iterator gdcm::SequenceOfFragments::Begin ( )  [inline]
```

10.271.4.3 Begin() [2/2]

```
ConstIterator gdcm::SequenceOfFragments::Begin ( ) const  [inline]
```

10.271.4.4 Clear()

```
void gdcm::SequenceOfFragments::Clear ( )  [override], [virtual]
```

Clear.

Implements gdcm::Value.

10.271.4.5 ComputeByteLength()

```
unsigned long gdcm::SequenceOfFragments::ComputeByteLength ( ) const
```

10.271.4.6 ComputeLength()

```
VL gdcm::SequenceOfFragments::ComputeLength ( ) const
```

10.271.4.7 End() [1/2]

```
Iterator gdcm::SequenceOfFragments::End ( )  [inline]
```

10.271.4.8 End() [2/2]

```
ConstIterator gdcm::SequenceOfFragments::End ( ) const  [inline]
```

10.271.4.9 GetBuffer()

```
bool gdcm::SequenceOfFragments::GetBuffer (
            char * buffer,
            unsigned long length ) const
```

10.271.4.10 GetFragBuffer()

```
bool gdcm::SequenceOfFragments::GetFragBuffer (
            unsigned int fragNb,
            char * buffer,
            unsigned long & length ) const
```

10.271.4.11 GetFragment()

```
const Fragment & gdcm::SequenceOfFragments::GetFragment (
            SizeType num ) const
```

Examples

DecompressImage.cs, FixBrokenJ2K.cxx, and FixJAIBugJPEGLS.cxx.

10.271.4.12 GetLength()

```
VL gdcm::SequenceOfFragments::GetLength ( ) const  [inline], [override], [virtual]
```

Returns the SQ length, as read from disk.

Implements gdcm::Value.

10.271.4.13 GetNumberOfFragments()

```
SizeType gdcm::SequenceOfFragments::GetNumberOfFragments ( ) const
```

Examples

FixJAIBugJPEGLS.cxx.

10.271.4.14 GetTable() [1/2]

```
BasicOffsetTable & gdcm::SequenceOfFragments::GetTable ( )  [inline]
```

10.271.4.15 GetTable() [2/2]

```
const BasicOffsetTable & gdcm::SequenceOfFragments::GetTable ( ) const  [inline]
```

10.271.4.16 New()

```
static SmartPointer< SequenceOfFragments > gdcm::SequenceOfFragments::New ( )  [inline], [static]
```

Examples

DecompressImageMultiframe.cs, DecompressJPEGFile.cs, and MpegVideoInfo.cs.

10.271.4.17 operator==()

```
bool gdcm::SequenceOfFragments::operator== (
            const Value & val ) const  [inline], [override], [virtual]
```

Implements gdcm::Value.

10.271.4.18 Print()

```
void gdcm::SequenceOfFragments::Print (
            std::ostream & os ) const  [inline], [override], [virtual]
```

Reimplemented from gdcm::Object.

10.271.4.19 Read()

```
template<typename TSwap >
std::istream & gdcm::SequenceOfFragments::Read (
            std::istream & is,
            bool readvalues = true )  [inline]
```

10.271.4.20 ReadPreValue()

```
template<typename TSwap >
std::istream & gdcm::SequenceOfFragments::ReadPreValue (
            std::istream & is )  [inline]
```

References gdcmDebugMacro.

10.271.4.21 ReadValue()

```
template<typename TSwap >
std::istream & gdcm::SequenceOfFragments::ReadValue (
            std::istream & is,
            bool  )  [inline]
```

References gdcmAssertAlwaysMacro, gdcmDebugMacro, gdcmWarningMacro, gdcm::Tag::GetElement(), gdcm::Tag::GetGroup(), gdcm::ByteValue::GetLength(), gdcm::ByteValue::GetPointer(), gdcm::DataElement::GetTag(), gdcm::DataElement::GetVL(), gdcm::Fragment::Read(), gdcm::Fragment::ReadBacktrack(), and gdcm::Exception::what().

10.271.4.22 SetLength()

```
void gdcm::SequenceOfFragments::SetLength (
            VL length )  [inline], [override], [virtual]
```

Sets the actual SQ length.

Implements gdcm::Value.

10.271.4.23 Write()

```
template<typename TSwap >
std::ostream const & gdcm::SequenceOfFragments::Write (
            std::ostream & os ) const  [inline]
```

References gdcm::Tag::Write(), and gdcm::VL::Write().

10.271.4.24 WriteBuffer()

```
bool gdcm::SequenceOfFragments::WriteBuffer (
            std::ostream & os ) const
```

Examples

GetJPEGSamplePrecision.cxx.

The documentation for this class was generated from the following file:

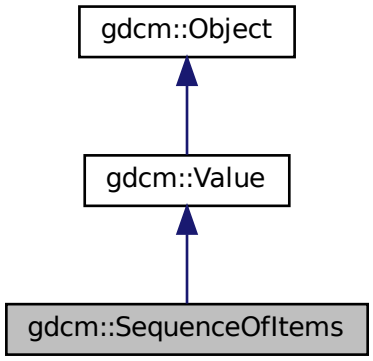
- gdcmSequenceOfFragments.h

10.272 gdcm::SequenceOfItems Class Reference

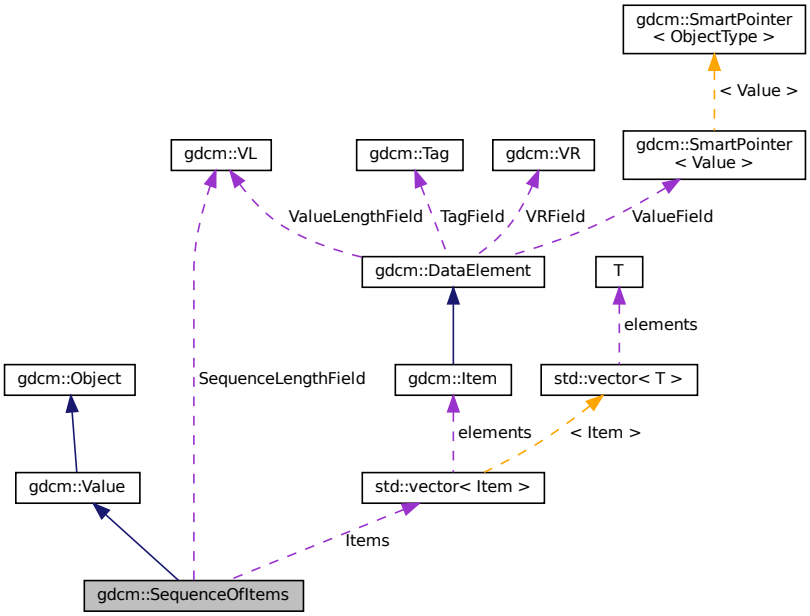
Class to represent a Sequence Of Items.

`#include <gdcmSequenceOfItems.h>`

Inheritance diagram for gdcm::SequenceOfItems:



Collaboration diagram for gdcm::SequenceOfItems:



Public Types

- typedef ItemVector::const_iterator ConstIterator
- typedef std::vector< Item > ItemVector
- typedef ItemVector::iterator Iterator
- typedef ItemVector::size_type SizeType

Public Member Functions

- SequenceOfItems ()

  constructor (UndefinedLength by default)
- void AddItem (Item const &item)

  Appends an Item to the already added ones.
- Item & AddNewUndefinedLengthItem ()

  Appends an Item to the already added ones.
- Iterator Begin ()
- ConstIterator Begin () const
- void Clear () override

  remove all items within the sequence
- template<typename TDE >
  VL ComputeLength () const
- Iterator End ()
- ConstIterator End () const
- bool FindDataElement (const Tag &t) const
- Item & GetItem (SizeType position)
- const Item & GetItem (SizeType position) const
- VL GetLength () const override

  Returns the SQ length, as read from disk.
- SizeType GetNumberOfItems () const
- bool IsEmpty () const
- bool IsUndefinedLength () const

  return if Value Length if of undefined length
- SequenceOfItems & operator= (const SequenceOfItems &val)
- bool operator== (const Value &val) const override
- void Print (std::ostream &os) const override
- template<typename TDE , typename TSwap >
  std::istream & Read (std::istream &is, bool readvalues=true)
- bool RemoveItemByIndex (const SizeType index)
- void SetLength (VL length) override

  Sets the actual SQ length.
- void SetLengthToUndefined ()

  Properly set the Sequence of Item to be undefined length.
- void SetNumberOfItems (SizeType n)
- template<typename TDE , typename TSwap >
  std::ostream const & Write (std::ostream &os) const

Static Public Member Functions

- static SmartPointer< SequenceOfItems > New ()

Public Attributes

- ItemVector Items

  Vector of Sequence Items.
- VL SequenceLengthField

  Total length of the Sequence (or 0xffffffff) if undefined.

Additional Inherited Members

10.272.1 Detailed Description

Class to represent a Sequence Of Items.

(value representation : SQ)

- a Value Representation for Data Elements that contains a sequence of Data Sets.
- Sequence of Item allows for Nested Data Sets

See PS 3.5, 7.4.6 Data Element Type Within a Sequence

Note

SEQUENCE OF ITEMS (VALUE REPRESENTATION SQ) A Value Representation for Data Elements that contain a sequence of Data Sets. Sequence of Items allows for Nested Data Sets.

Examples

DumpExamCard.cxx, DumpGEMSMovieGroup.cxx, ExtractEncryptedContent.cxx, Extracting_All_Resolution.cxx, Fake_Image_Using_Stream_Image_Writer.cxx, GenAllVR.cxx, GenFakeIdentifyFile.cxx, GenLongSeqs.cxx, GenSeqs.cxx, GetSequenceUltrasound.cxx, and ReadExplicitLengthSQIVR.cxx.

10.272.2 Member Typedef Documentation

10.272.2.1 ConstIterator

```
typedef ItemVector::const_iterator gdcm::SequenceOfItems::ConstIterator
```

10.272.2.2 ItemVector

```
typedef std::vector< Item > gdcm::SequenceOfItems::ItemVector
```

10.272.2.3 Iterator

```
typedef ItemVector::iterator gdcm::SequenceOfItems::Iterator
```

10.272.2.4 SizeType

```
typedef ItemVector::size_type gdcm::SequenceOfItems::SizeType
```

10.272.3 Constructor & Destructor Documentation

10.272.3.1 SequenceOfItems()

```
gdcm::SequenceOfItems::SequenceOfItems ( )  [inline]
```

constructor (UndefinedLength by default)

10.272.4 Member Function Documentation

10.272.4.1 AddItem()

```
void gdcm::SequenceOfItems::AddItem (
            Item const & item )
```

Appends an Item to the already added ones.

Examples

Extracting_All_Resolution.cxx.

10.272.4.2 AddNewUndefinedLengthItem()

```
Item & gdcm::SequenceOfItems::AddNewUndefinedLengthItem ( )
```

Appends an Item to the already added ones.

10.272.4.3 Begin() [1/2]

```
Iterator gdcm::SequenceOfItems::Begin ( )  [inline]
```

10.272.4.4 Begin() [2/2]

```
ConstIterator gdcm::SequenceOfItems::Begin ( ) const  [inline]
```

10.272.4.5 Clear()

```
void gdcm::SequenceOfItems::Clear ( )  [override], [virtual]
```

remove all items within the sequence

Implements gdcm::Value.

10.272.4.6 ComputeLength()

```
template<typename TDE >
VL gdcm::SequenceOfItems::ComputeLength ( ) const
```

10.272.4.7 End() [1/2]

```
Iterator gdcm::SequenceOfItems::End ( )  [inline]
```

10.272.4.8 End() [2/2]

```
ConstIterator gdcm::SequenceOfItems::End ( ) const  [inline]
```

10.272.4.9 FindDataElement()

```
bool gdcm::SequenceOfItems::FindDataElement (
            const Tag & t ) const
```

10.272.4.10 GetItem() [1/2]

```
Item & gdcm::SequenceOfItems::GetItem (
            SizeType position )
```

10.272.4.11 GetItem() [2/2]

```
const Item & gdcm::SequenceOfItems::GetItem (
            SizeType position ) const
```

Examples

DumpExamCard.cxx, DumpGEMSMovieGroup.cxx, ExtractEncryptedContent.cxx, and GetSequenceUltrasound.cxx.

10.272.4.12 GetLength()

```
VL gdcm::SequenceOfItems::GetLength ( ) const  [inline], [override], [virtual]
```

Returns the SQ length, as read from disk.

Implements gdcm::Value.

10.272.4.13 GetNumberOfItems()

```
SizeType gdcm::SequenceOfItems::GetNumberOfItems ( ) const  [inline]
```

Examples

DumpExamCard.cxx, DumpGEMSMovieGroup.cxx, ExtractEncryptedContent.cxx, and GetSequenceUltrasound.cxx.

10.272.4.14 IsEmpty()

```
bool gdcm::SequenceOfItems::IsEmpty ( ) const  [inline]
```

10.272.4.15 IsUndefinedLength()

```
bool gdcm::SequenceOfItems::IsUndefinedLength ( ) const  [inline]
```

return if Value Length if of undefined length

10.272.4.16 New()

```
static SmartPointer< SequenceOfItems > gdcm::SequenceOfItems::New ( )  [inline], [static]
```

Examples

NewSequence.cs.

10.272.4.17 operator=()

```
SequenceOfItems & gdcm::SequenceOfItems::operator= (
            const SequenceOfItems & val )  [inline]
```

References Items, and SequenceLengthField.

10.272.4.18 operator==()

```
bool gdcm::SequenceOfItems::operator== (
            const Value & val ) const  [inline], [override], [virtual]
```

Implements gdcm::Value.

References Items, and SequenceLengthField.

10.272.4.19 Print()

```
void gdcm::SequenceOfItems::Print (
            std::ostream & os ) const  [inline], [override], [virtual]
```

Reimplemented from gdcm::Object.

10.272.4.20 Read()

```
template<typename TDE , typename TSwap >
std::istream & gdcm::SequenceOfItems::Read (
            std::istream & is,
            bool readvalues = true )  [inline]
```

References gdcm::Item::Clear(), gdcmDebugMacro, gdcmWarningMacro, gdcm::Exception::GetDescription(), gdcm::Item::GetNestedDataSet(), gdcm::DataElement::GetTag(), gdcm::DataElement::GetVL(), gdcm::Item::Read(), and gdcm::DataSet::Size().

10.272.4.21 RemoveItemByIndex()

```
bool gdcm::SequenceOfItems::RemoveItemByIndex (
            const SizeType index )
```

Remove an Item as specified by its index, if index > size, false is returned Index starts at 1 not 0

10.272.4.22 SetLength()

```
void gdcm::SequenceOfItems::SetLength (
            VL length )  [inline], [override], [virtual]
```

Sets the actual SQ length.

Implements gdcm::Value.

10.272.4.23 SetLengthToUndefined()

```
void gdcm::SequenceOfItems::SetLengthToUndefined ( )
```

Properly set the Sequence of Item to be undefined length.

10.272.4.24 SetNumberOfItems()

```
void gdcm::SequenceOfItems::SetNumberOfItems (
            SizeType n )  [inline]
```

10.272.4.25 Write()

```
template<typename TDE , typename TSwap >
std::ostream const & gdcm::SequenceOfItems::Write (
            std::ostream & os ) const  [inline]
```

References gdcm::Tag::Write(), and gdcm::VL::Write().

10.272.5 Member Data Documentation

10.272.5.1 Items

```
ItemVector gdcm::SequenceOfItems::Items
```

Vector of Sequence Items.

Referenced by operator=(), and operator==().

10.272.5.2 SequenceLengthField

```
VL gdcm::SequenceOfItems::SequenceLengthField
```

Total length of the Sequence (or 0xffffffff) if undefined.

Referenced by operator=(), and operator==().

The documentation for this class was generated from the following file:

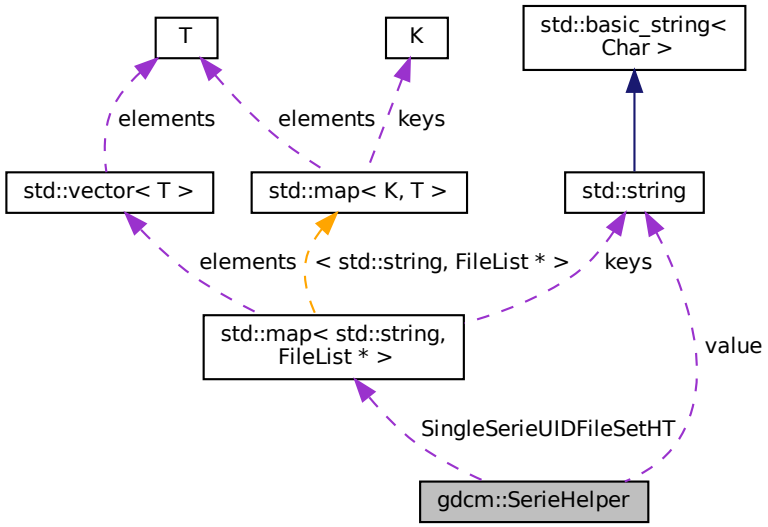
- gdcmSequenceOfItems.h

10.273 gdcm::SerieHelper Class Reference

SerieHelper DO NOT USE this class, it is only a temporary solution for ITK migration from GDCM 1.x to GDCM 2.x It will disappear soon, you've been warned.

`#include <gdcmSerieHelper.h>`

Collaboration diagram for gdcm::SerieHelper:



Public Member Functions

- SerieHelper ()
- ~SerieHelper ()
- void AddRestriction (const std::string &tag)
- void AddRestriction (uint16_t group, uint16_t elem, std::string const &value, int op)
- void Clear ()
- void CreateDefaultUniqueSeriesIdentifier ()
- std::string CreateUniqueSeriesIdentifier (File *inFile)
- FileList * GetFirstSingleSerieUIDFileSet ()
- FileList * GetNextSingleSerieUIDFileSet ()
- void OrderFileList (FileList *fileSet)
- void SetDirectory (std::string const &dir, bool recursive=false)
- void SetLoadMode (int)
- void SetUseSeriesDetails (bool useSeriesDetails)

Protected Types

- using Rule = RuleStructure{ uint16_t group
- typedef std::vector< Rule > SerieRestrictions
- typedef std::map< std::string, FileList * > SingleSerieUIDFileSetmap

Protected Member Functions

- bool AddFile (FileWithName &header)
- void AddFileName (std::string const &filename)
- void AddRestriction (const Tag &tag)
- bool FileNameOrdering (FileList *fileList)
- bool ImageNumberOrdering (FileList *fileList)
- bool ImagePositionPatientOrdering (FileList *fileSet)
- bool UserOrdering (FileList *fileSet)

Protected Attributes

- uint16_t elem
- SingleSerieUIDFileSetmap::iterator ItFileSetHt
- int op
- SingleSerieUIDFileSetmap SingleSerieUIDFileSetHT
- std::string value

10.273.1 Detailed Description

SerieHelper DO NOT USE this class, it is only a temporary solution for ITK migration from GDCM 1.x to GDCM 2.x It will disappear soon, you've been warned.

Instead see ImageHelper or IPPSorter

10.273.2 Member Typedef Documentation

10.273.2.1 Rule

```
using gdcm::SerieHelper::Rule = RuleStructure{ uint16_t group  [protected]
```

10.273.2.2 SerieRestrictions

```
typedef std::vector<Rule> gdcm::SerieHelper::SerieRestrictions  [protected]
```

10.273.2.3 SingleSerieUIDFileSetmap

```
typedef std::map<std::string, FileList *> gdcm::SerieHelper::SingleSerieUIDFileSetmap  [protected]
```

10.273.3 Constructor & Destructor Documentation

10.273.3.1 SerieHelper()

```
gdcm::SerieHelper::SerieHelper ( )
```

10.273.3.2 ~SerieHelper()

```
gdcm::SerieHelper::~SerieHelper ( )
```

10.273.4 Member Function Documentation

10.273.4.1 AddFile()

```
bool gdcm::SerieHelper::AddFile (
            FileWithName & header )  [protected]
```

10.273.4.2 AddFileName()

```
void gdcm::SerieHelper::AddFileName (
            std::string const & filename )  [protected]
```

10.273.4.3 AddRestriction() [1/3]

```
void gdcm::SerieHelper::AddRestriction (
            const std::string & tag )
```

10.273.4.4 AddRestriction() [2/3]

```
void gdcm::SerieHelper::AddRestriction (
            const Tag & tag )  [protected]
```

10.273.4.5 AddRestriction() [3/3]

```
void gdcm::SerieHelper::AddRestriction (
            uint16_t group,
            uint16_t elem,
            std::string const & value,
            int op )
```

10.273.4.6 Clear()

```
void gdcm::SerieHelper::Clear ( )
```

10.273.4.7 CreateDefaultUniqueSeriesIdentifier()

```
void gdcm::SerieHelper::CreateDefaultUniqueSeriesIdentifier ( )
```

10.273.4.8 CreateUniqueSeriesIdentifier()

```
std::string gdcm::SerieHelper::CreateUniqueSeriesIdentifier (
            File * inFile )
```

10.273.4.9 FileNameOrdering()

```
bool gdcm::SerieHelper::FileNameOrdering (
            FileList * fileList )  [protected]
```

10.273.4.10 GetFirstSingleSerieUIDFileSet()

```
FileList * gdcm::SerieHelper::GetFirstSingleSerieUIDFileSet ( )
```

10.273.4.11 GetNextSingleSerieUIDFileSet()

```
FileList * gdcm::SerieHelper::GetNextSingleSerieUIDFileSet ( )
```

10.273.4.12 ImageNumberOrdering()

```
bool gdcm::SerieHelper::ImageNumberOrdering (
            FileList * fileList )  [protected]
```

10.273.4.13 ImagePositionPatientOrdering()

```
bool gdcm::SerieHelper::ImagePositionPatientOrdering (
            FileList * fileSet )  [protected]
```

10.273.4.14 OrderFileList()

```
void gdcm::SerieHelper::OrderFileList (
            FileList * fileSet )
```

10.273.4.15 SetDirectory()

```
void gdcm::SerieHelper::SetDirectory (
            std::string const & dir,
            bool recursive = false )
```

10.273.4.16 SetLoadMode()

```
void gdcm::SerieHelper::SetLoadMode (
            int  )  [inline]
```

10.273.4.17 SetUseSeriesDetails()

```
void gdcm::SerieHelper::SetUseSeriesDetails (
            bool useSeriesDetails )
```

10.273.4.18 UserOrdering()

```
bool gdcm::SerieHelper::UserOrdering (
            FileList * fileSet )  [protected]
```

10.273.5 Member Data Documentation

10.273.5.1 elem

```
uint16_t gdcm::SerieHelper::elem  [protected]
```

10.273.5.2 ItFileSetHt

```
SingleSerieUIDFileSetmap::iterator gdcm::SerieHelper::ItFileSetHt  [protected]
```

10.273.5.3 op

```
int gdcm::SerieHelper::op  [protected]
```

10.273.5.4 SingleSerieUIDFileSetHT

```
SingleSerieUIDFileSetmap gdcm::SerieHelper::SingleSerieUIDFileSetHT  [protected]
```

10.273.5.5 value

```
std::string gdcm::SerieHelper::value  [protected]
```

The documentation for this class was generated from the following file:

- gdcmSerieHelper.h

10.274 gdcm::Series Class Reference

Series.

```
#include <gdcmSeries.h>
```

Public Member Functions

- Series ()=default

10.274.1 Detailed Description

Series.

10.274.2 Constructor & Destructor Documentation

10.274.2.1 Series()

```
gdcm::Series::Series ( )  [default]
```

The documentation for this class was generated from the following file:

- gdcmSeries.h

10.275 gdcm::network::ServiceClassApplicationInformation Class Reference

```
#include <gdcmServiceClassApplicationInformation.h>
```

Public Member Functions

- ServiceClassApplicationInformation ()
- void Print (std::ostream &os) const
- std::istream & Read (std::istream &is)
- void SetTuple (uint8_t levelofsupport, uint8_t levelofdigitalsig, uint8_t elementcoercion)
- size_t Size () const
- const std::ostream & Write (std::ostream &os) const

10.275.1 Detailed Description

PS 3.4 Table B.3-1 SERVICE-CLASS-APPLICATION-INFORMATION (A-ASSOCIATE-RQ)

10.275.2 Constructor & Destructor Documentation

10.275.2.1 ServiceClassApplicationInformation()

```
gdcm::network::ServiceClassApplicationInformation::ServiceClassApplicationInformation ( )
```

10.275.3 Member Function Documentation

10.275.3.1 Print()

```
void gdcm::network::ServiceClassApplicationInformation::Print (
            std::ostream & os ) const
```

10.275.3.2 Read()

```
std::istream & gdcm::network::ServiceClassApplicationInformation::Read (
            std::istream & is )
```

10.275.3.3 SetTuple()

```
void gdcm::network::ServiceClassApplicationInformation::SetTuple (
            uint8_t levelofsupport,
            uint8_t levelofdigitalsig,
            uint8_t elementcoercion )
```

10.275.3.4 Size()

```
size_t gdcm::network::ServiceClassApplicationInformation::Size ( ) const
```

10.275.3.5 Write()

```
const std::ostream & gdcm::network::ServiceClassApplicationInformation::Write (
            std::ostream & os ) const
```

The documentation for this class was generated from the following file:

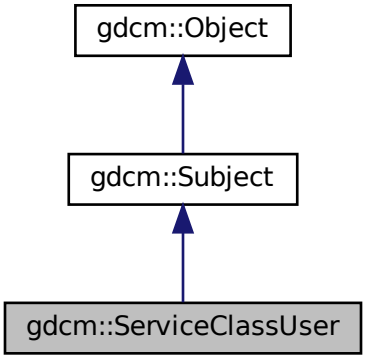
- gdcmServiceClassApplicationInformation.h

10.276 gdcm::ServiceClassUser Class Reference

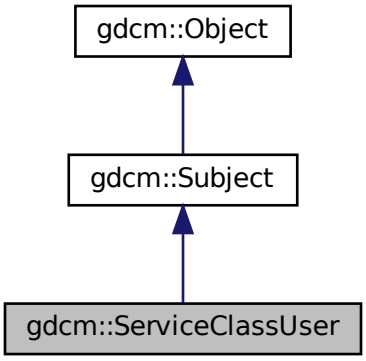
ServiceClassUser.

`#include <gdcmServiceClassUser.h>`

Inheritance diagram for gdcm::ServiceClassUser:



Collaboration diagram for gdcm::ServiceClassUser:



Public Member Functions

- ServiceClassUser ()
- ServiceClassUser (const ServiceClassUser &)=delete
- ~ServiceClassUser () override
- const char * GetAETitle () const
- const char * GetCalledAETitle () const
- double GetTimeout () const
- bool InitializeConnection ()
- bool IsPresentationContextAccepted (const PresentationContext &pc) const

  Return if the passed in presentation was accepted during association negotiation.
- void operator= (const ServiceClassUser &)=delete
- bool SendEcho ()

  C-ECHO.
- bool SendFind (const BaseRootQuery *query, std::vector< DataSet > &retDatasets)

  C-FIND a query, return result are in retDatasets.
- bool SendMove (const BaseRootQuery *query, const char *outputdir)

  Execute a C-MOVE, based on query, return files are written in outputdir.
- bool SendMove (const BaseRootQuery *query, std::vector< DataSet > &retDatasets)

  Execute a C-MOVE, based on query, returned dataset are Implicit.
- bool SendMove (const BaseRootQuery *query, std::vector< File > &retFile)

  Execute a C-MOVE, based on query, returned Files are stored in vector.
- bool SendStore (const char *filename)

  Execute a C-STORE on file on disk, named filename.
- bool SendStore (DataSet const &ds)

  Execute a C-STORE on a DataSet, the transfer syntax used will be Implicit.
- bool SendStore (File const &file)
- void SetAETitle (const char *aetitle)

  set calling ae title
- void SetCalledAETitle (const char *aetitle)

  set called ae title
- void SetHostname (const char *hostname)

  Set the name of the called hostname (hostname or IP address)
- void SetPort (uint16_t port)

  Set port of remote host (called application)
- void SetPortSCP (uint16_t portscp)

  Set the port for any incoming C-STORE-SCP operation (typically in a return of C-MOVE)
- void SetPresentationContexts (std::vector< PresentationContext > const &pcs)

  Set the Presentation Context used for the Association.
- void SetTimeout (double t)

  set/get Timeout
- bool StartAssociation ()

  Start the association. Need to call SetPresentationContexts before.
- bool StopAssociation ()

  Stop the running association.

Static Public Member Functions

- static SmartPointer< ServiceClassUser > New ()

  for wrapped language: instantiate a reference counted object

Additional Inherited Members

10.276.1 Detailed Description

ServiceClassUser.

Examples

CStoreQtProgress.cxx.

10.276.2 Constructor & Destructor Documentation

10.276.2.1 ServiceClassUser() [1/2]

```
gdcm::ServiceClassUser::ServiceClassUser ( )
```

Construct a SCU with default:

- hostname = localhost
- port = 104

10.276.2.2 ~ServiceClassUser()

```
gdcm::ServiceClassUser::~ServiceClassUser ( )  [override]
```

10.276.2.3 ServiceClassUser() [2/2]

```
gdcm::ServiceClassUser::ServiceClassUser (
            const ServiceClassUser &  )  [delete]
```

10.276.3 Member Function Documentation

10.276.3.1 GetAETitle()

```
const char * gdcm::ServiceClassUser::GetAETitle ( ) const
```

10.276.3.2 GetCalledAETitle()

```
const char * gdcm::ServiceClassUser::GetCalledAETitle ( ) const
```

10.276.3.3 GetTimeout()

```
double gdcm::ServiceClassUser::GetTimeout ( ) const
```

10.276.3.4 InitializeConnection()

```
bool gdcm::ServiceClassUser::InitializeConnection ( )
```

Will try to connect This will setup the actual timeout used during the whole connection time. Need to call SetTimeout first

Examples

CStoreQtProgress.cxx.

10.276.3.5 IsPresentationContextAccepted()

```
bool gdcm::ServiceClassUser::IsPresentationContextAccepted (
            const PresentationContext & pc ) const
```

Return if the passed in presentation was accepted during association negotiation.

10.276.3.6 New()

```
static SmartPointer< ServiceClassUser > gdcm::ServiceClassUser::New ( )  [inline], [static]
```

for wrapped language: instantiate a reference counted object

10.276.3.7 operator=()

```
void gdcm::ServiceClassUser::operator= (
            const ServiceClassUser &  )  [delete]
```

10.276.3.8 SendEcho()

```
bool gdcm::ServiceClassUser::SendEcho ( )
```

C-ECHO.

10.276.3.9 SendFind()

```
bool gdcm::ServiceClassUser::SendFind (
            const BaseRootQuery * query,
            std::vector< DataSet > & retDatasets )
```

C-FIND a query, return result are in retDatasets.

10.276.3.10 SendMove() [1/3]

```
bool gdcm::ServiceClassUser::SendMove (
            const BaseRootQuery * query,
            const char * outputdir )
```

Execute a C-MOVE, based on query, return files are written in outputdir.

10.276.3.11 SendMove() [2/3]

```
bool gdcm::ServiceClassUser::SendMove (
            const BaseRootQuery * query,
            std::vector< DataSet > & retDatasets )
```

Execute a C-MOVE, based on query, returned dataset are Implicit.

10.276.3.12 SendMove() [3/3]

```
bool gdcm::ServiceClassUser::SendMove (
            const BaseRootQuery * query,
            std::vector< File > & retFile )
```

Execute a C-MOVE, based on query, returned Files are stored in vector.

10.276.3.13 SendStore() [1/3]

```
bool gdcm::ServiceClassUser::SendStore (
            const char * filename )
```

Execute a C-STORE on file on disk, named filename.

Examples

CStoreQtProgress.cxx.

10.276.3.14 SendStore() [2/3]

```
bool gdcm::ServiceClassUser::SendStore (
            DataSet const & ds )
```

Execute a C-STORE on a DataSet, the transfer syntax used will be Implicit.

10.276.3.15 SendStore() [3/3]

```
bool gdcm::ServiceClassUser::SendStore (
            File const & file )
```

Execute a C-STORE on a File, the transfer syntax used for the query is based on the file.

10.276.3.16 SetAETitle()

```
void gdcm::ServiceClassUser::SetAETitle (
            const char * aetitle )
```

set calling ae title

10.276.3.17 SetCalledAETitle()

```
void gdcm::ServiceClassUser::SetCalledAETitle (
            const char * aetitle )
```

set called ae title

Examples

CStoreQtProgress.cxx.

10.276.3.18 SetHostname()

```
void gdcm::ServiceClassUser::SetHostname (
            const char * hostname )
```

Set the name of the called hostname (hostname or IP address)

Examples

CStoreQtProgress.cxx.

10.276.3.19 SetPort()

```
void gdcm::ServiceClassUser::SetPort (
            uint16_t port )
```

Set port of remote host (called application)

Examples

CStoreQtProgress.cxx.

10.276.3.20 SetPortSCP()

```
void gdcm::ServiceClassUser::SetPortSCP (
            uint16_t portscp )
```

Set the port for any incoming C-STORE-SCP operation (typically in a return of C-MOVE)

10.276.3.21 SetPresentationContexts()

```
void gdcm::ServiceClassUser::SetPresentationContexts (
            std::vector< PresentationContext > const & pcs )
```

Set the Presentation Context used for the Association.

Examples

CStoreQtProgress.cxx.

10.276.3.22 SetTimeout()

```
void gdcm::ServiceClassUser::SetTimeout (
            double t )
```

set/get Timeout

Examples

CStoreQtProgress.cxx.

10.276.3.23 StartAssociation()

```
bool gdcm::ServiceClassUser::StartAssociation ( )
```

Start the association. Need to call SetPresentationContexts before.

Examples

CStoreQtProgress.cxx.

10.276.3.24 StopAssociation()

```
bool gdcm::ServiceClassUser::StopAssociation ( )
```

Stop the running association.

Examples

CStoreQtProgress.cxx.

The documentation for this class was generated from the following file:

- gdcmServiceClassUser.h

10.277 gdcm::SHA1 Class Reference

Class for SHA1.

```
#include <gdcmSHA1.h>
```

Public Member Functions

- SHA1 ()
- SHA1 (const SHA1 &)=delete
- ~SHA1 ()
- void operator= (const SHA1 &)=delete

Static Public Member Functions

- static bool Compute (const char *buffer, unsigned long buf_len, char digest_str[20 *2+1])
- static bool ComputeFile (const char *filename, char digest_str[20 *2+1])

10.277.1 Detailed Description

Class for SHA1.

Warning

this class is able to pick from one implementation:

1. the one from OpenSSL (when GDCM_USE_SYSTEM_OPENSSL is turned ON)

In all other cases it will return an error

10.277.2 Constructor & Destructor Documentation

10.277.2.1 SHA1() [1/2]

```
gdcm::SHA1::SHA1 ( )
```

10.277.2.2 ~SHA1()

```
gdcm::SHA1::~SHA1 ( )
```

10.277.2.3 SHA1() [2/2]

```
gdcm::SHA1::SHA1 (
            const SHA1 &  )  [delete]
```

10.277.3 Member Function Documentation

10.277.3.1 Compute()

```
static bool gdcm::SHA1::Compute (
            const char * buffer,
            unsigned long buf_len,
            char digest_str[20 *2+1] )  [static]
```

10.277.3.2 ComputeFile()

```
static bool gdcm::SHA1::ComputeFile (
            const char * filename,
            char digest_str[20 *2+1] )  [static]
```

10.277.3.3 operator=()

```
void gdcm::SHA1::operator= (
            const SHA1 &  )  [delete]
```

The documentation for this class was generated from the following file:

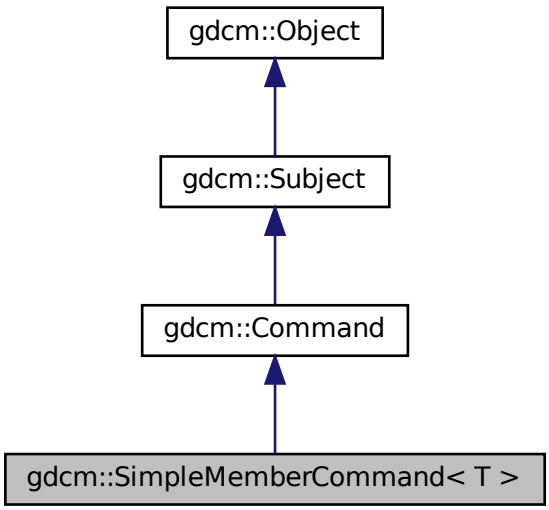
- gdcmSHA1.h

10.278 gdcm::SimpleMemberCommand< T > Class Template Reference

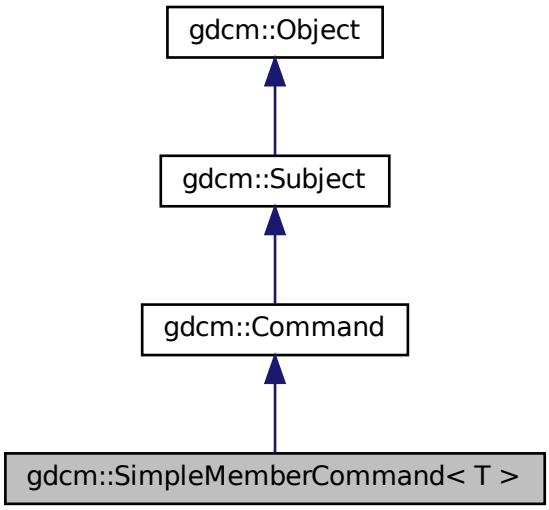
Command subclass that calls a pointer to a member function.

`#include <gdcmCommand.h>`

Inheritance diagram for gdcm::SimpleMemberCommand< T >:



Collaboration diagram for gdcm::SimpleMemberCommand< T >:



Public Types

- typedef SimpleMemberCommand Self
- typedef void(T::* TMemberFunctionPointer) ()

Public Member Functions

- SimpleMemberCommand (const Self &)=delete
- void Execute (const Subject *, const Event &) override
- void Execute (Subject *, const Event &) override
- void operator= (const Self &)=delete
- void SetCallbackFunction (T *object, TMemberFunctionPointer memberFunction)

Static Public Member Functions

- static SmartPointer< SimpleMemberCommand > New ()

Protected Member Functions

- SimpleMemberCommand ()
- ~SimpleMemberCommand () override=default

Protected Attributes

- TMemberFunctionPointer m_MemberFunction
- T * m_This

10.278.1 Detailed Description

template<typename T>
class gdcm::SimpleMemberCommand< T >

Command subclass that calls a pointer to a member function.

SimpleMemberCommand calls a pointer to a member function with no arguments.

10.278.2 Member Typedef Documentation

10.278.2.1 Self

```
template<typename T >
typedef SimpleMemberCommand gdcm::SimpleMemberCommand< T >::Self
```

Standard class typedefs.

10.278.2.2 TMemberFunctionPointer

```
template<typename T >
typedef void(T::* gdcm::SimpleMemberCommand< T >::TMemberFunctionPointer) ()
```

A method callback.

10.278.3 Constructor & Destructor Documentation

10.278.3.1 SimpleMemberCommand() [1/2]

```
template<typename T >
gdcm::SimpleMemberCommand< T >::SimpleMemberCommand (
            const Self &  )  [delete]
```

10.278.3.2 SimpleMemberCommand() [2/2]

```
template<typename T >
gdcm::SimpleMemberCommand< T >::SimpleMemberCommand ( )  [inline], [protected]
```

Referenced by gdcm::SimpleMemberCommand< T >::New().

10.278.3.3 ~SimpleMemberCommand()

```
template<typename T >
gdcm::SimpleMemberCommand< T >::~SimpleMemberCommand ( )  [override], [protected], [default]
```

10.278.4 Member Function Documentation

10.278.4.1 Execute() [1/2]

```
template<typename T >
void gdcm::SimpleMemberCommand< T >::Execute (
            const Subject * caller,
            const Event & event )  [inline], [override], [virtual]
```

Abstract method that defines the action to be taken by the command. This variant is expected to be used when requests comes from a const Object

Implements gdcm::Command.

References gdcm::SimpleMemberCommand< T >::m_MemberFunction.

10.278.4.2 Execute() [2/2]

```
template<typename T >
void gdcm::SimpleMemberCommand< T >::Execute (
            Subject * ,
            const Event &  )  [inline], [override], [virtual]
```

Invoke the callback function.

Implements gdcm::Command.

References gdcm::SimpleMemberCommand< T >::m_MemberFunction.

10.278.4.3 New()

```
template<typename T >
static SmartPointer< SimpleMemberCommand > gdcm::SimpleMemberCommand< T >::New ( )  [inline],
[static]
```

Run-time type information (and related methods). Method for creation through the object factory.

References gdcm::SimpleMemberCommand< T >::SimpleMemberCommand().

10.278.4.4 operator=()

```
template<typename T >
void gdcm::SimpleMemberCommand< T >::operator= (
            const Self &  )  [delete]
```

10.278.4.5 SetCallbackFunction()

```
template<typename T >
void gdcm::SimpleMemberCommand< T >::SetCallbackFunction (
            T * object,
            TMemberFunctionPointer memberFunction )  [inline]
```

Specify the callback function.

References gdcm::SimpleMemberCommand< T >::m_MemberFunction, and gdcm::SimpleMemberCommand< T >::m_This.

10.278.5 Member Data Documentation

10.278.5.1 m_MemberFunction

```
template<typename T >
TMemberFunctionPointer gdcm::SimpleMemberCommand< T >::m_MemberFunction  [protected]
```

Referenced by gdcm::SimpleMemberCommand< T >::Execute(), and gdcm::SimpleMemberCommand< T >::SetCallbackFunction().

10.278.5.2 m_This

```
template<typename T >
T* gdcm::SimpleMemberCommand< T >::m_This  [protected]
```

Referenced by gdcm::SimpleMemberCommand< T >::SetCallbackFunction().

The documentation for this class was generated from the following file:

- gdcmCommand.h

10.279 gdcm::SimpleSubjectWatcher Class Reference

SimpleSubjectWatcher.

```
#include <gdcmSimpleSubjectWatcher.h>
```

Public Member Functions

- SimpleSubjectWatcher (const SimpleSubjectWatcher &)=delete
- SimpleSubjectWatcher (Subject *s, const char *comment="")
- virtual ~SimpleSubjectWatcher ()
- void operator= (const SimpleSubjectWatcher &)=delete

Protected Member Functions

- virtual void EndFilter ()
- virtual void ShowAbort ()
- virtual void ShowAnonymization (Subject *caller, const Event &evt)
- virtual void ShowData (Subject *caller, const Event &evt)
- virtual void ShowDataSet (Subject *caller, const Event &evt)
- virtual void ShowFileName (Subject *caller, const Event &evt)
- virtual void ShowIteration ()
- virtual void ShowProgress (Subject *caller, const Event &evt)
- virtual void StartFilter ()
- void TestAbortOff ()
- void TestAbortOn ()

10.279.1 Detailed Description

SimpleSubjectWatcher.

This is a typical Subject Watcher class. It will observe all events.

Examples

BasicAnonymizer.cs, ClinicalTrialIdentificationWorkflow.cs, FileChangeTS.cs, FileChangeTSLossy.cs, ScanDirectory.cs, and SimpleScanner.cxx.

10.279.2 Constructor & Destructor Documentation

10.279.2.1 SimpleSubjectWatcher() [1/2]

```
gdcm::SimpleSubjectWatcher::SimpleSubjectWatcher (
            Subject * s,
            const char * comment = "" )
```

10.279.2.2 ~SimpleSubjectWatcher()

```
virtual gdcm::SimpleSubjectWatcher::~SimpleSubjectWatcher ( )  [virtual]
```

10.279.2.3 SimpleSubjectWatcher() [2/2]

```
gdcm::SimpleSubjectWatcher::SimpleSubjectWatcher (
            const SimpleSubjectWatcher &  )  [delete]
```

10.279.3 Member Function Documentation

10.279.3.1 EndFilter()

```
virtual void gdcm::SimpleSubjectWatcher::EndFilter ( )  [protected], [virtual]
```

10.279.3.2 operator=()

```
void gdcm::SimpleSubjectWatcher::operator= (
            const SimpleSubjectWatcher &  )  [delete]
```

10.279.3.3 ShowAbort()

```
virtual void gdcm::SimpleSubjectWatcher::ShowAbort ( )  [protected], [virtual]
```

10.279.3.4 ShowAnonymization()

```
virtual void gdcm::SimpleSubjectWatcher::ShowAnonymization (
            Subject * caller,
            const Event & evt )  [protected], [virtual]
```

10.279.3.5 ShowData()

```
virtual void gdcm::SimpleSubjectWatcher::ShowData (
            Subject * caller,
            const Event & evt )  [protected], [virtual]
```

10.279.3.6 ShowDataSet()

```
virtual void gdcm::SimpleSubjectWatcher::ShowDataSet (
            Subject * caller,
            const Event & evt )  [protected], [virtual]
```

10.279.3.7 ShowFileName()

```
virtual void gdcm::SimpleSubjectWatcher::ShowFileName (
            Subject * caller,
            const Event & evt )  [protected], [virtual]
```

Examples

SimpleScanner.cxx.

10.279.3.8 ShowIteration()

```
virtual void gdcm::SimpleSubjectWatcher::ShowIteration ( )  [protected], [virtual]
```

10.279.3.9 ShowProgress()

```
virtual void gdcm::SimpleSubjectWatcher::ShowProgress (
            Subject * caller,
            const Event & evt )  [protected], [virtual]
```

10.279.3.10 StartFilter()

```
virtual void gdcm::SimpleSubjectWatcher::StartFilter ( )  [protected], [virtual]
```

10.279.3.11 TestAbortOff()

```
void gdcm::SimpleSubjectWatcher::TestAbortOff ( )  [protected]
```

10.279.3.12 TestAbortOn()

```
void gdcm::SimpleSubjectWatcher::TestAbortOn ( )  [protected]
```

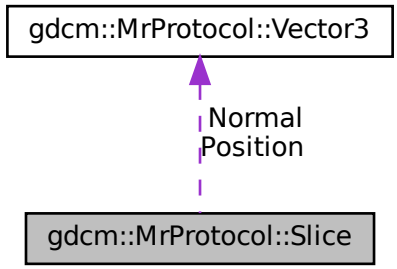
The documentation for this class was generated from the following file:

- gdcmSimpleSubjectWatcher.h

10.280 gdcm::MrProtocol::Slice Struct Reference

`#include <gdcmMrProtocol.h>`

Collaboration diagram for gdcm::MrProtocol::Slice:



Public Attributes

- Vector3 Normal
- Vector3 Position

10.280.1 Member Data Documentation

10.280.1.1 Normal

`Vector3 gdcm::MrProtocol::Slice::Normal`

10.280.1.2 Position

`Vector3 gdcm::MrProtocol::Slice::Position`

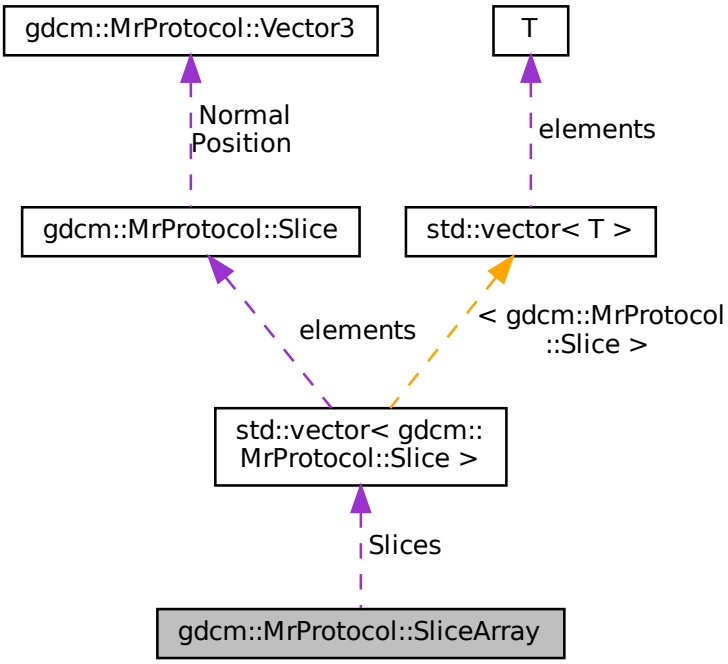
The documentation for this struct was generated from the following file:

- gdcmMrProtocol.h

10.281 gdcm::MrProtocol::SliceArray Struct Reference

`#include <gdcmMrProtocol.h>`

Collaboration diagram for gdcm::MrProtocol::SliceArray:



Public Attributes

- std::vector< Slice > Slices

10.281.1 Member Data Documentation

10.281.1.1 Slices

`std::vector< Slice > gdcm::MrProtocol::SliceArray::Slices`

The documentation for this struct was generated from the following file:

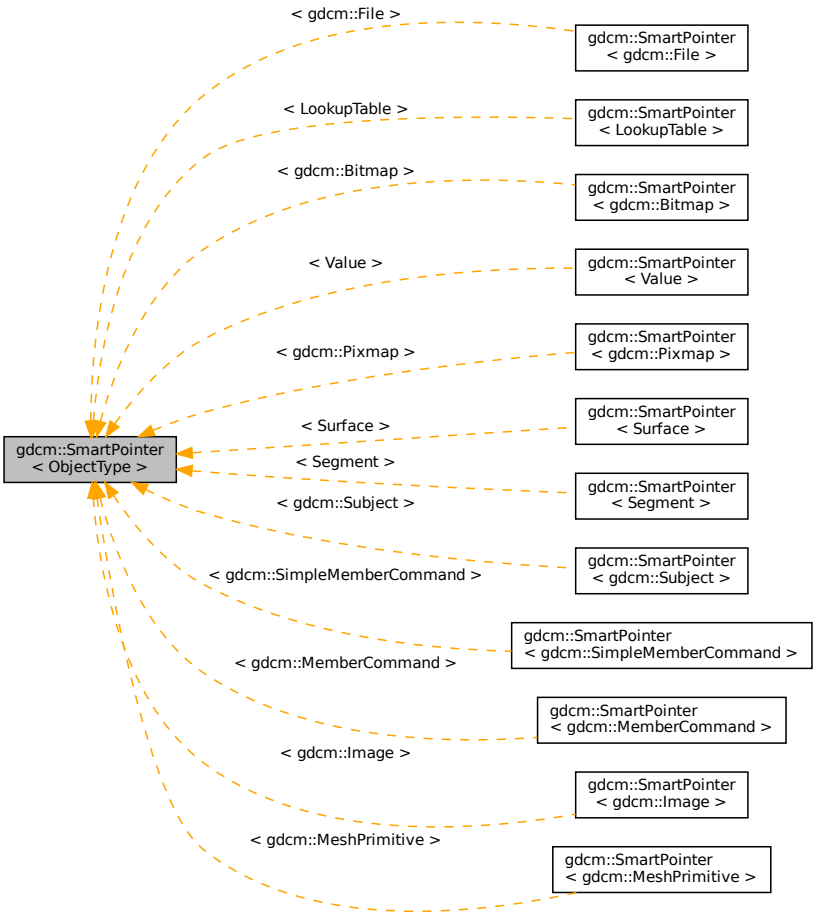
- gdcmMrProtocol.h

10.282 gdcm::SmartPointer< ObjectType > Class Template Reference

Class for Smart Pointer.

`#include <gdcmSmartPointer.h>`

Inheritance diagram for gdcm::SmartPointer< ObjectType >:



Public Member Functions

- SmartPointer ()
- SmartPointer (const SmartPointer< ObjectType > &p)
- SmartPointer (ObjectType *p)
- SmartPointer (ObjectType const &p)
- ~SmartPointer ()
- ObjectType * GetPointer () const

  Explicit function to retrieve the pointer.

- operator ObjectType * () const

  Return pointer to object.
- ObjectType & operator* () const
- ObjectType * operator-> () const

  Overload operator ->
- SmartPointer & operator= (ObjectType *r)

  Overload operator assignment.
- SmartPointer & operator= (ObjectType const &r)
- SmartPointer & operator= (SmartPointer const &r)

  Overload operator assignment.

10.282.1 Detailed Description

template<class ObjectType>
class gdcm::SmartPointer< ObjectType >

Class for Smart Pointer.

Will only work for subclass of gdcm::Object See tr1/shared_ptr for a more general approach (not invasive) #include <tr1/memory> { shared_ptr<Bla> b(new Bla); }

Note

Class partly based on post by Bill Hubauer: `http://groups.google.com/group/comp.lang.c++/msg/173ddc38a827a930`

See also

`http://www.davethehat.com/articles/smartp.htm`

and itk::SmartPointer

Examples

CStoreQtProgress.cxx, ChangeSequenceUltrasound.cxx, DumpGEMSMovieGroup.cxx, DumpPhilipsECHO.cxx, DumpToshibaDTI.cxx, DumpToshibaDTI2.cxx, DumpVisusChange.cxx, Extracting_All_Resolution.cxx, Fake_Image_Using_Stream_ FixBrokenJ2K.cxx, GenAllVR.cxx, GenFakeIdentifyFile.cxx, GenLongSeqs.cxx, GenSeqs.cxx, GetSubSequenceData.cxx, LargeVRDSExplicit.cxx, ReadAndDumpDICOMDIR.cxx, ReadExplicitLengthSQIVR.cxx, SimpleScanner.cxx, gdcmrtionplan.cxx, and gdcmrtplan.cxx.

10.282.2 Constructor & Destructor Documentation

10.282.2.1 SmartPointer() [1/4]

```
template<class ObjectType >
gdcm::SmartPointer< ObjectType >::SmartPointer ( )  [inline]
```

10.282.2.2 SmartPointer() [2/4]

```
template<class ObjectType >
gdcm::SmartPointer< ObjectType >::SmartPointer (
            const SmartPointer< ObjectType > & p )  [inline]
```

10.282.2.3 SmartPointer() [3/4]

```
template<class ObjectType >
gdcm::SmartPointer< ObjectType >::SmartPointer (
            ObjectType * p )  [inline]
```

10.282.2.4 SmartPointer() [4/4]

```
template<class ObjectType >
gdcm::SmartPointer< ObjectType >::SmartPointer (
            ObjectType const & p )  [inline]
```

10.282.2.5 ~SmartPointer()

```
template<class ObjectType >
gdcm::SmartPointer< ObjectType >::~SmartPointer ( )  [inline]
```

10.282.3 Member Function Documentation

10.282.3.1 GetPointer()

```
template<class ObjectType >
ObjectType * gdcm::SmartPointer< ObjectType >::GetPointer ( ) const  [inline]
```

Explicit function to retrieve the pointer.

10.282.3.2 operator ObjectType *()

```
template<class ObjectType >
gdcm::SmartPointer< ObjectType >::operator ObjectType * ( ) const  [inline]
```

Return pointer to object.

10.282.3.3 operator*()

```
template<class ObjectType >
ObjectType & gdcm::SmartPointer< ObjectType >::operator* ( ) const  [inline]
```

10.282.3.4 operator->()

```
template<class ObjectType >
ObjectType * gdcm::SmartPointer< ObjectType >::operator-> ( ) const  [inline]
```

Overload operator ->

10.282.3.5 operator=() [1/3]

```
template<class ObjectType >
SmartPointer & gdcm::SmartPointer< ObjectType >::operator= (
            ObjectType * r )  [inline]
```

Overload operator assignment.

10.282.3.6 operator=() [2/3]

```
template<class ObjectType >
SmartPointer & gdcm::SmartPointer< ObjectType >::operator= (
            ObjectType const & r )  [inline]
```

References gdcm::SmartPointer< ObjectType >::operator=().

10.282.3.7 operator=() [3/3]

```
template<class ObjectType >
SmartPointer & gdcm::SmartPointer< ObjectType >::operator= (
            SmartPointer< ObjectType > const & r )  [inline]
```

Overload operator assignment.

References gdcm::SmartPointer< ObjectType >::operator=().

Referenced by gdcm::SmartPointer< ObjectType >::operator=().

The documentation for this class was generated from the following files:

- gdcmObject.h
- gdcmSmartPointer.h

10.283 gdcm::network::SOPClassExtendedNegociationSub Class Reference

SOPClassExtendedNegociationSub.

```
#include <gdcmSOPClassExtendedNegociationSub.h>
```

Public Member Functions

- SOPClassExtendedNegociationSub ()
- void Print (std::ostream &os) const
- std::istream & Read (std::istream &is)
- void SetTuple (const char *uid, uint8_t levelofsupport=3, uint8_t levelofdigitalsig=0, uint8_t elementcoercion=2)
- size_t Size () const
- const std::ostream & Write (std::ostream &os) const

10.283.1 Detailed Description

SOPClassExtendedNegociationSub.

PS 3.7 Table D.3-11 SOP CLASS EXTENDED NEGOTIATION SUB-ITEM FIELDS (A-ASSOCIATE-RQ and A-↩ ASSOCIATE-AC)

10.283.2 Constructor & Destructor Documentation

10.283.2.1 SOPClassExtendedNegociationSub()

```
gdcm::network::SOPClassExtendedNegociationSub::SOPClassExtendedNegociationSub ( )
```

10.283.3 Member Function Documentation

10.283.3.1 Print()

```
void gdcm::network::SOPClassExtendedNegociationSub::Print (
            std::ostream & os ) const
```

10.283.3.2 Read()

```
std::istream & gdcm::network::SOPClassExtendedNegociationSub::Read (
            std::istream & is )
```

10.283.3.3 SetTuple()

```
void gdcm::network::SOPClassExtendedNegociationSub::SetTuple (
            const char * uid,
            uint8_t levelofsupport = 3,
            uint8_t levelofdigitalsig = 0,
            uint8_t elementcoercion = 2 )
```

10.283.3.4 Size()

```
size_t gdcm::network::SOPClassExtendedNegociationSub::Size ( ) const
```

10.283.3.5 Write()

```
const std::ostream & gdcm::network::SOPClassExtendedNegociationSub::Write (
            std::ostream & os ) const
```

The documentation for this class was generated from the following file:

- gdcmSOPClassExtendedNegociationSub.h

10.284 gdcm::SOPClassUIDToIOD Class Reference

Class convert a class SOP Class UID into IOD.

```
#include <gdcmSOPClassUIDToIOD.h>
```

Public Types

- typedef const char * const(SOPClassUIDToIODType)[2]

Static Public Member Functions

- static const char * GetIOD (UIDs const &uid)
- static const char * GetIODFromSOPClassUID (const char *sopclassuid)
- static unsigned int GetNumberOfSOPClassToIOD ()

  Return the number of SOP Class UID listed internally.
- static const char * GetSOPClassUIDFromIOD (const char *iod)
- static SOPClassUIDToIODType & GetSOPClassUIDToIOD (unsigned int i)
- static SOPClassUIDToIODType * GetSOPClassUIDToIODs ()

10.284.1 Detailed Description

Class convert a class SOP Class UID into IOD.

Reference PS 3.4 Table B.5-1 STANDARD SOP CLASSES

10.284.2 Member Typedef Documentation

10.284.2.1 const

```
typedef const char * gdcm::SOPClassUIDToIOD::const(SOPClassUIDToIODType)[2]
```

10.284.3 Member Function Documentation

10.284.3.1 GetIOD()

```
static const char * gdcm::SOPClassUIDToIOD::GetIOD (
            UIDs const & uid )  [static]
```

Return the associated IOD based on a SOP Class UID uid (there is a one-to-one mapping from SOP Class UID to matching IOD)

Examples

GenerateStandardSOPClasses.cxx.

10.284.3.2 GetIODFromSOPClassUID()

```
static const char * gdcm::SOPClassUIDToIOD::GetIODFromSOPClassUID (
            const char * sopclassuid )  [static]
```

10.284.3.3 GetNumberOfSOPClassToIOD()

```
static unsigned int gdcm::SOPClassUIDToIOD::GetNumberOfSOPClassToIOD ( )  [static]
```

Return the number of SOP Class UID listed internally.

10.284.3.4 GetSOPClassUIDFromIOD()

```
static const char * gdcm::SOPClassUIDToIOD::GetSOPClassUIDFromIOD (
            const char * iod )  [static]
```

10.284.3.5 GetSOPClassUIDToIOD()

```
static SOPClassUIDToIODType & gdcm::SOPClassUIDToIOD::GetSOPClassUIDToIOD (
            unsigned int i )  [static]
```

10.284.3.6 GetSOPClassUIDToIODs()

```
static SOPClassUIDToIODType * gdcm::SOPClassUIDToIOD::GetSOPClassUIDToIODs ( )  [static]
```

The documentation for this class was generated from the following file:

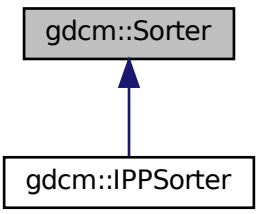
- gdcmSOPClassUIDToIOD.h

10.285 gdcm::Sorter Class Reference

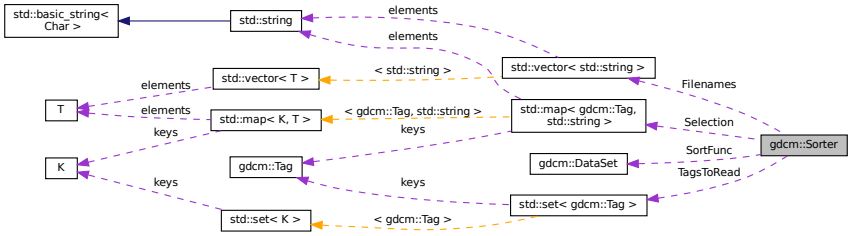
Sorter.

```
#include <gdcmSorter.h>
```

Inheritance diagram for gdcm::Sorter:



Collaboration diagram for gdcm::Sorter:



Public Types

- typedef bool(∗ SortFunction) (DataSet const &, DataSet const &)

  Set the sort function which compares one dataset to the other.

Public Member Functions

- Sorter ()
- virtual ∼Sorter ()
- bool AddSelect (Tag const &tag, const char ∗value)

  UNSUPPORTED FOR NOW.
- const std::vector< std::string > & GetFilenames () const
- void Print (std::ostream &os) const

  Print.
- void SetSortFunction (SortFunction f)
- void SetTagsToRead (std::set< Tag > const &tags)
- virtual bool Sort (std::vector< std::string > const &filenames)

  Typically the output of Directory::GetFilenames()
- virtual bool StableSort (std::vector< std::string > const &filenames)

Protected Types

- typedef std::map< Tag, std::string > SelectionMap

Protected Attributes

- std::vector< std::string > Filenames
- std::map< Tag, std::string > Selection
- SortFunction SortFunc
- std::set< Tag > TagsToRead

Friends

- std::ostream & operator<< (std::ostream &_os, const Sorter &s)

10.285.1 Detailed Description

Sorter.

General class to do sorting using a custom function You simply need to provide a function of type: Sorter::SortFunction

Warning

implementation details. For now there is no cache mechanism. Which means that every time you call Sort, all files specified as input parameter are *read*

See also

Scanner

Examples

SortImage.cxx, SortImage2.cs, and VolumeSorter.cxx.

10.285.2 Member Typedef Documentation

10.285.2.1 SelectionMap

```
typedef std::map<Tag,std::string> gdcm::Sorter::SelectionMap  [protected]
```

10.285.2.2 SortFunction

```
typedef bool(* gdcm::Sorter::SortFunction) (DataSet const &, DataSet const &)
```

Set the sort function which compares one dataset to the other.

10.285.3 Constructor & Destructor Documentation

10.285.3.1 Sorter()

```
gdcm::Sorter::Sorter ( )
```

10.285.3.2 ~Sorter()

```
virtual gdcm::Sorter::~Sorter ( )  [virtual]
```

10.285.4 Member Function Documentation

10.285.4.1 AddSelect()

```
bool gdcm::Sorter::AddSelect (
            Tag const & tag,
            const char * value )
```

UNSUPPORTED FOR NOW.

10.285.4.2 GetFilenames()

```
const std::vector< std::string > & gdcm::Sorter::GetFilenames ( ) const  [inline]
```

Return the list of filenames as sorted by the specific algorithm used. Empty by default (before Sort() is called)

Examples

Compute3DSpacing.cxx, SortImage.cxx, VolumeSorter.cxx, gdcmorthoplanes.cxx, and reslicesphere.cxx.

10.285.4.3 Print()

```
void gdcm::Sorter::Print (
            std::ostream & os ) const
```

Print.

Examples

SortImage.cxx, VolumeSorter.cxx, and gdcmorthoplanes.cxx.

10.285.4.4 SetSortFunction()

```
void gdcm::Sorter::SetSortFunction (
            SortFunction f )
```

Examples

SortImage.cxx, SortImage2.cs, and VolumeSorter.cxx.

10.285.4.5 SetTagsToRead()

```
void gdcm::Sorter::SetTagsToRead (
            std::set< Tag > const & tags )
```

Specify a set of tags to be read in during the sort procedure. By default this set is empty, in which case the entire image, including pixel data, is read in.

10.285.4.6 Sort()

```
virtual bool gdcm::Sorter::Sort (
            std::vector< std::string > const & filenames )  [virtual]
```

Typically the output of Directory::GetFilenames()

Reimplemented in gdcm::IPPSorter.

Examples

SortImage.cxx.

10.285.4.7 StableSort()

```
virtual bool gdcm::Sorter::StableSort (
            std::vector< std::string > const & filenames )  [virtual]
```

Examples

SortImage.cxx, and VolumeSorter.cxx.

10.285.5 Friends And Related Function Documentation

10.285.5.1 operator<<

```
std::ostream & operator<< (
            std::ostream & _os,
            const Sorter & s )  [friend]
```

10.285.6 Member Data Documentation

10.285.6.1 Filenames

```
std::vector<std::string> gdcm::Sorter::Filenames  [protected]
```

10.285.6.2 Selection

```
std::map<Tag,std::string> gdcm::Sorter::Selection  [protected]
```

10.285.6.3 SortFunc

```
SortFunction gdcm::Sorter::SortFunc  [protected]
```

10.285.6.4 TagsToRead

```
std::set<Tag> gdcm::Sorter::TagsToRead  [protected]
```

The documentation for this class was generated from the following file:

- gdcmSorter.h

10.286 gdcm::Spacing Class Reference

Class for Spacing.

```
#include <gdcmSpacing.h>
```

Public Types

- enum SpacingType {
  DETECTOR = 0 ,
  MAGNIFIED ,
  CALIBRATED ,
  UNKNOWN }

Public Member Functions

- Spacing ()
- ~Spacing ()

Static Public Member Functions

- static Attribute< 0x28, 0x34 > ComputePixelAspectRatioFromPixelSpacing (const Attribute< 0x28, 0x30 > &pixelspacing)

10.286.1 Detailed Description

Class for Spacing.

It all began with a mail to WG6:

Subject: Imager Pixel Spacing vs Pixel Spacing Body: [Apologies for the duplicate post, namely to David Clunie & OFFIS team]

I have been trying to understand CP-586 in the following two cases:

On the one hand:

- DISCIMG/IMAGES/CRIMAGE taken from `http://dclunie.com/images/pixelspacingtestimages.zip`

And on the other hand:

- `http://gdcm.sourceforge.net/thingies/cr_pixelspacing.dcm`

If I understand correctly the CP, one is required to use Pixel Spacing for measurement ('true size' print) instead of Imager Pixel Spacing, since the two attributes are present and Pixel Spacing is different from Imager Pixel Spacing.

If this is correct, then the test data DISCIMG/IMAGES/CRIMAGE is incorrect. If this is incorrect (ie. I need to use Imager Pixel Spacing), then the display of cr_pixelspacing.dcm for measurement will be incorrect.

Could someone please let me know what am I missing here? I could not find any information in any header that would allow me to differentiate those.

Thank you for your time,

Ref: `http://lists.nema.org/scripts/lyris.pl?sub=488573&id=400720477`

See PS 3.3-2008, Table C.7-11b IMAGE PIXEL MACRO ATTRIBUTES

Ratio of the vertical size and horizontal size of the pixels in the image specified by a pair of integer values where the first value is the vertical pixel size, and the second value is the horizontal pixel size. Required if the aspect ratio values do not have a ratio of 1:1 and the physical pixel spacing is not specified by Pixel Spacing (0028,0030), or Imager Pixel Spacing (0018,1164) or Nominal Scanned Pixel Spacing (0018,2010), either for the entire Image or per-frame in a Functional Group Macro. See C.7.6.3.1.7.

PS 3.3-2008 10.7.1.3 Pixel Spacing Value Order and Valid Values All pixel spacing related attributes shall have non-zero values, except when there is only a single row or column or pixel of data present, in which case the corresponding value may be zero.

Ref: `http://gdcm.sourceforge.net/wiki/index.php/Imager_Pixel_Spacing`

10.286.2 Member Enumeration Documentation

10.286.2.1 SpacingType

`enum gdcm::Spacing::SpacingType`

Enumerator

DETECTOR	
MAGNIFIED	
CALIBRATED	
UNKNOWN	

10.286.3 Constructor & Destructor Documentation

10.286.3.1 Spacing()

```
gdcm::Spacing::Spacing ( )
```

10.286.3.2 ~Spacing()

```
gdcm::Spacing::~Spacing ( )
```

10.286.4 Member Function Documentation

10.286.4.1 ComputePixelAspectRatioFromPixelSpacing()

```
static Attribute< 0x28, 0x34 > gdcm::Spacing::ComputePixelAspectRatioFromPixelSpacing (
            const Attribute< 0x28, 0x30 > & pixelspacing )  [static]
```

The documentation for this class was generated from the following file:

- gdcmSpacing.h

10.287 gdcm::Spectroscopy Class Reference

Spectroscopy class.

```
#include <gdcmSpectroscopy.h>
```

Public Member Functions

- Spectroscopy ()=default

10.287.1 Detailed Description

Spectroscopy class.

10.287.2 Constructor & Destructor Documentation

10.287.2.1 Spectroscopy()

```
gdcm::Spectroscopy::Spectroscopy ( )  [default]
```

The documentation for this class was generated from the following file:

- gdcmSpectroscopy.h

10.288 gdcm::SplitMosaicFilter Class Reference

SplitMosaicFilter class.

```
#include <gdcmSplitMosaicFilter.h>
```

Public Member Functions

- SplitMosaicFilter ()
- ~SplitMosaicFilter ()
- bool ComputeMOSAICDimensions (unsigned int dims[3])
- bool ComputeMOSAICSliceNormal (double dims[3], bool &inverted)

  Extract the value for SliceNormalVector (CSA header)
- bool ComputeMOSAICSlicePosition (double pos[3], bool inverted)

  Extract the value for ImagePositionPatient (requires inverted flag)
- File & GetFile ()
- const File & GetFile () const
- Image & GetImage ()
- const Image & GetImage () const
- void SetFile (const File &f)
- void SetImage (const Image &image)
- bool Split ()

  Split the SIEMENS MOSAIC image.

Static Public Member Functions

- static bool GetAcquisitionSize (unsigned int size[2], DataSet const &ds)

  Get the Acquisition Matrix (non zero value):
- static unsigned int GetNumberOfImagesInMosaic (File const &file)

  Return the value for NumberOfImagesInMosaic, or compute it from Acquisition Size.

10.288.1 Detailed Description

SplitMosaicFilter class.

Class to reshuffle bytes for a SIEMENS Mosaic image Siemens CSA Image Header CSA:= Common Siemens Architecture, sometimes also known as Common syngo Architecture

Warning

when private attributes are not found, the acquisition matrix is used to compute the NumberOfImagesInMosaic. This means trailing black slices will be considered in the volume (instead of discarded). CSA 0029,1010 is needed for correct NumberOfImagesInMosaic CSA 0029,1020 is needed to compute the correct origin without above info default are taken (may not be accurate).

10.288.2 Constructor & Destructor Documentation

10.288.2.1 SplitMosaicFilter()

```
gdcm::SplitMosaicFilter::SplitMosaicFilter ( )
```

10.288.2.2 ~SplitMosaicFilter()

```
gdcm::SplitMosaicFilter::~SplitMosaicFilter ( )
```

10.288.3 Member Function Documentation

10.288.3.1 ComputeMOSAICDimensions()

```
bool gdcm::SplitMosaicFilter::ComputeMOSAICDimensions (
            unsigned int dims[3] )
```

Compute the new dimensions according to private information stored in the MOSAIC header.

10.288.3.2 ComputeMOSAICSliceNormal()

```
bool gdcm::SplitMosaicFilter::ComputeMOSAICSliceNormal (
            double dims[3],
            bool & inverted )
```

Extract the value for SliceNormalVector (CSA header)

10.288.3.3 ComputeMOSAICSlicePosition()

```
bool gdcm::SplitMosaicFilter::ComputeMOSAICSlicePosition (
            double pos[3],
            bool inverted )
```

Extract the value for ImagePositionPatient (requires inverted flag)

10.288.3.4 GetAcquisitionSize()

```
static bool gdcm::SplitMosaicFilter::GetAcquisitionSize (
            unsigned int size[2],
            DataSet const & ds )  [static]
```

Get the Acquisition Matrix (non zero value):

10.288.3.5 GetFile() [1/2]

```
File & gdcm::SplitMosaicFilter::GetFile ( )  [inline]
```

10.288.3.6 GetFile() [2/2]

```
const File & gdcm::SplitMosaicFilter::GetFile ( ) const  [inline]
```

10.288.3.7 GetImage() [1/2]

```
Image & gdcm::SplitMosaicFilter::GetImage ( )  [inline]
```

10.288.3.8 GetImage() [2/2]

```
const Image & gdcm::SplitMosaicFilter::GetImage ( ) const  [inline]
```

10.288.3.9 GetNumberOfImagesInMosaic()

```
static unsigned int gdcm::SplitMosaicFilter::GetNumberOfImagesInMosaic (
            File const & file )  [static]
```

Return the value for NumberOfImagesInMosaic, or compute it from Acquisition Size.

10.288.3.10 SetFile()

```
void gdcm::SplitMosaicFilter::SetFile (
            const File & f )  [inline]
```

10.288.3.11 SetImage()

```
void gdcm::SplitMosaicFilter::SetImage (
            const Image & image )
```

10.288.3.12 Split()

```
bool gdcm::SplitMosaicFilter::Split ( )
```

Split the SIEMENS MOSAIC image.

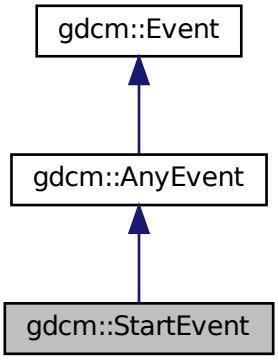
The documentation for this class was generated from the following file:

- gdcmSplitMosaicFilter.h

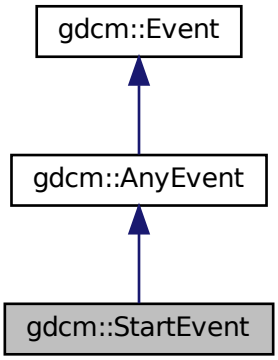
10.289 gdcm::StartEvent Class Reference

`#include <gdcmEvent.h>`

Inheritance diagram for gdcm::StartEvent:



Collaboration diagram for gdcm::StartEvent:



Additional Inherited Members

The documentation for this class was generated from the following file:

- gdcmEvent.h

10.290 gdcm::static_assert_test< x > Struct Template Reference

`#include <gdcmStaticAssert.h>`

The documentation for this struct was generated from the following file:

- gdcmStaticAssert.h

10.291 gdcm::STATIC_ASSERTION_FAILURE< x > Struct Template Reference

The documentation for this struct was generated from the following file:

- gdcmStaticAssert.h

10.292 gdcm::STATIC_ASSERTION_FAILURE< true > Struct Reference

`#include <gdcmStaticAssert.h>`

Public Types

- enum { value = 1 }

10.292.1 Member Enumeration Documentation

10.292.1.1 anonymous enum

`anonymous enum`

Enumerator

value	

The documentation for this struct was generated from the following file:

- gdcmStaticAssert.h

10.293 gdcm::StreamImageReader Class Reference

StreamImageReader.

`#include <gdcmStreamImageReader.h>`

Public Member Functions

- StreamImageReader ()
- virtual ~StreamImageReader ()
- bool CanReadImage () const
- void DefinePixelExtent (uint16_t inXMin, uint16_t inXMax, uint16_t inYMin, uint16_t inYMax, uint16_t inZMin=0, uint16_t inZMax=1)
- uint32_t DefineProperBufferLength () const
- std::vector< unsigned int > GetDimensionsValueForResolution (unsigned int)
- File const & GetFile () const
- bool Read (char *inReadBuffer, const std::size_t &inBufferLength)
- virtual bool ReadImageInformation ()
- void SetFileName (const char *inFileName)
- void SetStream (std::istream &inStream)

10.293.1 Detailed Description

StreamImageReader.

Note

its role is to convert the DICOM DataSet into a Image representation via an ITK streaming (ie, multithreaded) interface Image is different from Pixmap has it has a position and a direction in Space. Currently, this class is thread safe in that it can read a single extent in a single thread. Multiple versions can be used for multiple extents/threads.

See also

Image

Examples

ExtractOneFrame.cs, and StreamImageReaderTest.cxx.

10.293.2 Constructor & Destructor Documentation

10.293.2.1 StreamImageReader()

```
gdcm::StreamImageReader::StreamImageReader ( )
```

10.293.2.2 ~StreamImageReader()

```
virtual gdcm::StreamImageReader::~StreamImageReader ( )  [virtual]
```

10.293.3 Member Function Documentation

10.293.3.1 CanReadImage()

```
bool gdcm::StreamImageReader::CanReadImage ( ) const
```

Only RAW images are currently readable by the stream reader. As more streaming codecs are added, then this function will be updated to reflect those changes. Calling this function prior to reading will ensure that only streamable files are streamed. Make sure to call ReadImageInformation prior to calling this function.

Examples

StreamImageReaderTest.cxx.

10.293.3.2 DefinePixelExtent()

```
void gdcm::StreamImageReader::DefinePixelExtent (
            uint16_t inXMin,
            uint16_t inXMax,
            uint16_t inYMin,
            uint16_t inYMax,
            uint16_t inZMin = 0,
            uint16_t inZMax = 1 )
```

Defines an image extent for the Read function. DICOM states that an image can have no more than 2^16 pixels per edge (as of 2009) In this case, the pixel extents ignore the direction cosines entirely, and assumes that the origin of the image is at location 0,0 (regardless of the definition in space per the tags). So, if the first 100 pixels of the first row are to be read in, this function should be called with DefinePixelExtent(0, 100, 0, 1), regardless of pixel size or orientation.

Examples

ExtractOneFrame.cs, and StreamImageReaderTest.cxx.

10.293.3.3 DefineProperBufferLength()

```
uint32_t gdcm::StreamImageReader::DefineProperBufferLength ( ) const
```

Paying attention to the pixel format and so forth, define the proper buffer length for the user. The return amount is in bytes. Call this function to determine the size of the char* buffer that will need to be passed in to ReadImageSubregion(). If the return is 0, then that means that the pixel extent was not defined prior

Examples

ExtractOneFrame.cs, and StreamImageReaderTest.cxx.

10.293.3.4 GetDimensionsValueForResolution()

```
std::vector< unsigned int > gdcm::StreamImageReader::GetDimensionsValueForResolution (
            unsigned int  )
```

10.293.3.5 GetFile()

```
File const & gdcm::StreamImageReader::GetFile ( ) const
```

Returns the dataset read by ReadImageInformation Couple this with the ImageHelper to get statistics about the image, like pixel extent, to be able to initialize buffers for reading

Examples

ExtractOneFrame.cs, and StreamImageReaderTest.cxx.

10.293.3.6 Read()

```
bool gdcm::StreamImageReader::Read (
            char * inReadBuffer,
            const std::size_t & inBufferLength )
```

Read the DICOM image. There are three reasons for failure:

1. The extent is not set
2. the conversion from char* to std::ostream (internally) fails
3. the given buffer isn't large enough to accommodate the desired pixel extent. This method has been implemented to look similar to the metaimageio in itk MUST have an extent defined, or else Read will return false. If no particular extent is required, use ImageReader instead.

Examples

ExtractOneFrame.cs, and StreamImageReaderTest.cxx.

10.293.3.7 ReadImageInformation()

```
virtual bool gdcm::StreamImageReader::ReadImageInformation ( )  [virtual]
```

Set the spacing and dimension information for the set filename. returns false if the file is not initialized or not an image, with the pixel (7fe0,0010) tag.

Examples

ExtractOneFrame.cs, and StreamImageReaderTest.cxx.

10.293.3.8 SetFileName()

```
void gdcm::StreamImageReader::SetFileName (
            const char * inFileName )
```

One of either SetFileName or SetStream must be called prior to any other functions. These initialize an internal Reader class to be able to get non-pixel image information.

Examples

ExtractOneFrame.cs, and StreamImageReaderTest.cxx.

10.293.3.9 SetStream()

```
void gdcm::StreamImageReader::SetStream (
            std::istream & inStream )
```

The documentation for this class was generated from the following file:

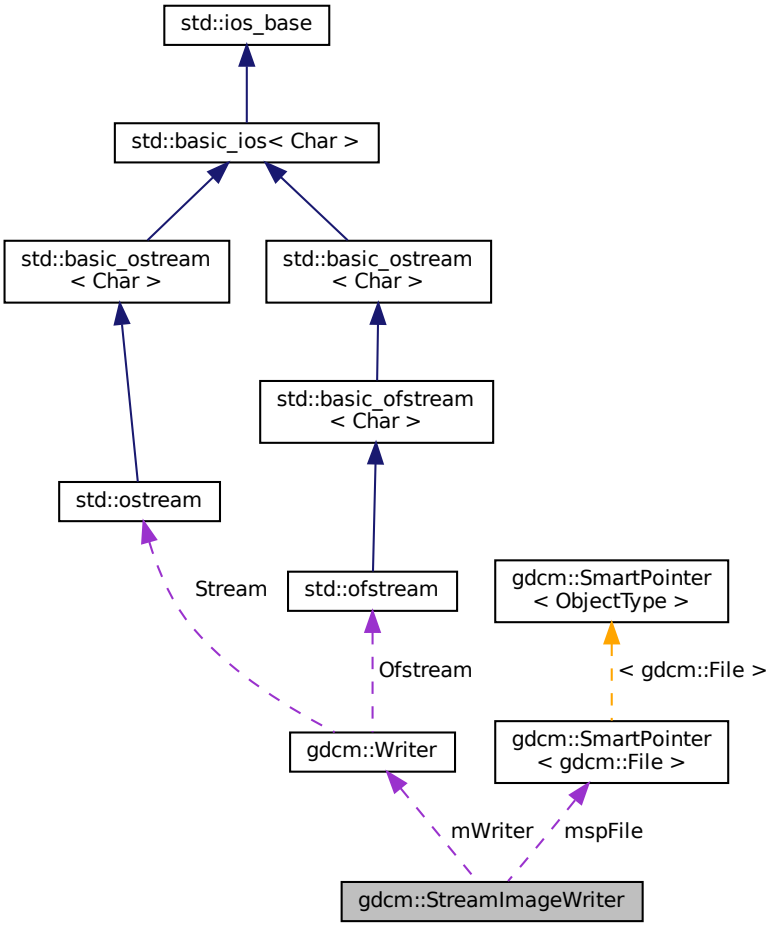
- gdcmStreamImageReader.h

10.294 gdcm::StreamImageWriter Class Reference

StreamImageReader.

`#include <gdcmStreamImageWriter.h>`

Collaboration diagram for gdcm::StreamImageWriter:



Public Member Functions

- StreamImageWriter ()
- virtual ~StreamImageWriter ()
- bool CanWriteFile () const
- void DefinePixelExtent (uint16_t inXMin, uint16_t inXMax, uint16_t inYMin, uint16_t inYMax, uint16_t inZMin=0, uint16_t inZMax=1)
- uint32_t DefineProperBufferLength ()

- void SetFile (const File &inFile)
- void SetFileName (const char *inFileName)
- void SetStream (std::ostream &inStream)
- bool Write (void *inWriteBuffer, const std::size_t &inBufferLength)
- virtual bool WriteImageInformation ()

Protected Member Functions

- virtual bool WriteImageSubregionRAW (char *inWriteBuffer, const std::size_t &inBufferLength)
- int WriteRawHeader (RAWCodec *inCodec, std::ostream *inStream)

Protected Attributes

- int mElementOffsets
- int mElementOffsets1
- SmartPointer< File > mspFile
- Writer mWriter
- uint16_t mXMax
- uint16_t mXMin
- uint16_t mYMax
- uint16_t mYMin
- uint16_t mZMax
- uint16_t mZMin

10.294.1 Detailed Description

StreamImageReader.

Note

its role is to convert the DICOM DataSet into a Image representation via an ITK streaming (ie, multithreaded) interface Image is different from Pixmap has it has a position and a direction in Space. Currently, this class is threadsafe in that it can read a single extent in a single thread. Multiple versions can be used for multiple extents/threads.

See also

Image

Examples

Extracting_All_Resolution.cxx, Fake_Image_Using_Stream_Image_Writer.cxx, and StreamImageReaderTest.cxx.

10.294.2 Constructor & Destructor Documentation

10.294.2.1 StreamImageWriter()

```
gdcm::StreamImageWriter::StreamImageWriter ( )
```

10.294.2.2 ~StreamImageWriter()

```
virtual gdcm::StreamImageWriter::~StreamImageWriter ( )  [virtual]
```

10.294.3 Member Function Documentation

10.294.3.1 CanWriteFile()

```
bool gdcm::StreamImageWriter::CanWriteFile ( ) const
```

This function determines if a file can even be written using the streaming writer unlike the reader, can be called before WriteImageInformation, but must be called after SetFile.

Examples

Extracting_All_Resolution.cxx, and Fake_Image_Using_Stream_Image_Writer.cxx.

10.294.3.2 DefinePixelExtent()

```
void gdcm::StreamImageWriter::DefinePixelExtent (
            uint16_t inXMin,
            uint16_t inXMax,
            uint16_t inYMin,
            uint16_t inYMax,
            uint16_t inZMin = 0,
            uint16_t inZMax = 1 )
```

Defines an image extent for the Read function. DICOM states that an image can have no more than 2^16 pixels per edge (as of 2009) In this case, the pixel extents ignore the direction cosines entirely, and assumes that the origin of the image is at location 0,0 (regardless of the definition in space per the tags). So, if the first 100 pixels of the first row are to be read in, this function should be called with DefinePixelExtent(0, 100, 0, 1), regardless of pixel size or orientation. 15 nov 2010: added z dimension, defaults to being 1 plane large

Examples

Extracting_All_Resolution.cxx, Fake_Image_Using_Stream_Image_Writer.cxx, and StreamImageReaderTest.cxx.

10.294.3.3 DefineProperBufferLength()

```
uint32_t gdcm::StreamImageWriter::DefineProperBufferLength ( )
```

Paying attention to the pixel format and so forth, define the proper buffer length for the user. The return amount is in bytes. If the return is 0, then that means that the pixel extent was not defined prior this return is for RAW inputs which are then encoded by the writer, but are used to ensure that the writer gets the proper buffer size

Examples

Extracting_All_Resolution.cxx, Fake_Image_Using_Stream_Image_Writer.cxx, and StreamImageReaderTest.cxx.

10.294.3.4 SetFile()

```
void gdcm::StreamImageWriter::SetFile (
            const File & inFile )
```

Set the image information to be written to disk that is everything but the pixel information: (7fe0,0010) PixelData

Examples

Extracting_All_Resolution.cxx, Fake_Image_Using_Stream_Image_Writer.cxx, and StreamImageReaderTest.cxx.

10.294.3.5 SetFileName()

```
void gdcm::StreamImageWriter::SetFileName (
            const char * inFileName )
```

One of either SetFileName or SetStream must be called prior to any other functions. These initialize an internal Reader class to be able to get non-pixel image information.

10.294.3.6 SetStream()

```
void gdcm::StreamImageWriter::SetStream (
            std::ostream & inStream )
```

Examples

Extracting_All_Resolution.cxx, Fake_Image_Using_Stream_Image_Writer.cxx, and StreamImageReaderTest.cxx.

10.294.3.7 Write()

```
bool gdcm::StreamImageWriter::Write (
            void * inWriteBuffer,
            const std::size_t & inBufferLength )
```

Read the DICOM image. There are three reasons for failure:

1. The extent is not set
2. the conversion from void* to std::ostream (internally) fails
3. the given buffer isn't large enough to accommodate the desired pixel extent. This method has been implemented to look similar to the metaimageio in itk MUST have an extent defined, or else Read will return false. If no particular extent is required, use ImageReader instead.

Examples

Extracting_All_Resolution.cxx, Fake_Image_Using_Stream_Image_Writer.cxx, and StreamImageReaderTest.cxx.

10.294.3.8 WriteImageInformation()

```
virtual bool gdcm::StreamImageWriter::WriteImageInformation ( )  [virtual]
```

Write the header information to disk, and a bunch of zeros for the actual pixel information Of course, if we're doing a non-compressed format, that works but if it's compressed, we have to force the ordering of chunks that are written.

Examples

Extracting_All_Resolution.cxx, Fake_Image_Using_Stream_Image_Writer.cxx, and StreamImageReaderTest.cxx.

10.294.3.9 WriteImageSubregionRAW()

```
virtual bool gdcm::StreamImageWriter::WriteImageSubregionRAW (
            char * inWriteBuffer,
            const std::size_t & inBufferLength )  [protected], [virtual]
```

Using the min, max, etc set by DefinePixelExtent, this will fill the given buffer Make sure to call DefinePixelExtent and to initialize the buffer with the amount given by DefineProperBufferLength prior to calling this. reads by the RAW codec; other codecs are added once implemented

10.294.3.10 WriteRawHeader()

```
int gdcm::StreamImageWriter::WriteRawHeader (
            RAWCodec * inCodec,
            std::ostream * inStream )  [protected]
```

when writing a raw file, we know the full extent, and can just write the first 12 bytes out (the tag, the VR, and the size) when we do compressed files, we'll do it in chunks, as described in 2009-3, part 5, Annex A, section 4. Pass the raw codec so that in the rare case of a bigendian explicit raw, the first 12 bytes written out should still be kosher. returns -1 if there's any failure, or the complete offset (12 bytes) if it works. Those 12 bytes are then added to the position in order to determine where to write.

10.294.4 Member Data Documentation

10.294.4.1 mElementOffsets

```
int gdcm::StreamImageWriter::mElementOffsets  [protected]
```

The result of WriteRawHeader (or another header, when that's implemented) This result is saved so that the first N bytes aren't constantly being rewritten for each chunk that's passed in. For compressed data, the offset table will require rewrites of data.

10.294.4.2 mElementOffsets1

```
int gdcm::StreamImageWriter::mElementOffsets1  [protected]
```

10.294.4.3 mspFile

```
SmartPointer<File> gdcm::StreamImageWriter::mspFile  [protected]
```

10.294.4.4 mWriter

```
Writer gdcm::StreamImageWriter::mWriter  [protected]
```

10.294.4.5 mXMax

```
uint16_t gdcm::StreamImageWriter::mXMax  [protected]
```

10.294.4.6 mXMin

```
uint16_t gdcm::StreamImageWriter::mXMin  [protected]
```

10.294.4.7 mYMax

```
uint16_t gdcm::StreamImageWriter::mYMax  [protected]
```

10.294.4.8 mYMin

```
uint16_t gdcm::StreamImageWriter::mYMin  [protected]
```

10.294.4.9 mZMax

```
uint16_t gdcm::StreamImageWriter::mZMax  [protected]
```

10.294.4.10 mZMin

```
uint16_t gdcm::StreamImageWriter::mZMin  [protected]
```

The documentation for this class was generated from the following file:

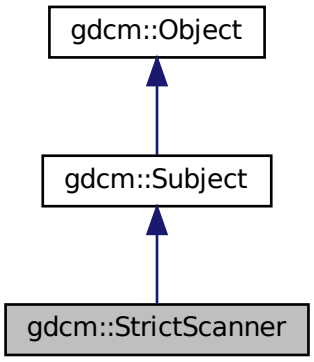
- gdcmStreamImageWriter.h

10.295 gdcm::StrictScanner Class Reference

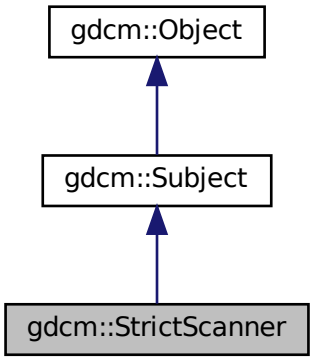
StrictScanner.

`#include <gdcmStrictScanner.h>`

Inheritance diagram for gdcm::StrictScanner:



Collaboration diagram for gdcm::StrictScanner:



Classes

- struct ltstr

Public Types

- typedef MappingType::const_iterator ConstIterator
- typedef std::map< const char *, TagToValue, ltstr > MappingType
- typedef std::map< Tag, const char * > TagToValue
- typedef TagToValue::value_type TagToValueValueType
- typedef std::set< std::string > ValuesType

Public Member Functions

- StrictScanner ()
- ~StrictScanner () override
- void AddPrivateTag (PrivateTag const &t)
- void AddSkipTag (Tag const &t)

  Add a tag that will need to be skipped. Those are root level skip tags.
- void AddTag (Tag const &t)

  Add a tag that will need to be read. Those are root level skip tags.
- ConstIterator Begin () const
- void ClearSkipTags ()
- void ClearTags ()
- ConstIterator End () const
- Directory::FilenamesType GetAllFilenamesFromTagToValue (Tag const &t, const char *valueref) const
- const char * GetFilenameFromTagToValue (Tag const &t, const char *valueref) const
- Directory::FilenamesType const & GetFilenames () const
- Directory::FilenamesType GetKeys () const
- TagToValue const & GetMapping (const char *filename) const

  Get the std::map mapping filenames to value for file 'filename'.
- TagToValue const & GetMappingFromTagToValue (Tag const &t, const char *value) const

  See GetFilenameFromTagToValue(). This is simply GetFilenameFromTagToValue followed.
- MappingType const & GetMappings () const

  Mappings are the mapping from a particular tag to the map, mapping filename to value:
- Directory::FilenamesType GetOrderedValues (Tag const &t) const
- const char * GetValue (const char *filename, Tag const &t) const
- ValuesType const & GetValues () const

  Get all the values found (in lexicographic order)
- ValuesType GetValues (Tag const &t) const

  Get all the values found (in lexicographic order) associated with Tag 't'.
- bool IsKey (const char *filename) const
- void Print (std::ostream &os) const override

  Print result.
- void PrintTable (std::ostream &os) const
- bool Scan (Directory::FilenamesType const &filenames)

  Start the scan !

Static Public Member Functions

- static SmartPointer< StrictScanner > New ()

  for wrapped language: instantiate a reference counted object

Protected Member Functions

- void ProcessPublicTag (StringFilter &sf, const char *filename)

Friends

- std::ostream & operator<< (std::ostream &_os, const StrictScanner &s)

10.295.1 Detailed Description

StrictScanner.

This filter is meant for quickly browsing a FileSet (a set of files on disk). Special consideration are taken so as to read the minimum amount of information in each file in order to retrieve the user specified set of DICOM Attribute.

This filter is dealing with both VRASCII and VRBINARY element, thanks to the help of StringFilter

Warning

IMPORTANT In case of file where tags are not ordered (illegal as per DICOM specification), the output will be missing information

Note

implementation details. All values are stored in a std::set of std::string. Then the address of the cstring underlying the std::string is used in the std::map.

This class implement the Subject/Observer pattern trigger the following events:

- ProgressEvent
- StartEvent
- EndEvent

Examples

ScanDirectory.cs, and SimpleScanner.cxx.

10.295.2 Member Typedef Documentation

10.295.2.1 ConstIterator

```
typedef MappingType::const_iterator gdcm::StrictScanner::ConstIterator
```

10.295.2.2 MappingType

```
typedef std::map<const char *,TagToValue, ltstr> gdcm::StrictScanner::MappingType
```

10.295.2.3 TagToValue

```
typedef std::map<Tag, const char*> gdcm::StrictScanner::TagToValue
```

struct to map a filename to a value Implementation note: all std::map in this class will be using const char * and not std::string since we are pointing to existing std::string (hold in a std::vector) this avoid an extra copy of the byte array. Tag are used as Tag class since sizeof(tag) <= sizeof(pointer)

10.295.2.4 TagToValueValueType

```
typedef TagToValue::value_type gdcm::StrictScanner::TagToValueValueType
```

10.295.2.5 ValuesType

```
typedef std::set< std::string > gdcm::StrictScanner::ValuesType
```

10.295.3 Constructor & Destructor Documentation

10.295.3.1 StrictScanner()

```
gdcm::StrictScanner::StrictScanner ( )  [inline]
```

10.295.3.2 ~StrictScanner()

```
gdcm::StrictScanner::~StrictScanner ( )  [override]
```

10.295.4 Member Function Documentation

10.295.4.1 AddPrivateTag()

```
void gdcm::StrictScanner::AddPrivateTag (
            PrivateTag const & t )
```

10.295.4.2 AddSkipTag()

```
void gdcm::StrictScanner::AddSkipTag (
            Tag const & t )
```

Add a tag that will need to be skipped. Those are root level skip tags.

10.295.4.3 AddTag()

```
void gdcm::StrictScanner::AddTag (
            Tag const & t )
```

Add a tag that will need to be read. Those are root level skip tags.

Examples

ScanDirectory.cs, and SimpleScanner.cxx.

10.295.4.4 Begin()

```
ConstIterator gdcm::StrictScanner::Begin ( ) const  [inline]
```

10.295.4.5 ClearSkipTags()

```
void gdcm::StrictScanner::ClearSkipTags ( )
```

10.295.4.6 ClearTags()

```
void gdcm::StrictScanner::ClearTags ( )
```

10.295.4.7 End()

```
ConstIterator gdcm::StrictScanner::End ( ) const  [inline]
```

10.295.4.8 GetAllFilenamesFromTagToValue()

```
Directory::FilenamesType gdcm::StrictScanner::GetAllFilenamesFromTagToValue (
            Tag const & t,
            const char * valueref ) const
```

Will loop over all files and return a vector of std::strings of filenames where value match the reference value 'valueref'

10.295.4.9 GetFilenameFromTagToValue()

```
const char * gdcm::StrictScanner::GetFilenameFromTagToValue (
            Tag const & t,
            const char * valueref ) const
```

Will loop over all files and return the first file where value match the reference value 'valueref'

10.295.4.10 GetFilenames()

```
Directory::FilenamesType const & gdcm::StrictScanner::GetFilenames ( ) const  [inline]
```

10.295.4.11 GetKeys()

```
Directory::FilenamesType gdcm::StrictScanner::GetKeys ( ) const
```

Return the list of filename that are key in the internal map, which means those filename were properly parsed

10.295.4.12 GetMapping()

```
TagToValue const & gdcm::StrictScanner::GetMapping (
            const char * filename ) const
```

Get the std::map mapping filenames to value for file 'filename'.

Examples

SimpleScanner.cxx.

10.295.4.13 GetMappingFromTagToValue()

```
TagToValue const & gdcm::StrictScanner::GetMappingFromTagToValue (
            Tag const & t,
            const char * value ) const
```

See GetFilenameFromTagToValue(). This is simply GetFilenameFromTagToValue followed.

10.295.4.14 GetMappings()

```
MappingType const & gdcm::StrictScanner::GetMappings ( ) const  [inline]
```

Mappings are the mapping from a particular tag to the map, mapping filename to value:

10.295.4.15 GetOrderedValues()

```
Directory::FilenamesType gdcm::StrictScanner::GetOrderedValues (
            Tag const & t ) const
```

Get all the values found (in a vector) associated with Tag 't' This function is identical to GetValues, but is accessible from the wrapped layer (python, C#, java)

10.295.4.16 GetValue()

```
const char * gdcm::StrictScanner::GetValue (
            const char * filename,
            Tag const & t ) const
```

Retrieve the value found for tag: t associated with file: filename This is meant for a single short call. If multiple calls (multiple tags) should be done, prefer the GetMapping function, and then reuse the TagToValue hash table.

Warning

Tag 't' should have been added via AddTag() prior to the Scan() call !

10.295.4.17 GetValues() [1/2]

```
ValuesType const & gdcm::StrictScanner::GetValues ( ) const  [inline]
```

Get all the values found (in lexicographic order)

10.295.4.18 GetValues() [2/2]

```
ValuesType gdcm::StrictScanner::GetValues (
            Tag const & t ) const
```

Get all the values found (in lexicographic order) associated with Tag 't'.

10.295.4.19 IsKey()

```
bool gdcm::StrictScanner::IsKey (
            const char * filename ) const
```

Check if filename is a key in the Mapping table. returns true only of file can be found, which means the file was indeed a DICOM file that could be processed

Examples

ScanDirectory.cs, and SimpleScanner.cxx.

10.295.4.20 New()

```
static SmartPointer< StrictScanner > gdcm::StrictScanner::New ( )  [inline], [static]
```

for wrapped language: instantiate a reference counted object

Examples

ScanDirectory.cs.

10.295.4.21 Print()

```
void gdcm::StrictScanner::Print (
            std::ostream & os ) const  [override], [virtual]
```

Print result.

Reimplemented from gdcm::Object.

10.295.4.22 PrintTable()

```
void gdcm::StrictScanner::PrintTable (
            std::ostream & os ) const
```

10.295.4.23 ProcessPublicTag()

```
void gdcm::StrictScanner::ProcessPublicTag (
            StringFilter & sf,
            const char * filename )  [protected]
```

10.295.4.24 Scan()

```
bool gdcm::StrictScanner::Scan (
            Directory::FilenamesType const & filenames )
```

Start the scan !

Examples

ScanDirectory.cs, and SimpleScanner.cxx.

10.295.5 Friends And Related Function Documentation

10.295.5.1 operator<<

```
std::ostream & operator<< (
            std::ostream & _os,
            const StrictScanner & s )  [friend]
```

The documentation for this class was generated from the following file:

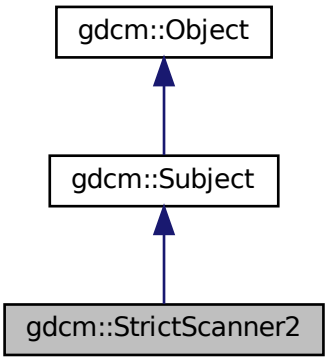
- gdcmStrictScanner.h

10.296 gdcm::StrictScanner2 Class Reference

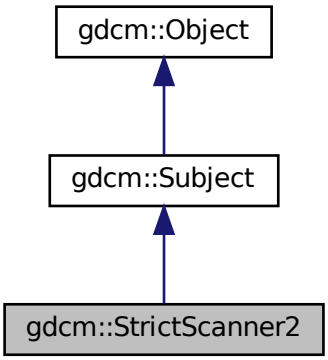
StrictScanner2.

`#include <gdcmStrictScanner2.h>`

Inheritance diagram for gdcm::StrictScanner2:



Collaboration diagram for gdcm::StrictScanner2:



Classes

- struct ltstr

Public Types

- typedef PrivateMappingType::const_iterator PrivateConstIterator
- typedef std::map< const char *, PrivateTagToValue, ltstr > PrivateMappingType
- typedef std::map< PrivateTag, const char * > PrivateTagToValue
- typedef PrivateTagToValue::value_type PrivateTagToValueValueType
- typedef PublicMappingType::const_iterator PublicConstIterator
- typedef std::map< const char *, PublicTagToValue, ltstr > PublicMappingType
- typedef std::map< Tag, const char * > PublicTagToValue
- typedef PublicTagToValue::value_type PublicTagToValueValueType
- typedef std::set< std::string > ValuesType

Public Member Functions

- StrictScanner2 ()
- ~StrictScanner2 () override
- bool AddPrivateTag (PrivateTag const &pt)
- bool AddPublicTag (Tag const &t)

  Add a tag that will need to be read. Those are root level tags.
- bool AddSkipTag (Tag const &t)

  Add a tag that will need to be skipped. Those are root level skip tags.
- PublicConstIterator Begin () const
- void ClearPrivateTags ()
- void ClearPublicTags ()
- void ClearSkipTags ()
- PublicConstIterator End () const
- Directory::FilenamesType GetAllFilenamesFromPrivateTagToValue (PrivateTag const &pt, const char *valueref) const
- Directory::FilenamesType GetAllFilenamesFromPublicTagToValue (Tag const &t, const char *valueref) const
- const char * GetFilenameFromPrivateTagToValue (PrivateTag const &pt, const char *valueref) const
- const char * GetFilenameFromPublicTagToValue (Tag const &t, const char *valueref) const
- Directory::FilenamesType const & GetFilenames () const

  Return the list of filenames.
- Directory::FilenamesType GetKeys () const
- PrivateTagToValue const & GetMappingFromPrivateTagToValue (PrivateTag const &pt, const char *value) const
- PublicTagToValue const & GetMappingFromPublicTagToValue (Tag const &t, const char *value) const
- PrivateTagToValue const & GetPrivateMapping (const char *filename) const
- PrivateMappingType const & GetPrivateMappings () const
- Directory::FilenamesType GetPrivateOrderedValues (PrivateTag const &pt) const
- const char * GetPrivateValue (const char *filename, PrivateTag const &t) const
- ValuesType GetPrivateValues (PrivateTag const &pt) const
- PublicTagToValue const & GetPublicMapping (const char *filename) const

  Get the std::map mapping filenames to value for file 'filename'.
- PublicMappingType const & GetPublicMappings () const
- Directory::FilenamesType GetPublicOrderedValues (Tag const &t) const
- const char * GetPublicValue (const char *filename, Tag const &t) const
- ValuesType GetPublicValues (Tag const &t) const

  Get all the values found (in lexicographic order) associated with Tag 't'.
- ValuesType const & GetValues () const

Get all the values found (in lexicographic order)

- bool IsKey (const char *filename) const
- void Print (std::ostream &os) const override

  Print result.
- void PrintTable (std::ostream &os, bool header=false) const

  Print result as CSV table.
- PrivateConstIterator PrivateBegin () const
- PrivateConstIterator PrivateEnd () const
- bool Scan (Directory::FilenamesType const &filenames)

  Start the scan !

Static Public Member Functions

- static SmartPointer< StrictScanner2 > New ()

  for wrapped language: instantiate a reference counted object

Protected Member Functions

- void ProcessPrivateTag (StringFilter &sf, const char *filename)
- void ProcessPublicTag (StringFilter &sf, const char *filename)

Friends

- std::ostream & operator<< (std::ostream &_os, const StrictScanner2 &s)

10.296.1 Detailed Description

StrictScanner2.

This filter is meant for quickly browsing a FileSet (a set of files on disk). Special consideration are taken so as to read the minimum amount of information in each file in order to retrieve the user specified set of DICOM Attribute.

This filter is dealing with both VRASCII and VRBINARY element, thanks to the help of StringFilter

Warning

IMPORTANT In case of file where tags are not ordered (illegal as per DICOM specification), the output will be missing information

Note

implementation details. All values are stored in a std::set of std::string. Then the address of the cstring underlying the std::string is used in the std::map.

This class implement the Subject/Observer pattern trigger the following events:

- ProgressEvent
- StartEvent
- EndEvent

10.296.2 Member Typedef Documentation

10.296.2.1 PrivateConstIterator

```
typedef PrivateMappingType::const_iterator gdcm::StrictScanner2::PrivateConstIterator
```

10.296.2.2 PrivateMappingType

```
typedef std::map<const char *, PrivateTagToValue, ltstr> gdcm::StrictScanner2::PrivateMappingType
```

10.296.2.3 PrivateTagToValue

```
typedef std::map<PrivateTag, const char *> gdcm::StrictScanner2::PrivateTagToValue
```

10.296.2.4 PrivateTagToValueValueType

```
typedef PrivateTagToValue::value_type gdcm::StrictScanner2::PrivateTagToValueValueType
```

10.296.2.5 PublicConstIterator

```
typedef PublicMappingType::const_iterator gdcm::StrictScanner2::PublicConstIterator
```

10.296.2.6 PublicMappingType

```
typedef std::map<const char *, PublicTagToValue, ltstr> gdcm::StrictScanner2::PublicMappingType
```

10.296.2.7 PublicTagToValue

```
typedef std::map<Tag, const char *> gdcm::StrictScanner2::PublicTagToValue
```

struct to map a filename to a value Implementation note: all std::map in this class will be using const char * and not std::string since we are pointing to existing std::string (held in a std::vector) this avoid an extra copy of the byte array. Tag are used as Tag class since sizeof(tag) <= sizeof(pointer)

10.296.2.8 PublicTagToValueValueType

```
typedef PublicTagToValue::value_type gdcm::StrictScanner2::PublicTagToValueValueType
```

10.296.2.9 ValuesType

```
typedef std::set<std::string> gdcm::StrictScanner2::ValuesType
```

10.296.3 Constructor & Destructor Documentation

10.296.3.1 StrictScanner2()

```
gdcm::StrictScanner2::StrictScanner2 ( )  [inline]
```

10.296.3.2 ~StrictScanner2()

```
gdcm::StrictScanner2::~StrictScanner2 ( )  [override]
```

10.296.4 Member Function Documentation

10.296.4.1 AddPrivateTag()

```
bool gdcm::StrictScanner2::AddPrivateTag (
            PrivateTag const & pt )
```

10.296.4.2 AddPublicTag()

```
bool gdcm::StrictScanner2::AddPublicTag (
            Tag const & t )
```

Add a tag that will need to be read. Those are root level tags.

10.296.4.3 AddSkipTag()

```
bool gdcm::StrictScanner2::AddSkipTag (
            Tag const & t )
```

Add a tag that will need to be skipped. Those are root level skip tags.

10.296.4.4 Begin()

```
PublicConstIterator gdcm::StrictScanner2::Begin ( ) const  [inline]
```

10.296.4.5 ClearPrivateTags()

```
void gdcm::StrictScanner2::ClearPrivateTags ( )
```

10.296.4.6 ClearPublicTags()

```
void gdcm::StrictScanner2::ClearPublicTags ( )
```

10.296.4.7 ClearSkipTags()

```
void gdcm::StrictScanner2::ClearSkipTags ( )
```

10.296.4.8 End()

```
PublicConstIterator gdcm::StrictScanner2::End ( ) const  [inline]
```

10.296.4.9 GetAllFilenamesFromPrivateTagToValue()

```
Directory::FilenamesType gdcm::StrictScanner2::GetAllFilenamesFromPrivateTagToValue (
            PrivateTag const & pt,
            const char * valueref ) const
```

10.296.4.10 GetAllFilenamesFromPublicTagToValue()

```
Directory::FilenamesType gdcm::StrictScanner2::GetAllFilenamesFromPublicTagToValue (
            Tag const & t,
            const char * valueref ) const
```

Will loop over all files and return a vector of std::strings of filenames where value match the reference value 'valueref'

10.296.4.11 GetFilenameFromPrivateTagToValue()

```
const char * gdcm::StrictScanner2::GetFilenameFromPrivateTagToValue (
            PrivateTag const & pt,
            const char * valueref ) const
```

10.296.4.12 GetFilenameFromPublicTagToValue()

```
const char * gdcm::StrictScanner2::GetFilenameFromPublicTagToValue (
            Tag const & t,
            const char * valueref ) const
```

Will loop over all files and return the first file where value match the reference value 'valueref'

10.296.4.13 GetFilenames()

```
Directory::FilenamesType const & gdcm::StrictScanner2::GetFilenames ( ) const  [inline]
```

Return the list of filenames.

10.296.4.14 GetKeys()

```
Directory::FilenamesType gdcm::StrictScanner2::GetKeys ( ) const
```

Return the list of filename that are key in the internal map, which means those filename were properly parsed

10.296.4.15 GetMappingFromPrivateTagToValue()

```
PrivateTagToValue const & gdcm::StrictScanner2::GetMappingFromPrivateTagToValue (
            PrivateTag const & pt,
            const char * value ) const
```

10.296.4.16 GetMappingFromPublicTagToValue()

```
PublicTagToValue const & gdcm::StrictScanner2::GetMappingFromPublicTagToValue (
            Tag const & t,
            const char * value ) const
```

See GetFilenameFromTagToValue(). This is simply GetFilenameFromTagToValue followed

10.296.4.17 GetPrivateMapping()

```
PrivateTagToValue const & gdcm::StrictScanner2::GetPrivateMapping (
            const char * filename ) const
```

10.296.4.18 GetPrivateMappings()

```
PrivateMappingType const & gdcm::StrictScanner2::GetPrivateMappings ( ) const  [inline]
```

10.296.4.19 GetPrivateOrderedValues()

```
Directory::FilenamesType gdcm::StrictScanner2::GetPrivateOrderedValues (
            PrivateTag const & pt ) const
```

10.296.4.20 GetPrivateValue()

```
const char * gdcm::StrictScanner2::GetPrivateValue (
            const char * filename,
            PrivateTag const & t ) const
```

10.296.4.21 GetPrivateValues()

```
ValuesType gdcm::StrictScanner2::GetPrivateValues (
            PrivateTag const & pt ) const
```

Get all the values found (in lexicographic order) associated with PrivateTag 'pt'

10.296.4.22 GetPublicMapping()

```
PublicTagToValue const & gdcm::StrictScanner2::GetPublicMapping (
            const char * filename ) const
```

Get the std::map mapping filenames to value for file 'filename'.

10.296.4.23 GetPublicMappings()

```
PublicMappingType const & gdcm::StrictScanner2::GetPublicMappings ( ) const  [inline]
```

Mappings are the mapping from a particular tag to the map, mapping filename to value:

10.296.4.24 GetPublicOrderedValues()

```
Directory::FilenamesType gdcm::StrictScanner2::GetPublicOrderedValues (
            Tag const & t ) const
```

Get all the values found (in a vector) associated with Tag 't' This function is identical to GetValues, but is accessible from the wrapped layer (python, C#, java)

10.296.4.25 GetPublicValue()

```
const char * gdcm::StrictScanner2::GetPublicValue (
            const char * filename,
            Tag const & t ) const
```

Retrieve the value found for tag: t associated with file: filename This is meant for a single short call. If multiple calls (multiple tags) should be done, prefer the GetMapping function, and then reuse the TagToValue hash table.

Warning

Tag 't' should have been added via AddTag() prior to the Scan() call !

10.296.4.26 GetPublicValues()

```
ValuesType gdcm::StrictScanner2::GetPublicValues (
            Tag const & t ) const
```

Get all the values found (in lexicographic order) associated with Tag 't'.

10.296.4.27 GetValues()

```
ValuesType const & gdcm::StrictScanner2::GetValues ( ) const  [inline]
```

Get all the values found (in lexicographic order)

10.296.4.28 IsKey()

```
bool gdcm::StrictScanner2::IsKey (
            const char * filename ) const
```

Check if filename is a key in the Mapping table. returns true only of file can be found, which means the file was indeed a DICOM file that could be processed

10.296.4.29 New()

```
static SmartPointer< StrictScanner2 > gdcm::StrictScanner2::New ( )  [inline], [static]
```

for wrapped language: instantiate a reference counted object

10.296.4.30 Print()

```
void gdcm::StrictScanner2::Print (
            std::ostream & os ) const  [override], [virtual]
```

Print result.

Reimplemented from gdcm::Object.

10.296.4.31 PrintTable()

```
void gdcm::StrictScanner2::PrintTable (
            std::ostream & os,
            bool header = false ) const
```

Print result as CSV table.

10.296.4.32 PrivateBegin()

```
PrivateConstIterator gdcm::StrictScanner2::PrivateBegin ( ) const  [inline]
```

10.296.4.33 PrivateEnd()

```
PrivateConstIterator gdcm::StrictScanner2::PrivateEnd ( ) const  [inline]
```

10.296.4.34 ProcessPrivateTag()

```
void gdcm::StrictScanner2::ProcessPrivateTag (
            StringFilter & sf,
            const char * filename )  [protected]
```

10.296.4.35 ProcessPublicTag()

```
void gdcm::StrictScanner2::ProcessPublicTag (
            StringFilter & sf,
            const char * filename )  [protected]
```

10.296.4.36 Scan()

```
bool gdcm::StrictScanner2::Scan (
            Directory::FilenamesType const & filenames )
```

Start the scan !

10.296.5 Friends And Related Function Documentation

10.296.5.1 operator<<

```
std::ostream & operator<< (
            std::ostream & _os,
            const StrictScanner2 & s )  [friend]
```

The documentation for this class was generated from the following file:

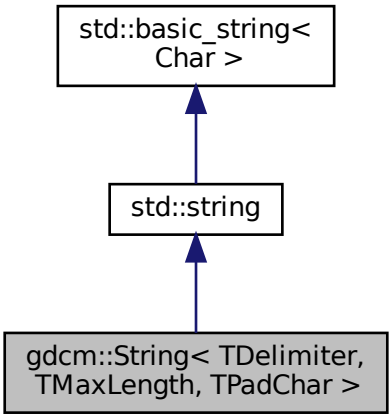
- gdcmStrictScanner2.h

10.297 gdcm::String< TDelimiter, TMaxLength, TPadChar > Class Template Reference

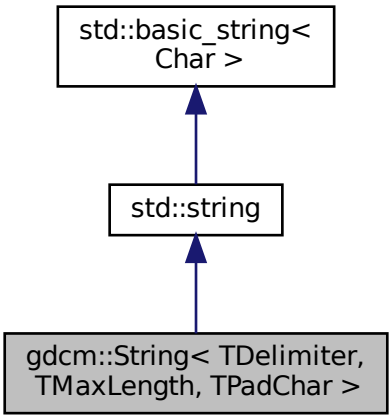
String.

```
#include <gdcmString.h>
```

Inheritance diagram for gdcm::String< TDelimiter, TMaxLength, TPadChar >:



Collaboration diagram for gdcm::String< TDelimiter, TMaxLength, TPadChar >:



Public Types

- typedef std::string::const_iterator const_iterator
- typedef std::string::const_reference const_reference
- typedef std::string::const_reverse_iterator const_reverse_iterator
- typedef std::string::difference_type difference_type
- typedef std::string::iterator iterator
- typedef std::string::pointer pointer
- typedef std::string::reference reference
- typedef std::string::reverse_iterator reverse_iterator
- typedef std::string::size_type size_type
- typedef std::string::value_type value_type

Public Member Functions

- String ()

  String constructors.
- String (const std::string &s, size_type pos=0, size_type n=npos)
- String (const value_type *s)
- String (const value_type *s, size_type n)
- bool IsValid () const

  return if string is valid
- operator const char * () const

  WARNING: Trailing \0 might be lost in this operation:
- std::string Trim () const
- gdcm::String< TDelimiter, TMaxLength, TPadChar > Truncate () const

Static Public Member Functions

- static std::string Trim (const char *input)

10.297.1 Detailed Description

template<char TDelimiter = '\\', unsigned int TMaxLength = 64, char TPadChar = ' '>
class gdcm::String< TDelimiter, TMaxLength, TPadChar >

String.

Note

TDelimiter template parameter is used to separate multiple String (VM1 >) TMaxLength is only a hint. No one actually respect the max length TPadChar is the string padding (0 or space)

Examples

TemplateEmptyImage.cxx.

10.297.2 Member Typedef Documentation

10.297.2.1 const_iterator

```
template<char TDelimiter = '\\', unsigned int TMaxLength = 64, char TPadChar = ' '>
typedef std::string::const_iterator gdcm::String< TDelimiter, TMaxLength, TPadChar >::const_↩
iterator
```

10.297.2.2 const_reference

```
template<char TDelimiter = '\\', unsigned int TMaxLength = 64, char TPadChar = ' '>
typedef std::string::const_reference gdcm::String< TDelimiter, TMaxLength, TPadChar >::const_↩
reference
```

10.297.2.3 const_reverse_iterator

```
template<char TDelimiter = '\\', unsigned int TMaxLength = 64, char TPadChar = ' '>
typedef std::string::const_reverse_iterator gdcm::String< TDelimiter, TMaxLength, TPadChar >↩
::const_reverse_iterator
```

10.297.2.4 difference_type

```
template<char TDelimiter = '\\', unsigned int TMaxLength = 64, char TPadChar = ' '>
typedef std::string::difference_type gdcm::String< TDelimiter, TMaxLength, TPadChar >::difference↩
_type
```

10.297.2.5 iterator

```
template<char TDelimiter = '\\', unsigned int TMaxLength = 64, char TPadChar = ' '>
typedef std::string::iterator gdcm::String< TDelimiter, TMaxLength, TPadChar >::iterator
```

10.297.2.6 pointer

```
template<char TDelimiter = '\\', unsigned int TMaxLength = 64, char TPadChar = ' '>
typedef std::string::pointer gdcm::String< TDelimiter, TMaxLength, TPadChar >::pointer
```

10.297.2.7 reference

```
template<char TDelimiter = '\\', unsigned int TMaxLength = 64, char TPadChar = ' '>
typedef std::string::reference gdcm::String< TDelimiter, TMaxLength, TPadChar >::reference
```

10.297.2.8 reverse_iterator

```
template<char TDelimiter = '\\', unsigned int TMaxLength = 64, char TPadChar = ' '>
typedef std::string::reverse_iterator gdcm::String< TDelimiter, TMaxLength, TPadChar >::reverse↩
_iterator
```

10.297.2.9 size_type

```
template<char TDelimiter = '\\', unsigned int TMaxLength = 64, char TPadChar = ' '>
typedef std::string::size_type gdcm::String< TDelimiter, TMaxLength, TPadChar >::size_type
```

10.297.2.10 value_type

```
template<char TDelimiter = '\\', unsigned int TMaxLength = 64, char TPadChar = ' '>
typedef std::string::value_type gdcm::String< TDelimiter, TMaxLength, TPadChar >::value_type
```

10.297.3 Constructor & Destructor Documentation

10.297.3.1 String() [1/4]

```
template<char TDelimiter = '\\', unsigned int TMaxLength = 64, char TPadChar = ' '>
gdcm::String< TDelimiter, TMaxLength, TPadChar >::String ( )  [inline]
```

String constructors.

10.297.3.2 String() [2/4]

```
template<char TDelimiter = '\\', unsigned int TMaxLength = 64, char TPadChar = ' '>
gdcm::String< TDelimiter, TMaxLength, TPadChar >::String (
            const value_type * s )  [inline]
```

10.297.3.3 String() [3/4]

```
template<char TDelimiter = '\\', unsigned int TMaxLength = 64, char TPadChar = ' '>
gdcm::String< TDelimiter, TMaxLength, TPadChar >::String (
            const value_type * s,
            size_type n )  [inline]
```

10.297.3.4 String() [4/4]

```
template<char TDelimiter = '\\', unsigned int TMaxLength = 64, char TPadChar = ' '>
gdcm::String< TDelimiter, TMaxLength, TPadChar >::String (
            const std::string & s,
            size_type pos = 0,
            size_type n = npos )  [inline]
```

10.297.4 Member Function Documentation

10.297.4.1 IsValid()

```
template<char TDelimiter = '\\', unsigned int TMaxLength = 64, char TPadChar = ' '>
bool gdcm::String< TDelimiter, TMaxLength, TPadChar >::IsValid ( ) const  [inline]
```

return if string is valid

Referenced by gdcm::LO::IsValid(), and gdcm::String< TDelimiter, TMaxLength, TPadChar >::Truncate().

10.297.4.2 operator const char *()

```
template<char TDelimiter = '\\', unsigned int TMaxLength = 64, char TPadChar = ' '>
gdcm::String< TDelimiter, TMaxLength, TPadChar >::operator const char * ( ) const  [inline]
```

WARNING: Trailing \0 might be lost in this operation:

10.297.4.3 Trim() [1/2]

```
template<char TDelimiter = '\\', unsigned int TMaxLength = 64, char TPadChar = ' '>
std::string gdcm::String< TDelimiter, TMaxLength, TPadChar >::Trim ( ) const  [inline]
```

Trim function is required to return a std::string object, otherwise we could not create a gdcm::String object with an odd number of bytes...

Examples

DumpExamCard.cxx.

10.297.4.4 Trim() [2/2]

```
template<char TDelimiter = '\\', unsigned int TMaxLength = 64, char TPadChar = ' '>
static std::string gdcm::String< TDelimiter, TMaxLength, TPadChar >::Trim (
            const char * input )  [inline], [static]
```

10.297.4.5 Truncate()

```
template<char TDelimiter = '\\', unsigned int TMaxLength = 64, char TPadChar = ' '>
gdcm::String< TDelimiter, TMaxLength, TPadChar > gdcm::String< TDelimiter, TMaxLength, TPadChar
>::Truncate ( ) const  [inline]
```

References gdcm::String< TDelimiter, TMaxLength, TPadChar >::IsValid().

The documentation for this class was generated from the following file:

- gdcmString.h

10.298 gdcm::StringFilter Class Reference

StringFilter.

```
#include <gdcmStringFilter.h>
```

Public Member Functions

- StringFilter ()
- ~StringFilter ()
- bool ExecuteQuery (std::string const &query, std::string &value) const
- std::string FromString (const Tag &t, const char *value, size_t len)

  Convert to string the char array defined by the pair (value,len)
- File & GetFile ()
- const File & GetFile () const
- void SetDicts (const Dicts &dicts)

  Allow user to pass in there own dicts.
- void SetFile (const File &f)

  Set/Get File.
- std::string ToString (const DataElement &de) const
- std::string ToString (const PrivateTag &t) const
- std::string ToString (const Tag &t) const

  Directly from a Tag:
- std::pair< std::string, std::string > ToStringPair (const DataElement &de) const
- std::pair< std::string, std::string > ToStringPair (const Tag &t) const

  Directly from a Tag:
- void UseDictAlways (bool)

Protected Member Functions

- bool ExecuteQuery (std::string const &query, DataSet const &ds, std::string &value) const
- std::pair< std::string, std::string > ToStringPair (const Tag &t, DataSet const &ds) const

10.298.1 Detailed Description

StringFilter.

StringFilter is the class that make gdcm2.x looks more like gdcm1 and transform the binary blob contained in a DataElement into a string, typically this is a nice feature to have for wrapped language

Examples

DumpVisusChange.cxx, ReadAndPrintAttributes.cxx, and SimplePrintPatientName.cs.

10.298.2 Constructor & Destructor Documentation

10.298.2.1 StringFilter()

```
gdcm::StringFilter::StringFilter ( )
```

10.298.2.2 ~StringFilter()

```
gdcm::StringFilter::~StringFilter ( )
```

10.298.3 Member Function Documentation

10.298.3.1 ExecuteQuery() [1/2]

```
bool gdcm::StringFilter::ExecuteQuery (
            std::string const & query,
            DataSet const & ds,
            std::string & value ) const  [protected]
```

10.298.3.2 ExecuteQuery() [2/2]

```
bool gdcm::StringFilter::ExecuteQuery (
            std::string const & query,
            std::string & value ) const
```

Execute the XPATH query to find a value (as string) return false when attribute is not found (or an error in the XPATH query) You need to make sure that your XPATH query is syntatically correct

10.298.3.3 FromString()

```
std::string gdcm::StringFilter::FromString (
            const Tag & t,
            const char * value,
            size_t len )
```

Convert to string the char array defined by the pair (value,len)

10.298.3.4 GetFile() [1/2]

```
File & gdcm::StringFilter::GetFile ( )  [inline]
```

10.298.3.5 GetFile() [2/2]

```
const File & gdcm::StringFilter::GetFile ( ) const  [inline]
```

10.298.3.6 SetDicts()

```
void gdcm::StringFilter::SetDicts (
            const Dicts & dicts )
```

Allow user to pass in there own dicts.

10.298.3.7 SetFile()

```
void gdcm::StringFilter::SetFile (
            const File & f )  [inline]
```

Set/Get File.

Examples

DumpVisusChange.cxx, ReadAndPrintAttributes.cxx, and SimplePrintPatientName.cs.

10.298.3.8 ToString() [1/3]

```
std::string gdcm::StringFilter::ToString (
            const DataElement & de ) const
```

Convert to string the ByteValue contained in a DataElement. The DataElement must be coming from the actual DataSet associated with File (see SetFile).

Examples

DumpVisusChange.cxx, ReadAndPrintAttributes.cxx, and SimplePrintPatientName.cs.

10.298.3.9 ToString() [2/3]

```
std::string gdcm::StringFilter::ToString (
            const PrivateTag & t ) const
```

10.298.3.10 ToString() [3/3]

```
std::string gdcm::StringFilter::ToString (
            const Tag & t ) const
```

Directly from a Tag:

10.298.3.11 ToStringPair() [1/3]

```
std::pair< std::string, std::string > gdcm::StringFilter::ToStringPair (
            const DataElement & de ) const
```

Convert to string the ByteValue contained in a DataElement the returned elements are: pair.first : the name as found in the dictionary of DataElement pari.second : the value encoded into a string (US,UL...) are properly converted

Examples

ReadAndPrintAttributes.cxx.

10.298.3.12 ToStringPair() [2/3]

```
std::pair< std::string, std::string > gdcm::StringFilter::ToStringPair (
            const Tag & t ) const
```

Directly from a Tag:

10.298.3.13 ToStringPair() [3/3]

```
std::pair< std::string, std::string > gdcm::StringFilter::ToStringPair (
            const Tag & t,
            DataSet const & ds ) const  [protected]
```

10.298.3.14 UseDictAlways()

```
void gdcm::StringFilter::UseDictAlways (
            bool  )  [inline]
```

The documentation for this class was generated from the following file:

- gdcmStringFilter.h

10.299 gdcm::Study Class Reference

Study.

`#include <gdcmStudy.h>`

Public Member Functions

- Study ()=default

10.299.1 Detailed Description

Study.

10.299.2 Constructor & Destructor Documentation

10.299.2.1 Study()

```
gdcm::Study::Study ( )  [default]
```

The documentation for this class was generated from the following file:

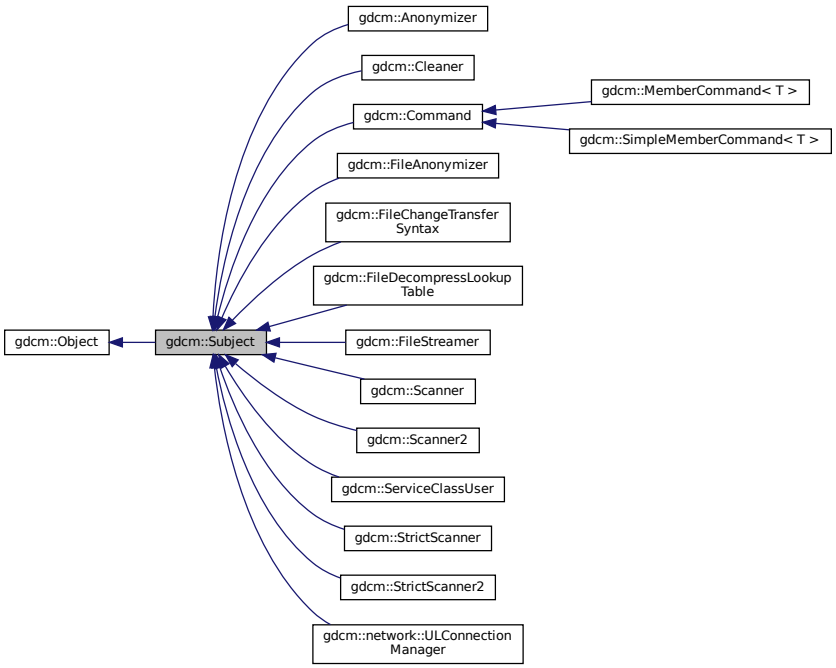
- gdcmStudy.h

10.300 gdcm::Subject Class Reference

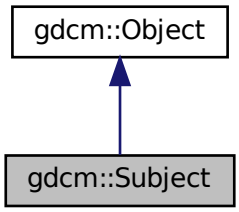
Subject.

```
#include <gdcmSubject.h>
```

Inheritance diagram for gdcm::Subject:



Collaboration diagram for gdcm::Subject:



Public Member Functions

- Subject ()
- ~Subject () override
- unsigned long AddObserver (const Event &event, Command *)
- unsigned long AddObserver (const Event &event, Command *) const
- Command * GetCommand (unsigned long tag)
- bool HasObserver (const Event &event) const
- void InvokeEvent (const Event &)
- void InvokeEvent (const Event &) const
- void RemoveAllObservers ()
- void RemoveObserver (unsigned long tag)

Additional Inherited Members

10.300.1 Detailed Description

Subject.

See also

Command Event

Examples

BasicAnonymizer.cs, ClinicalTrialIdentificationWorkflow.cs, ScanDirectory.cs, and SimpleScanner.cxx.

10.300.2 Constructor & Destructor Documentation

10.300.2.1 Subject()

```
gdcm::Subject::Subject ( )
```

10.300.2.2 ~Subject()

```
gdcm::Subject::~Subject ( )  [override]
```

10.300.3 Member Function Documentation

10.300.3.1 AddObserver() [1/2]

```
unsigned long gdcm::Subject::AddObserver (
            const Event & event,
            Command *  )
```

Allow people to add/remove/invoke observers (callbacks) to any GDCM object. This is an implementation of the subject/observer design pattern. An observer is added by specifying an event to respond to and an gdcm::Command to execute. It returns an unsigned long tag which can be used later to remove the event or retrieve the command. The memory for the Command becomes the responsibility of this object, so don't pass the same instance of a command to two different objects

10.300.3.2 AddObserver() [2/2]

```
unsigned long gdcm::Subject::AddObserver (
            const Event & event,
            Command *  ) const
```

10.300.3.3 GetCommand()

```
Command * gdcm::Subject::GetCommand (
            unsigned long tag )
```

Get the command associated with the given tag. NOTE: This returns a pointer to a Command, but it is safe to assign this to a Command::Pointer. Since Command inherits from LightObject, at this point in the code, only a pointer or a reference to the Command can be used.

10.300.3.4 HasObserver()

```
bool gdcm::Subject::HasObserver (
            const Event & event ) const
```

Return true if an observer is registered for this event.

10.300.3.5 InvokeEvent() [1/2]

```
void gdcm::Subject::InvokeEvent (
            const Event &  )
```

Call Execute on all the Commands observing this event id.

10.300.3.6 InvokeEvent() [2/2]

```
void gdcm::Subject::InvokeEvent (
            const Event &  ) const
```

Call Execute on all the Commands observing this event id. The actions triggered by this call doesn't modify this object.

10.300.3.7 RemoveAllObservers()

```
void gdcm::Subject::RemoveAllObservers ( )
```

Remove all observers .

10.300.3.8 RemoveObserver()

```
void gdcm::Subject::RemoveObserver (
            unsigned long tag )
```

Remove the observer with this tag value.

The documentation for this class was generated from the following file:

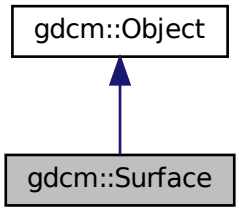
- gdcmSubject.h

10.301 gdcm::Surface Class Reference

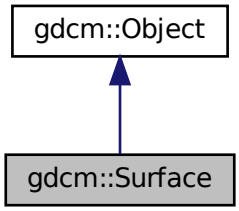
This class defines a SURFACE IE.

`#include <gdcmSurface.h>`

Inheritance diagram for gdcm::Surface:



Collaboration diagram for gdcm::Surface:



Public Types

- enum STATES {
  NO = 0 ,
  YES ,
  UNKNOWN ,
  STATES_END }
- enum VIEWType {
  SURFACE = 0 ,
  WIREFRAME ,
  POINTS ,
  VIEWType_END }

  Enumeration for Recommended Presentation Type.

Public Member Functions

- Surface ()
- ~Surface () override
- SegmentHelper::BasicCodedEntry & GetAlgorithmFamily ()
- SegmentHelper::BasicCodedEntry const & GetAlgorithmFamily () const
- const char * GetAlgorithmName () const
- const char * GetAlgorithmVersion () const
- const float * GetAxisOfRotation () const
- const float * GetCenterOfRotation () const
- STATES GetFiniteVolume () const
- STATES GetManifold () const
- float GetMaximumPointDistance () const
- float GetMeanPointDistance () const
- MeshPrimitive & GetMeshPrimitive ()
- MeshPrimitive const & GetMeshPrimitive () const
- unsigned long GetNumberOfSurfacePoints () const
- unsigned long GetNumberOfVectors () const
- DataElement & GetPointCoordinatesData ()
- const DataElement & GetPointCoordinatesData () const
- const float * GetPointPositionAccuracy () const
- const float * GetPointsBoundingBoxCoordinates () const
- SegmentHelper::BasicCodedEntry & GetProcessingAlgorithm ()
- SegmentHelper::BasicCodedEntry const & GetProcessingAlgorithm () const
- const unsigned short * GetRecommendedDisplayCIELabValue () const
- unsigned short GetRecommendedDisplayCIELabValue (const unsigned int idx) const
- unsigned short GetRecommendedDisplayGrayscaleValue () const
- float GetRecommendedPresentationOpacity () const
- VIEWType GetRecommendedPresentationType () const
- const char * GetSurfaceComments () const
- unsigned long GetSurfaceNumber () const
- bool GetSurfaceProcessing () const
- const char * GetSurfaceProcessingDescription () const
- float GetSurfaceProcessingRatio () const
- const float * GetVectorAccuracy () const
- DataElement & GetVectorCoordinateData ()
- const DataElement & GetVectorCoordinateData () const
- unsigned short GetVectorDimensionality () const
- void SetAlgorithmFamily (SegmentHelper::BasicCodedEntry const &BSE)
- void SetAlgorithmName (const char *str)
- void SetAlgorithmVersion (const char *str)
- void SetAxisOfRotation (const float *axis)
- void SetCenterOfRotation (const float *center)
- void SetFiniteVolume (STATES state)
- void SetManifold (STATES state)
- void SetMaximumPointDistance (float maximum)
- void SetMeanPointDistance (float average)
- void SetMeshPrimitive (MeshPrimitive &mp)
- void SetNumberOfSurfacePoints (const unsigned long nb)
- void SetNumberOfVectors (const unsigned long nb)
- void SetPointCoordinatesData (DataElement const &de)

- void SetPointPositionAccuracy (const float *accuracies)
- void SetPointsBoundingBoxCoordinates (const float *coordinates)
- void SetProcessingAlgorithm (SegmentHelper::BasicCodedEntry const &BSE)
- void SetRecommendedDisplayCIELabValue (const std::vector< unsigned short > &vl)
- void SetRecommendedDisplayCIELabValue (const unsigned short vl, const unsigned int idx=0)
- void SetRecommendedDisplayCIELabValue (const unsigned short vl[3])
- void SetRecommendedDisplayGrayscaleValue (const unsigned short vl)
- void SetRecommendedPresentationOpacity (const float opacity)
- void SetRecommendedPresentationType (VIEWType type)
- void SetSurfaceComments (const char *comment)
- void SetSurfaceNumber (const unsigned long nb)
- void SetSurfaceProcessing (bool b)
- void SetSurfaceProcessingDescription (const char *description)
- void SetSurfaceProcessingRatio (const float ratio)
- void SetVectorAccuracy (const float *accuracy)
- void SetVectorCoordinateData (DataElement const &de)
- void SetVectorDimensionality (const unsigned short dim)

Static Public Member Functions

- static STATES GetSTATES (const char *state)
- static const char * GetSTATESString (STATES state)
- static VIEWType GetVIEWType (const char *type)
- static const char * GetVIEWTypeString (VIEWType type)

Additional Inherited Members

10.301.1 Detailed Description

This class defines a SURFACE IE.

This members are taken from required surface mesh module attributes.

See also

PS 3.3 A.1.2.18 , A.57 and C.27

10.301.2 Member Enumeration Documentation

10.301.2.1 STATES

```
enum gdcm::Surface::STATES
```

Enumerator

NO	
YES	
UNKNOWN	
STATES_END	

10.301.2.2 VIEWType

```
enum gdcm::Surface::VIEWType
```

Enumeration for Recommended Presentation Type.

See also

Tag(0x0066, 0x000D) and PS 3.3 C.27.1.1.3

Enumerator

SURFACE	
WIREFRAME	
POINTS	
VIEWType_END	

10.301.3 Constructor & Destructor Documentation

10.301.3.1 Surface()

```
gdcm::Surface::Surface ( )
```

10.301.3.2 ~Surface()

```
gdcm::Surface::~Surface ( )  [override]
```

10.301.4 Member Function Documentation

10.301.4.1 GetAlgorithmFamily() [1/2]

```
SegmentHelper::BasicCodedEntry & gdcm::Surface::GetAlgorithmFamily ( )
```

10.301.4.2 GetAlgorithmFamily() [2/2]

```
SegmentHelper::BasicCodedEntry const & gdcm::Surface::GetAlgorithmFamily ( ) const
```

10.301.4.3 GetAlgorithmName()

```
const char * gdcm::Surface::GetAlgorithmName ( ) const
```

10.301.4.4 GetAlgorithmVersion()

```
const char * gdcm::Surface::GetAlgorithmVersion ( ) const
```

10.301.4.5 GetAxisOfRotation()

```
const float * gdcm::Surface::GetAxisOfRotation ( ) const
```

Note

Pointer is null if undefined

10.301.4.6 GetCenterOfRotation()

```
const float * gdcm::Surface::GetCenterOfRotation ( ) const
```

Note

Pointer is null if undefined

10.301.4.7 GetFiniteVolume()

```
STATES gdcm::Surface::GetFiniteVolume ( ) const
```

10.301.4.8 GetManifold()

```
STATES gdcm::Surface::GetManifold ( ) const
```

10.301.4.9 GetMaximumPointDistance()

```
float gdcm::Surface::GetMaximumPointDistance ( ) const
```

10.301.4.10 GetMeanPointDistance()

```
float gdcm::Surface::GetMeanPointDistance ( ) const
```

10.301.4.11 GetMeshPrimitive() [1/2]

```
MeshPrimitive & gdcm::Surface::GetMeshPrimitive ( )
```

10.301.4.12 GetMeshPrimitive() [2/2]

```
MeshPrimitive const & gdcm::Surface::GetMeshPrimitive ( ) const
```

10.301.4.13 GetNumberOfSurfacePoints()

```
unsigned long gdcm::Surface::GetNumberOfSurfacePoints ( ) const
```

10.301.4.14 GetNumberOfVectors()

```
unsigned long gdcm::Surface::GetNumberOfVectors ( ) const
```

10.301.4.15 GetPointCoordinatesData() [1/2]

```
DataElement & gdcm::Surface::GetPointCoordinatesData ( )
```

10.301.4.16 GetPointCoordinatesData() [2/2]

```
const DataElement & gdcm::Surface::GetPointCoordinatesData ( ) const
```

10.301.4.17 GetPointPositionAccuracy()

```
const float * gdcm::Surface::GetPointPositionAccuracy ( ) const
```

Note

Pointer is null if undefined

10.301.4.18 GetPointsBoundingBoxCoordinates()

```
const float * gdcm::Surface::GetPointsBoundingBoxCoordinates ( ) const
```

Note

Pointer is null if undefined

10.301.4.19 GetProcessingAlgorithm() [1/2]

```
SegmentHelper::BasicCodedEntry & gdcm::Surface::GetProcessingAlgorithm ( )
```

10.301.4.20 GetProcessingAlgorithm() [2/2]

```
SegmentHelper::BasicCodedEntry const & gdcm::Surface::GetProcessingAlgorithm ( ) const
```

10.301.4.21 GetRecommendedDisplayCIELabValue() [1/2]

```
const unsigned short * gdcm::Surface::GetRecommendedDisplayCIELabValue ( ) const
```

10.301.4.22 GetRecommendedDisplayCIELabValue() [2/2]

```
unsigned short gdcm::Surface::GetRecommendedDisplayCIELabValue (
            const unsigned int idx ) const
```

10.301.4.23 GetRecommendedDisplayGrayscaleValue()

```
unsigned short gdcm::Surface::GetRecommendedDisplayGrayscaleValue ( ) const
```

10.301.4.24 GetRecommendedPresentationOpacity()

```
float gdcm::Surface::GetRecommendedPresentationOpacity ( ) const
```

10.301.4.25 GetRecommendedPresentationType()

```
VIEWType gdcm::Surface::GetRecommendedPresentationType ( ) const
```

10.301.4.26 GetSTATES()

```
static STATES gdcm::Surface::GetSTATES (
            const char * state )  [static]
```

10.301.4.27 GetSTATESString()

```
static const char * gdcm::Surface::GetSTATESString (
            STATES state )  [static]
```

10.301.4.28 GetSurfaceComments()

```
const char * gdcm::Surface::GetSurfaceComments ( ) const
```

10.301.4.29 GetSurfaceNumber()

```
unsigned long gdcm::Surface::GetSurfaceNumber ( ) const
```

10.301.4.30 GetSurfaceProcessing()

```
bool gdcm::Surface::GetSurfaceProcessing ( ) const
```

10.301.4.31 GetSurfaceProcessingDescription()

```
const char * gdcm::Surface::GetSurfaceProcessingDescription ( ) const
```

10.301.4.32 GetSurfaceProcessingRatio()

```
float gdcm::Surface::GetSurfaceProcessingRatio ( ) const
```

10.301.4.33 GetVectorAccuracy()

```
const float * gdcm::Surface::GetVectorAccuracy ( ) const
```

10.301.4.34 GetVectorCoordinateData() [1/2]

```
DataElement & gdcm::Surface::GetVectorCoordinateData ( )
```

10.301.4.35 GetVectorCoordinateData() [2/2]

```
const DataElement & gdcm::Surface::GetVectorCoordinateData ( ) const
```

10.301.4.36 GetVectorDimensionality()

```
unsigned short gdcm::Surface::GetVectorDimensionality ( ) const
```

10.301.4.37 GetVIEWType()

```
static VIEWType gdcm::Surface::GetVIEWType (
            const char * type )  [static]
```

10.301.4.38 GetVIEWTypeString()

```
static const char * gdcm::Surface::GetVIEWTypeString (
            VIEWType type )  [static]
```

10.301.4.39 SetAlgorithmFamily()

```
void gdcm::Surface::SetAlgorithmFamily (
            SegmentHelper::BasicCodedEntry const & BSE )
```

10.301.4.40 SetAlgorithmName()

```
void gdcm::Surface::SetAlgorithmName (
            const char * str )
```

10.301.4.41 SetAlgorithmVersion()

```
void gdcm::Surface::SetAlgorithmVersion (
            const char * str )
```

10.301.4.42 SetAxisOfRotation()

```
void gdcm::Surface::SetAxisOfRotation (
            const float * axis )
```

10.301.4.43 SetCenterOfRotation()

```
void gdcm::Surface::SetCenterOfRotation (
            const float * center )
```

10.301.4.44 SetFiniteVolume()

```
void gdcm::Surface::SetFiniteVolume (
            STATES state )
```

10.301.4.45 SetManifold()

```
void gdcm::Surface::SetManifold (
            STATES state )
```

10.301.4.46 SetMaximumPointDistance()

```
void gdcm::Surface::SetMaximumPointDistance (
            float maximum )
```

10.301.4.47 SetMeanPointDistance()

```
void gdcm::Surface::SetMeanPointDistance (
            float average )
```

10.301.4.48 SetMeshPrimitive()

```
void gdcm::Surface::SetMeshPrimitive (
            MeshPrimitive & mp )
```

10.301.4.49 SetNumberOfSurfacePoints()

```
void gdcm::Surface::SetNumberOfSurfacePoints (
            const unsigned long nb )
```

10.301.4.50 SetNumberOfVectors()

```
void gdcm::Surface::SetNumberOfVectors (
            const unsigned long nb )
```

10.301.4.51 SetPointCoordinatesData()

```
void gdcm::Surface::SetPointCoordinatesData (
            DataElement const & de )
```

10.301.4.52 SetPointPositionAccuracy()

```
void gdcm::Surface::SetPointPositionAccuracy (
            const float * accuracies )
```

10.301.4.53 SetPointsBoundingBoxCoordinates()

```
void gdcm::Surface::SetPointsBoundingBoxCoordinates (
            const float * coordinates )
```

10.301.4.54 SetProcessingAlgorithm()

```
void gdcm::Surface::SetProcessingAlgorithm (
            SegmentHelper::BasicCodedEntry const & BSE )
```

10.301.4.55 SetRecommendedDisplayCIELabValue() [1/3]

```
void gdcm::Surface::SetRecommendedDisplayCIELabValue (
            const std::vector< unsigned short > & vl )
```

10.301.4.56 SetRecommendedDisplayCIELabValue() [2/3]

```
void gdcm::Surface::SetRecommendedDisplayCIELabValue (
            const unsigned short vl,
            const unsigned int idx = 0 )
```

10.301.4.57 SetRecommendedDisplayCIELabValue() [3/3]

```
void gdcm::Surface::SetRecommendedDisplayCIELabValue (
            const unsigned short vl[3] )
```

10.301.4.58 SetRecommendedDisplayGrayscaleValue()

```
void gdcm::Surface::SetRecommendedDisplayGrayscaleValue (
            const unsigned short vl )
```

10.301.4.59 SetRecommendedPresentationOpacity()

```
void gdcm::Surface::SetRecommendedPresentationOpacity (
            const float opacity )
```

10.301.4.60 SetRecommendedPresentationType()

```
void gdcm::Surface::SetRecommendedPresentationType (
            VIEWType type )
```

10.301.4.61 SetSurfaceComments()

```
void gdcm::Surface::SetSurfaceComments (
            const char * comment )
```

10.301.4.62 SetSurfaceNumber()

```
void gdcm::Surface::SetSurfaceNumber (
            const unsigned long nb )
```

10.301.4.63 SetSurfaceProcessing()

```
void gdcm::Surface::SetSurfaceProcessing (
            bool b )
```

10.301.4.64 SetSurfaceProcessingDescription()

```
void gdcm::Surface::SetSurfaceProcessingDescription (
            const char * description )
```

10.301.4.65 SetSurfaceProcessingRatio()

```
void gdcm::Surface::SetSurfaceProcessingRatio (
            const float ratio )
```

10.301.4.66 SetVectorAccuracy()

```
void gdcm::Surface::SetVectorAccuracy (
            const float * accuracy )
```

10.301.4.67 SetVectorCoordinateData()

```
void gdcm::Surface::SetVectorCoordinateData (
            DataElement const & de )
```

10.301.4.68 SetVectorDimensionality()

```
void gdcm::Surface::SetVectorDimensionality (
            const unsigned short dim )
```

The documentation for this class was generated from the following file:

- gdcmSurface.h

10.302 gdcm::SurfaceHelper Class Reference

SurfaceHelper.

`#include <gdcmSurfaceHelper.h>`

Public Types

- typedef std::vector< unsigned short > ColorArray

Static Public Member Functions

- template<typename T , typename U >
  static std::vector< T > RecommendedDisplayCIELabToRGB (const ColorArray &CIELab, const U rangeMax=255)

  Convert a DICOM CIE-Lab (after reading) color into RGB.
- template<typename U >
  static std::vector< float > RecommendedDisplayCIELabToRGB (const ColorArray &CIELab, const U rangeMax=255)

  Convert a DICOM CIE-Lab (after reading) color into RGB.
- template<typename T , typename U >
  static ColorArray RGBToRecommendedDisplayCIELab (const std::vector< T > &RGB, const U rangeMax=255)

  Convert a RGB color into DICOM CIE-Lab (ready to write).
- template<typename T , typename U >
  static unsigned short RGBToRecommendedDisplayGrayscale (const std::vector< T > &RGB, const U rangeMax=255)

  Convert a RGB color into DICOM grayscale (ready to write).

10.302.1 Detailed Description

SurfaceHelper.

Helper class for Surface object

10.302.2 Member Typedef Documentation

10.302.2.1 ColorArray

```
typedef std::vector< unsigned short > gdcm::SurfaceHelper::ColorArray
```

10.302.3 Member Function Documentation

10.302.3.1 RecommendedDisplayCIELabToRGB() [1/2]

```
template<typename T , typename U >
std::vector< T > gdcm::SurfaceHelper::RecommendedDisplayCIELabToRGB (
            const ColorArray & CIELab,
            const U rangeMax = 255 )  [static]
```

Convert a DICOM CIE-Lab (after reading) color into RGB.

See also

PS 3.3 C.10.7.1.1

Parameters

CIELab	DICOM CIE-Lab array.
rangeMax	Max value of the RGB range.

Template Parameters

T	Type of CIELab components.
U	Type of rangeMax value.

10.302.3.2 RecommendedDisplayCIELabToRGB() [2/2]

```
template<typename U >
std::vector< float > gdcm::SurfaceHelper::RecommendedDisplayCIELabToRGB (
```

```
            const ColorArray & CIELab,
            const U rangeMax = 255 )  [static]
```

Convert a DICOM CIE-Lab (after reading) color into RGB.

See also

PS 3.3 C.10.7.1.1

Parameters

CIELab	DICOM CIE-Lab array.
rangeMax	Max value of the RGB range.

Template Parameters

U	Type of rangeMax value.

10.302.3.3 RGBToRecommendedDisplayCIELab()

```
template<typename T , typename U >
SurfaceHelper::ColorArray gdcm::SurfaceHelper::RGBToRecommendedDisplayCIELab (
            const std::vector< T > & RGB,
            const U rangeMax = 255 )  [static]
```

Convert a RGB color into DICOM CIE-Lab (ready to write).

See also

PS 3.3 C.10.7.1.1

Parameters

RGB	RGB array.
rangeMax	Max value of the RGB range.

Template Parameters

T	Type of RGB components.
U	Type of rangeMax value.

10.302.3.4 RGBToRecommendedDisplayGrayscale()

```
template<typename T , typename U >
unsigned short gdcm::SurfaceHelper::RGBToRecommendedDisplayGrayscale (
             const std::vector< T > & RGB,
             const U rangeMax = 255 )  [static]
```

Convert a RGB color into DICOM grayscale (ready to write).

See also

PS 3.3 C.27.1 tag(0062,000C)

Parameters

RGB	RGB array.
rangeMax	Max value of the RGB range.

Template Parameters

T	Type of RGB components.
U	Type of rangeMax value.

The documentation for this class was generated from the following file:

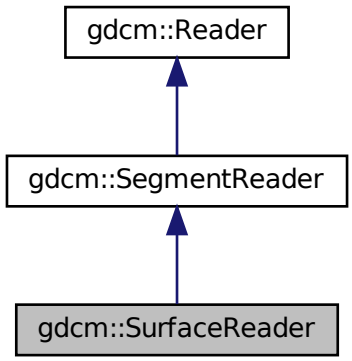
- gdcmSurfaceHelper.h

10.303 gdcm::SurfaceReader Class Reference

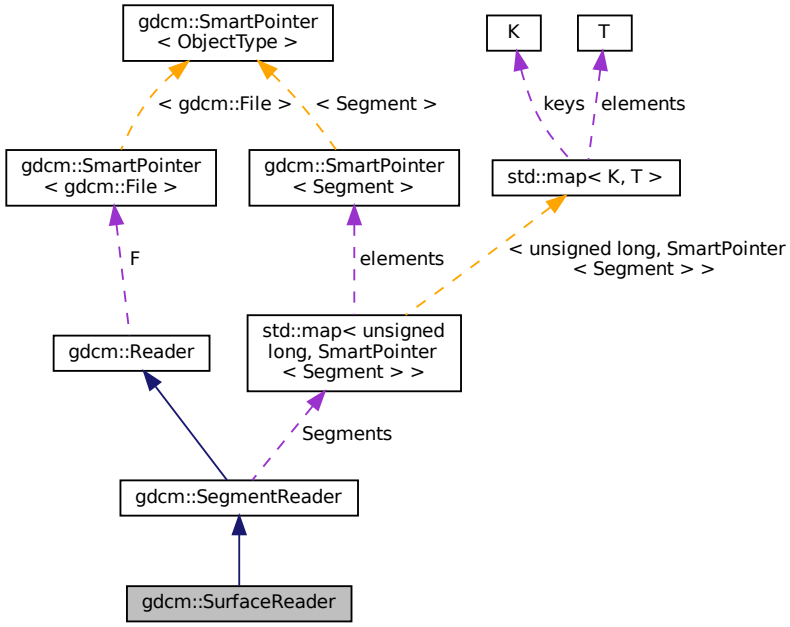
This class defines a SURFACE IE reader.

```
#include <gdcmSurfaceReader.h>
```

Inheritance diagram for gdcm::SurfaceReader:



Collaboration diagram for gdcm::SurfaceReader:



Public Member Functions

- SurfaceReader ()

- ~SurfaceReader () override
- unsigned long GetNumberOfSurfaces () const
- bool Read () override

  Read.

Protected Member Functions

- bool ReadPointMacro (SmartPointer< Surface > surface, const DataSet &surfaceDS)
- bool ReadSurface (const Item &surfaceItem, const unsigned long idx)
- bool ReadSurfaces ()

Additional Inherited Members

10.303.1 Detailed Description

This class defines a SURFACE IE reader.

It reads surface mesh module attributes.

See also

PS 3.3 A.1.2.18 , A.57 and C.27

10.303.2 Constructor & Destructor Documentation

10.303.2.1 SurfaceReader()

```
gdcm::SurfaceReader::SurfaceReader ( )
```

10.303.2.2 ~SurfaceReader()

```
gdcm::SurfaceReader::~SurfaceReader ( )  [override]
```

10.303.3 Member Function Documentation

10.303.3.1 GetNumberOfSurfaces()

```
unsigned long gdcm::SurfaceReader::GetNumberOfSurfaces ( ) const
```

10.303.3.2 Read()

```
bool gdcm::SurfaceReader::Read ( )  [override], [virtual]
```

Read.

Reimplemented from gdcm::SegmentReader.

10.303.3.3 ReadPointMacro()

```
bool gdcm::SurfaceReader::ReadPointMacro (
            SmartPointer< Surface > surface,
            const DataSet & surfaceDS )  [protected]
```

10.303.3.4 ReadSurface()

```
bool gdcm::SurfaceReader::ReadSurface (
            const Item & surfaceItem,
            const unsigned long idx )  [protected]
```

10.303.3.5 ReadSurfaces()

```
bool gdcm::SurfaceReader::ReadSurfaces ( )  [protected]
```

The documentation for this class was generated from the following file:

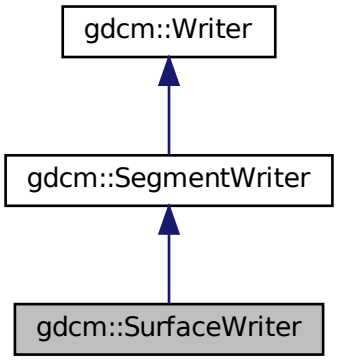
- gdcmSurfaceReader.h

10.304 gdcm::SurfaceWriter Class Reference

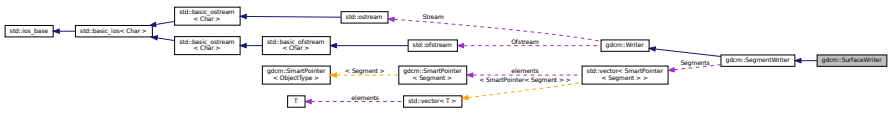
This class defines a SURFACE IE writer.

`#include <gdcmSurfaceWriter.h>`

Inheritance diagram for gdcm::SurfaceWriter:



Collaboration diagram for gdcm::SurfaceWriter:



Public Member Functions

- SurfaceWriter ()
- ~SurfaceWriter () override
- unsigned long GetNumberOfSurfaces ()
- void SetNumberOfSurfaces (const unsigned long nb)
- bool Write () override

  Write.

Protected Member Functions

- void ComputeNumberOfSurfaces ()
- bool PrepareWrite ()
- bool PrepareWritePointMacro (SmartPointer< Surface > surface, DataSet &surfaceDS, const TransferSyntax &ts)

Protected Attributes

- unsigned long NumberOfSurfaces

Additional Inherited Members

10.304.1 Detailed Description

This class defines a SURFACE IE writer.

It writes surface mesh module attributes.

See also

PS 3.3 A.1.2.18 , A.57 and C.27

10.304.2 Constructor & Destructor Documentation

10.304.2.1 SurfaceWriter()

```
gdcm::SurfaceWriter::SurfaceWriter ( )
```

10.304.2.2 ~SurfaceWriter()

```
gdcm::SurfaceWriter::~SurfaceWriter ( )  [override]
```

10.304.3 Member Function Documentation

10.304.3.1 ComputeNumberOfSurfaces()

```
void gdcm::SurfaceWriter::ComputeNumberOfSurfaces ( )  [protected]
```

10.304.3.2 GetNumberOfSurfaces()

```
unsigned long gdcm::SurfaceWriter::GetNumberOfSurfaces ( )
```

10.304.3.3 PrepareWrite()

```
bool gdcm::SurfaceWriter::PrepareWrite ( )  [protected]
```

10.304.3.4 PrepareWritePointMacro()

```
bool gdcm::SurfaceWriter::PrepareWritePointMacro (
            SmartPointer< Surface > surface,
            DataSet & surfaceDS,
            const TransferSyntax & ts )  [protected]
```

10.304.3.5 SetNumberOfSurfaces()

```
void gdcm::SurfaceWriter::SetNumberOfSurfaces (
            const unsigned long nb )
```

10.304.3.6 Write()

```
bool gdcm::SurfaceWriter::Write ( )  [override], [virtual]
```

Write.

Reimplemented from gdcm::SegmentWriter.

10.304.4 Member Data Documentation

10.304.4.1 NumberOfSurfaces

```
unsigned long gdcm::SurfaceWriter::NumberOfSurfaces  [protected]
```

The documentation for this class was generated from the following file:

- gdcmSurfaceWriter.h

10.305 gdcm::SwapCode Class Reference

SwapCode representation.

```
#include <gdcmSwapCode.h>
```

Public Types

- enum SwapCodeType {
  Unknown = 0 ,
  LittleEndian = 1234 ,
  BigEndian = 4321 ,
  BadLittleEndian = 3412 ,
  BadBigEndian = 2143 }

Public Member Functions

- SwapCode (SwapCodeType sc=Unknown)
- operator SwapCode::SwapCodeType () const

Static Public Member Functions

- static const char * GetSwapCodeString (SwapCode const &sc)

Static Protected Member Functions

- static int GetIndex (SwapCode const &sc)

Friends

- std::ostream & operator<< (std::ostream &os, const SwapCode &sc)

10.305.1 Detailed Description

SwapCode representation.

Examples

TestByteSwap.cxx.

10.305.2 Member Enumeration Documentation

10.305.2.1 SwapCodeType

```
enum gdcm::SwapCode::SwapCodeType
```

Enumerator

Unknown	
LittleEndian	
BigEndian	
BadLittleEndian	
BadBigEndian	

10.305.3 Constructor & Destructor Documentation

10.305.3.1 SwapCode()

```
gdcm::SwapCode::SwapCode (
            SwapCodeType sc = Unknown )  [inline]
```

10.305.4 Member Function Documentation

10.305.4.1 GetIndex()

```
static int gdcm::SwapCode::GetIndex (
            SwapCode const & sc )  [static], [protected]
```

10.305.4.2 GetSwapCodeString()

```
static const char * gdcm::SwapCode::GetSwapCodeString (
            SwapCode const & sc )  [static]
```

10.305.4.3 operator SwapCode::SwapCodeType()

```
gdcm::SwapCode::operator SwapCode::SwapCodeType ( ) const  [inline]
```

10.305.5 Friends And Related Function Documentation

10.305.5.1 operator<<

```
std::ostream & operator<< (
            std::ostream & os,
            const SwapCode & sc )  [friend]
```

The documentation for this class was generated from the following file:

- gdcmSwapCode.h

10.306 gdcm::SwapperDoOp Class Reference

```
#include <gdcmSwapper.h>
```

Static Public Member Functions

- template<typename T >
  static T Swap (T val)
- template<typename T >
  static void SwapArray (T *array, size_t n)

10.306.1 Member Function Documentation

10.306.1.1 Swap()

```
template<typename T >
static T gdcm::SwapperDoOp::Swap (
            T val )  [static]
```

10.306.1.2 SwapArray()

```
template<typename T >
static void gdcm::SwapperDoOp::SwapArray (
            T * array,
            size_t n )  [inline], [static]
```

The documentation for this class was generated from the following file:

- gdcmSwapper.h

10.307 gdcm::SwapperNoOp Class Reference

```
#include <gdcmSwapper.h>
```

Static Public Member Functions

- template<typename T >
  static T Swap (T val)
- template<typename T >
  static void SwapArray (T *, size_t)

10.307.1 Detailed Description

Examples

DumpSiemensBase64.cxx, DumpToshibaDTI.cxx, DumpToshibaDTI2.cxx, and ReadExplicitLengthSQIVR.cxx.

10.307.2 Member Function Documentation

10.307.2.1 Swap()

```
template<typename T >
static T gdcm::SwapperNoOp::Swap (
             T val )  [inline], [static]
```

10.307.2.2 SwapArray()

```
template<typename T >
static void gdcm::SwapperNoOp::SwapArray (
             T * ,
             size_t  )  [inline], [static]
```

The documentation for this class was generated from the following file:

- gdcmSwapper.h

10.308 gdcm::System Class Reference

Class to do system operation.

`#include <gdcmSystem.h>`

Static Public Member Functions

- static std::wstring ConvertToUNC (const char *utf8path)
- static bool DeleteDirectory (const char *source)

  remove a directory named source
- static size_t EncodeBytes (char *out, const unsigned char *data, int size)
- static bool FileExists (const char *filename)

  Check whether the specified file exist on the system.
- static bool FileIsDirectory (const char *name)

  Check whether the file specified is a directory:
- static bool FileIsSymlink (const char *name)

  Check whether name is a symlink.
- static size_t FileSize (const char *filename)
- static time_t FileTime (const char *filename)
- static bool FormatDateTime (char date[22], time_t t, long milliseconds=0)
- static bool GetCurrentDateTime (char date[22])
- static const char * GetCurrentModuleFileName ()
- static const char * GetCurrentProcessFileName ()
- static const char * GetCurrentResourcesDirectory ()
- static const char * GetCWD ()

- static bool GetHostName (char hostname[255])
- static const char * GetLastSystemError ()

  Return the last error.
- static const char * GetLocaleCharset ()

  return `locale charmap`
- static const char * GetTimezoneOffsetFromUTC ()
- static bool MakeDirectory (const char *path)

  Create a directory name path.
- static bool ParseDateTime (time_t &timep, const char date[22])

  Parse a date stored as ASCII text into a time_t structured (discard millisecond if any)
- static bool ParseDateTime (time_t &timep, long &milliseconds, const char date[22])
- static bool RemoveFile (const char *source)

  remove a file named source
- static int StrCaseCmp (const char *s1, const char *s2)

  consistent func for C99 spec of strcasecmp/strncasecmp
- static int StrNCaseCmp (const char *s1, const char *s2, size_t n)
- static char * StrSep (char **stringp, const char *delim)
- static char * StrTokR (char *ptr, const char *sep, char **end)

  strtok_r

Static Protected Member Functions

- static bool GetPermissions (const char *file, unsigned short &mode)

  NOT THREAD SAFE.
- static bool SetPermissions (const char *file, unsigned short mode)

10.308.1 Detailed Description

Class to do system operation.

OS independent functionalities

Examples

BasicAnonymizer.cs, BasicImageAnonymizer.cs, ClinicalTrialIdentificationWorkflow.cs, CompressLossyJPEG.cs, DecompressImageMultiframe.cs, DecompressJPEGFile.cs, DumpCSA.cs, ExtractEncapsulatedFile.cs, ExtractImageRegion.cs, ExtractImageRegionWithLUT.cs, ExtractOneFrame.cs, FileAnonymize.cs, FileChangeTS.cs, FileChangeTSLossy.cs, FileStreaming.cs, GetArray.cs, MetaImageMD5Activiz.cs, MpegVideoInfo.cs, ReformatFile.cs, RescaleImage.cs, ScanDirectory.cs, SimplePrint.cs, and StandardizeFiles.cs.

10.308.2 Member Function Documentation

10.308.2.1 ConvertToUNC()

```
static std::wstring gdcm::System::ConvertToUNC (
            const char * utf8path )  [static]
```

When needed convert a PATH into a UNC equivalent. This allow transparent support for path longer that MAX_PATH. Only on _MSC_VER compiler, return empty string otherwise.

10.308.2.2 DeleteDirectory()

```
static bool gdcm::System::DeleteDirectory (
            const char * source )  [static]
```

remove a directory named source

10.308.2.3 EncodeBytes()

```
static size_t gdcm::System::EncodeBytes (
            char * out,
            const unsigned char * data,
            int size )  [static]
```

Used internally by the UIDGenerator class to convert a uuid tape to a DICOM VR:UI type

10.308.2.4 FileExists()

```
static bool gdcm::System::FileExists (
            const char * filename )  [static]
```

Check whether the specified file exist on the system.

Examples

DumpVisusChange.cxx, EncapsulateFileInRawData.cxx, MagnifyFile.cxx, and gdcmorthoplanes.cxx.

10.308.2.5 FileIsDirectory()

```
static bool gdcm::System::FileIsDirectory (
            const char * name )  [static]
```

Check whether the file specified is a directory:

Examples

DumpVisusChange.cxx, gdcmorthoplanes.cxx, and threadgdcm.cxx.

10.308.2.6 FileIsSymlink()

```
static bool gdcm::System::FileIsSymlink (
            const char * name )  [static]
```

Check whether name is a symlink.

10.308.2.7 FileSize()

```
static size_t gdcm::System::FileSize (
            const char * filename )  [static]
```

Return the filesize. 0 if file does not exist.

Warning

you need to use FileExists to differentiate between empty file and missing file.

for very large size file and on system where size_t is not appropriate to store off_t value the function will return 0.

Examples

CheckBigEndianBug.cxx, CreateARGBImage.cxx, CreateCMYKImage.cxx, EncapsulateFileInRawData.cxx, and SimpleScanner.cxx.

10.308.2.8 FileTime()

```
static time_t gdcm::System::FileTime (
            const char * filename )  [static]
```

Return the time of last modification of file 0 if the file does not exist

10.308.2.9 FormatDateTime()

```
static bool gdcm::System::FormatDateTime (
            char date[22],
            time_t t,
            long milliseconds = 0 )  [static]
```

format as ASCII text a time_t with milliseconds See VR::DT from DICOM PS 3.5 milliseconds is in the range [0, 999999]

10.308.2.10 GetCurrentDateTime()

```
static bool gdcm::System::GetCurrentDateTime (
            char date[22] )  [static]
```

Return the current data time, and format it as ASCII text. This is simply a call to gettimeofday + FormatDateTime, since WIN32 do not have an implementation for gettimeofday, this is more portable. The call time(0) is not precise for our resolution

Examples

TemplateEmptyImage.cxx.

10.308.2.11 GetCurrentModuleFileName()

```
static const char * gdcm::System::GetCurrentModuleFileName ( )  [static]
```

Return the directory the current module is located: NOT THREAD SAFE

10.308.2.12 GetCurrentProcessFileName()

```
static const char * gdcm::System::GetCurrentProcessFileName ( )  [static]
```

Return the directory the current process (executable) is located: NOT THREAD SAFE

10.308.2.13 GetCurrentResourcesDirectory()

```
static const char * gdcm::System::GetCurrentResourcesDirectory ( )  [static]
```

On some system (Apple) return the path to the current bundled 'Resources' directory NOT THREAD SAFE

10.308.2.14 GetCWD()

```
static const char * gdcm::System::GetCWD ( )  [static]
```

Return current working directory Warning: if current working path is too long (>2048 bytes) the call will fail and call will return NULL NOT THREAD SAFE

10.308.2.15 GetHostName()

```
static bool gdcm::System::GetHostName (
            char hostname[255] )  [static]
```

Retrieve the hostname, only the first 255 byte are copied. This may come handy to specify the Station Name

10.308.2.16 GetLastSystemError()

```
static const char * gdcm::System::GetLastSystemError ( )  [static]
```

Return the last error.

10.308.2.17 GetLocaleCharset()

```
static const char * gdcm::System::GetLocaleCharset ( )  [static]
```

return `locale charmap`

10.308.2.18 GetPermissions()

```
static bool gdcm::System::GetPermissions (
            const char * file,
            unsigned short & mode )  [static], [protected]
```

NOT THREAD SAFE.

10.308.2.19 GetTimezoneOffsetFromUTC()

```
static const char * gdcm::System::GetTimezoneOffsetFromUTC ( )  [static]
```

Return the value for Timezone Offset From UTC as string.

Warning

not thread safe

10.308.2.20 MakeDirectory()

```
static bool gdcm::System::MakeDirectory (
            const char * path )  [static]
```

Create a directory name path.

10.308.2.21 ParseDateTime() [1/2]

```
static bool gdcm::System::ParseDateTime (
            time_t & timep,
            const char date[22] )  [static]
```

Parse a date stored as ASCII text into a time_t structured (discard millisecond if any)

10.308.2.22 ParseDateTime() [2/2]

```
static bool gdcm::System::ParseDateTime (
            time_t & timep,
            long & milliseconds,
            const char date[22] )  [static]
```

Parse a date stored as ASCII text into a time_t structured and millisecond

See also

FormatDateTime

10.308.2.23 RemoveFile()

```
static bool gdcm::System::RemoveFile (
            const char * source )  [static]
```

remove a file named source

10.308.2.24 SetPermissions()

```
static bool gdcm::System::SetPermissions (
            const char * file,
            unsigned short mode )  [static], [protected]
```

10.308.2.25 StrCaseCmp()

```
static int gdcm::System::StrCaseCmp (
             const char * s1,
             const char * s2 )  [static]
```

consistent func for C99 spec of strcasecmp/strncasecmp

10.308.2.26 StrNCaseCmp()

```
static int gdcm::System::StrNCaseCmp (
             const char * s1,
             const char * s2,
             size_t n )  [static]
```

Precondition

n != 0

10.308.2.27 StrSep()

```
static char * gdcm::System::StrSep (
             char ** stringp,
             const char * delim )  [static]
```

strsep param stringp is passed by pointer, it may be modified, you'll need to make a copy, in case you want to free the memory pointed at

10.308.2.28 StrTokR()

```
static char * gdcm::System::StrTokR (
             char * ptr,
             const char * sep,
             char ** end )  [static]
```

strtok_r

The documentation for this class was generated from the following file:

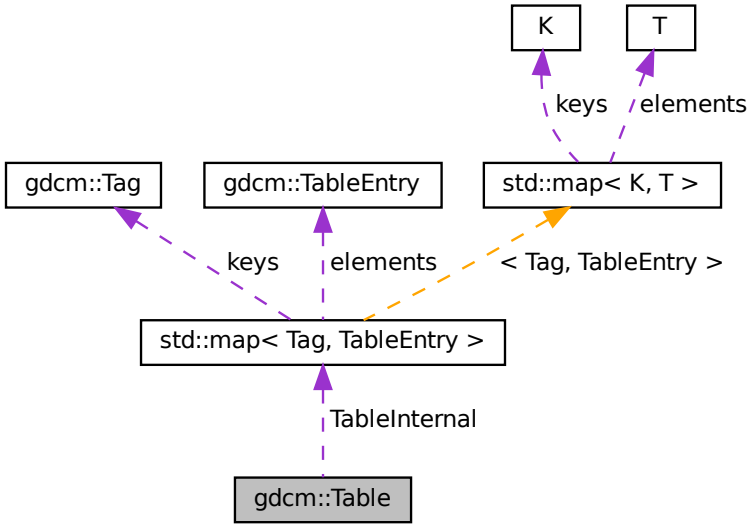
- gdcmSystem.h

10.309 gdcm::Table Class Reference

Table.

`#include <gdcmTable.h>`

Collaboration diagram for gdcm::Table:



Public Types

- typedef std::map< Tag, TableEntry > MapTableEntry

Public Member Functions

- Table ()=default
- Table (const Table &_val)=delete
- ~Table ()=default
- const TableEntry & GetTableEntry (const Tag &tag) const
- void InsertEntry (Tag const &tag, TableEntry const &te)
- Table & operator= (const Table &_val)=delete

Public Attributes

- MapTableEntry TableInternal

Friends

- std::ostream & operator<< (std::ostream &_os, const Table &_val)

10.309.1 Detailed Description

Table.

10.309.2 Member Typedef Documentation

10.309.2.1 MapTableEntry

```
typedef std::map<Tag, TableEntry> gdcm::Table::MapTableEntry
```

10.309.3 Constructor & Destructor Documentation

10.309.3.1 Table() [1/2]

```
gdcm::Table::Table ( )  [default]
```

10.309.3.2 ~Table()

```
gdcm::Table::~Table ( )  [default]
```

10.309.3.3 Table() [2/2]

```
gdcm::Table::Table (
            const Table & _val )  [delete]
```

10.309.4 Member Function Documentation

10.309.4.1 GetTableEntry()

```
const TableEntry & gdcm::Table::GetTableEntry (
            const Tag & tag ) const  [inline]
```

References GetTableEntry(), and TableInternal.

Referenced by GetTableEntry().

10.309.4.2 InsertEntry()

```
void gdcm::Table::InsertEntry (
            Tag const & tag,
            TableEntry const & te )  [inline]
```

References TableInternal.

10.309.4.3 operator=()

```
Table & gdcm::Table::operator= (
            const Table & _val )  [delete]
```

10.309.5 Friends And Related Function Documentation

10.309.5.1 operator<<

```
std::ostream & operator<< (
            std::ostream & _os,
            const Table & _val )  [friend]
```

10.309.6 Member Data Documentation

10.309.6.1 TableInternal

```
MapTableEntry gdcm::Table::TableInternal
```

Referenced by GetTableEntry(), and InsertEntry().

The documentation for this class was generated from the following file:

- gdcmTable.h

10.310 gdcm::TableEntry Class Reference

TableEntry.

```
#include <gdcmTableEntry.h>
```

Public Member Functions

- TableEntry (const char *attribute=nullptr, Type const &type=Type(), const char *des=nullptr)
- ~TableEntry ()=default

10.310.1 Detailed Description

TableEntry.

10.310.2 Constructor & Destructor Documentation

10.310.2.1 TableEntry()

```
gdcm::TableEntry::TableEntry (
            const char * attribute = nullptr,
            Type const & type = Type(),
            const char * des = nullptr )  [inline]
```

10.310.2.2 ~TableEntry()

```
gdcm::TableEntry::~TableEntry ( )  [default]
```

The documentation for this class was generated from the following file:

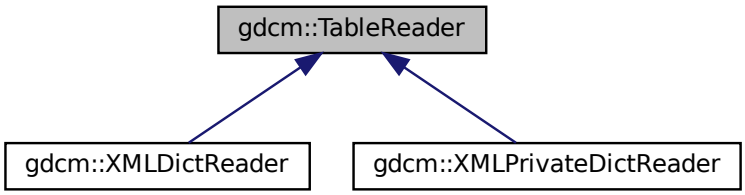
- gdcmTableEntry.h

10.311 gdcm::TableReader Class Reference

Class for representing a TableReader.

```
#include <gdcmTableReader.h>
```

Inheritance diagram for gdcm::TableReader:



Public Member Functions

- TableReader (Defs &defs)
- virtual ~TableReader ()=default
- virtual void CharacterDataHandler (const char *data, int length)
- virtual void EndElement (const char *name)
- const Defs & GetDefs () const
- const char * GetFilename ()
- void HandleIOD (const char **atts)
- void HandleIODEntry (const char **atts)
- void HandleMacro (const char **atts)
- void HandleMacroEntry (const char **atts)
- void HandleMacroEntryDescription (const char **atts)
- void HandleModule (const char **atts)
- void HandleModuleEntry (const char **atts)
- void HandleModuleEntryDescription (const char **atts)
- void HandleModuleInclude (const char **atts)
- int Read ()
- void SetFilename (const char *filename)
- virtual void StartElement (const char *name, const char **atts)

10.311.1 Detailed Description

Class for representing a TableReader.

Note

This class is an empty shell meant to be derived

10.311.2 Constructor & Destructor Documentation

10.311.2.1 TableReader()

```
gdcm::TableReader::TableReader (
            Defs & defs )  [inline]
```

10.311.2.2 ~TableReader()

```
virtual gdcm::TableReader::~TableReader ( )  [virtual], [default]
```

10.311.3 Member Function Documentation

10.311.3.1 CharacterDataHandler()

```
virtual void gdcm::TableReader::CharacterDataHandler (
            const char * data,
            int length )  [virtual]
```

Reimplemented in gdcm::XMLDictReader, and gdcm::XMLPrivateDictReader.

10.311.3.2 EndElement()

```
virtual void gdcm::TableReader::EndElement (
            const char * name )  [virtual]
```

Reimplemented in gdcm::XMLDictReader, and gdcm::XMLPrivateDictReader.

10.311.3.3 GetDefs()

```
const Defs & gdcm::TableReader::GetDefs ( ) const  [inline]
```

10.311.3.4 GetFilename()

```
const char * gdcm::TableReader::GetFilename ( )  [inline]
```

10.311.3.5 HandleIOD()

```
void gdcm::TableReader::HandleIOD (
            const char ** atts )
```

10.311.3.6 HandleIODEntry()

```
void gdcm::TableReader::HandleIODEntry (
            const char ** atts )
```

10.311.3.7 HandleMacro()

```
void gdcm::TableReader::HandleMacro (
            const char ** atts )
```

10.311.3.8 HandleMacroEntry()

```
void gdcm::TableReader::HandleMacroEntry (
            const char ** atts )
```

10.311.3.9 HandleMacroEntryDescription()

```
void gdcm::TableReader::HandleMacroEntryDescription (
            const char ** atts )
```

10.311.3.10 HandleModule()

```
void gdcm::TableReader::HandleModule (
            const char ** atts )
```

10.311.3.11 HandleModuleEntry()

```
void gdcm::TableReader::HandleModuleEntry (
            const char ** atts )
```

10.311.3.12 HandleModuleEntryDescription()

```
void gdcm::TableReader::HandleModuleEntryDescription (
            const char ** atts )
```

10.311.3.13 HandleModuleInclude()

```
void gdcm::TableReader::HandleModuleInclude (
            const char ** atts )
```

10.311.3.14 Read()

```
int gdcm::TableReader::Read ( )
```

10.311.3.15 SetFilename()

```
void gdcm::TableReader::SetFilename (
            const char * filename )  [inline]
```

10.311.3.16 StartElement()

```
virtual void gdcm::TableReader::StartElement (
            const char * name,
            const char ** atts )  [virtual]
```

Reimplemented in gdcm::XMLDictReader, and gdcm::XMLPrivateDictReader.

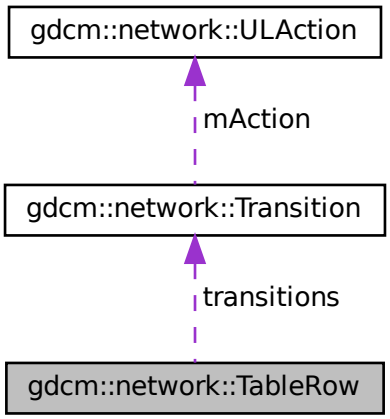
The documentation for this class was generated from the following file:

- gdcmTableReader.h

10.312 gdcm::network::TableRow Class Reference

```
#include <gdcmULTransitionTable.h>
```

Collaboration diagram for gdcm::network::TableRow:



Public Member Functions

- TableRow ()
- ~TableRow ()

Public Attributes

- Transition * transitions [cMaxStateID]

10.312.1 Constructor & Destructor Documentation

10.312.1.1 TableRow()

```
gdcm::network::TableRow::TableRow ( )  [inline]
```

References gdcm::network::cMaxStateID, and transitions.

10.312.1.2 ~TableRow()

```
gdcm::network::TableRow::~TableRow ( )  [inline]
```

References gdcm::network::cMaxStateID, and transitions.

10.312.2 Member Data Documentation

10.312.2.1 transitions

```
Transition* gdcm::network::TableRow::transitions[cMaxStateID]
```

Referenced by TableRow(), and ~TableRow().

The documentation for this class was generated from the following file:

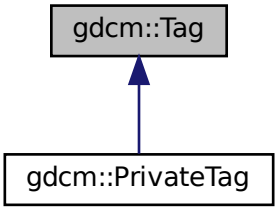
- gdcmULTransitionTable.h

10.313 gdcm::Tag Class Reference

Class to represent a DICOM Data Element (Attribute) Tag (Group, Element).

```
#include <gdcmTag.h>
```

Inheritance diagram for gdcm::Tag:



Public Member Functions

- Tag (const Tag &_val)
- Tag (uint16_t group, uint16_t element)

  Constructor with 2∗uint16_t.
- Tag (uint32_t tag=0)

  Constructor with 1∗uint32_t Prefer the cstor that takes two uint16_t.
- uint16_t GetElement () const

  Returns the 'Element number' of the given Tag.
- uint32_t GetElementTag () const

  Returns the full tag value of the given Tag.
- uint16_t GetGroup () const

  Returns the 'Group number' of the given Tag.
- uint32_t GetLength () const

  return the length of tag (read: size on disk)
- Tag GetPrivateCreator () const

  Return the Private Creator Data Element tag of a private data element.
- bool IsGroupLength () const

  return whether the tag correspond to a group length tag:
- bool IsGroupXX (const Tag &t) const

  e.g 6002,3000 belong to groupXX: 6000,3000
- bool IsIllegal () const

  return if the tag is considered to be an illegal tag
- bool IsPrivate () const
- bool IsPrivateCreator () const
- bool IsPublic () const
- bool operator!= (const Tag &_val) const
- bool operator< (const Tag &_val) const
- bool operator<= (const Tag &t2) const
- Tag & operator= (const Tag &_val)
- bool operator== (const Tag &_val) const
- uint16_t & operator[ ] (const unsigned int &_id)

  Returns the Group or Element of the given Tag, depending on id (0/1)
- const uint16_t & operator[ ] (const unsigned int &_id) const

  Returns the Group or Element of the given Tag, depending on id (0/1)
- std::string PrintAsContinuousString () const
- std::string PrintAsContinuousUpperCaseString () const

  Same as PrintAsContinuousString, but hexadecimal [a-f] are printed using upper case.
- std::string PrintAsPipeSeparatedString () const
- template<typename TSwap >
  std::istream & Read (std::istream &is)

  Read a tag from binary representation.
- bool ReadFromCommaSeparatedString (const char ∗str)
- bool ReadFromContinuousString (const char ∗str)
- bool ReadFromPipeSeparatedString (const char ∗str)
- void SetElement (uint16_t element)

  Sets the 'Element number' of the given Tag.
- void SetElementTag (uint16_t group, uint16_t element)

Sets the 'Group number' & 'Element number' of the given Tag.

- void SetElementTag (uint32_t tag)

  Sets the full tag value of the given Tag.
- void SetGroup (uint16_t group)

  Sets the 'Group number' of the given Tag.
- void SetPrivateCreator (Tag const &t)

  Set private creator:
- template<typename TSwap >
  const std::ostream & Write (std::ostream &os) const

  Write a tag in binary rep.

Friends

- std::ostream & operator<< (std::ostream &_os, const Tag &_val)
- std::istream & operator>> (std::istream &_is, Tag &_val)

10.313.1 Detailed Description

Class to represent a DICOM Data Element (Attribute) Tag (Group, Element).

Basically an uint32_t which can also be expressed as two uint16_t (group and element)

Note

DATA ELEMENT TAG: A unique identifier for a Data Element composed of an ordered pair of numbers (a Group Number followed by an Element Number). GROUP NUMBER: The first number in the ordered pair of numbers that makes up a Data Element Tag. ELEMENT NUMBER: The second number in the ordered pair of numbers that makes up a Data Element Tag.

Examples

BasicAnonymizer.cs, BasicImageAnonymizer.cs, ChangeSequenceUltrasound.cxx, ClinicalTrialAnnotate.cxx, ClinicalTrialIdentificationWorkflow.cs, CreateARGBImage.cxx, CreateCMYKImage.cxx, CreateFakeRTDOSE.cxx, CreateJPIPDataSet.cxx, DecompressImage.cs, DeriveSeries.cxx, DumpToSQLITE3.cxx, DumpVisusChange.cxx, DuplicatePCDE.cxx, EncapsulateFileInRawData.cxx, ExtractEncapsulatedFile.cs, ExtractEncryptedContent.cxx, Extracting_All_Resolution.cxx, Fake_Image_Using_Stream_Image_Writer.cxx, FileAnonymize.cs, FileChangeTS.cs, FileChangeTSLossy.cs, FixBrokenJ2K.cxx, FixJAIBugJPEGLS.cxx, GenAllVR.cxx, GenFakeIdentifyFile.cxx, GenFakeImage.cxx, GenLongSeqs.cxx, GenSeqs.cxx, GetJPEGSamplePrecision.cxx, GetSequenceUltrasound.cxx, GetSubSequenceData.cxx, LargeVRDSExplicit.cxx, MakeTemplate.cxx, ManipulateFile.cs, MergeTwoFiles.cxx, MpegVideoInfo.cs, PatchFile.cxx, PublicDict.cxx, ReadAndDumpDICOMDIR.cxx, ReadAndDumpDICOMDIR2.cxx, ReadAndPrintAttributes.cxx, ReadExplicitLengthSQIVR.cxx, ScanDirectory.cs, SimpleScanner.cxx, SortImage.cxx, StreamImageReaderTest.cxx, TraverseModules.cxx, VolumeSorter.cxx, gdcmrtionplan.cxx, gdcmrtplan.cxx, iU22tomultisc.cxx, pmsct_rgb1.cxx, and rle2img.cxx.

10.313.2 Constructor & Destructor Documentation

10.313.2.1 Tag() [1/3]

```
gdcm::Tag::Tag (
            uint16_t group,
            uint16_t element )  [inline]
```

Constructor with 2∗uint16_t.

10.313.2.2 Tag() [2/3]

```
gdcm::Tag::Tag (
            uint32_t tag = 0 )  [inline]
```

Constructor with 1∗uint32_t Prefer the cstor that takes two uint16_t.

10.313.2.3 Tag() [3/3]

```
gdcm::Tag::Tag (
            const Tag & _val )  [inline]
```

References tag.

10.313.3 Member Function Documentation

10.313.3.1 GetElement()

```
uint16_t gdcm::Tag::GetElement ( ) const  [inline]
```

Returns the 'Element number' of the given Tag.

Examples

DuplicatePCDE.cxx, and PublicDict.cxx.

Referenced by gdcm::PrivateTag::PrivateTag(), gdcm::DataSet::ComputeGroupLength(), IsGroupXX(), gdcm::PrivateDict::PrintXML(), gdcm::SequenceOfFragments::ReadValue(), and SetPrivateCreator().

10.313.3.2 GetElementTag()

```
uint32_t gdcm::Tag::GetElementTag ( ) const  [inline]
```

Returns the full tag value of the given Tag.

Referenced by gdcm::PrivateTag::operator!=(), gdcm::PrivateTag::operator=(), and gdcm::PrivateTag::operator==().

10.313.3.3 GetGroup()

```
uint16_t gdcm::Tag::GetGroup ( ) const  [inline]
```

Returns the 'Group number' of the given Tag.

Examples

DuplicatePCDE.cxx, and GenAllVR.cxx.

Referenced by gdcm::DataSet::ComputeGroupLength(), gdcm::DataSet::Insert(), gdcm::FileMetaInformation::Insert(), gdcm::CommandDataSet::Insert(), IsGroupXX(), gdcm::PrivateDict::PrintXML(), gdcm::SequenceOfFragments::ReadValue(), and SetPrivateCreator().

10.313.3.4 GetLength()

```
uint32_t gdcm::Tag::GetLength ( ) const  [inline]
```

return the length of tag (read: size on disk)

10.313.3.5 GetPrivateCreator()

```
Tag gdcm::Tag::GetPrivateCreator ( ) const  [inline]
```

Return the Private Creator Data Element tag of a private data element.

References SetElement().

10.313.3.6 IsGroupLength()

```
bool gdcm::Tag::IsGroupLength ( ) const  [inline]
```

return whether the tag correspond to a group length tag:

10.313.3.7 IsGroupXX()

```
bool gdcm::Tag::IsGroupXX (
             const Tag & t ) const  [inline]
```

e.g 6002,3000 belong to groupXX: 6000,3000

References GetElement(), GetGroup(), and IsPrivate().

10.313.3.8 IsIllegal()

```
bool gdcm::Tag::IsIllegal ( ) const  [inline]
```

return if the tag is considered to be an illegal tag

10.313.3.9 IsPrivate()

```
bool gdcm::Tag::IsPrivate ( ) const  [inline]
```

PRIVATE DATA ELEMENT: Additional Data Element, defined by an implementor, to communicate information that is not contained in Standard Data Elements. Private Data elements have odd Group Numbers.

Examples

DuplicatePCDE.cxx.

Referenced by IsGroupXX(), and SetPrivateCreator().

10.313.3.10 IsPrivateCreator()

```
bool gdcm::Tag::IsPrivateCreator ( ) const  [inline]
```

Returns if tag is a Private Creator (xxxx,00yy), where xxxx is odd number and yy in [0x10,0xFF]

Examples

DuplicatePCDE.cxx.

10.313.3.11 IsPublic()

```
bool gdcm::Tag::IsPublic ( ) const  [inline]
```

STANDARD DATA ELEMENT: A Data Element defined in the DICOM Standard, and therefore listed in the DICOM Data Element Dictionary in PS 3.6. Is the Tag from the Public dict...well the implementation is buggy it does not prove the element is indeed in the dict...

10.313.3.12 operator"!=()

```
bool gdcm::Tag::operator!= (
            const Tag & _val ) const  [inline]
```

References tag.

10.313.3.13 operator<()

```
bool gdcm::Tag::operator< (
            const Tag & _val ) const  [inline]
```

DICOM Standard expects the Data Element to be sorted by Tags All other comparison can be constructed from this one and operator ==

References tag, and tags.

10.313.3.14 operator<=()

```
bool gdcm::Tag::operator<= (
            const Tag & t2 ) const  [inline]
```

10.313.3.15 operator=()

```
Tag & gdcm::Tag::operator= (
            const Tag & _val )  [inline]
```

References tag.

10.313.3.16 operator==()

```
bool gdcm::Tag::operator== (
            const Tag & _val ) const  [inline]
```

References tag.

10.313.3.17 operator[]() [1/2]

```
uint16_t & gdcm::Tag::operator[] (
            const unsigned int & _id )  [inline]
```

Returns the Group or Element of the given Tag, depending on id (0/1)

10.313.3.18 operator[]() [2/2]

```
const uint16_t & gdcm::Tag::operator[] (
            const unsigned int & _id ) const  [inline]
```

Returns the Group or Element of the given Tag, depending on id (0/1)

10.313.3.19 PrintAsContinuousString()

```
std::string gdcm::Tag::PrintAsContinuousString ( ) const
```

Print tag value with no separating comma: eg. tag = "12345678" It comes in useful when reading tag values from XML file(in NativeDICOMModel)

10.313.3.20 PrintAsContinuousUpperCaseString()

```
std::string gdcm::Tag::PrintAsContinuousUpperCaseString ( ) const
```

Same as PrintAsContinuousString, but hexadecimal [a-f] are printed using upper case.

10.313.3.21 PrintAsPipeSeparatedString()

```
std::string gdcm::Tag::PrintAsPipeSeparatedString ( ) const
```

Print as a pipe separated string (GDCM 1.x compat only). Do not use in newer code

See also

ReadFromPipeSeparatedString

10.313.3.22 Read()

```
template<typename TSwap >
std::istream & gdcm::Tag::Read (
             std::istream & is )  [inline]
```

Read a tag from binary representation.

10.313.3.23 ReadFromCommaSeparatedString()

```
bool gdcm::Tag::ReadFromCommaSeparatedString (
             const char * str )
```

Read from a comma separated string. This is a highly user oriented function, the string should be formatted as↩ : 1234,5678 to specify the tag (0x1234,0x5678) The notation comes from the DICOM standard, and is handy to use from a command line program

10.313.3.24 ReadFromContinuousString()

```
bool gdcm::Tag::ReadFromContinuousString (
             const char * str )
```

Read From XML formatted tag value eg. tag = "12345678" It comes in useful when reading tag values from XML file(in NativeDICOMModel)

10.313.3.25 ReadFromPipeSeparatedString()

```
bool gdcm::Tag::ReadFromPipeSeparatedString (
            const char * str )
```

Read from a pipe separated string (GDCM 1.x compat only). Do not use in newer code

See also

ReadFromCommaSeparatedString

10.313.3.26 SetElement()

```
void gdcm::Tag::SetElement (
            uint16_t element )  [inline]
```

Sets the 'Element number' of the given Tag.

Examples

DuplicatePCDE.cxx, and PublicDict.cxx.

Referenced by GetPrivateCreator().

10.313.3.27 SetElementTag() [1/2]

```
void gdcm::Tag::SetElementTag (
            uint16_t group,
            uint16_t element )  [inline]
```

Sets the 'Group number' & 'Element number' of the given Tag.

10.313.3.28 SetElementTag() [2/2]

```
void gdcm::Tag::SetElementTag (
            uint32_t tag )  [inline]
```

Sets the full tag value of the given Tag.

10.313.3.29 SetGroup()

```
void gdcm::Tag::SetGroup (
            uint16_t group )  [inline]
```

Sets the 'Group number' of the given Tag.

10.313.3.30 SetPrivateCreator()

```
void gdcm::Tag::SetPrivateCreator (
            Tag const & t )  [inline]
```

Set private creator:

Examples

DuplicatePCDE.cxx.

References GetElement(), GetGroup(), and IsPrivate().

10.313.3.31 Write()

```
template<typename TSwap >
const std::ostream & gdcm::Tag::Write (
            std::ostream & os ) const  [inline]
```

Write a tag in binary rep.

Referenced by gdcm::Item::Write(), gdcm::SequenceOfFragments::Write(), and gdcm::SequenceOfItems::Write().

10.313.4 Friends And Related Function Documentation

10.313.4.1 operator<<

```
std::ostream & operator<< (
            std::ostream & _os,
            const Tag & _val )  [friend]
```

10.313.4.2 operator>>

```
std::istream & operator>> (
            std::istream & _is,
            Tag & _val )  [friend]
```

10.313.5 Member Data Documentation

10.313.5.1 bytes

```
char gdcm::Tag::bytes[4]
```

10.313.5.2 tag

```
uint32_t gdcm::Tag::tag
```

Referenced by Tag(), operator!=(), operator<(), operator=(), and operator==().

10.313.5.3 tags

```
uint16_t gdcm::Tag::tags[2]
```

Referenced by operator<().

The documentation for this class was generated from the following file:

- gdcmTag.h

10.314 gdcm::TagPath Class Reference

class to handle a path of tag.

```
#include <gdcmTagPath.h>
```

Public Member Functions

- TagPath ()
- ~TagPath ()
- bool ConstructFromString (const char *path)
- bool ConstructFromTagList (Tag const *l, unsigned int n)

  Construct from a list of tags.
- void Print (std::ostream &) const
- bool Push (Tag const &t)
- bool Push (unsigned int itemnum)

Static Public Member Functions

- static bool IsValid (const char *path)

  Return if path is valid or not.

10.314.1 Detailed Description

class to handle a path of tag.

Any Resemblance to Existing XPath is Purely Coincidental `ftp://medical.nema.org/medical/dicom/supps/sup118_pc.pdf`

10.314.2 Constructor & Destructor Documentation

10.314.2.1 TagPath()

```
gdcm::TagPath::TagPath ( )
```

10.314.2.2 ~TagPath()

```
gdcm::TagPath::~TagPath ( )
```

10.314.3 Member Function Documentation

10.314.3.1 ConstructFromString()

```
bool gdcm::TagPath::ConstructFromString (
            const char * path )
```

"/0018,0018/"... No space allowed, comma is use to separate tag group from tag element and slash is used to separate tag return false if invalid

10.314.3.2 ConstructFromTagList()

```
bool gdcm::TagPath::ConstructFromTagList (
            Tag const * l,
            unsigned int n )
```

Construct from a list of tags.

10.314.3.3 IsValid()

```
static bool gdcm::TagPath::IsValid (
            const char * path )  [static]
```

Return if path is valid or not.

10.314.3.4 Print()

```
void gdcm::TagPath::Print (
            std::ostream &  ) const
```

10.314.3.5 Push() [1/2]

```
bool gdcm::TagPath::Push (
            Tag const & t )
```

10.314.3.6 Push() [2/2]

```
bool gdcm::TagPath::Push (
            unsigned int itemnum )
```

The documentation for this class was generated from the following file:

- gdcmTagPath.h

10.315 gdcm::Testing Class Reference

class for testing

`#include <gdcmTesting.h>`

Public Types

- typedef const char ∗const (∗ MD5DataImagesType)[2]
- typedef const char ∗const (∗ MediaStorageDataFilesType)[2]

  return the table that map the media storage (as string) of a filename (gdcmData)

Public Member Functions

- Testing ()=default
- ∼Testing ()=default
- void Print (std::ostream &os=std::cout)

  Print.

Static Public Member Functions

- static bool ComputeFileMD5 (const char ∗filename, char digest_str[33])
- static bool ComputeMD5 (const char ∗buffer, size_t buf_len, char digest_str[33])
- static const char ∗ GetDataExtraRoot ()

  Return the GDCM DATA EXTRA ROOT.
- static const char ∗ GetDataRoot ()

  Return the GDCM DATA ROOT.
- static const char ∗ GetFileName (unsigned int file)
- static const char ∗const ∗ GetFileNames ()

  return the table of fullpath to gdcmData DICOM files:
- static int GetLossyFlagFromFile (const char ∗filepath)
- static const char ∗const ∗ GetMD5DataImage (unsigned int file)
- static MD5DataImagesType GetMD5DataImages ()
- static const char ∗ GetMD5FromBrokenFile (const char ∗filepath)
- static const char ∗ GetMD5FromFile (const char ∗filepath)

- static const char ∗const ∗ GetMediaStorageDataFile (unsigned int file)
- static MediaStorageDataFilesType GetMediaStorageDataFiles ()
- static const char ∗ GetMediaStorageFromFile (const char ∗filepath)
- static unsigned int GetNumberOfFileNames ()
- static unsigned int GetNumberOfMD5DataImages ()
- static unsigned int GetNumberOfMediaStorageDataFiles ()
- static const char ∗ GetPixelSpacingDataRoot ()

  Return the GDCM PIXEL SPACING DATA ROOT (See David Clunie website for dataset)
- static std::streamoff GetSelectedPrivateGroupOffsetFromFile (const char ∗filepath)
- static std::streamoff GetSelectedTagsOffsetFromFile (const char ∗filepath)
- static const char ∗ GetSourceDirectory ()
- static std::streamoff GetStreamOffsetFromFile (const char ∗filepath)
- static const char ∗ GetTempDirectory (const char ∗subdir=nullptr)
- static const wchar_t ∗ GetTempDirectoryW (const wchar_t ∗subdir=nullptr)

  NOT THREAD SAFE.
- static const char ∗ GetTempFilename (const char ∗filename, const char ∗subdir=nullptr)

  NOT THREAD SAFE.
- static const wchar_t ∗ GetTempFilenameW (const wchar_t ∗filename, const wchar_t ∗subdir=nullptr)

  NOT THREAD SAFE.

10.315.1 Detailed Description

class for testing

this class is used for the nightly regression system for GDCM It makes heavily use of md5 computation

See also

gdcm::MD5 class for md5 computation

10.315.2 Member Typedef Documentation

10.315.2.1 MD5DataImagesType

```
typedef const char* const(* gdcm::Testing::MD5DataImagesType)[2]
```

return the table that map the md5 (as in md5sum) of the Pixel Data associated to a filename

10.315.2.2 MediaStorageDataFilesType

```
typedef const char* const(* gdcm::Testing::MediaStorageDataFilesType)[2]
```

return the table that map the media storage (as string) of a filename (gdcmData)

10.315.3 Constructor & Destructor Documentation

10.315.3.1 Testing()

```
gdcm::Testing::Testing ( )  [default]
```

10.315.3.2 ~Testing()

```
gdcm::Testing::~Testing ( )  [default]
```

10.315.4 Member Function Documentation

10.315.4.1 ComputeFileMD5()

```
static bool gdcm::Testing::ComputeFileMD5 (
            const char * filename,
            char digest_str[33] )  [static]
```

Examples

MetaImageMD5Activiz.cs.

10.315.4.2 ComputeMD5()

```
static bool gdcm::Testing::ComputeMD5 (
            const char * buffer,
            size_t buf_len,
            char digest_str[33] )  [static]
```

MD5 stuff digest_str needs to be at least : strlen = [2∗16+1]; string will be \0 padded. (md5 are 32 bytes long) Testing is not meant to be shipped with an installed GDCM release, always prefer the gdcm::MD5 API when doing md5 computation.

10.315.4.3 GetDataExtraRoot()

```
static const char * gdcm::Testing::GetDataExtraRoot ( )  [static]
```

Return the GDCM DATA EXTRA ROOT.

Examples

DiscriminateVolume.cxx, VolumeSorter.cxx, and reslicesphere.cxx.

10.315.4.4 GetDataRoot()

```
static const char * gdcm::Testing::GetDataRoot ( )  [static]
```

Return the GDCM DATA ROOT.

Examples

Convert16BitsTo8Bits.cxx, ConvertMultiFrameToSingleFrame.cxx, ConvertRGBToLuminance.cxx, and MagnifyFile.cxx.

10.315.4.5 GetFileName()

```
static const char * gdcm::Testing::GetFileName (
            unsigned int file )  [static]
```

Examples

MetaImageMD5Activiz.cs.

10.315.4.6 GetFileNames()

```
static const char *const * gdcm::Testing::GetFileNames ( )  [static]
```

return the table of fullpath to gdcmData DICOM files:

Examples

TestReader.cxx.

10.315.4.7 GetLossyFlagFromFile()

```
static int gdcm::Testing::GetLossyFlagFromFile (
            const char * filepath )  [static]
```

Return the lossy flag of the given filename -1 -> Error 0 -> Lossless 1 -> Lossy

10.315.4.8 GetMD5DataImage()

```
static const char *const * gdcm::Testing::GetMD5DataImage (
            unsigned int file )  [static]
```

10.315.4.9 GetMD5DataImages()

```
static MD5DataImagesType gdcm::Testing::GetMD5DataImages ( )  [static]
```

10.315.4.10 GetMD5FromBrokenFile()

```
static const char * gdcm::Testing::GetMD5FromBrokenFile (
            const char * filepath )  [static]
```

Return what should have been the md5 of file 'filepath' This is based on current GDCM implementation to decipher a broken DICOM file.

10.315.4.11 GetMD5FromFile()

```
static const char * gdcm::Testing::GetMD5FromFile (
            const char * filepath )  [static]
```

10.315.4.12 GetMediaStorageDataFile()

```
static const char *const * gdcm::Testing::GetMediaStorageDataFile (
            unsigned int file )  [static]
```

10.315.4.13 GetMediaStorageDataFiles()

```
static MediaStorageDataFilesType gdcm::Testing::GetMediaStorageDataFiles ( )  [static]
```

10.315.4.14 GetMediaStorageFromFile()

```
static const char * gdcm::Testing::GetMediaStorageFromFile (
            const char * filepath )  [static]
```

Examples

MetaImageMD5Activiz.cs, and TestReader.cxx.

10.315.4.15 GetNumberOfFileNames()

```
static unsigned int gdcm::Testing::GetNumberOfFileNames ( )  [static]
```

Examples

MetaImageMD5Activiz.cs.

10.315.4.16 GetNumberOfMD5DataImages()

```
static unsigned int gdcm::Testing::GetNumberOfMD5DataImages ( )  [static]
```

10.315.4.17 GetNumberOfMediaStorageDataFiles()

```
static unsigned int gdcm::Testing::GetNumberOfMediaStorageDataFiles ( )  [static]
```

10.315.4.18 GetPixelSpacingDataRoot()

```
static const char * gdcm::Testing::GetPixelSpacingDataRoot ( )  [static]
```

Return the GDCM PIXEL SPACING DATA ROOT (See David Clunie website for dataset)

10.315.4.19 GetSelectedPrivateGroupOffsetFromFile()

```
static std::streamoff gdcm::Testing::GetSelectedPrivateGroupOffsetFromFile (
            const char * filepath )  [static]
```

Return the offset just after private attribute (0009,0010,"GEMS_IDEN_01") if found. Otherwise the offset of the next attribute -1 if not found

10.315.4.20 GetSelectedTagsOffsetFromFile()

```
static std::streamoff gdcm::Testing::GetSelectedTagsOffsetFromFile (
            const char * filepath )  [static]
```

Return the offset just after Pixel Data Length (7fe0,0000) if found. Otherwise the offset of the very first pixel cell in Pixel Data -1 if not found

10.315.4.21 GetSourceDirectory()

```
static const char * gdcm::Testing::GetSourceDirectory ( )  [static]
```

Examples

BasicAnonymizer.cs, and ClinicalTrialIdentificationWorkflow.cs.

10.315.4.22 GetStreamOffsetFromFile()

```
static std::streamoff gdcm::Testing::GetStreamOffsetFromFile (
            const char * filepath )  [static]
```

Return the offset of the very first pixel cell in the PixelData -1 if not found

10.315.4.23 GetTempDirectory()

```
static const char * gdcm::Testing::GetTempDirectory (
            const char * subdir = nullptr )  [static]
```

NOT THREAD SAFE Returns the temp directory as used in testing needing to output data:

Examples

MetaImageMD5Activiz.cs.

10.315.4.24 GetTempDirectoryW()

```
static const wchar_t * gdcm::Testing::GetTempDirectoryW (
            const wchar_t * subdir = nullptr )  [static]
```

NOT THREAD SAFE.

10.315.4.25 GetTempFilename()

```
static const char * gdcm::Testing::GetTempFilename (
            const char * filename,
            const char * subdir = nullptr )  [static]
```

NOT THREAD SAFE.

Examples

MetaImageMD5Activiz.cs.

10.315.4.26 GetTempFilenameW()

```
static const wchar_t * gdcm::Testing::GetTempFilenameW (
            const wchar_t * filename,
            const wchar_t * subdir = nullptr )  [static]
```

NOT THREAD SAFE.

10.315.4.27 Print()

```
void gdcm::Testing::Print (
            std::ostream & os = std::cout )
```

Print.

The documentation for this class was generated from the following file:

- gdcmTesting.h

10.316 gdcm::Trace Class Reference

Trace.

`#include <gdcmTrace.h>`

Public Member Functions

- Trace ()
- ~Trace ()

Static Public Member Functions

- static void DebugOff ()
- static void DebugOn ()
- static void ErrorOff ()
- static void ErrorOn ()
- static bool GetDebugFlag ()
- static std::ostream & GetDebugStream ()
- static bool GetErrorFlag ()
- static std::ostream & GetErrorStream ()
- static std::ostream & GetStream ()
- static bool GetWarningFlag ()
- static std::ostream & GetWarningStream ()
- static void SetDebug (bool debug)

  Turn debug messages on (default: false)
- static void SetDebugStream (std::ostream &os)

  Explicitly set the stream which receive Debug messages:
- static void SetError (bool debug)

  Turn error messages on (default: true)
- static void SetErrorStream (std::ostream &os)

  Explicitly set the stream which receive Error messages:
- static void SetStream (std::ostream &os)
- static void SetStreamToFile (const char *filename)
- static void SetWarning (bool debug)

  Turn warning messages on (default: true)
- static void SetWarningStream (std::ostream &os)

  Explicitly set the stream which receive Warning messages:
- static void WarningOff ()
- static void WarningOn ()

10.316.1 Detailed Description

Trace.

Debug / Warning and Error are encapsulated in this class by default the Trace class will redirect any debug/warning/error to std::cerr. Unless SetStream was specified with another (open) stream or SetStreamToFile was specified to a writable file on the system.

Warning

All string messages are removed during compilation time when compiled with CMAKE_BUILD_TYPE being set to either:

- Release
- MinSizeRel It is recommended to compile with RelWithDebInfo and/or Debug during prototyping of applications.

Examples

DecompressJPEGFile.cs.

10.316.2 Constructor & Destructor Documentation

10.316.2.1 Trace()

```
gdcm::Trace::Trace ( )
```

10.316.2.2 ~Trace()

```
gdcm::Trace::~Trace ( )
```

10.316.3 Member Function Documentation

10.316.3.1 DebugOff()

```
static void gdcm::Trace::DebugOff ( )  [static]
```

Examples

MetaImageMD5Activiz.cs, and TestReader.cxx.

10.316.3.2 DebugOn()

```
static void gdcm::Trace::DebugOn ( )  [static]
```

Examples

CreateFakePET.cxx, DecompressJPEGFile.cs, Fake_Image_Using_Stream_Image_Writer.cxx, and StreamImageReaderTest.cxx.

10.316.3.3 ErrorOff()

```
static void gdcm::Trace::ErrorOff ( )  [static]
```

Examples

MetaImageMD5Activiz.cs.

10.316.3.4 ErrorOn()

```
static void gdcm::Trace::ErrorOn ( )  [static]
```

10.316.3.5 GetDebugFlag()

```
static bool gdcm::Trace::GetDebugFlag ( )  [static]
```

10.316.3.6 GetDebugStream()

```
static std::ostream & gdcm::Trace::GetDebugStream ( )  [static]
```

10.316.3.7 GetErrorFlag()

```
static bool gdcm::Trace::GetErrorFlag ( )  [static]
```

10.316.3.8 GetErrorStream()

```
static std::ostream & gdcm::Trace::GetErrorStream ( )  [static]
```

10.316.3.9 GetStream()

```
static std::ostream & gdcm::Trace::GetStream ( )  [static]
```

10.316.3.10 GetWarningFlag()

```
static bool gdcm::Trace::GetWarningFlag ( )  [static]
```

10.316.3.11 GetWarningStream()

```
static std::ostream & gdcm::Trace::GetWarningStream ( )  [static]
```

10.316.3.12 SetDebug()

```
static void gdcm::Trace::SetDebug (
            bool debug )  [static]
```

Turn debug messages on (default: false)

Examples

DumpToSQLITE3.cxx.

10.316.3.13 SetDebugStream()

```
static void gdcm::Trace::SetDebugStream (
            std::ostream & os )  [static]
```

Explicitly set the stream which receive Debug messages:

10.316.3.14 SetError()

```
static void gdcm::Trace::SetError (
            bool debug )  [static]
```

Turn error messages on (default: true)

10.316.3.15 SetErrorStream()

```
static void gdcm::Trace::SetErrorStream (
            std::ostream & os )  [static]
```

Explicitly set the stream which receive Error messages:

Examples

CStoreQtProgress.cxx.

10.316.3.16 SetStream()

```
static void gdcm::Trace::SetStream (
            std::ostream & os )  [static]
```

Explicitly set the ostream for gdcm::Trace to report to This will set the DebugStream, WarningStream and ErrorStream at once:

10.316.3.17 SetStreamToFile()

```
static void gdcm::Trace::SetStreamToFile (
            const char * filename )  [static]
```

Explicitly set the filename for gdcm::Trace to report to The file will be created (it will not append to existing file)

10.316.3.18 SetWarning()

```
static void gdcm::Trace::SetWarning (
            bool debug )  [static]
```

Turn warning messages on (default: true)

Examples

DumpToSQLITE3.cxx.

10.316.3.19 SetWarningStream()

```
static void gdcm::Trace::SetWarningStream (
            std::ostream & os )  [static]
```

Explicitly set the stream which receive Warning messages:

10.316.3.20 WarningOff()

```
static void gdcm::Trace::WarningOff ( )  [static]
```

Examples

MetaImageMD5Activiz.cs, and TestReader.cxx.

10.316.3.21 WarningOn()

```
static void gdcm::Trace::WarningOn ( )  [static]
```

Examples

Fake_Image_Using_Stream_Image_Writer.cxx, and StreamImageReaderTest.cxx.

The documentation for this class was generated from the following file:

- gdcmTrace.h

10.317 gdcm::TransferSyntax Class Reference

Class to manipulate Transfer Syntax.

```
#include <gdcmTransferSyntax.h>
```

Public Types

- enum NegociatedType {
  Unknown = 0 ,
  Explicit ,
  Implicit }
- enum TSType {
  ImplicitVRLittleEndian = 0 ,
  ImplicitVRBigEndianPrivateGE ,
  ExplicitVRLittleEndian ,
  DeflatedExplicitVRLittleEndian ,
  ExplicitVRBigEndian ,
  JPEGBaselineProcess1 ,
  JPEGExtendedProcess2_4 ,
  JPEGExtendedProcess3_5 ,
  JPEGSpectralSelectionProcess6_8 ,
  JPEGFullProgressionProcess10_12 ,
  JPEGLosslessProcess14 ,
  JPEGLosslessProcess14_1 ,
  JPEGLSLossless ,
  JPEGLSNearLossless ,
  JPEG2000Lossless ,
  JPEG2000 ,
  JPEG2000Part2Lossless ,
  JPEG2000Part2 ,
  RLELossless ,
  MPEG2MainProfile ,
  ImplicitVRBigEndianACRNEMA ,
  WeirdPapryus ,
  CT_private_ELE ,
  JPIPReferenced ,
  MPEG2MainProfileHighLevel ,
  MPEG4AVCH264HighProfileLevel4_1 ,
  MPEG4AVCH264BDcompatibleHighProfileLevel4_1 ,
  TS_END }

Public Member Functions

- TransferSyntax (TSType type=ImplicitVRLittleEndian)
- bool CanStoreLossy () const
- NegociatedType GetNegociatedType () const
- const char * GetString () const
- SwapCode GetSwapCode () const
- bool IsEncapsulated () const
- bool IsEncoded () const
- bool IsExplicit () const
- bool IsImplicit () const
- bool IsLossless () const
- bool IsLossy () const
- bool IsValid () const
- operator TSType () const

Static Public Member Functions

- static const char * GetTSString (TSType ts)
- static TSType GetTSType (const char *str)

Friends

- std::ostream & operator<< (std::ostream &os, const TransferSyntax &ts)

10.317.1 Detailed Description

Class to manipulate Transfer Syntax.

Note

TRANSFER SYNTAX (Standard and Private): A set of encoding rules that allow Application Entities to unambiguously negotiate the encoding techniques (e.g., Data Element structure, byte ordering, compression) they are able to support, thereby allowing these Application Entities to communicate.

Todo : The implementation is completely retarded -> see gdcm::UIDs for a replacement We need: IsSupported We need preprocess of raw/xml file We need GetFullName()

Need a notion of Private Syntax. As defined in PS 3.5. Section 9.2

See also

UIDs

Examples

BasicImageAnonymizer.cs, CompressLossyJPEG.cs, DecompressImageMultiframe.cs, DecompressJPEGFile.cs, FileChangeTS.cs, FileChangeTSLossy.cs, GetJPEGSamplePrecision.cxx, LargeVRDSExplicit.cxx, MakeTemplate.cxx, MpegVideoInfo.cs, and StandardizeFiles.cs.

10.317.2 Member Enumeration Documentation

10.317.2.1 NegociatedType

`enum gdcm::TransferSyntax::NegociatedType`

Enumerator

Unknown	
Explicit	
Implicit	

10.317.2.2 TSType

enum gdcm::TransferSyntax::TSType

Enumerator

ImplicitVRLittleEndian	
ImplicitVRBigEndianPrivateGE	
ExplicitVRLittleEndian	
DeflatedExplicitVRLittleEndian	
ExplicitVRBigEndian	
JPEGBaselineProcess1	
JPEGExtendedProcess2_4	
JPEGExtendedProcess3_5	
JPEGSpectralSelectionProcess6_8	
JPEGFullProgressionProcess10_12	
JPEGLosslessProcess14	
JPEGLosslessProcess14_1	
JPEGLSLossless	
JPEGLSNearLossless	
JPEG2000Lossless	
JPEG2000	
JPEG2000Part2Lossless	
JPEG2000Part2	
RLELossless	
MPEG2MainProfile	
ImplicitVRBigEndianACRNEMA	
WeirdPapryus	
CT_private_ELE	
JPIPReferenced	
MPEG2MainProfileHighLevel	
MPEG4AVCH264HighProfileLevel4_1	
MPEG4AVCH264BDcompatibleHighProfileLevel4_1	
TS_END	

Examples

BasicImageAnonymizer.cs, CompressLossyJPEG.cs, DecompressImageMultiframe.cs, DecompressJPEGFile.cs, FileChangeTS.cs, FileChangeTSLossy.cs, MpegVideoInfo.cs, and StandardizeFiles.cs.

10.317.3 Constructor & Destructor Documentation

10.317.3.1 TransferSyntax()

```
gdcm::TransferSyntax::TransferSyntax (
            TSType type = ImplicitVRLittleEndian )  [inline]
```

10.317.4 Member Function Documentation

10.317.4.1 CanStoreLossy()

```
bool gdcm::TransferSyntax::CanStoreLossy ( ) const
```

return true if TransFer Syntax Allow storing of Lossy Pixel Data

10.317.4.2 GetNegociatedType()

```
NegociatedType gdcm::TransferSyntax::GetNegociatedType ( ) const
```

10.317.4.3 GetString()

```
const char * gdcm::TransferSyntax::GetString ( ) const  [inline]
```

References GetTSString().

10.317.4.4 GetSwapCode()

```
SwapCode gdcm::TransferSyntax::GetSwapCode ( ) const
```

Deprecated Return the SwapCode associated with the Transfer Syntax. Be careful with the special GE private syntax the DataSet is written in little endian but the Pixel Data is in Big Endian.

10.317.4.5 GetTSString()

```
static const char * gdcm::TransferSyntax::GetTSString (
            TSType ts )  [static]
```

Examples

LargeVRDSExplicit.cxx.

Referenced by GetString().

10.317.4.6 GetTSType()

```
static TSType gdcm::TransferSyntax::GetTSType (
            const char * str )  [static]
```

10.317.4.7 IsEncapsulated()

```
bool gdcm::TransferSyntax::IsEncapsulated ( ) const
```

Examples

ExtractIconFromFile.cxx.

10.317.4.8 IsEncoded()

```
bool gdcm::TransferSyntax::IsEncoded ( ) const
```

10.317.4.9 IsExplicit()

```
bool gdcm::TransferSyntax::IsExplicit ( ) const
```

10.317.4.10 IsImplicit()

```
bool gdcm::TransferSyntax::IsImplicit ( ) const
```

10.317.4.11 IsLossless()

```
bool gdcm::TransferSyntax::IsLossless ( ) const
```

Return true if the transfer syntax algorithm is a lossless algorithm

10.317.4.12 IsLossy()

```
bool gdcm::TransferSyntax::IsLossy ( ) const
```

Return true if the transfer syntax algorithm is a lossy algorithm

10.317.4.13 IsValid()

```
bool gdcm::TransferSyntax::IsValid ( ) const  [inline]
```

10.317.4.14 operator TSType()

```
gdcm::TransferSyntax::operator TSType ( ) const  [inline]
```

10.317.5 Friends And Related Function Documentation

10.317.5.1 operator<<

```
std::ostream & operator<< (
            std::ostream & os,
            const TransferSyntax & ts )  [friend]
```

The documentation for this class was generated from the following file:

- gdcmTransferSyntax.h

10.318 gdcm::network::TransferSyntaxSub Class Reference

TransferSyntaxSub.

`#include <gdcmTransferSyntaxSub.h>`

Public Member Functions

- TransferSyntaxSub ()
- const char * GetName () const
- bool operator== (const TransferSyntaxSub &ts) const
- void Print (std::ostream &os) const
- std::istream & Read (std::istream &is)
- void SetName (const char *name)
- void SetNameFromUID (UIDs::TSName tsname)
- size_t Size () const
- const std::ostream & Write (std::ostream &os) const

10.318.1 Detailed Description

TransferSyntaxSub.

Table 9-15 TRANSFER SYNTAX SUB-ITEM FIELDS

TODO what is the goal of :

Table 9-19 TRANSFER SYNTAX SUB-ITEM FIELDS

10.318.2 Constructor & Destructor Documentation

10.318.2.1 TransferSyntaxSub()

`gdcm::network::TransferSyntaxSub::TransferSyntaxSub ( )`

10.318.3 Member Function Documentation

10.318.3.1 GetName()

```
const char * gdcm::network::TransferSyntaxSub::GetName ( ) const  [inline]
```

10.318.3.2 operator==()

```
bool gdcm::network::TransferSyntaxSub::operator== (
             const TransferSyntaxSub & ts ) const  [inline]
```

10.318.3.3 Print()

```
void gdcm::network::TransferSyntaxSub::Print (
             std::ostream & os ) const
```

10.318.3.4 Read()

```
std::istream & gdcm::network::TransferSyntaxSub::Read (
             std::istream & is )
```

10.318.3.5 SetName()

```
void gdcm::network::TransferSyntaxSub::SetName (
             const char * name )
```

10.318.3.6 SetNameFromUID()

```
void gdcm::network::TransferSyntaxSub::SetNameFromUID (
             UIDs::TSName tsname )
```

10.318.3.7 Size()

```
size_t gdcm::network::TransferSyntaxSub::Size ( ) const
```

10.318.3.8 Write()

```
const std::ostream & gdcm::network::TransferSyntaxSub::Write (
            std::ostream & os ) const
```

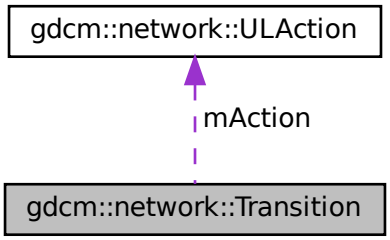
The documentation for this class was generated from the following file:

- gdcmTransferSyntaxSub.h

10.319 gdcm::network::Transition Struct Reference

```
#include <gdcmULTransitionTable.h>
```

Collaboration diagram for gdcm::network::Transition:



Public Member Functions

- Transition ()
- Transition (int inEndState, ULAction ∗inAction)
- ∼Transition ()

Static Public Member Functions

- static Transition ∗ MakeNew (int inEndState, ULAction ∗inAction)

Public Attributes

- ULAction * mAction
- int mEnd

10.319.1 Constructor & Destructor Documentation

10.319.1.1 Transition() [1/2]

```
gdcm::network::Transition::Transition ( )  [inline]
```

References gdcm::network::eStaDoesNotExist, mAction, and mEnd.

Referenced by MakeNew().

10.319.1.2 ~Transition()

```
gdcm::network::Transition::~Transition ( )  [inline]
```

References mAction.

10.319.1.3 Transition() [2/2]

```
gdcm::network::Transition::Transition (
            int inEndState,
            ULAction * inAction )  [inline]
```

References mAction, and mEnd.

10.319.2 Member Function Documentation

10.319.2.1 MakeNew()

```
static Transition * gdcm::network::Transition::MakeNew (
            int inEndState,
            ULAction * inAction )  [inline], [static]
```

References Transition().

10.319.3 Member Data Documentation

10.319.3.1 mAction

```
ULAction* gdcm::network::Transition::mAction
```

Referenced by Transition(), and ~Transition().

10.319.3.2 mEnd

```
int gdcm::network::Transition::mEnd
```

Referenced by Transition().

The documentation for this struct was generated from the following file:

- gdcmULTransitionTable.h

10.320 gdcm::Type Class Reference

Type.

```
#include <gdcmType.h>
```

Public Types

- enum TypeType {
  T1 = 0 ,
  T1C ,
  T2 ,
  T2C ,
  T3 ,
  UNKNOWN }

Public Member Functions

- Type (TypeType type=UNKNOWN)
- operator TypeType () const

Static Public Member Functions

- static const char * GetTypeString (TypeType type)
- static TypeType GetTypeType (const char *type)

Friends

- std::ostream & operator<< (std::ostream &os, const Type &vr)

10.320.1 Detailed Description

Type.

Note

PS 3.5 7.4 DATA ELEMENT TYPE 7.4.1 TYPE 1 REQUIRED DATA ELEMENTS 7.4.2 TYPE 1C CONDITIONAL DATA ELEMENTS 7.4.3 TYPE 2 REQUIRED DATA ELEMENTS 7.4.4 TYPE 2C CONDITIONAL DATA ELEMENTS 7.4.5 TYPE 3 OPTIONAL DATA ELEMENTS

The intent of Type 2 Data Elements is to allow a zero length to be conveyed when the operator or application does not know its value or has a specific reason for not specifying its value. It is the intent that the device should support these Data Elements.

Examples

TraverseModules.cxx.

10.320.2 Member Enumeration Documentation

10.320.2.1 TypeType

`enum gdcm::Type::TypeType`

Enumerator

T1	
T1C	
T2	
T2C	
T3	
UNKNOWN	

10.320.3 Constructor & Destructor Documentation

10.320.3.1 Type()

```
gdcm::Type::Type (
            TypeType type = UNKNOWN )  [inline]
```

10.320.4 Member Function Documentation

10.320.4.1 GetTypeString()

```
static const char * gdcm::Type::GetTypeString (
            TypeType type )  [static]
```

10.320.4.2 GetTypeType()

```
static TypeType gdcm::Type::GetTypeType (
            const char * type )  [static]
```

Referenced by gdcm::ModuleEntry::ModuleEntry().

10.320.4.3 operator TypeType()

```
gdcm::Type::operator TypeType ( ) const  [inline]
```

10.320.5 Friends And Related Function Documentation

10.320.5.1 operator<<

```
std::ostream & operator<< (
            std::ostream & os,
            const Type & vr )  [friend]
```

The documentation for this class was generated from the following file:

- gdcmType.h

10.321 gdcm::UI Struct Reference

`#include <gdcmVR.h>`

Public Attributes

- char Internal [64+1]

Friends

- std::ostream & operator<< (std::ostream &_os, const UI &_val)

10.321.1 Friends And Related Function Documentation

10.321.1.1 operator<<

```
std::ostream & operator<< (
            std::ostream & _os,
            const UI & _val )  [friend]
```

10.321.2 Member Data Documentation

10.321.2.1 Internal

```
char gdcm::UI::Internal[64+1]
```

The documentation for this struct was generated from the following file:

- gdcmVR.h

10.322 gdcm::UIDGenerator Class Reference

Class for generating unique UID.

`#include <gdcmUIDGenerator.h>`

Public Member Functions

- UIDGenerator ()

  By default the root of a UID is a GDCM Root...
- const char * Generate ()

Static Public Member Functions

- static const char * GetGDCMUID ()

  Return the default (GDCM) root UID:
- static const char * GetRoot ()
- static bool IsValid (const char *uid)
- static void SetRoot (const char *root)

Static Protected Member Functions

- static bool GenerateUUID (unsigned char *uuid_data)

10.322.1 Detailed Description

Class for generating unique UID.

Note

bla Usage: When constructing a Series or Study UID, user *has* to keep around the UID, otherwise the UID Generator will simply forget the value and create a new UID.

Examples

CreateJPIPDataSet.cxx, EncapsulateFileInRawData.cxx, Extracting_All_Resolution.cxx, Fake_Image_Using_Stream_Image_Writer
GenAllVR.cxx, GenFakeIdentifyFile.cxx, GenFakeImage.cxx, GetSubSequenceData.cxx, ManipulateFile.cs,
MpegVideoInfo.cs, ReformatFile.cs, StreamImageReaderTest.cxx, TemplateEmptyImage.cxx, and uid_unique.cxx.

10.322.2 Constructor & Destructor Documentation

10.322.2.1 UIDGenerator()

```
gdcm::UIDGenerator::UIDGenerator ( )  [inline]
```

By default the root of a UID is a GDCM Root...

10.322.3 Member Function Documentation

10.322.3.1 Generate()

```
const char * gdcm::UIDGenerator::Generate ( )
```

Internally uses a std::string, so two calls have the same pointer ! save into a std::string In summary do not write code like that: const char *uid1 = uid.Generate(); const char *uid2 = uid.Generate(); since uid1 == uid2

Examples

CreateJPIPDataSet.cxx, EncapsulateFileInRawData.cxx, Extracting_All_Resolution.cxx, Fake_Image_Using_Stream_Image_Writer GenAllVR.cxx, GenFakeIdentifyFile.cxx, GenFakeImage.cxx, GetSubSequenceData.cxx, ManipulateFile.cs, ReformatFile.cs, StreamImageReaderTest.cxx, TemplateEmptyImage.cxx, and uid_unique.cxx.

10.322.3.2 GenerateUUID()

```
static bool gdcm::UIDGenerator::GenerateUUID (
            unsigned char * uuid_data )  [static], [protected]
```

10.322.3.3 GetGDCMUID()

```
static const char * gdcm::UIDGenerator::GetGDCMUID ( )  [static]
```

Return the default (GDCM) root UID:

10.322.3.4 GetRoot()

```
static const char * gdcm::UIDGenerator::GetRoot ( )  [static]
```

Examples

ClinicalTrialIdentificationWorkflow.cs, ReformatFile.cs, and StandardizeFiles.cs.

10.322.3.5 IsValid()

```
static bool gdcm::UIDGenerator::IsValid (
            const char * uid )  [static]
```

Find out if the string is a valid UID or not

Todo : Move that in DataStructureAndEncoding (see FileMetaInformation::CheckFileMetaInformation)

10.322.3.6 SetRoot()

```
static void gdcm::UIDGenerator::SetRoot (
            const char * root )  [static]
```

The current implementation in GDCM make use of the UUID implementation (RFC 4122) and has been successfully been tested for a root of size 26 bytes. Any longer root should work (the Generate() function will return a string), but will truncate the high bits of the 128bits UUID until the generated string fits on 64 bits. The authors disclaims any responsabitlity for garanteeing uniqueness of UIDs when the root is longer than 26 bytes.

Examples

ClinicalTrialIdentificationWorkflow.cs, ReformatFile.cs, StandardizeFiles.cs, and uid_unique.cxx.

The documentation for this class was generated from the following file:

- gdcmUIDGenerator.h

10.323 gdcm::UIDs Class Reference

all known uids

```
#include <gdcmUIDs.h>
```

Public Types

- typedef const char *const (* TransferSyntaxStringsType)[2]
- enum TSName {
  VerificationSOPClass = 1 ,
  ImplicitVRLittleEndianDefaultTransferSyntaxforDICOM = 2 ,
  ExplicitVRLittleEndian = 3 ,
  DeflatedExplicitVRLittleEndian = 4 ,
  ExplicitVRBigEndian = 5 ,
  JPEGBaselineProcess1DefaultTransferSyntaxforLossyJPEG8BitImageCompression = 6 ,
  JPEGExtendedProcess24DefaultTransferSyntaxforLossyJPEG12BitImageCompressionProcess4only = 7 ,
  JPEGExtendedProcess35Retired = 8 ,
  JPEGSpectralSelectionNonHierarchicalProcess68Retired = 9 ,
  JPEGSpectralSelectionNonHierarchicalProcess79Retired = 10 ,
  JPEGFullProgressionNonHierarchicalProcess1012Retired = 11 ,
  JPEGFullProgressionNonHierarchicalProcess1113Retired = 12 ,
  JPEGLosslessNonHierarchicalProcess14 = 13 ,
  JPEGLosslessNonHierarchicalProcess15Retired = 14 ,
  JPEGExtendedHierarchicalProcess1618Retired = 15 ,
  JPEGExtendedHierarchicalProcess1719Retired = 16 ,
  JPEGSpectralSelectionHierarchicalProcess2022Retired = 17 ,
  JPEGSpectralSelectionHierarchicalProcess2123Retired = 18 ,
  JPEGFullProgressionHierarchicalProcess2426Retired = 19 ,
  JPEGFullProgressionHierarchicalProcess2527Retired = 20 ,
  JPEGLosslessHierarchicalProcess28Retired = 21 ,
  JPEGLosslessHierarchicalProcess29Retired = 22 ,
  JPEGLosslessNonHierarchicalFirstOrderPredictionProcess14SelectionValue1DefaultTransferSyntaxforLosslessJPEGImageCompr
  = 23 ,
  JPEGLSLosslessImageCompression = 24 ,
  JPEGLSLossyNearLosslessImageCompression = 25 ,
  JPEG2000ImageCompressionLosslessOnly = 26 ,
  JPEG2000ImageCompression = 27 ,
  JPEG2000Part2MulticomponentImageCompressionLosslessOnly = 28 ,
  JPEG2000Part2MulticomponentImageCompression = 29 ,
  JPIPReferenced = 30 ,
  JPIPReferencedDeflate = 31 ,
  MPEG2MainProfileMainLevel = 32 ,
  RLELossless = 33 ,
  RFC2557MIMEencapsulation = 34 ,
  XMLEncoding = 35 ,
  MediaStorageDirectoryStorage = 36 ,
  TalairachBrainAtlasFrameofReference = 37 ,
  SPM2T1FrameofReference = 38 ,
  SPM2T2FrameofReference = 39 ,
  SPM2PDFrameofReference = 40 ,
  SPM2EPIFrameofReference = 41 ,
  SPM2FILT1FrameofReference = 42 ,
  SPM2PETFrameofReference = 43 ,
  SPM2TRANSMFrameofReference = 44 ,
  SPM2SPECTFrameofReference = 45 ,
  SPM2GRAYFrameofReference = 46 ,
  SPM2WHITEFrameofReference = 47 ,
  SPM2CSFFrameofReference = 48 ,
  SPM2BRAINMASKFrameofReference = 49 ,

SPM2AVG305T1FrameofReference = 50 ,
SPM2AVG152T1FrameofReference = 51 ,
SPM2AVG152T2FrameofReference = 52 ,
SPM2AVG152PDFrameofReference = 53 ,
SPM2SINGLESUBJT1FrameofReference = 54 ,
ICBM452T1FrameofReference = 55 ,
ICBMSingleSubjectMRIFrameofReference = 56 ,
BasicStudyContentNotificationSOPClassRetired = 57 ,
StorageCommitmentPushModelSOPClass = 58 ,
StorageCommitmentPushModelSOPInstance = 59 ,
StorageCommitmentPullModelSOPClassRetired = 60 ,
StorageCommitmentPullModelSOPInstanceRetired = 61 ,
ProceduralEventLoggingSOPClass = 62 ,
ProceduralEventLoggingSOPInstance = 63 ,
SubstanceAdministrationLoggingSOPClass = 64 ,
SubstanceAdministrationLoggingSOPInstance = 65 ,
DICOMUIDRegistry = 66 ,
DICOMControlledTerminology = 67 ,
DICOMApplicationContextName = 68 ,
DetachedPatientManagementSOPClassRetired = 69 ,
DetachedPatientManagementMetaSOPClassRetired = 70 ,
DetachedVisitManagementSOPClassRetired = 71 ,
DetachedStudyManagementSOPClassRetired = 72 ,
StudyComponentManagementSOPClassRetired = 73 ,
ModalityPerformedProcedureStepSOPClass = 74 ,
ModalityPerformedProcedureStepRetrieveSOPClass = 75 ,
ModalityPerformedProcedureStepNotificationSOPClass = 76 ,
DetachedResultsManagementSOPClassRetired = 77 ,
DetachedResultsManagementMetaSOPClassRetired = 78 ,
DetachedStudyManagementMetaSOPClassRetired = 79 ,
DetachedInterpretationManagementSOPClassRetired = 80 ,
StorageServiceClass = 81 ,
BasicFilmSessionSOPClass = 82 ,
BasicFilmBoxSOPClass = 83 ,
BasicGrayscaleImageBoxSOPClass = 84 ,
BasicColorImageBoxSOPClass = 85 ,
ReferencedImageBoxSOPClassRetired = 86 ,
BasicGrayscalePrintManagementMetaSOPClass = 87 ,
ReferencedGrayscalePrintManagementMetaSOPClassRetired = 88 ,
PrintJobSOPClass = 89 ,
BasicAnnotationBoxSOPClass = 90 ,
PrinterSOPClass = 91 ,
PrinterConfigurationRetrievalSOPClass = 92 ,
PrinterSOPInstance = 93 ,
PrinterConfigurationRetrievalSOPInstance = 94 ,
BasicColorPrintManagementMetaSOPClass = 95 ,
ReferencedColorPrintManagementMetaSOPClassRetired = 96 ,
VOILUTBoxSOPClass = 97 ,
PresentationLUTSOPClass = 98 ,
ImageOverlayBoxSOPClassRetired = 99 ,
BasicPrintImageOverlayBoxSOPClassRetired = 100 ,
PrintQueueSOPInstanceRetired = 101 ,
PrintQueueManagementSOPClassRetired = 102 ,
StoredPrintStorageSOPClassRetired = 103 ,

HardcopyGrayscaleImageStorageSOPClassRetired = 104 ,
HardcopyColorImageStorageSOPClassRetired = 105 ,
PullPrintRequestSOPClassRetired = 106 ,
PullStoredPrintManagementMetaSOPClassRetired = 107 ,
MediaCreationManagementSOPClassUID = 108 ,
ComputedRadiographyImageStorage = 109 ,
DigitalXRayImageStorageForPresentation = 110 ,
DigitalXRayImageStorageForProcessing = 111 ,
DigitalMammographyXRayImageStorageForPresentation = 112 ,
DigitalMammographyXRayImageStorageForProcessing = 113 ,
DigitalIntraoralXRayImageStorageForPresentation = 114 ,
DigitalIntraoralXRayImageStorageForProcessing = 115 ,
CTImageStorage = 116 ,
EnhancedCTImageStorage = 117 ,
UltrasoundMultiframeImageStorageRetired = 118 ,
UltrasoundMultiframeImageStorage = 119 ,
MRImageStorage = 120 ,
EnhancedMRImageStorage = 121 ,
MRSpectroscopyStorage = 122 ,
NuclearMedicineImageStorageRetired = 123 ,
UltrasoundImageStorageRetired = 124 ,
UltrasoundImageStorage = 125 ,
SecondaryCaptureImageStorage = 126 ,
MultiframeSingleBitSecondaryCaptureImageStorage = 127 ,
MultiframeGrayscaleByteSecondaryCaptureImageStorage = 128 ,
MultiframeGrayscaleWordSecondaryCaptureImageStorage = 129 ,
MultiframeTrueColorSecondaryCaptureImageStorage = 130 ,
StandaloneOverlayStorageRetired = 131 ,
StandaloneCurveStorageRetired = 132 ,
WaveformStorageTrialRetired = 133 ,
ECG12leadWaveformStorage = 134 ,
GeneralECGWaveformStorage = 135 ,
AmbulatoryECGWaveformStorage = 136 ,
HemodynamicWaveformStorage = 137 ,
CardiacElectrophysiologyWaveformStorage = 138 ,
BasicVoiceAudioWaveformStorage = 139 ,
StandaloneModalityLUTStorageRetired = 140 ,
StandaloneVOILUTStorageRetired = 141 ,
GrayscaleSoftcopyPresentationStateStorageSOPClass = 142 ,
ColorSoftcopyPresentationStateStorageSOPClass = 143 ,
PseudoColorSoftcopyPresentationStateStorageSOPClass = 144 ,
BlendingSoftcopyPresentationStateStorageSOPClass = 145 ,
XRayAngiographicImageStorage = 146 ,
EnhancedXAImageStorage = 147 ,
XRayRadiofluoroscopicImageStorage = 148 ,
EnhancedXRFImageStorage = 149 ,
XRay3DAngiographicImageStorage = 150 ,
XRay3DCraniofacialImageStorage = 151 ,
XRayAngiographicBiPlaneImageStorageRetired = 152 ,
NuclearMedicineImageStorage = 153 ,
RawDataStorage = 154 ,
SpatialRegistrationStorage = 155 ,
SpatialFiducialsStorage = 156 ,
DeformableSpatialRegistrationStorage = 157 ,

SegmentationStorage = 158 ,
RealWorldValueMappingStorage = 159 ,
VLImageStorageTrialRetired = 160 ,
VLMultiframeImageStorageTrialRetired = 161 ,
VLEndoscopicImageStorage = 162 ,
VideoEndoscopicImageStorage = 163 ,
VLMicroscopicImageStorage = 164 ,
VideoMicroscopicImageStorage = 165 ,
VLSlideCoordinatesMicroscopicImageStorage = 166 ,
VLPhotographicImageStorage = 167 ,
VideoPhotographicImageStorage = 168 ,
OphthalmicPhotography8BitImageStorage = 169 ,
OphthalmicPhotography16BitImageStorage = 170 ,
StereometricRelationshipStorage = 171 ,
OphthalmicTomographyImageStorage = 172 ,
TextSRStorageTrialRetired = 173 ,
AudioSRStorageTrialRetired = 174 ,
DetailSRStorageTrialRetired = 175 ,
ComprehensiveSRStorageTrialRetired = 176 ,
BasicTextSRStorage = 177 ,
EnhancedSRStorage = 178 ,
ComprehensiveSRStorage = 179 ,
ProcedureLogStorage = 180 ,
MammographyCADSRStorage = 181 ,
KeyObjectSelectionDocumentStorage = 182 ,
ChestCADSRStorage = 183 ,
XRayRadiationDoseSRStorage = 184 ,
EncapsulatedPDFStorage = 185 ,
EncapsulatedCDAStorage = 186 ,
PositronEmissionTomographyImageStorage = 187 ,
StandalonePETCurveStorageRetired = 188 ,
RTImageStorage = 189 ,
RTDoseStorage = 190 ,
RTStructureSetStorage = 191 ,
RTBeamsTreatmentRecordStorage = 192 ,
RTPlanStorage = 193 ,
RTBrachyTreatmentRecordStorage = 194 ,
RTTreatmentSummaryRecordStorage = 195 ,
RTIonPlanStorage = 196 ,
RTIonBeamsTreatmentRecordStorage = 197 ,
PatientRootQueryRetrieveInformationModelFIND = 198 ,
PatientRootQueryRetrieveInformationModelMOVE = 199 ,
PatientRootQueryRetrieveInformationModelGET = 200 ,
StudyRootQueryRetrieveInformationModelFIND = 201 ,
StudyRootQueryRetrieveInformationModelMOVE = 202 ,
StudyRootQueryRetrieveInformationModelGET = 203 ,
PatientStudyOnlyQueryRetrieveInformationModelFINDRetired = 204 ,
PatientStudyOnlyQueryRetrieveInformationModelMOVERetired = 205 ,
PatientStudyOnlyQueryRetrieveInformationModelGETRetired = 206 ,
ModalityWorklistInformationModelFIND = 207 ,
GeneralPurposeWorklistInformationModelFIND = 208 ,
GeneralPurposeScheduledProcedureStepSOPClass = 209 ,
GeneralPurposePerformedProcedureStepSOPClass = 210 ,
GeneralPurposeWorklistManagementMetaSOPClass = 211 ,

InstanceAvailabilityNotificationSOPClass = 212 ,
RTBeamsDeliveryInstructionStorageSupplement74FrozenDraft = 213 ,
RTConventionalMachineVerificationSupplement74FrozenDraft = 214 ,
RTIonMachineVerificationSupplement74FrozenDraft = 215 ,
UnifiedWorklistandProcedureStepServiceClass = 216 ,
UnifiedProcedureStepPushSOPClass = 217 ,
UnifiedProcedureStepWatchSOPClass = 218 ,
UnifiedProcedureStepPullSOPClass = 219 ,
UnifiedProcedureStepEventSOPClass = 220 ,
UnifiedWorklistandProcedureStepSOPInstance = 221 ,
GeneralRelevantPatientInformationQuery = 222 ,
BreastImagingRelevantPatientInformationQuery = 223 ,
CardiacRelevantPatientInformationQuery = 224 ,
HangingProtocolStorage = 225 ,
HangingProtocolInformationModelFIND = 226 ,
HangingProtocolInformationModelMOVE = 227 ,
ProductCharacteristicsQuerySOPClass = 228 ,
SubstanceApprovalQuerySOPClass = 229 ,
dicomDeviceName = 230 ,
dicomDescription = 231 ,
dicomManufacturer = 232 ,
dicomManufacturerModelName = 233 ,
dicomSoftwareVersion = 234 ,
dicomVendorData = 235 ,
dicomAETitle = 236 ,
dicomNetworkConnectionReference = 237 ,
dicomApplicationCluster = 238 ,
dicomAssociationInitiator = 239 ,
dicomAssociationAcceptor = 240 ,
dicomHostname = 241 ,
dicomPort = 242 ,
dicomSOPClass = 243 ,
dicomTransferRole = 244 ,
dicomTransferSyntax = 245 ,
dicomPrimaryDeviceType = 246 ,
dicomRelatedDeviceReference = 247 ,
dicomPreferredCalledAETitle = 248 ,
dicomTLSCyphersuite = 249 ,
dicomAuthorizedNodeCertificateReference = 250 ,
dicomThisNodeCertificateReference = 251 ,
dicomInstalled = 252 ,
dicomStationName = 253 ,
dicomDeviceSerialNumber = 254 ,
dicomInstitutionName = 255 ,
dicomInstitutionAddress = 256 ,
dicomInstitutionDepartmentName = 257 ,
dicomIssuerOfPatientID = 258 ,
dicomPreferredCallingAETitle = 259 ,
dicomSupportedCharacterSet = 260 ,
dicomConfigurationRoot = 261 ,
dicomDevicesRoot = 262 ,
dicomUniqueAETitlesRegistryRoot = 263 ,
dicomDevice = 264 ,
dicomNetworkAE = 265 ,

dicomNetworkConnection = 266 ,
dicomUniqueAETitle = 267 ,
dicomTransferCapability = 268 ,
VLWholeSlideMicroscopyImageStorage = 269 ,
EnhancedUSVolumeStorage = 270 ,
SurfaceSegmentationStorage = 271 ,
BreastTomosynthesisImageStorage = 272 ,
LegacyConvertedEnhancedCTImageStorage = 273 ,
LegacyConvertedEnhancedMRImageStorage = 274 ,
LegacyConvertedEnhancedPETImageStorage = 275 ,
MPEG2MainProfileHighLevel = 276 ,
MPEG4AVCH_264HighProfileLevel4_1 = 277 ,
MPEG4AVCH_264BDcompatibleHighProfileLevel4_1 = 278 ,
PETColorPaletteSOPInstance = 279 ,
HotMetalBlueColorPaletteSOPInstance = 280 ,
PET20StepColorPaletteSOPInstance = 281 ,
SpringColorPaletteSOPInstance = 282 ,
SummerColorPaletteSOPInstance = 283 ,
FallColorPaletteSOPInstance = 284 ,
WinterColorPaletteSOPInstance = 285 ,
Papyrus3ImplicitVRLittleEndian = 286 ,
AdultMouseAnatomyOntology = 287 ,
UberonOntology = 288 ,
IntegratedTaxonomicInformationSystemITISTaxonomicSerialNumberTSN = 289 ,
MouseGenomeInitiativeMGI = 290 ,
PubChemCompoundCID = 291 ,
ICD11 = 292 ,
NewYorkUniversityMelanomaClinicalCooperativeGroup = 293 ,
MayoClinicNonradiologicalImagesSBSAnatomicalSurfaceRegionGuide = 294 ,
ImageBiomarkerStandardisationInitiative = 295 ,
RadiomicsOntology = 296 ,
DisplaySystemSOPClass = 297 ,
DisplaySystemSOPInstance = 298 ,
GeneralAudioWaveformStorage = 299 ,
ArterialPulseWaveformStorage = 300 ,
RespiratoryWaveformStorage = 301 ,
XAXRFGrayscaleSoftcopyPresentationStateStorage = 302 ,
GrayscalePlanarMPRVolumetricPresentationStateStorage = 303 ,
MPEG4AVCH_264HighProfileLevel4_2For2DVideo = 304 ,
MPEG4AVCH_264HighProfileLevel4_2For3DVideo = 305 ,
MPEG4AVCH_264StereoHighProfileLevel4_2 = 306 ,
HEVCH_265MainProfileLevel5_1 = 307 ,
HEVCH_265Main10ProfileLevel5_1 = 308 ,
HotIronColorPaletteSOPInstance = 309 ,
CompositingPlanarMPRVolumetricPresentationStateStorage = 310 ,
AdvancedBlendingPresentationStateStorage = 311 ,
VolumeRenderingVolumetricPresentationStateStorage = 312 ,
SegmentedVolumeRenderingVolumetricPresentationStateStorage = 313 ,
MultipleVolumeRenderingVolumetricPresentationStateStorage = 314 ,
Null0 = 315 ,
BreastProjectionXRayImageStorageForPresentation = 316 ,
BreastProjectionXRayImageStorageForProcessing = 317 ,
IntravascularOpticalCoherenceTomographyImageStorageForPresentation = 318 ,
IntravascularOpticalCoherenceTomographyImageStorageForProcessing = 319 ,

ParametricMapStorage = 320 ,
Null1 = 321 ,
TractographyResultsStorage = 322 ,
SurfaceScanMeshStorage = 323 ,
SurfaceScanPointCloudStorage = 324 ,
WideFieldOphthalmicPhotographyStereographicProjectionImageStorage = 325 ,
WideFieldOphthalmicPhotography3DCoordinatesImageStorage = 326 ,
OphthalmicOpticalCoherenceTomographyEnFaceImageStorage = 327 ,
OphthalmicOpticalCoherenceTomographyBscanVolumeAnalysisStorage = 328 ,
LensometryMeasurementsStorage = 329 ,
AutorefractionMeasurementsStorage = 330 ,
KeratometryMeasurementsStorage = 331 ,
SubjectiveRefractionMeasurementsStorage = 332 ,
VisualAcuityMeasurementsStorage = 333 ,
SpectaclePrescriptionReportStorage = 334 ,
OphthalmicAxialMeasurementsStorage = 335 ,
IntraocularLensCalculationsStorage = 336 ,
MacularGridThicknessandVolumeReportStorage = 337 ,
OphthalmicVisualFieldStaticPerimetryMeasurementsStorage = 338 ,
OphthalmicThicknessMapStorage = 339 ,
CornealTopographyMapStorage = 340 ,
Comprehensive3DSRStorage = 341 ,
ExtensibleSRStorage = 342 ,
RadiopharmaceuticalRadiationDoseSRStorage = 343 ,
ColonCADSRStorage = 344 ,
ImplantationPlanSRStorage = 345 ,
AcquisitionContextSRStorage = 346 ,
SimplifiedAdultEchoSRStorage = 347 ,
PatientRadiationDoseSRStorage = 348 ,
PlannedImagingAgentAdministrationSRStorage = 349 ,
PerformedImagingAgentAdministrationSRStorage = 350 ,
ContentAssessmentResultsStorage = 351 ,
EncapsulatedSTLStorage = 352 ,
EnhancedPETImageStorage = 353 ,
BasicStructuredDisplayStorage = 354 ,
CTDefinedProcedureProtocolStorage = 355 ,
CTPerformedProcedureProtocolStorage = 356 ,
ProtocolApprovalStorage = 357 ,
ProtocolApprovalInformationModelFIND = 358 ,
ProtocolApprovalInformationModelMOVE = 359 ,
ProtocolApprovalInformationModelGET = 360 ,
RTPhysicianIntentStorage = 361 ,
RTSegmentAnnotationStorage = 362 ,
DICOSCTImageStorage = 363 ,
DICOSDigitalXRayImageStorageForPresentation = 364 ,
DICOSDigitalXRayImageStorageForProcessing = 365 ,
DICOSThreatDetectionReportStorage = 366 ,
DICOS2DAITStorage = 367 ,
DICOS3DAITStorage = 368 ,
DICOSQuadrupoleResonanceQRStorage = 369 ,
EddyCurrentImageStorage = 370 ,
EddyCurrentMultiframeImageStorage = 371 ,
CompositeInstanceRootRetrieveMOVE = 372 ,
CompositeInstanceRootRetrieveGET = 373 ,

CompositeInstanceRetrieveWithoutBulkDataGET = 374 ,
DefinedProcedureProtocolInformationModelFIND = 375 ,
DefinedProcedureProtocolInformationModelMOVE = 376 ,
DefinedProcedureProtocolInformationModelGET = 377 ,
UPSFilteredGlobalSubscriptionSOPInstance = 378 ,
UnifiedWorklistandProcedureStepServiceClass1 = 379 ,
UnifiedProcedureStepPushSOPClass1 = 380 ,
UnifiedProcedureStepWatchSOPClass1 = 381 ,
UnifiedProcedureStepPullSOPClass1 = 382 ,
UnifiedProcedureStepEventSOPClass1 = 383 ,
RTBeamsDeliveryInstructionStorage = 384 ,
RTConventionalMachineVerification = 385 ,
RTIonMachineVerification = 386 ,
RTBrachyApplicationSetupDeliveryInstructionStorage = 387 ,
HangingProtocolInformationModelGET = 388 ,
ColorPaletteStorage = 389 ,
ColorPaletteQueryRetrieveInformationModelFIND = 390 ,
ColorPaletteQueryRetrieveInformationModelMOVE = 391 ,
ColorPaletteQueryRetrieveInformationModelGET = 392 ,
GenericImplantTemplateStorage = 393 ,
GenericImplantTemplateInformationModelFIND = 394 ,
GenericImplantTemplateInformationModelMOVE = 395 ,
GenericImplantTemplateInformationModelGET = 396 ,
ImplantAssemblyTemplateStorage = 397 ,
ImplantAssemblyTemplateInformationModelFIND = 398 ,
ImplantAssemblyTemplateInformationModelMOVE = 399 ,
ImplantAssemblyTemplateInformationModelGET = 400 ,
ImplantTemplateGroupStorage = 401 ,
ImplantTemplateGroupInformationModelFIND = 402 ,
ImplantTemplateGroupInformationModelMOVE = 403 ,
ImplantTemplateGroupInformationModelGET = 404 ,
NativeDICOMModel = 405 ,
AbstractMultiDimensionalImageModel = 406 ,
DICOMContentMappingResource = 407 ,
EnhancedMRColorImageStorage = 408 ,
UniversalCoordinatedTime = 409 }

- enum TSType {
uid_1_2_840_10008_1_1 = 1 ,
uid_1_2_840_10008_1_2 = 2 ,
uid_1_2_840_10008_1_2_1 = 3 ,
uid_1_2_840_10008_1_2_1_99 = 4 ,
uid_1_2_840_10008_1_2_2 = 5 ,
uid_1_2_840_10008_1_2_4_50 = 6 ,
uid_1_2_840_10008_1_2_4_51 = 7 ,
uid_1_2_840_10008_1_2_4_52 = 8 ,
uid_1_2_840_10008_1_2_4_53 = 9 ,
uid_1_2_840_10008_1_2_4_54 = 10 ,
uid_1_2_840_10008_1_2_4_55 = 11 ,
uid_1_2_840_10008_1_2_4_56 = 12 ,
uid_1_2_840_10008_1_2_4_57 = 13 ,
uid_1_2_840_10008_1_2_4_58 = 14 ,
uid_1_2_840_10008_1_2_4_59 = 15 ,
uid_1_2_840_10008_1_2_4_60 = 16 ,
uid_1_2_840_10008_1_2_4_61 = 17 ,

uid_1_2_840_10008_1_2_4_62 = 18 ,
uid_1_2_840_10008_1_2_4_63 = 19 ,
uid_1_2_840_10008_1_2_4_64 = 20 ,
uid_1_2_840_10008_1_2_4_65 = 21 ,
uid_1_2_840_10008_1_2_4_66 = 22 ,
uid_1_2_840_10008_1_2_4_70 = 23 ,
uid_1_2_840_10008_1_2_4_80 = 24 ,
uid_1_2_840_10008_1_2_4_81 = 25 ,
uid_1_2_840_10008_1_2_4_90 = 26 ,
uid_1_2_840_10008_1_2_4_91 = 27 ,
uid_1_2_840_10008_1_2_4_92 = 28 ,
uid_1_2_840_10008_1_2_4_93 = 29 ,
uid_1_2_840_10008_1_2_4_94 = 30 ,
uid_1_2_840_10008_1_2_4_95 = 31 ,
uid_1_2_840_10008_1_2_4_100 = 32 ,
uid_1_2_840_10008_1_2_5 = 33 ,
uid_1_2_840_10008_1_2_6_1 = 34 ,
uid_1_2_840_10008_1_2_6_2 = 35 ,
uid_1_2_840_10008_1_3_10 = 36 ,
uid_1_2_840_10008_1_4_1_1 = 37 ,
uid_1_2_840_10008_1_4_1_2 = 38 ,
uid_1_2_840_10008_1_4_1_3 = 39 ,
uid_1_2_840_10008_1_4_1_4 = 40 ,
uid_1_2_840_10008_1_4_1_5 = 41 ,
uid_1_2_840_10008_1_4_1_6 = 42 ,
uid_1_2_840_10008_1_4_1_7 = 43 ,
uid_1_2_840_10008_1_4_1_8 = 44 ,
uid_1_2_840_10008_1_4_1_9 = 45 ,
uid_1_2_840_10008_1_4_1_10 = 46 ,
uid_1_2_840_10008_1_4_1_11 = 47 ,
uid_1_2_840_10008_1_4_1_12 = 48 ,
uid_1_2_840_10008_1_4_1_13 = 49 ,
uid_1_2_840_10008_1_4_1_14 = 50 ,
uid_1_2_840_10008_1_4_1_15 = 51 ,
uid_1_2_840_10008_1_4_1_16 = 52 ,
uid_1_2_840_10008_1_4_1_17 = 53 ,
uid_1_2_840_10008_1_4_1_18 = 54 ,
uid_1_2_840_10008_1_4_2_1 = 55 ,
uid_1_2_840_10008_1_4_2_2 = 56 ,
uid_1_2_840_10008_1_9 = 57 ,
uid_1_2_840_10008_1_20_1 = 58 ,
uid_1_2_840_10008_1_20_1_1 = 59 ,
uid_1_2_840_10008_1_20_2 = 60 ,
uid_1_2_840_10008_1_20_2_1 = 61 ,
uid_1_2_840_10008_1_40 = 62 ,
uid_1_2_840_10008_1_40_1 = 63 ,
uid_1_2_840_10008_1_42 = 64 ,
uid_1_2_840_10008_1_42_1 = 65 ,
uid_1_2_840_10008_2_6_1 = 66 ,
uid_1_2_840_10008_2_16_4 = 67 ,
uid_1_2_840_10008_3_1_1_1 = 68 ,
uid_1_2_840_10008_3_1_2_1_1 = 69 ,
uid_1_2_840_10008_3_1_2_1_4 = 70 ,
uid_1_2_840_10008_3_1_2_2_1 = 71 ,

uid_1_2_840_10008_3_1_2_3_1 = 72 ,
uid_1_2_840_10008_3_1_2_3_2 = 73 ,
uid_1_2_840_10008_3_1_2_3_3 = 74 ,
uid_1_2_840_10008_3_1_2_3_4 = 75 ,
uid_1_2_840_10008_3_1_2_3_5 = 76 ,
uid_1_2_840_10008_3_1_2_5_1 = 77 ,
uid_1_2_840_10008_3_1_2_5_4 = 78 ,
uid_1_2_840_10008_3_1_2_5_5 = 79 ,
uid_1_2_840_10008_3_1_2_6_1 = 80 ,
uid_1_2_840_10008_4_2 = 81 ,
uid_1_2_840_10008_5_1_1_1 = 82 ,
uid_1_2_840_10008_5_1_1_2 = 83 ,
uid_1_2_840_10008_5_1_1_4 = 84 ,
uid_1_2_840_10008_5_1_1_4_1 = 85 ,
uid_1_2_840_10008_5_1_1_4_2 = 86 ,
uid_1_2_840_10008_5_1_1_9 = 87 ,
uid_1_2_840_10008_5_1_1_9_1 = 88 ,
uid_1_2_840_10008_5_1_1_14 = 89 ,
uid_1_2_840_10008_5_1_1_15 = 90 ,
uid_1_2_840_10008_5_1_1_16 = 91 ,
uid_1_2_840_10008_5_1_1_16_376 = 92 ,
uid_1_2_840_10008_5_1_1_17 = 93 ,
uid_1_2_840_10008_5_1_1_17_376 = 94 ,
uid_1_2_840_10008_5_1_1_18 = 95 ,
uid_1_2_840_10008_5_1_1_18_1 = 96 ,
uid_1_2_840_10008_5_1_1_22 = 97 ,
uid_1_2_840_10008_5_1_1_23 = 98 ,
uid_1_2_840_10008_5_1_1_24 = 99 ,
uid_1_2_840_10008_5_1_1_24_1 = 100 ,
uid_1_2_840_10008_5_1_1_25 = 101 ,
uid_1_2_840_10008_5_1_1_26 = 102 ,
uid_1_2_840_10008_5_1_1_27 = 103 ,
uid_1_2_840_10008_5_1_1_29 = 104 ,
uid_1_2_840_10008_5_1_1_30 = 105 ,
uid_1_2_840_10008_5_1_1_31 = 106 ,
uid_1_2_840_10008_5_1_1_32 = 107 ,
uid_1_2_840_10008_5_1_1_33 = 108 ,
uid_1_2_840_10008_5_1_4_1_1_1 = 109 ,
uid_1_2_840_10008_5_1_4_1_1_1_1 = 110 ,
uid_1_2_840_10008_5_1_4_1_1_1_1_1 = 111 ,
uid_1_2_840_10008_5_1_4_1_1_1_2 = 112 ,
uid_1_2_840_10008_5_1_4_1_1_1_2_1 = 113 ,
uid_1_2_840_10008_5_1_4_1_1_1_3 = 114 ,
uid_1_2_840_10008_5_1_4_1_1_1_3_1 = 115 ,
uid_1_2_840_10008_5_1_4_1_1_2 = 116 ,
uid_1_2_840_10008_5_1_4_1_1_2_1 = 117 ,
uid_1_2_840_10008_5_1_4_1_1_3 = 118 ,
uid_1_2_840_10008_5_1_4_1_1_3_1 = 119 ,
uid_1_2_840_10008_5_1_4_1_1_4 = 120 ,
uid_1_2_840_10008_5_1_4_1_1_4_1 = 121 ,
uid_1_2_840_10008_5_1_4_1_1_4_2 = 122 ,
uid_1_2_840_10008_5_1_4_1_1_5 = 123 ,
uid_1_2_840_10008_5_1_4_1_1_6 = 124 ,
uid_1_2_840_10008_5_1_4_1_1_6_1 = 125 ,

uid_1_2_840_10008_5_1_4_1_1_7 = 126 ,
uid_1_2_840_10008_5_1_4_1_1_7_1 = 127 ,
uid_1_2_840_10008_5_1_4_1_1_7_2 = 128 ,
uid_1_2_840_10008_5_1_4_1_1_7_3 = 129 ,
uid_1_2_840_10008_5_1_4_1_1_7_4 = 130 ,
uid_1_2_840_10008_5_1_4_1_1_8 = 131 ,
uid_1_2_840_10008_5_1_4_1_1_9 = 132 ,
uid_1_2_840_10008_5_1_4_1_1_9_1 = 133 ,
uid_1_2_840_10008_5_1_4_1_1_9_1_1 = 134 ,
uid_1_2_840_10008_5_1_4_1_1_9_1_2 = 135 ,
uid_1_2_840_10008_5_1_4_1_1_9_1_3 = 136 ,
uid_1_2_840_10008_5_1_4_1_1_9_2_1 = 137 ,
uid_1_2_840_10008_5_1_4_1_1_9_3_1 = 138 ,
uid_1_2_840_10008_5_1_4_1_1_9_4_1 = 139 ,
uid_1_2_840_10008_5_1_4_1_1_10 = 140 ,
uid_1_2_840_10008_5_1_4_1_1_11 = 141 ,
uid_1_2_840_10008_5_1_4_1_1_11_1 = 142 ,
uid_1_2_840_10008_5_1_4_1_1_11_2 = 143 ,
uid_1_2_840_10008_5_1_4_1_1_11_3 = 144 ,
uid_1_2_840_10008_5_1_4_1_1_11_4 = 145 ,
uid_1_2_840_10008_5_1_4_1_1_12_1 = 146 ,
uid_1_2_840_10008_5_1_4_1_1_12_1_1 = 147 ,
uid_1_2_840_10008_5_1_4_1_1_12_2 = 148 ,
uid_1_2_840_10008_5_1_4_1_1_12_2_1 = 149 ,
uid_1_2_840_10008_5_1_4_1_1_13_1_1 = 150 ,
uid_1_2_840_10008_5_1_4_1_1_13_1_2 = 151 ,
uid_1_2_840_10008_5_1_4_1_1_12_3 = 152 ,
uid_1_2_840_10008_5_1_4_1_1_20 = 153 ,
uid_1_2_840_10008_5_1_4_1_1_66 = 154 ,
uid_1_2_840_10008_5_1_4_1_1_66_1 = 155 ,
uid_1_2_840_10008_5_1_4_1_1_66_2 = 156 ,
uid_1_2_840_10008_5_1_4_1_1_66_3 = 157 ,
uid_1_2_840_10008_5_1_4_1_1_66_4 = 158 ,
uid_1_2_840_10008_5_1_4_1_1_67 = 159 ,
uid_1_2_840_10008_5_1_4_1_1_77_1 = 160 ,
uid_1_2_840_10008_5_1_4_1_1_77_2 = 161 ,
uid_1_2_840_10008_5_1_4_1_1_77_1_1 = 162 ,
uid_1_2_840_10008_5_1_4_1_1_77_1_1_1 = 163 ,
uid_1_2_840_10008_5_1_4_1_1_77_1_2 = 164 ,
uid_1_2_840_10008_5_1_4_1_1_77_1_2_1 = 165 ,
uid_1_2_840_10008_5_1_4_1_1_77_1_3 = 166 ,
uid_1_2_840_10008_5_1_4_1_1_77_1_4 = 167 ,
uid_1_2_840_10008_5_1_4_1_1_77_1_4_1 = 168 ,
uid_1_2_840_10008_5_1_4_1_1_77_1_5_1 = 169 ,
uid_1_2_840_10008_5_1_4_1_1_77_1_5_2 = 170 ,
uid_1_2_840_10008_5_1_4_1_1_77_1_5_3 = 171 ,
uid_1_2_840_10008_5_1_4_1_1_77_1_5_4 = 172 ,
uid_1_2_840_10008_5_1_4_1_1_88_1 = 173 ,
uid_1_2_840_10008_5_1_4_1_1_88_2 = 174 ,
uid_1_2_840_10008_5_1_4_1_1_88_3 = 175 ,
uid_1_2_840_10008_5_1_4_1_1_88_4 = 176 ,
uid_1_2_840_10008_5_1_4_1_1_88_11 = 177 ,
uid_1_2_840_10008_5_1_4_1_1_88_22 = 178 ,
uid_1_2_840_10008_5_1_4_1_1_88_33 = 179 ,

uid_1_2_840_10008_5_1_4_1_1_88_40 = 180 ,
uid_1_2_840_10008_5_1_4_1_1_88_50 = 181 ,
uid_1_2_840_10008_5_1_4_1_1_88_59 = 182 ,
uid_1_2_840_10008_5_1_4_1_1_88_65 = 183 ,
uid_1_2_840_10008_5_1_4_1_1_88_67 = 184 ,
uid_1_2_840_10008_5_1_4_1_1_104_1 = 185 ,
uid_1_2_840_10008_5_1_4_1_1_104_2 = 186 ,
uid_1_2_840_10008_5_1_4_1_1_128 = 187 ,
uid_1_2_840_10008_5_1_4_1_1_129 = 188 ,
uid_1_2_840_10008_5_1_4_1_1_481_1 = 189 ,
uid_1_2_840_10008_5_1_4_1_1_481_2 = 190 ,
uid_1_2_840_10008_5_1_4_1_1_481_3 = 191 ,
uid_1_2_840_10008_5_1_4_1_1_481_4 = 192 ,
uid_1_2_840_10008_5_1_4_1_1_481_5 = 193 ,
uid_1_2_840_10008_5_1_4_1_1_481_6 = 194 ,
uid_1_2_840_10008_5_1_4_1_1_481_7 = 195 ,
uid_1_2_840_10008_5_1_4_1_1_481_8 = 196 ,
uid_1_2_840_10008_5_1_4_1_1_481_9 = 197 ,
uid_1_2_840_10008_5_1_4_1_2_1_1 = 198 ,
uid_1_2_840_10008_5_1_4_1_2_1_2 = 199 ,
uid_1_2_840_10008_5_1_4_1_2_1_3 = 200 ,
uid_1_2_840_10008_5_1_4_1_2_2_1 = 201 ,
uid_1_2_840_10008_5_1_4_1_2_2_2 = 202 ,
uid_1_2_840_10008_5_1_4_1_2_2_3 = 203 ,
uid_1_2_840_10008_5_1_4_1_2_3_1 = 204 ,
uid_1_2_840_10008_5_1_4_1_2_3_2 = 205 ,
uid_1_2_840_10008_5_1_4_1_2_3_3 = 206 ,
uid_1_2_840_10008_5_1_4_31 = 207 ,
uid_1_2_840_10008_5_1_4_32_1 = 208 ,
uid_1_2_840_10008_5_1_4_32_2 = 209 ,
uid_1_2_840_10008_5_1_4_32_3 = 210 ,
uid_1_2_840_10008_5_1_4_32 = 211 ,
uid_1_2_840_10008_5_1_4_33 = 212 ,
uid_1_2_840_10008_5_1_4_34_1 = 213 ,
uid_1_2_840_10008_5_1_4_34_2 = 214 ,
uid_1_2_840_10008_5_1_4_34_3 = 215 ,
uid_1_2_840_10008_5_1_4_34_4 = 216 ,
uid_1_2_840_10008_5_1_4_34_4_1 = 217 ,
uid_1_2_840_10008_5_1_4_34_4_2 = 218 ,
uid_1_2_840_10008_5_1_4_34_4_3 = 219 ,
uid_1_2_840_10008_5_1_4_34_4_4 = 220 ,
uid_1_2_840_10008_5_1_4_34_5 = 221 ,
uid_1_2_840_10008_5_1_4_37_1 = 222 ,
uid_1_2_840_10008_5_1_4_37_2 = 223 ,
uid_1_2_840_10008_5_1_4_37_3 = 224 ,
uid_1_2_840_10008_5_1_4_38_1 = 225 ,
uid_1_2_840_10008_5_1_4_38_2 = 226 ,
uid_1_2_840_10008_5_1_4_38_3 = 227 ,
uid_1_2_840_10008_5_1_4_41 = 228 ,
uid_1_2_840_10008_5_1_4_42 = 229 ,
uid_1_2_840_10008_15_0_3_1 = 230 ,
uid_1_2_840_10008_15_0_3_2 = 231 ,
uid_1_2_840_10008_15_0_3_3 = 232 ,
uid_1_2_840_10008_15_0_3_4 = 233 ,

uid_1_2_840_10008_15_0_3_5 = 234 ,
uid_1_2_840_10008_15_0_3_6 = 235 ,
uid_1_2_840_10008_15_0_3_7 = 236 ,
uid_1_2_840_10008_15_0_3_8 = 237 ,
uid_1_2_840_10008_15_0_3_9 = 238 ,
uid_1_2_840_10008_15_0_3_10 = 239 ,
uid_1_2_840_10008_15_0_3_11 = 240 ,
uid_1_2_840_10008_15_0_3_12 = 241 ,
uid_1_2_840_10008_15_0_3_13 = 242 ,
uid_1_2_840_10008_15_0_3_14 = 243 ,
uid_1_2_840_10008_15_0_3_15 = 244 ,
uid_1_2_840_10008_15_0_3_16 = 245 ,
uid_1_2_840_10008_15_0_3_17 = 246 ,
uid_1_2_840_10008_15_0_3_18 = 247 ,
uid_1_2_840_10008_15_0_3_19 = 248 ,
uid_1_2_840_10008_15_0_3_20 = 249 ,
uid_1_2_840_10008_15_0_3_21 = 250 ,
uid_1_2_840_10008_15_0_3_22 = 251 ,
uid_1_2_840_10008_15_0_3_23 = 252 ,
uid_1_2_840_10008_15_0_3_24 = 253 ,
uid_1_2_840_10008_15_0_3_25 = 254 ,
uid_1_2_840_10008_15_0_3_26 = 255 ,
uid_1_2_840_10008_15_0_3_27 = 256 ,
uid_1_2_840_10008_15_0_3_28 = 257 ,
uid_1_2_840_10008_15_0_3_29 = 258 ,
uid_1_2_840_10008_15_0_3_30 = 259 ,
uid_1_2_840_10008_15_0_3_31 = 260 ,
uid_1_2_840_10008_15_0_4_1 = 261 ,
uid_1_2_840_10008_15_0_4_2 = 262 ,
uid_1_2_840_10008_15_0_4_3 = 263 ,
uid_1_2_840_10008_15_0_4_4 = 264 ,
uid_1_2_840_10008_15_0_4_5 = 265 ,
uid_1_2_840_10008_15_0_4_6 = 266 ,
uid_1_2_840_10008_15_0_4_7 = 267 ,
uid_1_2_840_10008_15_0_4_8 = 268 ,
uid_1_2_840_10008_5_1_4_1_1_77_1_6 = 269 ,
uid_1_2_840_10008_5_1_4_1_1_6_2 = 270 ,
uid_1_2_840_10008_5_1_4_1_1_66_5 = 271 ,
uid_1_2_840_10008_5_1_4_1_1_13_1_3 = 272 ,
uid_1_2_840_10008_5_1_4_1_1_2_2 = 273 ,
uid_1_2_840_10008_5_1_4_1_1_4_4 = 274 ,
uid_1_2_840_10008_5_1_4_1_1_128_1 = 275 ,
uid_1_2_840_10008_1_2_4_101 = 276 ,
uid_1_2_840_10008_1_2_4_102 = 277 ,
uid_1_2_840_10008_1_2_4_103 = 278 ,
uid_1_2_840_10008_1_5_2 = 279 ,
uid_1_2_840_10008_1_5_3 = 280 ,
uid_1_2_840_10008_1_5_4 = 281 ,
uid_1_2_840_10008_1_5_5 = 282 ,
uid_1_2_840_10008_1_5_6 = 283 ,
uid_1_2_840_10008_1_5_7 = 284 ,
uid_1_2_840_10008_1_5_8 = 285 ,
uid_1_2_840_10008_1_20 = 286 ,
uid_1_2_840_10008_2_16_5 = 287 ,

uid_1_2_840_10008_2_16_6 = 288 ,
uid_1_2_840_10008_2_16_7 = 289 ,
uid_1_2_840_10008_2_16_8 = 290 ,
uid_1_2_840_10008_2_16_9 = 291 ,
uid_1_2_840_10008_2_16_10 = 292 ,
uid_1_2_840_10008_2_16_11 = 293 ,
uid_1_2_840_10008_2_16_12 = 294 ,
uid_1_2_840_10008_2_16_13 = 295 ,
uid_1_2_840_10008_2_16_14 = 296 ,
uid_1_2_840_10008_5_1_1_40 = 297 ,
uid_1_2_840_10008_5_1_1_40_1 = 298 ,
uid_1_2_840_10008_5_1_4_1_1_9_4_2 = 299 ,
uid_1_2_840_10008_5_1_4_1_1_9_5_1 = 300 ,
uid_1_2_840_10008_5_1_4_1_1_9_6_1 = 301 ,
uid_1_2_840_10008_5_1_4_1_1_11_5 = 302 ,
uid_1_2_840_10008_5_1_4_1_1_11_6 = 303 ,
uid_1_2_840_10008_1_2_4_104 = 304 ,
uid_1_2_840_10008_1_2_4_105 = 305 ,
uid_1_2_840_10008_1_2_4_106 = 306 ,
uid_1_2_840_10008_1_2_4_107 = 307 ,
uid_1_2_840_10008_1_2_4_108 = 308 ,
uid_1_2_840_10008_1_5_1 = 309 ,
uid_1_2_840_10008_5_1_4_1_1_11_7 = 310 ,
uid_1_2_840_10008_5_1_4_1_1_11_8 = 311 ,
uid_1_2_840_10008_5_1_4_1_1_11_9 = 312 ,
uid_1_2_840_10008_5_1_4_1_1_11_10 = 313 ,
uid_1_2_840_10008_5_1_4_1_1_11_11 = 314 ,
uid_1_2_840_10008_5_1_4_1_1_12_77 = 315 ,
uid_1_2_840_10008_5_1_4_1_1_13_1_4 = 316 ,
uid_1_2_840_10008_5_1_4_1_1_13_1_5 = 317 ,
uid_1_2_840_10008_5_1_4_1_1_14_1 = 318 ,
uid_1_2_840_10008_5_1_4_1_1_14_2 = 319 ,
uid_1_2_840_10008_5_1_4_1_1_30 = 320 ,
uid_1_2_840_10008_5_1_4_1_1_40 = 321 ,
uid_1_2_840_10008_5_1_4_1_1_66_6 = 322 ,
uid_1_2_840_10008_5_1_4_1_1_68_1 = 323 ,
uid_1_2_840_10008_5_1_4_1_1_68_2 = 324 ,
uid_1_2_840_10008_5_1_4_1_1_77_1_5_5 = 325 ,
uid_1_2_840_10008_5_1_4_1_1_77_1_5_6 = 326 ,
uid_1_2_840_10008_5_1_4_1_1_77_1_5_7 = 327 ,
uid_1_2_840_10008_5_1_4_1_1_77_1_5_8 = 328 ,
uid_1_2_840_10008_5_1_4_1_1_78_1 = 329 ,
uid_1_2_840_10008_5_1_4_1_1_78_2 = 330 ,
uid_1_2_840_10008_5_1_4_1_1_78_3 = 331 ,
uid_1_2_840_10008_5_1_4_1_1_78_4 = 332 ,
uid_1_2_840_10008_5_1_4_1_1_78_5 = 333 ,
uid_1_2_840_10008_5_1_4_1_1_78_6 = 334 ,
uid_1_2_840_10008_5_1_4_1_1_78_7 = 335 ,
uid_1_2_840_10008_5_1_4_1_1_78_8 = 336 ,
uid_1_2_840_10008_5_1_4_1_1_79_1 = 337 ,
uid_1_2_840_10008_5_1_4_1_1_80_1 = 338 ,
uid_1_2_840_10008_5_1_4_1_1_81_1 = 339 ,
uid_1_2_840_10008_5_1_4_1_1_82_1 = 340 ,
uid_1_2_840_10008_5_1_4_1_1_88_34 = 341 ,

uid_1_2_840_10008_5_1_4_1_1_88_35 = 342 ,
uid_1_2_840_10008_5_1_4_1_1_88_68 = 343 ,
uid_1_2_840_10008_5_1_4_1_1_88_69 = 344 ,
uid_1_2_840_10008_5_1_4_1_1_88_70 = 345 ,
uid_1_2_840_10008_5_1_4_1_1_88_71 = 346 ,
uid_1_2_840_10008_5_1_4_1_1_88_72 = 347 ,
uid_1_2_840_10008_5_1_4_1_1_88_73 = 348 ,
uid_1_2_840_10008_5_1_4_1_1_88_74 = 349 ,
uid_1_2_840_10008_5_1_4_1_1_88_75 = 350 ,
uid_1_2_840_10008_5_1_4_1_1_90_1 = 351 ,
uid_1_2_840_10008_5_1_4_1_1_104_3 = 352 ,
uid_1_2_840_10008_5_1_4_1_1_130 = 353 ,
uid_1_2_840_10008_5_1_4_1_1_131 = 354 ,
uid_1_2_840_10008_5_1_4_1_1_200_1 = 355 ,
uid_1_2_840_10008_5_1_4_1_1_200_2 = 356 ,
uid_1_2_840_10008_5_1_4_1_1_200_3 = 357 ,
uid_1_2_840_10008_5_1_4_1_1_200_4 = 358 ,
uid_1_2_840_10008_5_1_4_1_1_200_5 = 359 ,
uid_1_2_840_10008_5_1_4_1_1_200_6 = 360 ,
uid_1_2_840_10008_5_1_4_1_1_481_10 = 361 ,
uid_1_2_840_10008_5_1_4_1_1_481_11 = 362 ,
uid_1_2_840_10008_5_1_4_1_1_501_1 = 363 ,
uid_1_2_840_10008_5_1_4_1_1_501_2_1 = 364 ,
uid_1_2_840_10008_5_1_4_1_1_501_2_2 = 365 ,
uid_1_2_840_10008_5_1_4_1_1_501_3 = 366 ,
uid_1_2_840_10008_5_1_4_1_1_501_4 = 367 ,
uid_1_2_840_10008_5_1_4_1_1_501_5 = 368 ,
uid_1_2_840_10008_5_1_4_1_1_501_6 = 369 ,
uid_1_2_840_10008_5_1_4_1_1_601_1 = 370 ,
uid_1_2_840_10008_5_1_4_1_1_601_2 = 371 ,
uid_1_2_840_10008_5_1_4_1_2_4_2 = 372 ,
uid_1_2_840_10008_5_1_4_1_2_4_3 = 373 ,
uid_1_2_840_10008_5_1_4_1_2_5_3 = 374 ,
uid_1_2_840_10008_5_1_4_20_1 = 375 ,
uid_1_2_840_10008_5_1_4_20_2 = 376 ,
uid_1_2_840_10008_5_1_4_20_3 = 377 ,
uid_1_2_840_10008_5_1_4_34_5_1 = 378 ,
uid_1_2_840_10008_5_1_4_34_6 = 379 ,
uid_1_2_840_10008_5_1_4_34_6_1 = 380 ,
uid_1_2_840_10008_5_1_4_34_6_2 = 381 ,
uid_1_2_840_10008_5_1_4_34_6_3 = 382 ,
uid_1_2_840_10008_5_1_4_34_6_4 = 383 ,
uid_1_2_840_10008_5_1_4_34_7 = 384 ,
uid_1_2_840_10008_5_1_4_34_8 = 385 ,
uid_1_2_840_10008_5_1_4_34_9 = 386 ,
uid_1_2_840_10008_5_1_4_34_10 = 387 ,
uid_1_2_840_10008_5_1_4_38_4 = 388 ,
uid_1_2_840_10008_5_1_4_39_1 = 389 ,
uid_1_2_840_10008_5_1_4_39_2 = 390 ,
uid_1_2_840_10008_5_1_4_39_3 = 391 ,
uid_1_2_840_10008_5_1_4_39_4 = 392 ,
uid_1_2_840_10008_5_1_4_43_1 = 393 ,
uid_1_2_840_10008_5_1_4_43_2 = 394 ,
uid_1_2_840_10008_5_1_4_43_3 = 395 ,

uid_1_2_840_10008_5_1_4_43_4 = 396 ,
uid_1_2_840_10008_5_1_4_44_1 = 397 ,
uid_1_2_840_10008_5_1_4_44_2 = 398 ,
uid_1_2_840_10008_5_1_4_44_3 = 399 ,
uid_1_2_840_10008_5_1_4_44_4 = 400 ,
uid_1_2_840_10008_5_1_4_45_1 = 401 ,
uid_1_2_840_10008_5_1_4_45_2 = 402 ,
uid_1_2_840_10008_5_1_4_45_3 = 403 ,
uid_1_2_840_10008_5_1_4_45_4 = 404 ,
uid_1_2_840_10008_7_1_1 = 405 ,
uid_1_2_840_10008_7_1_2 = 406 ,
uid_1_2_840_10008_8_1_1 = 407 ,
uid_1_2_840_10008_5_1_4_1_1_4_3 = 408 ,
uid_1_2_840_10008_15_1_1 = 409 }

Public Member Functions

- const char * GetName () const
- const char * GetString () const
- operator TSType () const
- bool SetFromUID (const char *str)

Static Public Member Functions

- static unsigned int GetNumberOfTransferSyntaxStrings ()
- static const char *const * GetTransferSyntaxString (unsigned int ts)
- static TransferSyntaxStringsType GetTransferSyntaxStrings ()
- static const char * GetUIDName (unsigned int ts)
- static const char * GetUIDString (unsigned int ts)

10.323.1 Detailed Description

all known uids

Examples

GenerateStandardSOPClasses.cxx.

10.323.2 Member Typedef Documentation

10.323.2.1 TransferSyntaxStringsType

```
typedef const char* const(* gdcm::UIDs::TransferSyntaxStringsType)[2]
```

10.323.3 Member Enumeration Documentation

10.323.3.1 TSName

enum gdcm::UIDs::TSName

Enumerator

VerificationSOPClass	
ImplicitVRLittleEndianDefaultTransferSyntaxforDICOM	
ExplicitVRLittleEndian	
DeflatedExplicitVRLittleEndian	
ExplicitVRBigEndian	
JPEGBaselineProcess1DefaultTransferSyntaxforLossyJPEG8BitImageCompression	
JPEGExtendedProcess24DefaultTransferSyntaxforLossyJPEG12BitImageCompressionProcess4only	
JPEGExtendedProcess35Retired	
JPEGSpectralSelectionNonHierarchicalProcess68Retired	
JPEGSpectralSelectionNonHierarchicalProcess79Retired	
JPEGFullProgressionNonHierarchicalProcess1012Retired	
JPEGFullProgressionNonHierarchicalProcess1113Retired	
JPEGLosslessNonHierarchicalProcess14	
JPEGLosslessNonHierarchicalProcess15Retired	
JPEGExtendedHierarchicalProcess1618Retired	
JPEGExtendedHierarchicalProcess1719Retired	
JPEGSpectralSelectionHierarchicalProcess2022Retired	
JPEGSpectralSelectionHierarchicalProcess2123Retired	
JPEGFullProgressionHierarchicalProcess2426Retired	
JPEGFullProgressionHierarchicalProcess2527Retired	
JPEGLosslessHierarchicalProcess28Retired	
JPEGLosslessHierarchicalProcess29Retired	
JPEGLosslessNonHierarchicalFirstOrderPredictionProcess14SelectionValue1DefaultTransferSyntaxforLossless↩ JPEGImageCompression	
JPEGLSLosslessImageCompression	
JPEGLSLossyNearLosslessImageCompression	
JPEG2000ImageCompressionLosslessOnly	
JPEG2000ImageCompression	
JPEG2000Part2MulticomponentImageCompressionLosslessOnly	
JPEG2000Part2MulticomponentImageCompression	
JPIPReferenced	
JPIPReferencedDeflate	
MPEG2MainProfileMainLevel	
RLELossless	
RFC2557MIMEencapsulation	
XMLEncoding	

Enumerator

MediaStorageDirectoryStorage	
TalairachBrainAtlasFrameofReference	
SPM2T1FrameofReference	
SPM2T2FrameofReference	
SPM2PDFrameofReference	
SPM2EPIFrameofReference	
SPM2FILT1FrameofReference	
SPM2PETFrameofReference	
SPM2TRANSMFrameofReference	
SPM2SPECTFrameofReference	
SPM2GRAYFrameofReference	
SPM2WHITEFrameofReference	
SPM2CSFFrameofReference	
SPM2BRAINMASKFrameofReference	
SPM2AVG305T1FrameofReference	
SPM2AVG152T1FrameofReference	
SPM2AVG152T2FrameofReference	
SPM2AVG152PDFrameofReference	
SPM2SINGLESUBJT1FrameofReference	
ICBM452T1FrameofReference	
ICBMSingleSubjectMRIFrameofReference	
BasicStudyContentNotificationSOPClassRetired	
StorageCommitmentPushModelSOPClass	
StorageCommitmentPushModelSOPInstance	
StorageCommitmentPullModelSOPClassRetired	
StorageCommitmentPullModelSOPInstanceRetired	
ProceduralEventLoggingSOPClass	
ProceduralEventLoggingSOPInstance	
SubstanceAdministrationLoggingSOPClass	
SubstanceAdministrationLoggingSOPInstance	
DICOMUIDRegistry	
DICOMControlledTerminology	
DICOMApplicationContextName	
DetachedPatientManagementSOPClassRetired	
DetachedPatientManagementMetaSOPClassRetired	
DetachedVisitManagementSOPClassRetired	
DetachedStudyManagementSOPClassRetired	
StudyComponentManagementSOPClassRetired	
ModalityPerformedProcedureStepSOPClass	
ModalityPerformedProcedureStepRetrieveSOPClass	
ModalityPerformedProcedureStepNotificationSOPClass	
DetachedResultsManagementSOPClassRetired	
DetachedResultsManagementMetaSOPClassRetired	
DetachedStudyManagementMetaSOPClassRetired	

Enumerator

DetachedInterpretationManagementSOPClassRetired	
StorageServiceClass	
BasicFilmSessionSOPClass	
BasicFilmBoxSOPClass	
BasicGrayscaleImageBoxSOPClass	
BasicColorImageBoxSOPClass	
ReferencedImageBoxSOPClassRetired	
BasicGrayscalePrintManagementMetaSOPClass	
ReferencedGrayscalePrintManagementMetaSOPClassRetired	
PrintJobSOPClass	
BasicAnnotationBoxSOPClass	
PrinterSOPClass	
PrinterConfigurationRetrievalSOPClass	
PrinterSOPInstance	
PrinterConfigurationRetrievalSOPInstance	
BasicColorPrintManagementMetaSOPClass	
ReferencedColorPrintManagementMetaSOPClassRetired	
VOILUTBoxSOPClass	
PresentationLUTSOPClass	
ImageOverlayBoxSOPClassRetired	
BasicPrintImageOverlayBoxSOPClassRetired	
PrintQueueSOPInstanceRetired	
PrintQueueManagementSOPClassRetired	
StoredPrintStorageSOPClassRetired	
HardcopyGrayscaleImageStorageSOPClassRetired	
HardcopyColorImageStorageSOPClassRetired	
PullPrintRequestSOPClassRetired	
PullStoredPrintManagementMetaSOPClassRetired	
MediaCreationManagementSOPClassUID	
ComputedRadiographyImageStorage	
DigitalXRayImageStorageForPresentation	
DigitalXRayImageStorageForProcessing	
DigitalMammographyXRayImageStorageForPresentation	
DigitalMammographyXRayImageStorageForProcessing	
DigitalIntraoralXRayImageStorageForPresentation	
DigitalIntraoralXRayImageStorageForProcessing	
CTImageStorage	
EnhancedCTImageStorage	
UltrasoundMultiframeImageStorageRetired	
UltrasoundMultiframeImageStorage	
MRImageStorage	
EnhancedMRImageStorage	
MRSpectroscopyStorage	

Enumerator

NuclearMedicineImageStorageRetired	
UltrasoundImageStorageRetired	
UltrasoundImageStorage	
SecondaryCaptureImageStorage	
MultiframeSingleBitSecondaryCaptureImageStorage	
MultiframeGrayscaleByteSecondaryCaptureImageStorage	
MultiframeGrayscaleWordSecondaryCaptureImageStorage	
MultiframeTrueColorSecondaryCaptureImageStorage	
StandaloneOverlayStorageRetired	
StandaloneCurveStorageRetired	
WaveformStorageTrialRetired	
ECG12leadWaveformStorage	
GeneralECGWaveformStorage	
AmbulatoryECGWaveformStorage	
HemodynamicWaveformStorage	
CardiacElectrophysiologyWaveformStorage	
BasicVoiceAudioWaveformStorage	
StandaloneModalityLUTStorageRetired	
StandaloneVOILUTStorageRetired	
GrayscaleSoftcopyPresentationStateStorageSOPClass	
ColorSoftcopyPresentationStateStorageSOPClass	
PseudoColorSoftcopyPresentationStateStorageSOPClass	
BlendingSoftcopyPresentationStateStorageSOPClass	
XRayAngiographicImageStorage	
EnhancedXAImageStorage	
XRayRadiofluoroscopicImageStorage	
EnhancedXRFImageStorage	
XRay3DAngiographicImageStorage	
XRay3DCraniofacialImageStorage	
XRayAngiographicBiPlaneImageStorageRetired	
NuclearMedicineImageStorage	
RawDataStorage	
SpatialRegistrationStorage	
SpatialFiducialsStorage	
DeformableSpatialRegistrationStorage	
SegmentationStorage	
RealWorldValueMappingStorage	
VLImageStorageTrialRetired	
VLMultiframeImageStorageTrialRetired	
VLEndoscopicImageStorage	
VideoEndoscopicImageStorage	
VLMicroscopicImageStorage	
VideoMicroscopicImageStorage	

Enumerator

VLSlideCoordinatesMicroscopicImageStorage	
VLPhotographicImageStorage	
VideoPhotographicImageStorage	
OphthalmicPhotography8BitImageStorage	
OphthalmicPhotography16BitImageStorage	
StereometricRelationshipStorage	
OphthalmicTomographyImageStorage	
TextSRStorageTrialRetired	
AudioSRStorageTrialRetired	
DetailSRStorageTrialRetired	
ComprehensiveSRStorageTrialRetired	
BasicTextSRStorage	
EnhancedSRStorage	
ComprehensiveSRStorage	
ProcedureLogStorage	
MammographyCADSRStorage	
KeyObjectSelectionDocumentStorage	
ChestCADSRStorage	
XRayRadiationDoseSRStorage	
EncapsulatedPDFStorage	
EncapsulatedCDAStorage	
PositronEmissionTomographyImageStorage	
StandalonePETCurveStorageRetired	
RTImageStorage	
RTDoseStorage	
RTStructureSetStorage	
RTBeamsTreatmentRecordStorage	
RTPlanStorage	
RTBrachyTreatmentRecordStorage	
RTTreatmentSummaryRecordStorage	
RTIonPlanStorage	
RTIonBeamsTreatmentRecordStorage	
PatientRootQueryRetrieveInformationModelFIND	
PatientRootQueryRetrieveInformationModelMOVE	
PatientRootQueryRetrieveInformationModelGET	
StudyRootQueryRetrieveInformationModelFIND	
StudyRootQueryRetrieveInformationModelMOVE	
StudyRootQueryRetrieveInformationModelGET	
PatientStudyOnlyQueryRetrieveInformationModelFINDRetired	
PatientStudyOnlyQueryRetrieveInformationModelMOVERetired	
PatientStudyOnlyQueryRetrieveInformationModelGETRetired	
ModalityWorklistInformationModelFIND	
GeneralPurposeWorklistInformationModelFIND	

Enumerator

GeneralPurposeScheduledProcedureStepSOPClass	
GeneralPurposePerformedProcedureStepSOPClass	
GeneralPurposeWorklistManagementMetaSOPClass	
InstanceAvailabilityNotificationSOPClass	
RTBeamsDeliveryInstructionStorageSupplement74FrozenDraft	
RTConventionalMachineVerificationSupplement74FrozenDraft	
RTIonMachineVerificationSupplement74FrozenDraft	
UnifiedWorklistandProcedureStepServiceClass	
UnifiedProcedureStepPushSOPClass	
UnifiedProcedureStepWatchSOPClass	
UnifiedProcedureStepPullSOPClass	
UnifiedProcedureStepEventSOPClass	
UnifiedWorklistandProcedureStepSOPInstance	
GeneralRelevantPatientInformationQuery	
BreastImagingRelevantPatientInformationQuery	
CardiacRelevantPatientInformationQuery	
HangingProtocolStorage	
HangingProtocolInformationModelFIND	
HangingProtocolInformationModelMOVE	
ProductCharacteristicsQuerySOPClass	
SubstanceApprovalQuerySOPClass	
dicomDeviceName	
dicomDescription	
dicomManufacturer	
dicomManufacturerModelName	
dicomSoftwareVersion	
dicomVendorData	
dicomAETitle	
dicomNetworkConnectionReference	
dicomApplicationCluster	
dicomAssociationInitiator	
dicomAssociationAcceptor	
dicomHostname	
dicomPort	
dicomSOPClass	
dicomTransferRole	
dicomTransferSyntax	
dicomPrimaryDeviceType	
dicomRelatedDeviceReference	
dicomPreferredCalledAETitle	
dicomTLSCyphersuite	
dicomAuthorizedNodeCertificateReference	
dicomThisNodeCertificateReference	
dicomInstalled	

Enumerator

dicomStationName	
dicomDeviceSerialNumber	
dicomInstitutionName	
dicomInstitutionAddress	
dicomInstitutionDepartmentName	
dicomIssuerOfPatientID	
dicomPreferredCallingAETitle	
dicomSupportedCharacterSet	
dicomConfigurationRoot	
dicomDevicesRoot	
dicomUniqueAETitlesRegistryRoot	
dicomDevice	
dicomNetworkAE	
dicomNetworkConnection	
dicomUniqueAETitle	
dicomTransferCapability	
VLWholeSlideMicroscopyImageStorage	
EnhancedUSVolumeStorage	
SurfaceSegmentationStorage	
BreastTomosynthesisImageStorage	
LegacyConvertedEnhancedCTImageStorage	
LegacyConvertedEnhancedMRImageStorage	
LegacyConvertedEnhancedPETImageStorage	
MPEG2MainProfileHighLevel	
MPEG4AVCH_264HighProfileLevel4_1	
MPEG4AVCH_264BDcompatibleHighProfileLevel4_1	
PETColorPaletteSOPInstance	
HotMetalBlueColorPaletteSOPInstance	
PET20StepColorPaletteSOPInstance	
SpringColorPaletteSOPInstance	
SummerColorPaletteSOPInstance	
FallColorPaletteSOPInstance	
WinterColorPaletteSOPInstance	
Papyrus3ImplicitVRLittleEndian	
AdultMouseAnatomyOntology	
UberonOntology	
IntegratedTaxonomicInformationSystemITISTaxonomicSerialNumberTSN	
MouseGenomeInitiativeMGI	
PubChemCompoundCID	
ICD11	
NewYorkUniversityMelanomaClinicalCooperativeGroup	
MayoClinicNonradiologicalImagesSBSAnatomicalSurfaceRegionGuide	
ImageBiomarkerStandardisationInitiative	
RadiomicsOntology	

Enumerator

DisplaySystemSOPClass	
DisplaySystemSOPInstance	
GeneralAudioWaveformStorage	
ArterialPulseWaveformStorage	
RespiratoryWaveformStorage	
XAXRFGrayscaleSoftcopyPresentationStateStorage	
GrayscalePlanarMPRVolumetricPresentationStateStorage	
MPEG4AVCH_264HighProfileLevel4_2For2DVideo	
MPEG4AVCH_264HighProfileLevel4_2For3DVideo	
MPEG4AVCH_264StereoHighProfileLevel4_2	
HEVCH_265MainProfileLevel5_1	
HEVCH_265Main10ProfileLevel5_1	
HotIronColorPaletteSOPInstance	
CompositingPlanarMPRVolumetricPresentationStateStorage	
AdvancedBlendingPresentationStateStorage	
VolumeRenderingVolumetricPresentationStateStorage	
SegmentedVolumeRenderingVolumetricPresentationStateStorage	
MultipleVolumeRenderingVolumetricPresentationStateStorage	
Null0	
BreastProjectionXRayImageStorageForPresentation	
BreastProjectionXRayImageStorageForProcessing	
IntravascularOpticalCoherenceTomographyImageStorageForPresentation	
IntravascularOpticalCoherenceTomographyImageStorageForProcessing	
ParametricMapStorage	
Null1	
TractographyResultsStorage	
SurfaceScanMeshStorage	
SurfaceScanPointCloudStorage	
WideFieldOphthalmicPhotographyStereographicProjectionImageStorage	
WideFieldOphthalmicPhotography3DCoordinatesImageStorage	
OphthalmicOpticalCoherenceTomographyEnFaceImageStorage	
OphthalmicOpticalCoherenceTomographyBscanVolumeAnalysisStorage	
LensometryMeasurementsStorage	
AutorefractionMeasurementsStorage	
KeratometryMeasurementsStorage	
SubjectiveRefractionMeasurementsStorage	
VisualAcuityMeasurementsStorage	
SpectaclePrescriptionReportStorage	
OphthalmicAxialMeasurementsStorage	
IntraocularLensCalculationsStorage	
MacularGridThicknessandVolumeReportStorage	
OphthalmicVisualFieldStaticPerimetryMeasurementsStorage	
OphthalmicThicknessMapStorage	

Enumerator

CornealTopographyMapStorage	
Comprehensive3DSRStorage	
ExtensibleSRStorage	
RadiopharmaceuticalRadiationDoseSRStorage	
ColonCADSRStorage	
ImplantationPlanSRStorage	
AcquisitionContextSRStorage	
SimplifiedAdultEchoSRStorage	
PatientRadiationDoseSRStorage	
PlannedImagingAgentAdministrationSRStorage	
PerformedImagingAgentAdministrationSRStorage	
ContentAssessmentResultsStorage	
EncapsulatedSTLStorage	
EnhancedPETImageStorage	
BasicStructuredDisplayStorage	
CTDefinedProcedureProtocolStorage	
CTPerformedProcedureProtocolStorage	
ProtocolApprovalStorage	
ProtocolApprovalInformationModelFIND	
ProtocolApprovalInformationModelMOVE	
ProtocolApprovalInformationModelGET	
RTPhysicianIntentStorage	
RTSegmentAnnotationStorage	
DICOSCTImageStorage	
DICOSDigitalXRayImageStorageForPresentation	
DICOSDigitalXRayImageStorageForProcessing	
DICOSThreatDetectionReportStorage	
DICOS2DAITStorage	
DICOS3DAITStorage	
DICOSQuadrupoleResonanceQRStorage	
EddyCurrentImageStorage	
EddyCurrentMultiframeImageStorage	
CompositeInstanceRootRetrieveMOVE	
CompositeInstanceRootRetrieveGET	
CompositeInstanceRetrieveWithoutBulkDataGET	
DefinedProcedureProtocolInformationModelFIND	
DefinedProcedureProtocolInformationModelMOVE	
DefinedProcedureProtocolInformationModelGET	
UPSFilteredGlobalSubscriptionSOPInstance	
UnifiedWorklistandProcedureStepServiceClass1	
UnifiedProcedureStepPushSOPClass1	
UnifiedProcedureStepWatchSOPClass1	
UnifiedProcedureStepPullSOPClass1	

Enumerator

UnifiedProcedureStepEventSOPClass1	
RTBeamsDeliveryInstructionStorage	
RTConventionalMachineVerification	
RTIonMachineVerification	
RTBrachyApplicationSetupDeliveryInstructionStorage	
HangingProtocolInformationModelGET	
ColorPaletteStorage	
ColorPaletteQueryRetrieveInformationModelFIND	
ColorPaletteQueryRetrieveInformationModelMOVE	
ColorPaletteQueryRetrieveInformationModelGET	
GenericImplantTemplateStorage	
GenericImplantTemplateInformationModelFIND	
GenericImplantTemplateInformationModelMOVE	
GenericImplantTemplateInformationModelGET	
ImplantAssemblyTemplateStorage	
ImplantAssemblyTemplateInformationModelFIND	
ImplantAssemblyTemplateInformationModelMOVE	
ImplantAssemblyTemplateInformationModelGET	
ImplantTemplateGroupStorage	
ImplantTemplateGroupInformationModelFIND	
ImplantTemplateGroupInformationModelMOVE	
ImplantTemplateGroupInformationModelGET	
NativeDICOMModel	
AbstractMultiDimensionalImageModel	
DICOMContentMappingResource	
EnhancedMRColorImageStorage	
UniversalCoordinatedTime	

10.323.3.2 TSType

`enum gdcm::UIDs::TSType`

Enumerator

uid_1_2_840_10008_1_1	
uid_1_2_840_10008_1_2	
uid_1_2_840_10008_1_2_1	
uid_1_2_840_10008_1_2_1_99	
uid_1_2_840_10008_1_2_2	
uid_1_2_840_10008_1_2_4_50	
uid_1_2_840_10008_1_2_4_51	

Enumerator

uid_1_2_840_10008_1_2_4_52	
uid_1_2_840_10008_1_2_4_53	
uid_1_2_840_10008_1_2_4_54	
uid_1_2_840_10008_1_2_4_55	
uid_1_2_840_10008_1_2_4_56	
uid_1_2_840_10008_1_2_4_57	
uid_1_2_840_10008_1_2_4_58	
uid_1_2_840_10008_1_2_4_59	
uid_1_2_840_10008_1_2_4_60	
uid_1_2_840_10008_1_2_4_61	
uid_1_2_840_10008_1_2_4_62	
uid_1_2_840_10008_1_2_4_63	
uid_1_2_840_10008_1_2_4_64	
uid_1_2_840_10008_1_2_4_65	
uid_1_2_840_10008_1_2_4_66	
uid_1_2_840_10008_1_2_4_70	
uid_1_2_840_10008_1_2_4_80	
uid_1_2_840_10008_1_2_4_81	
uid_1_2_840_10008_1_2_4_90	
uid_1_2_840_10008_1_2_4_91	
uid_1_2_840_10008_1_2_4_92	
uid_1_2_840_10008_1_2_4_93	
uid_1_2_840_10008_1_2_4_94	
uid_1_2_840_10008_1_2_4_95	
uid_1_2_840_10008_1_2_4_100	
uid_1_2_840_10008_1_2_5	
uid_1_2_840_10008_1_2_6_1	
uid_1_2_840_10008_1_2_6_2	
uid_1_2_840_10008_1_3_10	
uid_1_2_840_10008_1_4_1_1	
uid_1_2_840_10008_1_4_1_2	
uid_1_2_840_10008_1_4_1_3	
uid_1_2_840_10008_1_4_1_4	
uid_1_2_840_10008_1_4_1_5	
uid_1_2_840_10008_1_4_1_6	
uid_1_2_840_10008_1_4_1_7	
uid_1_2_840_10008_1_4_1_8	
uid_1_2_840_10008_1_4_1_9	
uid_1_2_840_10008_1_4_1_10	
uid_1_2_840_10008_1_4_1_11	
uid_1_2_840_10008_1_4_1_12	
uid_1_2_840_10008_1_4_1_13	
uid_1_2_840_10008_1_4_1_14	
uid_1_2_840_10008_1_4_1_15	
uid_1_2_840_10008_1_4_1_16	
uid_1_2_840_10008_1_4_1_17	

Enumerator

uid_1_2_840_10008_1_4_1_18	
uid_1_2_840_10008_1_4_2_1	
uid_1_2_840_10008_1_4_2_2	
uid_1_2_840_10008_1_9	
uid_1_2_840_10008_1_20_1	
uid_1_2_840_10008_1_20_1_1	
uid_1_2_840_10008_1_20_2	
uid_1_2_840_10008_1_20_2_1	
uid_1_2_840_10008_1_40	
uid_1_2_840_10008_1_40_1	
uid_1_2_840_10008_1_42	
uid_1_2_840_10008_1_42_1	
uid_1_2_840_10008_2_6_1	
uid_1_2_840_10008_2_16_4	
uid_1_2_840_10008_3_1_1_1	
uid_1_2_840_10008_3_1_2_1_1	
uid_1_2_840_10008_3_1_2_1_4	
uid_1_2_840_10008_3_1_2_2_1	
uid_1_2_840_10008_3_1_2_3_1	
uid_1_2_840_10008_3_1_2_3_2	
uid_1_2_840_10008_3_1_2_3_3	
uid_1_2_840_10008_3_1_2_3_4	
uid_1_2_840_10008_3_1_2_3_5	
uid_1_2_840_10008_3_1_2_5_1	
uid_1_2_840_10008_3_1_2_5_4	
uid_1_2_840_10008_3_1_2_5_5	
uid_1_2_840_10008_3_1_2_6_1	
uid_1_2_840_10008_4_2	
uid_1_2_840_10008_5_1_1_1	
uid_1_2_840_10008_5_1_1_2	
uid_1_2_840_10008_5_1_1_4	
uid_1_2_840_10008_5_1_1_4_1	
uid_1_2_840_10008_5_1_1_4_2	
uid_1_2_840_10008_5_1_1_9	
uid_1_2_840_10008_5_1_1_9_1	
uid_1_2_840_10008_5_1_1_14	
uid_1_2_840_10008_5_1_1_15	
uid_1_2_840_10008_5_1_1_16	
uid_1_2_840_10008_5_1_1_16_376	
uid_1_2_840_10008_5_1_1_17	
uid_1_2_840_10008_5_1_1_17_376	
uid_1_2_840_10008_5_1_1_18	
uid_1_2_840_10008_5_1_1_18_1	
uid_1_2_840_10008_5_1_1_22	
uid_1_2_840_10008_5_1_1_23	
uid_1_2_840_10008_5_1_1_24	

Enumerator

uid_1_2_840_10008_5_1_1_24_1	
uid_1_2_840_10008_5_1_1_25	
uid_1_2_840_10008_5_1_1_26	
uid_1_2_840_10008_5_1_1_27	
uid_1_2_840_10008_5_1_1_29	
uid_1_2_840_10008_5_1_1_30	
uid_1_2_840_10008_5_1_1_31	
uid_1_2_840_10008_5_1_1_32	
uid_1_2_840_10008_5_1_1_33	
uid_1_2_840_10008_5_1_4_1_1_1	
uid_1_2_840_10008_5_1_4_1_1_1_1	
uid_1_2_840_10008_5_1_4_1_1_1_1_1	
uid_1_2_840_10008_5_1_4_1_1_1_2	
uid_1_2_840_10008_5_1_4_1_1_1_2_1	
uid_1_2_840_10008_5_1_4_1_1_1_3	
uid_1_2_840_10008_5_1_4_1_1_1_3_1	
uid_1_2_840_10008_5_1_4_1_1_2	
uid_1_2_840_10008_5_1_4_1_1_2_1	
uid_1_2_840_10008_5_1_4_1_1_3	
uid_1_2_840_10008_5_1_4_1_1_3_1	
uid_1_2_840_10008_5_1_4_1_1_4	
uid_1_2_840_10008_5_1_4_1_1_4_1	
uid_1_2_840_10008_5_1_4_1_1_4_2	
uid_1_2_840_10008_5_1_4_1_1_5	
uid_1_2_840_10008_5_1_4_1_1_6	
uid_1_2_840_10008_5_1_4_1_1_6_1	
uid_1_2_840_10008_5_1_4_1_1_7	
uid_1_2_840_10008_5_1_4_1_1_7_1	
uid_1_2_840_10008_5_1_4_1_1_7_2	
uid_1_2_840_10008_5_1_4_1_1_7_3	
uid_1_2_840_10008_5_1_4_1_1_7_4	
uid_1_2_840_10008_5_1_4_1_1_8	
uid_1_2_840_10008_5_1_4_1_1_9	
uid_1_2_840_10008_5_1_4_1_1_9_1	
uid_1_2_840_10008_5_1_4_1_1_9_1_1	
uid_1_2_840_10008_5_1_4_1_1_9_1_2	
uid_1_2_840_10008_5_1_4_1_1_9_1_3	
uid_1_2_840_10008_5_1_4_1_1_9_2_1	
uid_1_2_840_10008_5_1_4_1_1_9_3_1	
uid_1_2_840_10008_5_1_4_1_1_9_4_1	
uid_1_2_840_10008_5_1_4_1_1_10	
uid_1_2_840_10008_5_1_4_1_1_11	
uid_1_2_840_10008_5_1_4_1_1_11_1	
uid_1_2_840_10008_5_1_4_1_1_11_2	
uid_1_2_840_10008_5_1_4_1_1_11_3	
uid_1_2_840_10008_5_1_4_1_1_11_4	

Enumerator

uid_1_2_840_10008_5_1_4_1_1_12_1	
uid_1_2_840_10008_5_1_4_1_1_12_1_1	
uid_1_2_840_10008_5_1_4_1_1_12_2	
uid_1_2_840_10008_5_1_4_1_1_12_2_1	
uid_1_2_840_10008_5_1_4_1_1_13_1_1	
uid_1_2_840_10008_5_1_4_1_1_13_1_2	
uid_1_2_840_10008_5_1_4_1_1_12_3	
uid_1_2_840_10008_5_1_4_1_1_20	
uid_1_2_840_10008_5_1_4_1_1_66	
uid_1_2_840_10008_5_1_4_1_1_66_1	
uid_1_2_840_10008_5_1_4_1_1_66_2	
uid_1_2_840_10008_5_1_4_1_1_66_3	
uid_1_2_840_10008_5_1_4_1_1_66_4	
uid_1_2_840_10008_5_1_4_1_1_67	
uid_1_2_840_10008_5_1_4_1_1_77_1	
uid_1_2_840_10008_5_1_4_1_1_77_2	
uid_1_2_840_10008_5_1_4_1_1_77_1_1	
uid_1_2_840_10008_5_1_4_1_1_77_1_1↩_1	
uid_1_2_840_10008_5_1_4_1_1_77_1_2	
uid_1_2_840_10008_5_1_4_1_1_77_1_2↩_1	
uid_1_2_840_10008_5_1_4_1_1_77_1_3	
uid_1_2_840_10008_5_1_4_1_1_77_1_4	
uid_1_2_840_10008_5_1_4_1_1_77_1_4↩_1	
uid_1_2_840_10008_5_1_4_1_1_77_1_5↩_1	
uid_1_2_840_10008_5_1_4_1_1_77_1_5↩_2	
uid_1_2_840_10008_5_1_4_1_1_77_1_5↩_3	
uid_1_2_840_10008_5_1_4_1_1_77_1_5↩_4	
uid_1_2_840_10008_5_1_4_1_1_88_1	
uid_1_2_840_10008_5_1_4_1_1_88_2	
uid_1_2_840_10008_5_1_4_1_1_88_3	
uid_1_2_840_10008_5_1_4_1_1_88_4	
uid_1_2_840_10008_5_1_4_1_1_88_11	
uid_1_2_840_10008_5_1_4_1_1_88_22	
uid_1_2_840_10008_5_1_4_1_1_88_33	
uid_1_2_840_10008_5_1_4_1_1_88_40	
uid_1_2_840_10008_5_1_4_1_1_88_50	
uid_1_2_840_10008_5_1_4_1_1_88_59	
uid_1_2_840_10008_5_1_4_1_1_88_65	
uid_1_2_840_10008_5_1_4_1_1_88_67	
uid_1_2_840_10008_5_1_4_1_1_104_1	

Enumerator

uid_1_2_840_10008_5_1_4_1_1_104_2	
uid_1_2_840_10008_5_1_4_1_1_128	
uid_1_2_840_10008_5_1_4_1_1_129	
uid_1_2_840_10008_5_1_4_1_1_481_1	
uid_1_2_840_10008_5_1_4_1_1_481_2	
uid_1_2_840_10008_5_1_4_1_1_481_3	
uid_1_2_840_10008_5_1_4_1_1_481_4	
uid_1_2_840_10008_5_1_4_1_1_481_5	
uid_1_2_840_10008_5_1_4_1_1_481_6	
uid_1_2_840_10008_5_1_4_1_1_481_7	
uid_1_2_840_10008_5_1_4_1_1_481_8	
uid_1_2_840_10008_5_1_4_1_1_481_9	
uid_1_2_840_10008_5_1_4_1_2_1_1	
uid_1_2_840_10008_5_1_4_1_2_1_2	
uid_1_2_840_10008_5_1_4_1_2_1_3	
uid_1_2_840_10008_5_1_4_1_2_2_1	
uid_1_2_840_10008_5_1_4_1_2_2_2	
uid_1_2_840_10008_5_1_4_1_2_2_3	
uid_1_2_840_10008_5_1_4_1_2_3_1	
uid_1_2_840_10008_5_1_4_1_2_3_2	
uid_1_2_840_10008_5_1_4_1_2_3_3	
uid_1_2_840_10008_5_1_4_31	
uid_1_2_840_10008_5_1_4_32_1	
uid_1_2_840_10008_5_1_4_32_2	
uid_1_2_840_10008_5_1_4_32_3	
uid_1_2_840_10008_5_1_4_32	
uid_1_2_840_10008_5_1_4_33	
uid_1_2_840_10008_5_1_4_34_1	
uid_1_2_840_10008_5_1_4_34_2	
uid_1_2_840_10008_5_1_4_34_3	
uid_1_2_840_10008_5_1_4_34_4	
uid_1_2_840_10008_5_1_4_34_4_1	
uid_1_2_840_10008_5_1_4_34_4_2	
uid_1_2_840_10008_5_1_4_34_4_3	
uid_1_2_840_10008_5_1_4_34_4_4	
uid_1_2_840_10008_5_1_4_34_5	
uid_1_2_840_10008_5_1_4_37_1	
uid_1_2_840_10008_5_1_4_37_2	
uid_1_2_840_10008_5_1_4_37_3	
uid_1_2_840_10008_5_1_4_38_1	
uid_1_2_840_10008_5_1_4_38_2	
uid_1_2_840_10008_5_1_4_38_3	
uid_1_2_840_10008_5_1_4_41	
uid_1_2_840_10008_5_1_4_42	
uid_1_2_840_10008_15_0_3_1	
uid_1_2_840_10008_15_0_3_2	

Enumerator

uid_1_2_840_10008_15_0_3_3	
uid_1_2_840_10008_15_0_3_4	
uid_1_2_840_10008_15_0_3_5	
uid_1_2_840_10008_15_0_3_6	
uid_1_2_840_10008_15_0_3_7	
uid_1_2_840_10008_15_0_3_8	
uid_1_2_840_10008_15_0_3_9	
uid_1_2_840_10008_15_0_3_10	
uid_1_2_840_10008_15_0_3_11	
uid_1_2_840_10008_15_0_3_12	
uid_1_2_840_10008_15_0_3_13	
uid_1_2_840_10008_15_0_3_14	
uid_1_2_840_10008_15_0_3_15	
uid_1_2_840_10008_15_0_3_16	
uid_1_2_840_10008_15_0_3_17	
uid_1_2_840_10008_15_0_3_18	
uid_1_2_840_10008_15_0_3_19	
uid_1_2_840_10008_15_0_3_20	
uid_1_2_840_10008_15_0_3_21	
uid_1_2_840_10008_15_0_3_22	
uid_1_2_840_10008_15_0_3_23	
uid_1_2_840_10008_15_0_3_24	
uid_1_2_840_10008_15_0_3_25	
uid_1_2_840_10008_15_0_3_26	
uid_1_2_840_10008_15_0_3_27	
uid_1_2_840_10008_15_0_3_28	
uid_1_2_840_10008_15_0_3_29	
uid_1_2_840_10008_15_0_3_30	
uid_1_2_840_10008_15_0_3_31	
uid_1_2_840_10008_15_0_4_1	
uid_1_2_840_10008_15_0_4_2	
uid_1_2_840_10008_15_0_4_3	
uid_1_2_840_10008_15_0_4_4	
uid_1_2_840_10008_15_0_4_5	
uid_1_2_840_10008_15_0_4_6	
uid_1_2_840_10008_15_0_4_7	
uid_1_2_840_10008_15_0_4_8	
uid_1_2_840_10008_5_1_4_1_1_77_1_6	
uid_1_2_840_10008_5_1_4_1_1_6_2	
uid_1_2_840_10008_5_1_4_1_1_66_5	
uid_1_2_840_10008_5_1_4_1_1_13_1_3	
uid_1_2_840_10008_5_1_4_1_1_2_2	
uid_1_2_840_10008_5_1_4_1_1_4_4	
uid_1_2_840_10008_5_1_4_1_1_128_1	
uid_1_2_840_10008_1_2_4_101	
uid_1_2_840_10008_1_2_4_102	

Enumerator

uid_1_2_840_10008_1_2_4_103	
uid_1_2_840_10008_1_5_2	
uid_1_2_840_10008_1_5_3	
uid_1_2_840_10008_1_5_4	
uid_1_2_840_10008_1_5_5	
uid_1_2_840_10008_1_5_6	
uid_1_2_840_10008_1_5_7	
uid_1_2_840_10008_1_5_8	
uid_1_2_840_10008_1_20	
uid_1_2_840_10008_2_16_5	
uid_1_2_840_10008_2_16_6	
uid_1_2_840_10008_2_16_7	
uid_1_2_840_10008_2_16_8	
uid_1_2_840_10008_2_16_9	
uid_1_2_840_10008_2_16_10	
uid_1_2_840_10008_2_16_11	
uid_1_2_840_10008_2_16_12	
uid_1_2_840_10008_2_16_13	
uid_1_2_840_10008_2_16_14	
uid_1_2_840_10008_5_1_1_40	
uid_1_2_840_10008_5_1_1_40_1	
uid_1_2_840_10008_5_1_4_1_1_9_4_2	
uid_1_2_840_10008_5_1_4_1_1_9_5_1	
uid_1_2_840_10008_5_1_4_1_1_9_6_1	
uid_1_2_840_10008_5_1_4_1_1_11_5	
uid_1_2_840_10008_5_1_4_1_1_11_6	
uid_1_2_840_10008_1_2_4_104	
uid_1_2_840_10008_1_2_4_105	
uid_1_2_840_10008_1_2_4_106	
uid_1_2_840_10008_1_2_4_107	
uid_1_2_840_10008_1_2_4_108	
uid_1_2_840_10008_1_5_1	
uid_1_2_840_10008_5_1_4_1_1_11_7	
uid_1_2_840_10008_5_1_4_1_1_11_8	
uid_1_2_840_10008_5_1_4_1_1_11_9	
uid_1_2_840_10008_5_1_4_1_1_11_10	
uid_1_2_840_10008_5_1_4_1_1_11_11	
uid_1_2_840_10008_5_1_4_1_1_12_77	
uid_1_2_840_10008_5_1_4_1_1_13_1_4	
uid_1_2_840_10008_5_1_4_1_1_13_1_5	
uid_1_2_840_10008_5_1_4_1_1_14_1	
uid_1_2_840_10008_5_1_4_1_1_14_2	
uid_1_2_840_10008_5_1_4_1_1_30	
uid_1_2_840_10008_5_1_4_1_1_40	
uid_1_2_840_10008_5_1_4_1_1_66_6	
uid_1_2_840_10008_5_1_4_1_1_68_1	

Enumerator

uid_1_2_840_10008_5_1_4_1_1_68_2	
uid_1_2_840_10008_5_1_4_1_1_77_1_5_5	
uid_1_2_840_10008_5_1_4_1_1_77_1_5_6	
uid_1_2_840_10008_5_1_4_1_1_77_1_5_7	
uid_1_2_840_10008_5_1_4_1_1_77_1_5_8	
uid_1_2_840_10008_5_1_4_1_1_78_1	
uid_1_2_840_10008_5_1_4_1_1_78_2	
uid_1_2_840_10008_5_1_4_1_1_78_3	
uid_1_2_840_10008_5_1_4_1_1_78_4	
uid_1_2_840_10008_5_1_4_1_1_78_5	
uid_1_2_840_10008_5_1_4_1_1_78_6	
uid_1_2_840_10008_5_1_4_1_1_78_7	
uid_1_2_840_10008_5_1_4_1_1_78_8	
uid_1_2_840_10008_5_1_4_1_1_79_1	
uid_1_2_840_10008_5_1_4_1_1_80_1	
uid_1_2_840_10008_5_1_4_1_1_81_1	
uid_1_2_840_10008_5_1_4_1_1_82_1	
uid_1_2_840_10008_5_1_4_1_1_88_34	
uid_1_2_840_10008_5_1_4_1_1_88_35	
uid_1_2_840_10008_5_1_4_1_1_88_68	
uid_1_2_840_10008_5_1_4_1_1_88_69	
uid_1_2_840_10008_5_1_4_1_1_88_70	
uid_1_2_840_10008_5_1_4_1_1_88_71	
uid_1_2_840_10008_5_1_4_1_1_88_72	
uid_1_2_840_10008_5_1_4_1_1_88_73	
uid_1_2_840_10008_5_1_4_1_1_88_74	
uid_1_2_840_10008_5_1_4_1_1_88_75	
uid_1_2_840_10008_5_1_4_1_1_90_1	
uid_1_2_840_10008_5_1_4_1_1_104_3	
uid_1_2_840_10008_5_1_4_1_1_130	
uid_1_2_840_10008_5_1_4_1_1_131	
uid_1_2_840_10008_5_1_4_1_1_200_1	
uid_1_2_840_10008_5_1_4_1_1_200_2	
uid_1_2_840_10008_5_1_4_1_1_200_3	
uid_1_2_840_10008_5_1_4_1_1_200_4	
uid_1_2_840_10008_5_1_4_1_1_200_5	
uid_1_2_840_10008_5_1_4_1_1_200_6	
uid_1_2_840_10008_5_1_4_1_1_481_10	
uid_1_2_840_10008_5_1_4_1_1_481_11	
uid_1_2_840_10008_5_1_4_1_1_501_1	
uid_1_2_840_10008_5_1_4_1_1_501_2_1	
uid_1_2_840_10008_5_1_4_1_1_501_2_2	
uid_1_2_840_10008_5_1_4_1_1_501_3	

Enumerator

uid_1_2_840_10008_5_1_4_1_1_501_4	
uid_1_2_840_10008_5_1_4_1_1_501_5	
uid_1_2_840_10008_5_1_4_1_1_501_6	
uid_1_2_840_10008_5_1_4_1_1_601_1	
uid_1_2_840_10008_5_1_4_1_1_601_2	
uid_1_2_840_10008_5_1_4_1_2_4_2	
uid_1_2_840_10008_5_1_4_1_2_4_3	
uid_1_2_840_10008_5_1_4_1_2_5_3	
uid_1_2_840_10008_5_1_4_20_1	
uid_1_2_840_10008_5_1_4_20_2	
uid_1_2_840_10008_5_1_4_20_3	
uid_1_2_840_10008_5_1_4_34_5_1	
uid_1_2_840_10008_5_1_4_34_6	
uid_1_2_840_10008_5_1_4_34_6_1	
uid_1_2_840_10008_5_1_4_34_6_2	
uid_1_2_840_10008_5_1_4_34_6_3	
uid_1_2_840_10008_5_1_4_34_6_4	
uid_1_2_840_10008_5_1_4_34_7	
uid_1_2_840_10008_5_1_4_34_8	
uid_1_2_840_10008_5_1_4_34_9	
uid_1_2_840_10008_5_1_4_34_10	
uid_1_2_840_10008_5_1_4_38_4	
uid_1_2_840_10008_5_1_4_39_1	
uid_1_2_840_10008_5_1_4_39_2	
uid_1_2_840_10008_5_1_4_39_3	
uid_1_2_840_10008_5_1_4_39_4	
uid_1_2_840_10008_5_1_4_43_1	
uid_1_2_840_10008_5_1_4_43_2	
uid_1_2_840_10008_5_1_4_43_3	
uid_1_2_840_10008_5_1_4_43_4	
uid_1_2_840_10008_5_1_4_44_1	
uid_1_2_840_10008_5_1_4_44_2	
uid_1_2_840_10008_5_1_4_44_3	
uid_1_2_840_10008_5_1_4_44_4	
uid_1_2_840_10008_5_1_4_45_1	
uid_1_2_840_10008_5_1_4_45_2	
uid_1_2_840_10008_5_1_4_45_3	
uid_1_2_840_10008_5_1_4_45_4	
uid_1_2_840_10008_7_1_1	
uid_1_2_840_10008_7_1_2	
uid_1_2_840_10008_8_1_1	
uid_1_2_840_10008_5_1_4_1_1_4_3	
uid_1_2_840_10008_15_1_1	

10.323.4 Member Function Documentation

10.323.4.1 GetName()

```
const char * gdcm::UIDs::GetName ( ) const
```

When object is Initialize function return the well known name associated with uid return NULL when not initialized

Examples

GenerateStandardSOPClasses.cxx.

Referenced by gdcm::operator<<().

10.323.4.2 GetNumberOfTransferSyntaxStrings()

```
static unsigned int gdcm::UIDs::GetNumberOfTransferSyntaxStrings ( )  [static]
```

10.323.4.3 GetString()

```
const char * gdcm::UIDs::GetString ( ) const
```

When object is Initialize function return the uid return NULL when not initialized

Examples

GenerateStandardSOPClasses.cxx.

Referenced by gdcm::operator<<().

10.323.4.4 GetTransferSyntaxString()

```
static const char *const * gdcm::UIDs::GetTransferSyntaxString (
            unsigned int ts )  [static]
```

10.323.4.5 GetTransferSyntaxStrings()

```
static TransferSyntaxStringsType gdcm::UIDs::GetTransferSyntaxStrings ( )  [static]
```

10.323.4.6 GetUIDName()

```
static const char * gdcm::UIDs::GetUIDName (
            unsigned int ts )  [static]
```

10.323.4.7 GetUIDString()

```
static const char * gdcm::UIDs::GetUIDString (
            unsigned int ts )  [static]
```

10.323.4.8 operator TSType()

```
gdcm::UIDs::operator TSType ( ) const  [inline]
```

10.323.4.9 SetFromUID()

```
bool gdcm::UIDs::SetFromUID (
            const char * str )
```

Initialize object from a string (a uid number) return false on error, and internal state is set to 0

Examples

GenerateStandardSOPClasses.cxx.

The documentation for this class was generated from the following file:

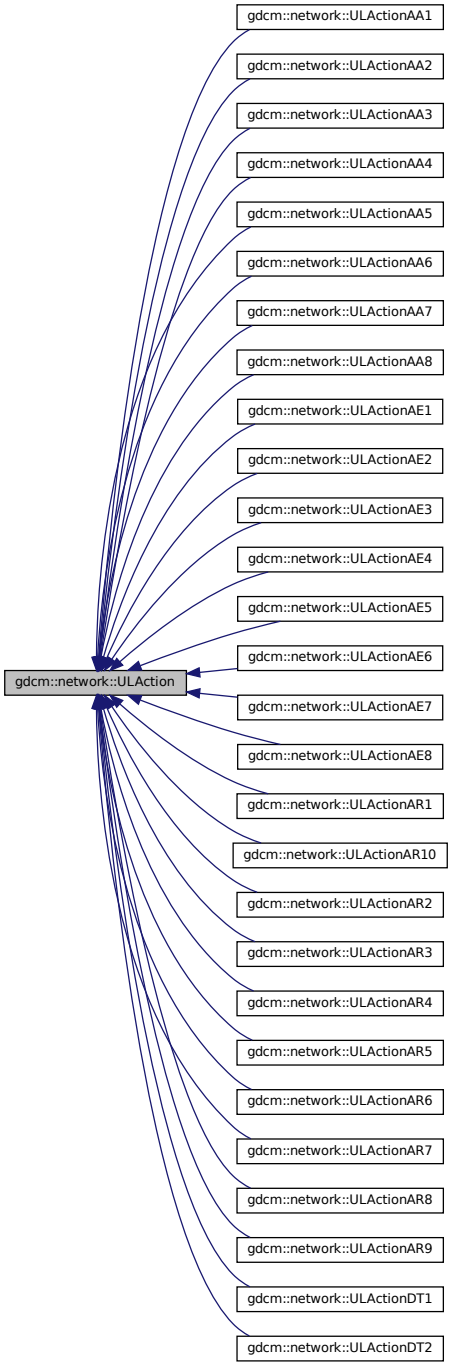
- gdcmUIDs.h

10.324 gdcm::network::ULAction Class Reference

ULAction.

`#include <gdcmULAction.h>`

Inheritance diagram for gdcm::network::ULAction:



Public Member Functions

- ULAction ()=default
- ULAction (const ULAction &inAction)=delete
- virtual ~ULAction ()=default
- void operator= (const ULAction &)=delete
- virtual EStateID PerformAction (Subject *s, ULEvent &inEvent, ULConnection &inConnection, bool &outWaiting↩ ForEvent, EEventID &outRaisedEvent)=0

10.324.1 Detailed Description

ULAction.

A ULConnection in a given ULState can perform certain ULActions. This base class provides the interface for running those ULActions on a given ULConnection.

Essentially, the ULConnectionManager will take this object, determined from the current ULState of the ULConnection, and pass the ULConnection object to the ULAction. The ULAction will then invoke whatever necessary commands are required by a given action.

The result of a ULAction is a ULEvent (ie, what happened as a result of the action).

This ULEvent is passed to the ULState, so that the transition to the next state can occur.

Actions are associated with Payloads – be those filestreams, AETitles to establish connections, whatever. The actual parameters that the user will pass via an action will come through a Payload object, which should, in itself, be some gdcm-based object (but not all objects can be payloads; sending a single dataelement as a payload isn't meaningful). As such, each action has its own particular payload.

For the sake of keeping files together, both the particular payload class and the action class will be defined in the same header file. Payloads should JUST be data (or streams), NO METHODS.

Some actions perform changes that should raise events on the local system, and some actions perform changes that will require waiting for events from the remote system.

Therefore, this base action has been modified so that those events are set by each action. When the event loop runs an action, it will then test to see if a local event was raised by the action, and if so, perform the appropriate subsequent action. If the action requires waiting for a response from the remote system, then the event loop will sit there (presumably with the ARTIM timer running) and wait for a response from the remote system. Once a response is obtained, then the the rest of the state transitions can happen.

10.324.2 Constructor & Destructor Documentation

10.324.2.1 ULAction() [1/2]

```
gdcm::network::ULAction::ULAction ( )  [default]
```

10.324.2.2 ~ULAction()

```
virtual gdcm::network::ULAction::~ULAction ( )  [virtual], [default]
```

10.324.2.3 ULAction() [2/2]

```
gdcm::network::ULAction::ULAction (
            const ULAction & inAction )  [delete]
```

10.324.3 Member Function Documentation

10.324.3.1 operator=()

```
void gdcm::network::ULAction::operator= (
            const ULAction &  )  [delete]
```

10.324.3.2 PerformAction()

```
virtual EStateID gdcm::network::ULAction::PerformAction (
            Subject * s,
            ULEvent & inEvent,
            ULConnection & inConnection,
            bool & outWaitingForEvent,
            EEventID & outRaisedEvent )  [pure virtual]
```

Implemented in gdcm::network::ULActionAA1, gdcm::network::ULActionAA2, gdcm::network::ULActionAA3, gdcm::network::ULActionAA4, gdcm::network::ULActionAA5, gdcm::network::ULActionAA6, gdcm::network::ULActionAA7, gdcm::network::ULActionAA8, gdcm::network::ULActionAE1, gdcm::network::ULActionAE2, gdcm::network::ULActionAE3, gdcm::network::ULActionAE4, gdcm::network::ULActionAE5, gdcm::network::ULActionAE6, gdcm::network::ULActionAE7, gdcm::network::ULActionAE8, gdcm::network::ULActionAR1, gdcm::network::ULActionAR2, gdcm::network::ULActionAR3, gdcm::network::ULActionAR4, gdcm::network::ULActionAR5, gdcm::network::ULActionAR6, gdcm::network::ULActionAR7, gdcm::network::ULActionAR8, gdcm::network::ULActionAR9, gdcm::network::ULActionAR10, gdcm::network::ULActionDT1, and gdcm::network::ULActionDT2.

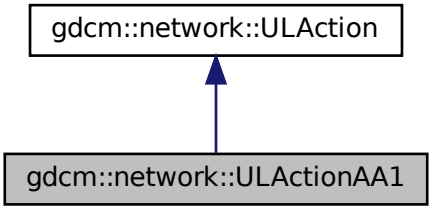
The documentation for this class was generated from the following file:

- gdcmULAction.h

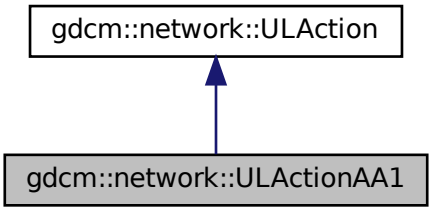
10.325 gdcm::network::ULActionAA1 Class Reference

`#include <gdcmULActionAA.h>`

Inheritance diagram for gdcm::network::ULActionAA1:



Collaboration diagram for gdcm::network::ULActionAA1:



Public Member Functions

- EStateID PerformAction (Subject *s, ULEvent &inEvent, ULConnection &inConnection, bool &outWaitingFor↩ Event, EEventID &outRaisedEvent) override

10.325.1 Member Function Documentation

10.325.1.1 PerformAction()

```
EStateID gdcm::network::ULActionAA1::PerformAction (
            Subject * s,
            ULEvent & inEvent,
            ULConnection & inConnection,
            bool & outWaitingForEvent,
            EEventID & outRaisedEvent )  [override], [virtual]
```

Implements gdcm::network::ULAction.

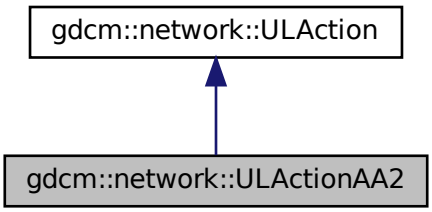
The documentation for this class was generated from the following file:

- gdcmULActionAA.h

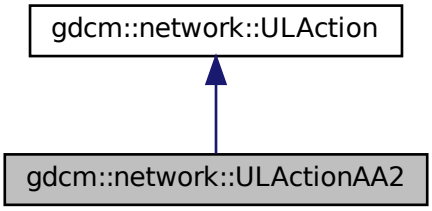
10.326 gdcm::network::ULActionAA2 Class Reference

```
#include <gdcmULActionAA.h>
```

Inheritance diagram for gdcm::network::ULActionAA2:



Collaboration diagram for gdcm::network::ULActionAA2:



Public Member Functions

- EStateID PerformAction (Subject *s, ULEvent &inEvent, ULConnection &inConnection, bool &outWaitingFor↩ Event, EEventID &outRaisedEvent) override

10.326.1 Member Function Documentation

10.326.1.1 PerformAction()

```
EStateID gdcm::network::ULActionAA2::PerformAction (
            Subject * s,
            ULEvent & inEvent,
            ULConnection & inConnection,
            bool & outWaitingForEvent,
            EEventID & outRaisedEvent )  [override], [virtual]
```

Implements gdcm::network::ULAction.

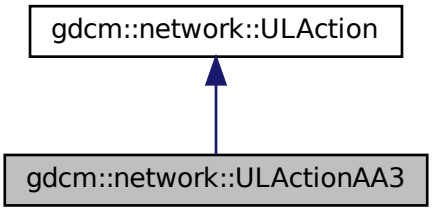
The documentation for this class was generated from the following file:

- gdcmULActionAA.h

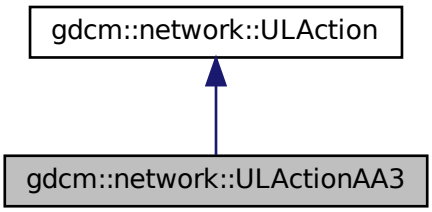
10.327 gdcm::network::ULActionAA3 Class Reference

```
#include <gdcmULActionAA.h>
```

Inheritance diagram for gdcm::network::ULActionAA3:



Collaboration diagram for gdcm::network::ULActionAA3:



Public Member Functions

- EStateID PerformAction (Subject *s, ULEvent &inEvent, ULConnection &inConnection, bool &outWaitingFor↩ Event, EEventID &outRaisedEvent) override

10.327.1 Member Function Documentation

10.327.1.1 PerformAction()

```
EStateID gdcm::network::ULActionAA3::PerformAction (
            Subject * s,
            ULEvent & inEvent,
            ULConnection & inConnection,
            bool & outWaitingForEvent,
            EEventID & outRaisedEvent )  [override], [virtual]
```

Implements gdcm::network::ULAction.

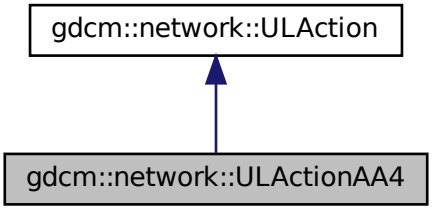
The documentation for this class was generated from the following file:

- gdcmULActionAA.h

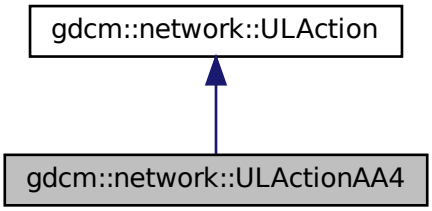
10.328 gdcm::network::ULActionAA4 Class Reference

`#include <gdcmULActionAA.h>`

Inheritance diagram for gdcm::network::ULActionAA4:



Collaboration diagram for gdcm::network::ULActionAA4:



Public Member Functions

- EStateID PerformAction (Subject *s, ULEvent &inEvent, ULConnection &inConnection, bool &outWaitingFor↩ Event, EEventID &outRaisedEvent) override

10.328.1 Member Function Documentation

10.328.1.1 PerformAction()

```
EStateID gdcm::network::ULActionAA4::PerformAction (
            Subject * s,
            ULEvent & inEvent,
            ULConnection & inConnection,
            bool & outWaitingForEvent,
            EEventID & outRaisedEvent )  [override], [virtual]
```

Implements gdcm::network::ULAction.

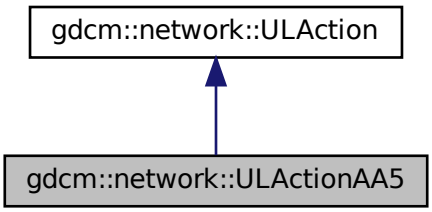
The documentation for this class was generated from the following file:

- gdcmULActionAA.h

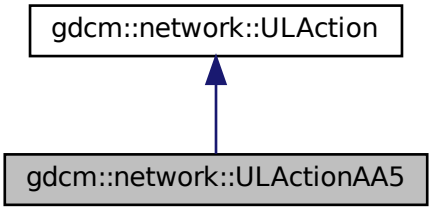
10.329 gdcm::network::ULActionAA5 Class Reference

```
#include <gdcmULActionAA.h>
```

Inheritance diagram for gdcm::network::ULActionAA5:



Collaboration diagram for gdcm::network::ULActionAA5:



Public Member Functions

- EStateID PerformAction (Subject *s, ULEvent &inEvent, ULConnection &inConnection, bool &outWaitingFor↩ Event, EEventID &outRaisedEvent) override

10.329.1 Member Function Documentation

10.329.1.1 PerformAction()

```
EStateID gdcm::network::ULActionAA5::PerformAction (
            Subject * s,
            ULEvent & inEvent,
            ULConnection & inConnection,
            bool & outWaitingForEvent,
            EEventID & outRaisedEvent )  [override], [virtual]
```

Implements gdcm::network::ULAction.

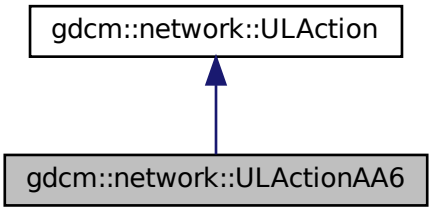
The documentation for this class was generated from the following file:

- gdcmULActionAA.h

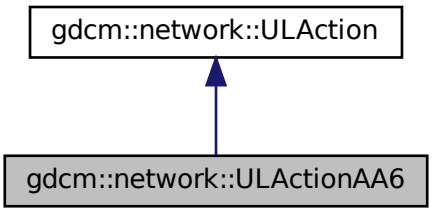
10.330 gdcm::network::ULActionAA6 Class Reference

```
#include <gdcmULActionAA.h>
```

Inheritance diagram for gdcm::network::ULActionAA6:



Collaboration diagram for gdcm::network::ULActionAA6:



Public Member Functions

- EStateID PerformAction (Subject *s, ULEvent &inEvent, ULConnection &inConnection, bool &outWaitingFor↩ Event, EEventID &outRaisedEvent) override

10.330.1 Member Function Documentation

10.330.1.1 PerformAction()

```
EStateID gdcm::network::ULActionAA6::PerformAction (
            Subject * s,
            ULEvent & inEvent,
            ULConnection & inConnection,
            bool & outWaitingForEvent,
            EEventID & outRaisedEvent )  [override], [virtual]
```

Implements gdcm::network::ULAction.

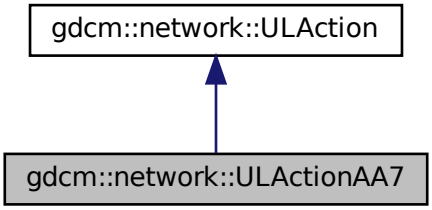
The documentation for this class was generated from the following file:

- gdcmULActionAA.h

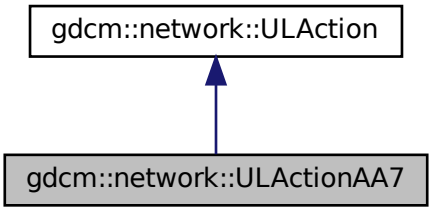
10.331 gdcm::network::ULActionAA7 Class Reference

`#include <gdcmULActionAA.h>`

Inheritance diagram for gdcm::network::ULActionAA7:



Collaboration diagram for gdcm::network::ULActionAA7:



Public Member Functions

- EStateID PerformAction (Subject *s, ULEvent &inEvent, ULConnection &inConnection, bool &outWaitingForEvent, EEventID &outRaisedEvent) override

10.331.1 Member Function Documentation

10.331.1.1 PerformAction()

```
EStateID gdcm::network::ULActionAA7::PerformAction (
            Subject * s,
            ULEvent & inEvent,
            ULConnection & inConnection,
            bool & outWaitingForEvent,
            EEventID & outRaisedEvent )  [override], [virtual]
```

Implements gdcm::network::ULAction.

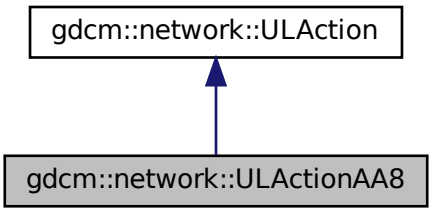
The documentation for this class was generated from the following file:

- gdcmULActionAA.h

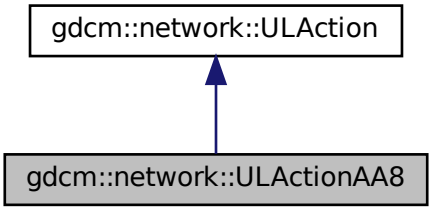
10.332 gdcm::network::ULActionAA8 Class Reference

```
#include <gdcmULActionAA.h>
```

Inheritance diagram for gdcm::network::ULActionAA8:



Collaboration diagram for gdcm::network::ULActionAA8:



Public Member Functions

- EStateID PerformAction (Subject *s, ULEvent &inEvent, ULConnection &inConnection, bool &outWaitingFor↩ Event, EEventID &outRaisedEvent) override

10.332.1 Member Function Documentation

10.332.1.1 PerformAction()

```
EStateID gdcm::network::ULActionAA8::PerformAction (
            Subject * s,
            ULEvent & inEvent,
            ULConnection & inConnection,
            bool & outWaitingForEvent,
            EEventID & outRaisedEvent )  [override], [virtual]
```

Implements gdcm::network::ULAction.

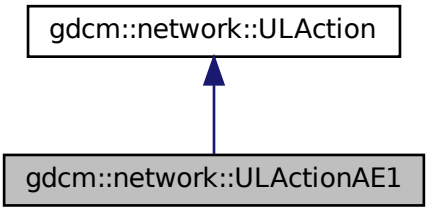
The documentation for this class was generated from the following file:

- gdcmULActionAA.h

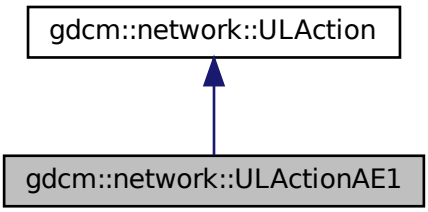
10.333 gdcm::network::ULActionAE1 Class Reference

```
#include <gdcmULActionAE.h>
```

Inheritance diagram for gdcm::network::ULActionAE1:



Collaboration diagram for gdcm::network::ULActionAE1:



Public Member Functions

- EStateID PerformAction (Subject *s, ULEvent &inEvent, ULConnection &inConnection, bool &outWaitingFor↩ Event, EEventID &outRaisedEvent) override

10.333.1 Member Function Documentation

10.333.1.1 PerformAction()

```
EStateID gdcm::network::ULActionAE1::PerformAction (
            Subject * s,
            ULEvent & inEvent,
            ULConnection & inConnection,
            bool & outWaitingForEvent,
            EEventID & outRaisedEvent )  [override], [virtual]
```

Implements gdcm::network::ULAction.

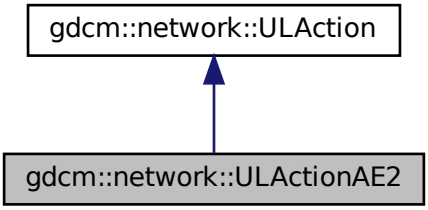
The documentation for this class was generated from the following file:

- gdcmULActionAE.h

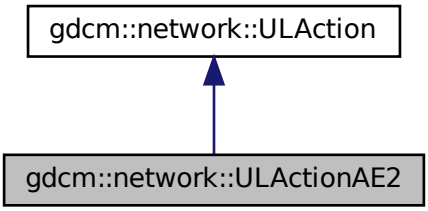
10.334 gdcm::network::ULActionAE2 Class Reference

`#include <gdcmULActionAE.h>`

Inheritance diagram for gdcm::network::ULActionAE2:



Collaboration diagram for gdcm::network::ULActionAE2:



Public Member Functions

- EStateID PerformAction (Subject *s, ULEvent &inEvent, ULConnection &inConnection, bool &outWaitingForEvent, EEventID &outRaisedEvent) override

10.334.1 Member Function Documentation

10.334.1.1 PerformAction()

```
EStateID gdcm::network::ULActionAE2::PerformAction (
            Subject * s,
            ULEvent & inEvent,
            ULConnection & inConnection,
            bool & outWaitingForEvent,
            EEventID & outRaisedEvent )  [override], [virtual]
```

Implements gdcm::network::ULAction.

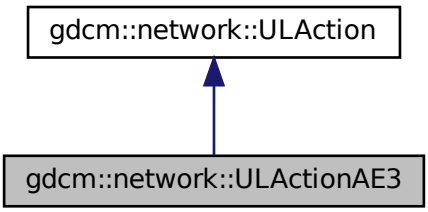
The documentation for this class was generated from the following file:

- gdcmULActionAE.h

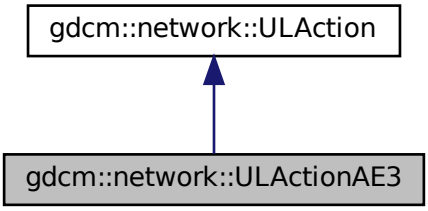
10.335 gdcm::network::ULActionAE3 Class Reference

```
#include <gdcmULActionAE.h>
```

Inheritance diagram for gdcm::network::ULActionAE3:



Collaboration diagram for gdcm::network::ULActionAE3:



Public Member Functions

- EStateID PerformAction (Subject *s, ULEvent &inEvent, ULConnection &inConnection, bool &outWaitingFor↩ Event, EEventID &outRaisedEvent) override

10.335.1 Member Function Documentation

10.335.1.1 PerformAction()

```
EStateID gdcm::network::ULActionAE3::PerformAction (
            Subject * s,
            ULEvent & inEvent,
            ULConnection & inConnection,
            bool & outWaitingForEvent,
            EEventID & outRaisedEvent )  [override], [virtual]
```

Implements gdcm::network::ULAction.

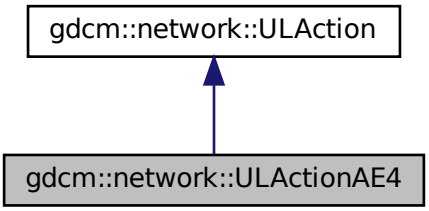
The documentation for this class was generated from the following file:

- gdcmULActionAE.h

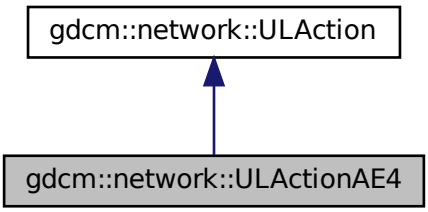
10.336 gdcm::network::ULActionAE4 Class Reference

```
#include <gdcmULActionAE.h>
```

Inheritance diagram for gdcm::network::ULActionAE4:



Collaboration diagram for gdcm::network::ULActionAE4:



Public Member Functions

- EStateID PerformAction (Subject *s, ULEvent &inEvent, ULConnection &inConnection, bool &outWaitingFor↩ Event, EEventID &outRaisedEvent) override

10.336.1 Member Function Documentation

10.336.1.1 PerformAction()

```
EStateID gdcm::network::ULActionAE4::PerformAction (
            Subject * s,
            ULEvent & inEvent,
            ULConnection & inConnection,
            bool & outWaitingForEvent,
            EEventID & outRaisedEvent )  [override], [virtual]
```

Implements gdcm::network::ULAction.

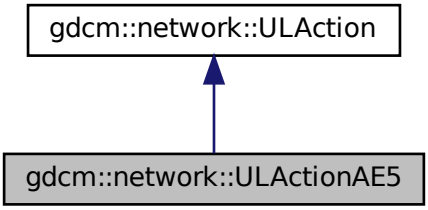
The documentation for this class was generated from the following file:

- gdcmULActionAE.h

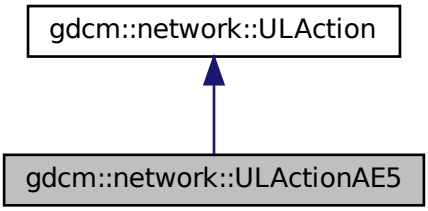
10.337 gdcm::network::ULActionAE5 Class Reference

`#include <gdcmULActionAE.h>`

Inheritance diagram for gdcm::network::ULActionAE5:



Collaboration diagram for gdcm::network::ULActionAE5:



Public Member Functions

- EStateID PerformAction (Subject *s, ULEvent &inEvent, ULConnection &inConnection, bool &outWaitingForEvent, EEventID &outRaisedEvent) override

10.337.1 Member Function Documentation

10.337.1.1 PerformAction()

```
EStateID gdcm::network::ULActionAE5::PerformAction (
            Subject * s,
            ULEvent & inEvent,
            ULConnection & inConnection,
            bool & outWaitingForEvent,
            EEventID & outRaisedEvent )  [override], [virtual]
```

Implements gdcm::network::ULAction.

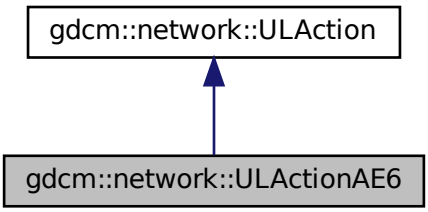
The documentation for this class was generated from the following file:

- gdcmULActionAE.h

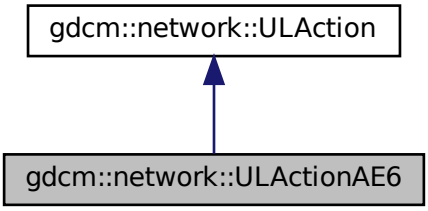
10.338 gdcm::network::ULActionAE6 Class Reference

```
#include <gdcmULActionAE.h>
```

Inheritance diagram for gdcm::network::ULActionAE6:



Collaboration diagram for gdcm::network::ULActionAE6:



Public Member Functions

- EStateID PerformAction (Subject *s, ULEvent &inEvent, ULConnection &inConnection, bool &outWaitingFor↩ Event, EEventID &outRaisedEvent) override

10.338.1 Member Function Documentation

10.338.1.1 PerformAction()

```
EStateID gdcm::network::ULActionAE6::PerformAction (
            Subject * s,
            ULEvent & inEvent,
            ULConnection & inConnection,
            bool & outWaitingForEvent,
            EEventID & outRaisedEvent )  [override], [virtual]
```

Implements gdcm::network::ULAction.

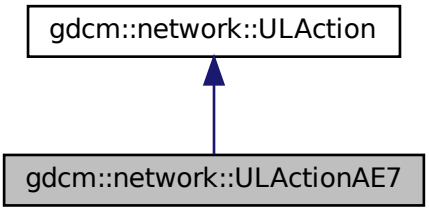
The documentation for this class was generated from the following file:

- gdcmULActionAE.h

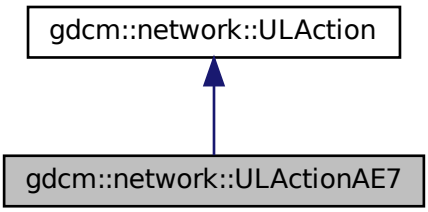
10.339 gdcm::network::ULActionAE7 Class Reference

```
#include <gdcmULActionAE.h>
```

Inheritance diagram for gdcm::network::ULActionAE7:



Collaboration diagram for gdcm::network::ULActionAE7:



Public Member Functions

- EStateID PerformAction (Subject *s, ULEvent &inEvent, ULConnection &inConnection, bool &outWaitingFor↩ Event, EEventID &outRaisedEvent) override

10.339.1 Member Function Documentation

10.339.1.1 PerformAction()

```
EStateID gdcm::network::ULActionAE7::PerformAction (
            Subject * s,
            ULEvent & inEvent,
            ULConnection & inConnection,
            bool & outWaitingForEvent,
            EEventID & outRaisedEvent )  [override], [virtual]
```

Implements gdcm::network::ULAction.

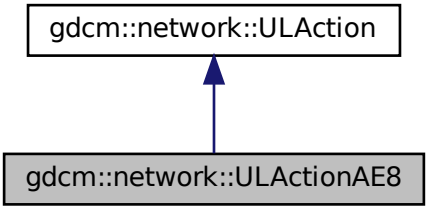
The documentation for this class was generated from the following file:

- gdcmULActionAE.h

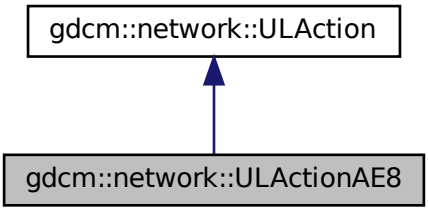
10.340 gdcm::network::ULActionAE8 Class Reference

`#include <gdcmULActionAE.h>`

Inheritance diagram for gdcm::network::ULActionAE8:



Collaboration diagram for gdcm::network::ULActionAE8:



Public Member Functions

- EStateID PerformAction (Subject *s, ULEvent &inEvent, ULConnection &inConnection, bool &outWaitingForEvent, EEventID &outRaisedEvent) override

10.340.1 Member Function Documentation

10.340.1.1 PerformAction()

```
EStateID gdcm::network::ULActionAE8::PerformAction (
            Subject * s,
            ULEvent & inEvent,
            ULConnection & inConnection,
            bool & outWaitingForEvent,
            EEventID & outRaisedEvent )  [override], [virtual]
```

Implements gdcm::network::ULAction.

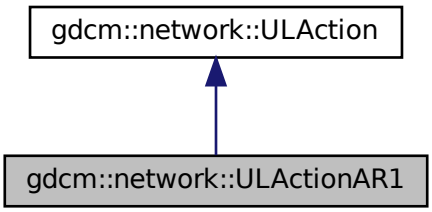
The documentation for this class was generated from the following file:

- gdcmULActionAE.h

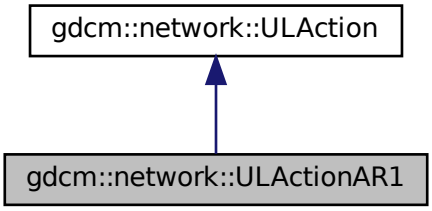
10.341 gdcm::network::ULActionAR1 Class Reference

```
#include <gdcmULActionAR.h>
```

Inheritance diagram for gdcm::network::ULActionAR1:



Collaboration diagram for gdcm::network::ULActionAR1:



Public Member Functions

- EStateID PerformAction (Subject *s, ULEvent &inEvent, ULConnection &inConnection, bool &outWaitingFor↩ Event, EEventID &outRaisedEvent) override

10.341.1 Member Function Documentation

10.341.1.1 PerformAction()

```
EStateID gdcm::network::ULActionAR1::PerformAction (
            Subject * s,
            ULEvent & inEvent,
            ULConnection & inConnection,
            bool & outWaitingForEvent,
            EEventID & outRaisedEvent )  [override], [virtual]
```

Implements gdcm::network::ULAction.

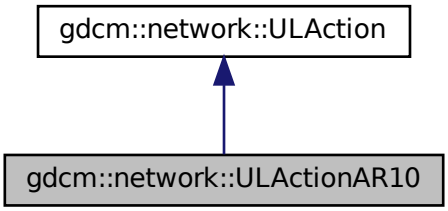
The documentation for this class was generated from the following file:

- gdcmULActionAR.h

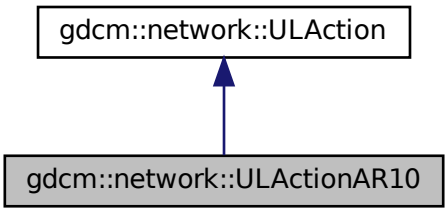
10.342 gdcm::network::ULActionAR10 Class Reference

```
#include <gdcmULActionAR.h>
```

Inheritance diagram for gdcm::network::ULActionAR10:



Collaboration diagram for gdcm::network::ULActionAR10:



Public Member Functions

- EStateID PerformAction (Subject *s, ULEvent &inEvent, ULConnection &inConnection, bool &outWaitingFor↩ Event, EEventID &outRaisedEvent) override

10.342.1 Member Function Documentation

10.342.1.1 PerformAction()

```
EStateID gdcm::network::ULActionAR10::PerformAction (
            Subject * s,
            ULEvent & inEvent,
            ULConnection & inConnection,
            bool & outWaitingForEvent,
            EEventID & outRaisedEvent )  [override], [virtual]
```

Implements gdcm::network::ULAction.

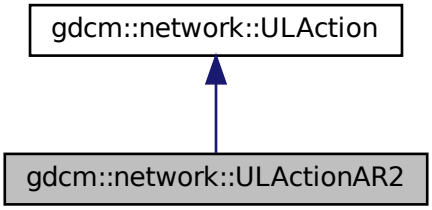
The documentation for this class was generated from the following file:

- gdcmULActionAR.h

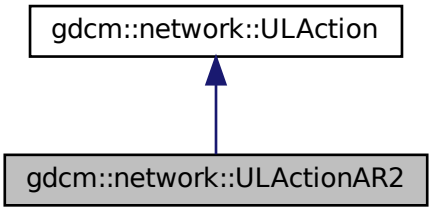
10.343 gdcm::network::ULActionAR2 Class Reference

`#include <gdcmULActionAR.h>`

Inheritance diagram for gdcm::network::ULActionAR2:



Collaboration diagram for gdcm::network::ULActionAR2:



Public Member Functions

- EStateID PerformAction (Subject *s, ULEvent &inEvent, ULConnection &inConnection, bool &outWaitingForEvent, EEventID &outRaisedEvent) override

10.343.1 Member Function Documentation

10.343.1.1 PerformAction()

```
EStateID gdcm::network::ULActionAR2::PerformAction (
            Subject * s,
            ULEvent & inEvent,
            ULConnection & inConnection,
            bool & outWaitingForEvent,
            EEventID & outRaisedEvent )  [override], [virtual]
```

Implements gdcm::network::ULAction.

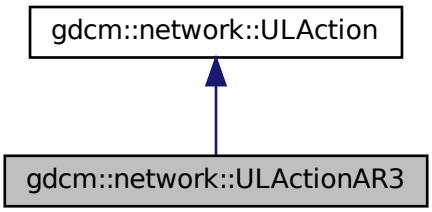
The documentation for this class was generated from the following file:

- gdcmULActionAR.h

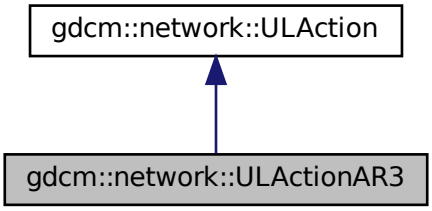
10.344 gdcm::network::ULActionAR3 Class Reference

```
#include <gdcmULActionAR.h>
```

Inheritance diagram for gdcm::network::ULActionAR3:



Collaboration diagram for gdcm::network::ULActionAR3:



Public Member Functions

- EStateID PerformAction (Subject *s, ULEvent &inEvent, ULConnection &inConnection, bool &outWaitingFor↩ Event, EEventID &outRaisedEvent) override

10.344.1 Member Function Documentation

10.344.1.1 PerformAction()

```
EStateID gdcm::network::ULActionAR3::PerformAction (
            Subject * s,
            ULEvent & inEvent,
            ULConnection & inConnection,
            bool & outWaitingForEvent,
            EEventID & outRaisedEvent )  [override], [virtual]
```

Implements gdcm::network::ULAction.

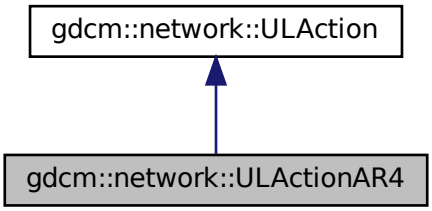
The documentation for this class was generated from the following file:

- gdcmULActionAR.h

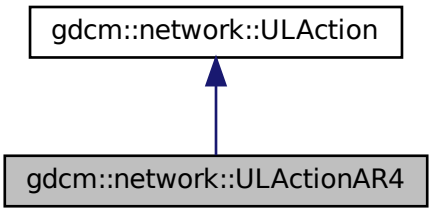
10.345 gdcm::network::ULActionAR4 Class Reference

```
#include <gdcmULActionAR.h>
```

Inheritance diagram for gdcm::network::ULActionAR4:



Collaboration diagram for gdcm::network::ULActionAR4:



Public Member Functions

- EStateID PerformAction (Subject *s, ULEvent &inEvent, ULConnection &inConnection, bool &outWaitingFor↩ Event, EEventID &outRaisedEvent) override

10.345.1 Member Function Documentation

10.345.1.1 PerformAction()

```
EStateID gdcm::network::ULActionAR4::PerformAction (
            Subject * s,
            ULEvent & inEvent,
            ULConnection & inConnection,
            bool & outWaitingForEvent,
            EEventID & outRaisedEvent )  [override], [virtual]
```

Implements gdcm::network::ULAction.

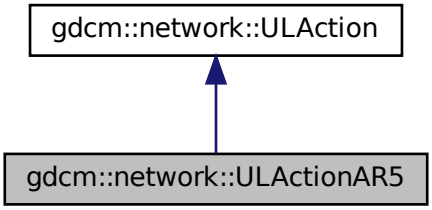
The documentation for this class was generated from the following file:

- gdcmULActionAR.h

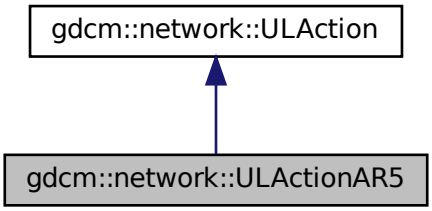
10.346 gdcm::network::ULActionAR5 Class Reference

`#include <gdcmULActionAR.h>`

Inheritance diagram for gdcm::network::ULActionAR5:



Collaboration diagram for gdcm::network::ULActionAR5:



Public Member Functions

- EStateID PerformAction (Subject *s, ULEvent &inEvent, ULConnection &inConnection, bool &outWaitingForEvent, EEventID &outRaisedEvent) override

10.346.1 Member Function Documentation

10.346.1.1 PerformAction()

```
EStateID gdcm::network::ULActionAR5::PerformAction (
            Subject * s,
            ULEvent & inEvent,
            ULConnection & inConnection,
            bool & outWaitingForEvent,
            EEventID & outRaisedEvent )  [override], [virtual]
```

Implements gdcm::network::ULAction.

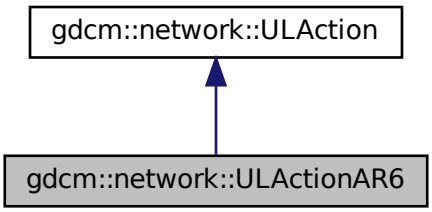
The documentation for this class was generated from the following file:

- gdcmULActionAR.h

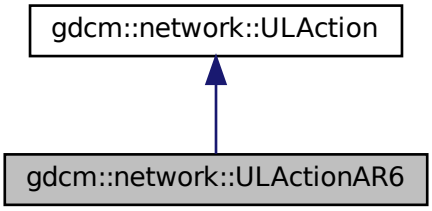
10.347 gdcm::network::ULActionAR6 Class Reference

```
#include <gdcmULActionAR.h>
```

Inheritance diagram for gdcm::network::ULActionAR6:



Collaboration diagram for gdcm::network::ULActionAR6:



Public Member Functions

- EStateID PerformAction (Subject *s, ULEvent &inEvent, ULConnection &inConnection, bool &outWaitingFor↩ Event, EEventID &outRaisedEvent) override

10.347.1 Member Function Documentation

10.347.1.1 PerformAction()

```
EStateID gdcm::network::ULActionAR6::PerformAction (
            Subject * s,
            ULEvent & inEvent,
            ULConnection & inConnection,
            bool & outWaitingForEvent,
            EEventID & outRaisedEvent )  [override], [virtual]
```

Implements gdcm::network::ULAction.

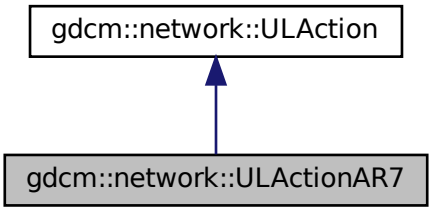
The documentation for this class was generated from the following file:

- gdcmULActionAR.h

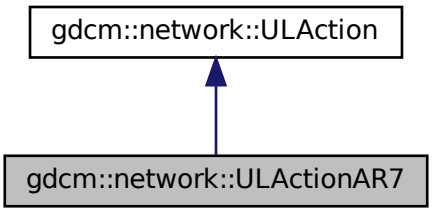
10.348 gdcm::network::ULActionAR7 Class Reference

```
#include <gdcmULActionAR.h>
```

Inheritance diagram for gdcm::network::ULActionAR7:



Collaboration diagram for gdcm::network::ULActionAR7:



Public Member Functions

- EStateID PerformAction (Subject *s, ULEvent &inEvent, ULConnection &inConnection, bool &outWaitingFor↩ Event, EEventID &outRaisedEvent) override

10.348.1 Member Function Documentation

10.348.1.1 PerformAction()

```
EStateID gdcm::network::ULActionAR7::PerformAction (
            Subject * s,
            ULEvent & inEvent,
            ULConnection & inConnection,
            bool & outWaitingForEvent,
            EEventID & outRaisedEvent )  [override], [virtual]
```

Implements gdcm::network::ULAction.

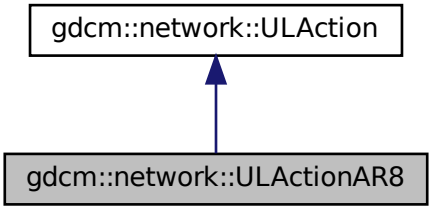
The documentation for this class was generated from the following file:

- gdcmULActionAR.h

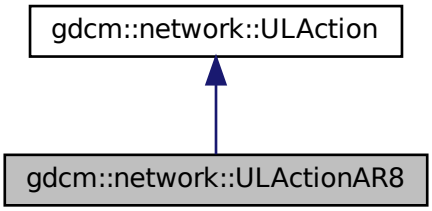
10.349 gdcm::network::ULActionAR8 Class Reference

`#include <gdcmULActionAR.h>`

Inheritance diagram for gdcm::network::ULActionAR8:



Collaboration diagram for gdcm::network::ULActionAR8:



Public Member Functions

- EStateID PerformAction (Subject *s, ULEvent &inEvent, ULConnection &inConnection, bool &outWaitingForEvent, EEventID &outRaisedEvent) override

10.349.1 Member Function Documentation

10.349.1.1 PerformAction()

```
EStateID gdcm::network::ULActionAR8::PerformAction (
            Subject * s,
            ULEvent & inEvent,
            ULConnection & inConnection,
            bool & outWaitingForEvent,
            EEventID & outRaisedEvent )  [override], [virtual]
```

Implements gdcm::network::ULAction.

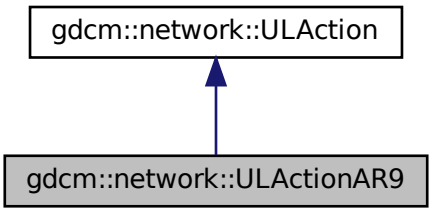
The documentation for this class was generated from the following file:

- gdcmULActionAR.h

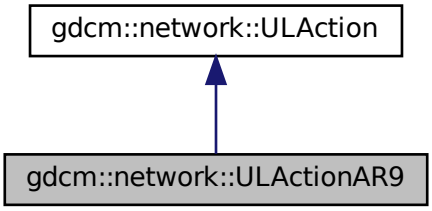
10.350 gdcm::network::ULActionAR9 Class Reference

```
#include <gdcmULActionAR.h>
```

Inheritance diagram for gdcm::network::ULActionAR9:



Collaboration diagram for gdcm::network::ULActionAR9:



Public Member Functions

- EStateID PerformAction (Subject *s, ULEvent &inEvent, ULConnection &inConnection, bool &outWaitingFor↩ Event, EEventID &outRaisedEvent) override

10.350.1 Member Function Documentation

10.350.1.1 PerformAction()

```
EStateID gdcm::network::ULActionAR9::PerformAction (
            Subject * s,
            ULEvent & inEvent,
            ULConnection & inConnection,
            bool & outWaitingForEvent,
            EEventID & outRaisedEvent )  [override], [virtual]
```

Implements gdcm::network::ULAction.

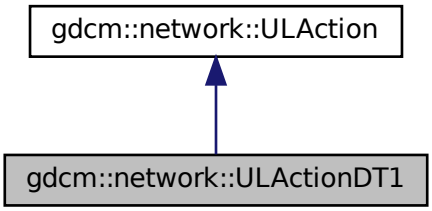
The documentation for this class was generated from the following file:

- gdcmULActionAR.h

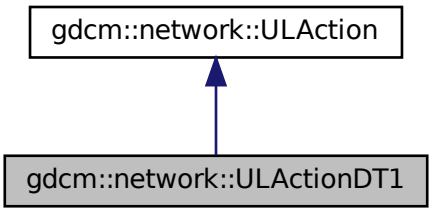
10.351 gdcm::network::ULActionDT1 Class Reference

```
#include <gdcmULActionDT.h>
```

Inheritance diagram for gdcm::network::ULActionDT1:



Collaboration diagram for gdcm::network::ULActionDT1:



Public Member Functions

- EStateID PerformAction (Subject *s, ULEvent &inEvent, ULConnection &inConnection, bool &outWaitingFor↩ Event, EEventID &outRaisedEvent) override

10.351.1 Member Function Documentation

10.351.1.1 PerformAction()

```
EStateID gdcm::network::ULActionDT1::PerformAction (
            Subject * s,
            ULEvent & inEvent,
            ULConnection & inConnection,
            bool & outWaitingForEvent,
            EEventID & outRaisedEvent )  [override], [virtual]
```

Implements gdcm::network::ULAction.

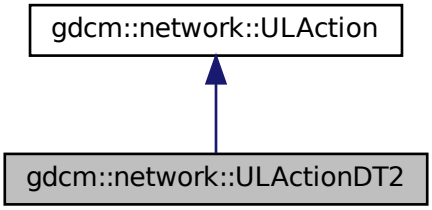
The documentation for this class was generated from the following file:

- gdcmULActionDT.h

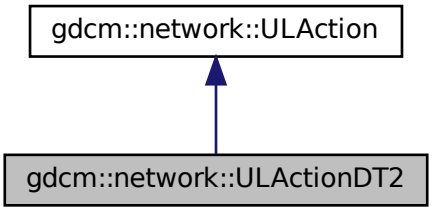
10.352 gdcm::network::ULActionDT2 Class Reference

`#include <gdcmULActionDT.h>`

Inheritance diagram for gdcm::network::ULActionDT2:



Collaboration diagram for gdcm::network::ULActionDT2:



Public Member Functions

- EStateID PerformAction (Subject *s, ULEvent &inEvent, ULConnection &inConnection, bool &outWaitingForEvent, EEventID &outRaisedEvent) override

10.352.1 Member Function Documentation

10.352.1.1 PerformAction()

```
EStateID gdcm::network::ULActionDT2::PerformAction (
            Subject * s,
            ULEvent & inEvent,
            ULConnection & inConnection,
            bool & outWaitingForEvent,
            EEventID & outRaisedEvent )  [override], [virtual]
```

Implements gdcm::network::ULAction.

The documentation for this class was generated from the following file:

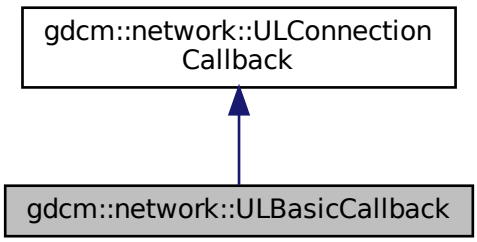
- gdcmULActionDT.h

10.353 gdcm::network::ULBasicCallback Class Reference

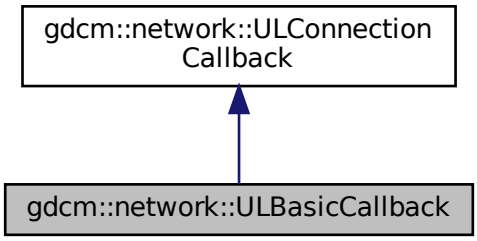
ULBasicCallback.

`#include <gdcmULBasicCallback.h>`

Inheritance diagram for gdcm::network::ULBasicCallback:



Collaboration diagram for gdcm::network::ULBasicCallback:



Public Member Functions

- ULBasicCallback ()=default
- ~ULBasicCallback () override=default
- std::vector< DataSet > const & GetDataSets () const
- std::vector< DataSet > const & GetResponses () const
- void HandleDataSet (const DataSet &inDataSet) override
- void HandleResponse (const DataSet &inDataSet) override

Additional Inherited Members

10.353.1 Detailed Description

ULBasicCallback.

This is the most basic of callbacks for how the ULConnectionManager handles incoming datasets. DataSets are just concatenated to the mDataSets vector, and the result can be pulled out of the vector by later code. Alternatives to this method include progress updates, saving to disk, etc. This class is NOT THREAD SAFE. Access the dataset vector after the entire set of datasets has been returned by the ULConnectionManager.

10.353.2 Constructor & Destructor Documentation

10.353.2.1 ULBasicCallback()

```
gdcm::network::ULBasicCallback::ULBasicCallback ( )  [default]
```

10.353.2.2 ~ULBasicCallback()

```
gdcm::network::ULBasicCallback::~ULBasicCallback ( )  [override], [default]
```

10.353.3 Member Function Documentation

10.353.3.1 GetDataSets()

```
std::vector< DataSet > const & gdcm::network::ULBasicCallback::GetDataSets ( ) const
```

10.353.3.2 GetResponses()

```
std::vector< DataSet > const & gdcm::network::ULBasicCallback::GetResponses ( ) const
```

10.353.3.3 HandleDataSet()

```
void gdcm::network::ULBasicCallback::HandleDataSet (
            const DataSet & inDataSet )  [override], [virtual]
```

Implements gdcm::network::ULConnectionCallback.

10.353.3.4 HandleResponse()

```
void gdcm::network::ULBasicCallback::HandleResponse (
            const DataSet & inDataSet )  [override], [virtual]
```

Implements gdcm::network::ULConnectionCallback.

The documentation for this class was generated from the following file:

- gdcmULBasicCallback.h

10.354 gdcm::network::ULConnection Class Reference

ULConnection.

```
#include <gdcmULConnection.h>
```

Public Member Functions

- ULConnection (const ULConnection &)=delete
- ULConnection (const ULConnectionInfo &inUserInformation)
- virtual ~ULConnection ()
- void AddAcceptedPresentationContext (const PresentationContextAC &inPC)
- PresentationContextRQ FindContext (const DataElement &de) const
- std::vector< PresentationContextAC > & GetAcceptedPresentationContexts ()
- std::vector< PresentationContextAC > const & GetAcceptedPresentationContexts () const
- const ULConnectionInfo & GetConnectionInfo () const
- uint32_t GetMaxPDUSize () const
- const PresentationContextAC * GetPresentationContextACByID (uint8_t id) const
- uint8_t GetPresentationContextIDFromPresentationContext (PresentationContextRQ const &pc) const

  return 0 upon error
- const PresentationContextRQ * GetPresentationContextRQByID (uint8_t id) const
- std::vector< PresentationContextRQ > const & GetPresentationContexts () const
- std::iostream * GetProtocol ()
- EStateID GetState () const
- ARTIMTimer & GetTimer ()
- bool InitializeConnection ()

  used to establish scu connections
- bool InitializeIncomingConnection ()

  used to establish scp connections
- void operator= (const ULConnection &)=delete
- void SetMaxPDUSize (uint32_t inSize)
- void SetPresentationContexts (const std::vector< PresentationContext > &inContexts)
- void SetPresentationContexts (const std::vector< PresentationContextRQ > &inContexts)
- void SetState (const EStateID &inState)
- void StopProtocol ()

Friends

- class ULActionAE6
- class ULConnectionManager

10.354.1 Detailed Description

ULConnection.

This is the class that contains the socket to another machine, and passes data through itself, as well as maintaining a sense of state.

The ULConnectionManager tells the ULConnection what data can actually be sent.

This class is done this way so that it can be eventually be replaced with a ULSecureConnection, if such a protocol is warranted, so that all data that passes through can be managed through a secure connection. For now, this class provides a simple pass-through mechanism to the socket itself.

So, for instance, a gdcm object will be passes to this object, and it will then get passed along the connection, if that connection is in the proper state to do so.

For right now, this class is not directly intended to be inherited from, but the potential for future ULSecureConnection warrants the addition, rather than having everything be managed from within the ULConnectionManager (or this class) without a wrapper.

10.354.2 Constructor & Destructor Documentation

10.354.2.1 ULConnection() [1/2]

```
gdcm::network::ULConnection::ULConnection (
            const ULConnectionInfo & inUserInformation )
```

10.354.2.2 ~ULConnection()

```
virtual gdcm::network::ULConnection::~ULConnection ( )  [virtual]
```

10.354.2.3 ULConnection() [2/2]

```
gdcm::network::ULConnection::ULConnection (
            const ULConnection &  )  [delete]
```

10.354.3 Member Function Documentation

10.354.3.1 AddAcceptedPresentationContext()

```
void gdcm::network::ULConnection::AddAcceptedPresentationContext (
            const PresentationContextAC & inPC )
```

10.354.3.2 FindContext()

```
PresentationContextRQ gdcm::network::ULConnection::FindContext (
            const DataElement & de ) const
```

10.354.3.3 GetAcceptedPresentationContexts() [1/2]

```
std::vector< PresentationContextAC > & gdcm::network::ULConnection::GetAcceptedPresentation↩
Contexts ( )
```

10.354.3.4 GetAcceptedPresentationContexts() [2/2]

```
std::vector< PresentationContextAC > const & gdcm::network::ULConnection::GetAcceptedPresentation↩
Contexts ( ) const
```

10.354.3.5 GetConnectionInfo()

```
const ULConnectionInfo & gdcm::network::ULConnection::GetConnectionInfo ( ) const
```

10.354.3.6 GetMaxPDUSize()

```
uint32_t gdcm::network::ULConnection::GetMaxPDUSize ( ) const
```

10.354.3.7 GetPresentationContextACByID()

```
const PresentationContextAC * gdcm::network::ULConnection::GetPresentationContextACByID (
            uint8_t id ) const
```

10.354.3.8 GetPresentationContextIDFromPresentationContext()

```
uint8_t gdcm::network::ULConnection::GetPresentationContextIDFromPresentationContext (
            PresentationContextRQ const & pc ) const
```

return 0 upon error

10.354.3.9 GetPresentationContextRQByID()

```
const PresentationContextRQ * gdcm::network::ULConnection::GetPresentationContextRQByID (
            uint8_t id ) const
```

10.354.3.10 GetPresentationContexts()

```
std::vector< PresentationContextRQ > const & gdcm::network::ULConnection::GetPresentationContexts
( ) const
```

10.354.3.11 GetProtocol()

```
std::iostream * gdcm::network::ULConnection::GetProtocol ( )
```

10.354.3.12 GetState()

```
EStateID gdcm::network::ULConnection::GetState ( ) const
```

10.354.3.13 GetTimer()

```
ARTIMTimer & gdcm::network::ULConnection::GetTimer ( )
```

10.354.3.14 InitializeConnection()

```
bool gdcm::network::ULConnection::InitializeConnection ( )
```

used to establish scu connections

10.354.3.15 InitializeIncomingConnection()

```
bool gdcm::network::ULConnection::InitializeIncomingConnection ( )
```

used to establish scp connections

10.354.3.16 operator=()

```
void gdcm::network::ULConnection::operator= (
            const ULConnection &  )  [delete]
```

10.354.3.17 SetMaxPDUSize()

```
void gdcm::network::ULConnection::SetMaxPDUSize (
            uint32_t inSize )
```

10.354.3.18 SetPresentationContexts() [1/2]

```
void gdcm::network::ULConnection::SetPresentationContexts (
            const std::vector< PresentationContext > & inContexts )
```

10.354.3.19 SetPresentationContexts() [2/2]

```
void gdcm::network::ULConnection::SetPresentationContexts (
            const std::vector< PresentationContextRQ > & inContexts )
```

10.354.3.20 SetState()

```
void gdcm::network::ULConnection::SetState (
            const EStateID & inState )
```

10.354.3.21 StopProtocol()

```
void gdcm::network::ULConnection::StopProtocol ( )
```

10.354.4 Friends And Related Function Documentation

10.354.4.1 ULActionAE6

```
friend class ULActionAE6  [friend]
```

10.354.4.2 ULConnectionManager

```
friend class ULConnectionManager  [friend]
```

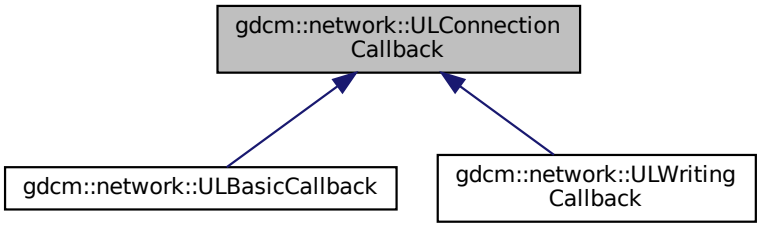
The documentation for this class was generated from the following file:

- gdcmULConnection.h

10.355 gdcm::network::ULConnectionCallback Class Reference

```
#include <gdcmULConnectionCallback.h>
```

Inheritance diagram for gdcm::network::ULConnectionCallback:



Public Member Functions

- ULConnectionCallback ()
- virtual ~ULConnectionCallback ()=default
- bool DataSetHandles () const
- virtual void HandleDataSet (const DataSet &inDataSet)=0
- virtual void HandleResponse (const DataSet &inDataSet)=0
- void ResetHandledDataSet ()
- void SetImplicitFlag (const bool imp)

Protected Member Functions

- void DataSetHandled ()

Protected Attributes

- bool mImplicit

10.355.1 Detailed Description

When a dataset comes back from a query/move/etc, the result can either be stored entirely in memory, or could be stored on disk. This class provides a mechanism to indicate what the ULConnectionManager should do with datasets that are produced through query results. The ULConnectionManager will call the HandleDataSet function during the course of receiving datasets. Particular implementations should fill in what that function does, including updating progress, etc. NOTE: since cmove requires that multiple event loops be employed, the callback function MUST set mHandledData↩ Set to true. otherwise, the cmove event loop handler will not know data was received, and proceed to end the loop prematurely.

10.355.2 Constructor & Destructor Documentation

10.355.2.1 ULConnectionCallback()

```
gdcm::network::ULConnectionCallback::ULConnectionCallback ( )  [inline]
```

10.355.2.2 ~ULConnectionCallback()

```
virtual gdcm::network::ULConnectionCallback::~ULConnectionCallback ( )  [virtual], [default]
```

10.355.3 Member Function Documentation

10.355.3.1 DataSetHandled()

```
void gdcm::network::ULConnectionCallback::DataSetHandled ( )  [inline], [protected]
```

10.355.3.2 DataSetHandles()

```
bool gdcm::network::ULConnectionCallback::DataSetHandles ( ) const  [inline]
```

10.355.3.3 HandleDataSet()

```
virtual void gdcm::network::ULConnectionCallback::HandleDataSet (
            const DataSet & inDataSet )  [pure virtual]
```

Implemented in gdcm::network::ULBasicCallback, and gdcm::network::ULWritingCallback.

10.355.3.4 HandleResponse()

```
virtual void gdcm::network::ULConnectionCallback::HandleResponse (
            const DataSet & inDataSet )  [pure virtual]
```

Implemented in gdcm::network::ULBasicCallback, and gdcm::network::ULWritingCallback.

10.355.3.5 ResetHandledDataSet()

```
void gdcm::network::ULConnectionCallback::ResetHandledDataSet ( )  [inline]
```

10.355.3.6 SetImplicitFlag()

```
void gdcm::network::ULConnectionCallback::SetImplicitFlag (
            const bool imp )  [inline]
```

10.355.4 Member Data Documentation

10.355.4.1 mImplicit

```
bool gdcm::network::ULConnectionCallback::mImplicit  [protected]
```

The documentation for this class was generated from the following file:

- gdcmULConnectionCallback.h

10.356 gdcm::network::ULConnectionInfo Class Reference

ULConnectionInfo.

```
#include <gdcmULConnectionInfo.h>
```

Public Member Functions

- ULConnectionInfo ()
- const char * GetCalledAETitle () const
- std::string GetCalledComputerName () const
- unsigned long GetCalledIPAddress () const
- int GetCalledIPPort () const
- const char * GetCallingAETitle () const
- unsigned long GetMaxPDULength () const
- bool Initialize (UserInformation const &inUserInformation, const char *inCalledAETitle, const char *inCalling↩ AETitle, unsigned long inCalledIPAddress, int inCalledIPPort, std::string inCalledComputerName)
- void SetMaxPDULength (unsigned long inMaxPDULength)

10.356.1 Detailed Description

ULConnectionInfo.

this class contains all the information about a particular connection as established by the user. That is, it's: User Information Calling AE Title Called AE Title IP address/computer name IP Port A connection must be established with this information, that's subsequently placed into various primitives for actual communication.

10.356.2 Constructor & Destructor Documentation

10.356.2.1 ULConnectionInfo()

```
gdcm::network::ULConnectionInfo::ULConnectionInfo ( )
```

10.356.3 Member Function Documentation

10.356.3.1 GetCalledAETitle()

```
const char * gdcm::network::ULConnectionInfo::GetCalledAETitle ( ) const
```

10.356.3.2 GetCalledComputerName()

```
std::string gdcm::network::ULConnectionInfo::GetCalledComputerName ( ) const
```

10.356.3.3 GetCalledIPAddress()

```
unsigned long gdcm::network::ULConnectionInfo::GetCalledIPAddress ( ) const
```

10.356.3.4 GetCalledIPPort()

```
int gdcm::network::ULConnectionInfo::GetCalledIPPort ( ) const
```

10.356.3.5 GetCallingAETitle()

```
const char * gdcm::network::ULConnectionInfo::GetCallingAETitle ( ) const
```

10.356.3.6 GetMaxPDULength()

```
unsigned long gdcm::network::ULConnectionInfo::GetMaxPDULength ( ) const
```

10.356.3.7 Initialize()

```
bool gdcm::network::ULConnectionInfo::Initialize (
             UserInformation const & inUserInformation,
             const char * inCalledAETitle,
             const char * inCallingAETitle,
             unsigned long inCalledIPAddress,
             int inCalledIPPort,
             std::string inCalledComputerName )
```

10.356.3.8 SetMaxPDULength()

```
void gdcm::network::ULConnectionInfo::SetMaxPDULength (
             unsigned long inMaxPDULength )
```

The documentation for this class was generated from the following file:

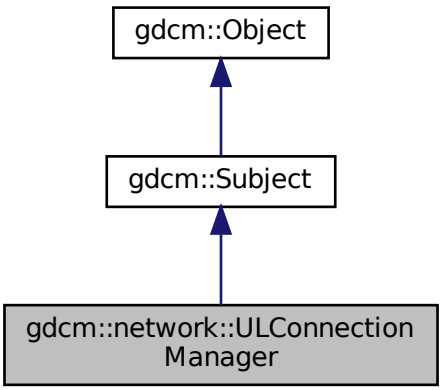
- gdcmULConnectionInfo.h

10.357 gdcm::network::ULConnectionManager Class Reference

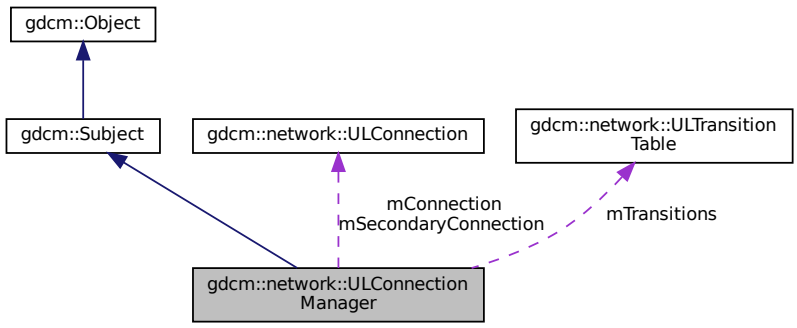
ULConnectionManager.

`#include <gdcmULConnectionManager.h>`

Inheritance diagram for gdcm::network::ULConnectionManager:



Collaboration diagram for gdcm::network::ULConnectionManager:



Public Member Functions

- ULConnectionManager ()
- ~ULConnectionManager () override
- bool BreakConnection (const double &inTimeout)
- void BreakConnectionNow ()
- bool EstablishConnection (const std::string &inAETitle, const std::string &inConnectAETitle, const std::string &inComputerName, long inIPAddress, uint16_t inConnectPort, double inTimeout, std::vector< PresentationContext > const &pcVector)
- bool EstablishConnectionMove (const std::string &inAETitle, const std::string &inConnectAETitle, const std::string &inComputerName, long inIPAddress, uint16_t inConnectPort, double inTimeout, uint16_t inReturnPort, std::vector< PresentationContext > const &pcVector)
- std::vector< PresentationDataValue > SendEcho ()
- std::vector< DataSet > SendFind (const BaseRootQuery *inRootQuery)
- void SendFind (const BaseRootQuery *inRootQuery, ULConnectionCallback *inCallback)
- std::vector< DataSet > SendMove (const BaseRootQuery *inRootQuery)
- bool SendMove (const BaseRootQuery *inRootQuery, ULConnectionCallback *inCallback)

  return false upon error
- std::vector< DataSet > SendNAction (const BaseQuery *inQuery)
- void SendNAction (const BaseQuery *inQuery, ULConnectionCallback *inCallback)
- std::vector< DataSet > SendNCreate (const BaseQuery *inQuery)
- void SendNCreate (const BaseQuery *inQuery, ULConnectionCallback *inCallback)
- std::vector< DataSet > SendNDelete (const BaseQuery *inQuery)
- void SendNDelete (const BaseQuery *inQuery, ULConnectionCallback *inCallback)
- std::vector< DataSet > SendNEventReport (const BaseQuery *inQuery)
- void SendNEventReport (const BaseQuery *inQuery, ULConnectionCallback *inCallback)
- std::vector< DataSet > SendNGet (const BaseQuery *inQuery)
- void SendNGet (const BaseQuery *inQuery, ULConnectionCallback *inCallback)
- std::vector< DataSet > SendNSet (const BaseQuery *inQuery)
- void SendNSet (const BaseQuery *inQuery, ULConnectionCallback *inCallback)
- std::vector< DataSet > SendStore (const File &file, std::istream *pStream=nullptr, std::streampos dataSetOffset=0)
- void SendStore (const File &file, ULConnectionCallback *inCallback, std::istream *pStream=nullptr, std::streampos dataSetOffset=0)

  callback based API

Protected Member Functions

- ULConnectionManager (const ULConnectionManager &inCM)
- EStateID RunEventLoop (ULEvent &inEvent, ULConnection *inWhichConnection, ULConnectionCallback *inCallback, const bool &startWaiting)
- EStateID RunMoveEventLoop (ULEvent &inEvent, ULConnectionCallback *inCallback)

Protected Attributes

- ULConnection * mConnection
- ULConnection * mSecondaryConnection
- ULTransitionTable mTransitions

10.357.1 Detailed Description

ULConnectionManager.

The ULConnectionManager performs actions on the ULConnection given inputs from the user and from the state of what's going on around the connection (ie, timeouts of the ARTIM timer, responses from the peer across the connection, etc).

Its inputs are ULEvents, and it performs ULActions.

10.357.2 Constructor & Destructor Documentation

10.357.2.1 ULConnectionManager() [1/2]

```
gdcm::network::ULConnectionManager::ULConnectionManager (
            const ULConnectionManager & inCM )  [protected]
```

10.357.2.2 ULConnectionManager() [2/2]

```
gdcm::network::ULConnectionManager::ULConnectionManager ( )
```

10.357.2.3 ~ULConnectionManager()

```
gdcm::network::ULConnectionManager::~ULConnectionManager ( )  [override]
```

10.357.3 Member Function Documentation

10.357.3.1 BreakConnection()

```
bool gdcm::network::ULConnectionManager::BreakConnection (
            const double & inTimeout )
```

10.357.3.2 BreakConnectionNow()

```
void gdcm::network::ULConnectionManager::BreakConnectionNow ( )
```

10.357.3.3 EstablishConnection()

```
bool gdcm::network::ULConnectionManager::EstablishConnection (
            const std::string & inAETitle,
            const std::string & inConnectAETitle,
            const std::string & inComputerName,
            long inIPAddress,
            uint16_t inConnectPort,
            double inTimeout,
            std::vector< PresentationContext > const & pcVector )
```

returns true if a connection of the given AETitle (ie, 'this' program) is able to connect to the given AETitle and Port in a certain amount of time providing the connection type will establish the proper exchange syntax with a server; if a different functionality is required, a different connection should be established. returns false if the connection type is 'move'– have to give a return port for move to work as specified.

10.357.3.4 EstablishConnectionMove()

```
bool gdcm::network::ULConnectionManager::EstablishConnectionMove (
            const std::string & inAETitle,
            const std::string & inConnectAETitle,
            const std::string & inComputerName,
            long inIPAddress,
            uint16_t inConnectPort,
            double inTimeout,
            uint16_t inReturnPort,
            std::vector< PresentationContext > const & pcVector )
```

returns true for above reasons, but contains the special 'move' port

10.357.3.5 RunEventLoop()

```
EStateID gdcm::network::ULConnectionManager::RunEventLoop (
            ULEvent & inEvent,
            ULConnection * inWhichConnection,
            ULConnectionCallback * inCallback,
            const bool & startWaiting )  [protected]
```

10.357.3.6 RunMoveEventLoop()

```
EStateID gdcm::network::ULConnectionManager::RunMoveEventLoop (
            ULEvent & inEvent,
            ULConnectionCallback * inCallback )  [protected]
```

10.357.3.7 SendEcho()

```
std::vector< PresentationDataValue > gdcm::network::ULConnectionManager::SendEcho ( )
```

10.357.3.8 SendFind() [1/2]

```
std::vector< DataSet > gdcm::network::ULConnectionManager::SendFind (
            const BaseRootQuery * inRootQuery )
```

10.357.3.9 SendFind() [2/2]

```
void gdcm::network::ULConnectionManager::SendFind (
            const BaseRootQuery * inRootQuery,
            ULConnectionCallback * inCallback )
```

10.357.3.10 SendMove() [1/2]

```
std::vector< DataSet > gdcm::network::ULConnectionManager::SendMove (
            const BaseRootQuery * inRootQuery )
```

10.357.3.11 SendMove() [2/2]

```
bool gdcm::network::ULConnectionManager::SendMove (
            const BaseRootQuery * inRootQuery,
            ULConnectionCallback * inCallback )
```

return false upon error

10.357.3.12 SendNAction() [1/2]

```
std::vector< DataSet > gdcm::network::ULConnectionManager::SendNAction (
            const BaseQuery * inQuery )
```

10.357.3.13 SendNAction() [2/2]

```
void gdcm::network::ULConnectionManager::SendNAction (
            const BaseQuery * inQuery,
            ULConnectionCallback * inCallback )
```

10.357.3.14 SendNCreate() [1/2]

```
std::vector< DataSet > gdcm::network::ULConnectionManager::SendNCreate (
            const BaseQuery * inQuery )
```

10.357.3.15 SendNCreate() [2/2]

```
void gdcm::network::ULConnectionManager::SendNCreate (
            const BaseQuery * inQuery,
            ULConnectionCallback * inCallback )
```

10.357.3.16 SendNDelete() [1/2]

```
std::vector< DataSet > gdcm::network::ULConnectionManager::SendNDelete (
            const BaseQuery * inQuery )
```

10.357.3.17 SendNDelete() [2/2]

```
void gdcm::network::ULConnectionManager::SendNDelete (
            const BaseQuery * inQuery,
            ULConnectionCallback * inCallback )
```

10.357.3.18 SendNEventReport() [1/2]

```
std::vector< DataSet > gdcm::network::ULConnectionManager::SendNEventReport (
            const BaseQuery * inQuery )
```

10.357.3.19 SendNEventReport() [2/2]

```
void gdcm::network::ULConnectionManager::SendNEventReport (
            const BaseQuery * inQuery,
            ULConnectionCallback * inCallback )
```

10.357.3.20 SendNGet() [1/2]

```
std::vector< DataSet > gdcm::network::ULConnectionManager::SendNGet (
            const BaseQuery * inQuery )
```

10.357.3.21 SendNGet() [2/2]

```
void gdcm::network::ULConnectionManager::SendNGet (
            const BaseQuery * inQuery,
            ULConnectionCallback * inCallback )
```

10.357.3.22 SendNSet() [1/2]

```
std::vector< DataSet > gdcm::network::ULConnectionManager::SendNSet (
            const BaseQuery * inQuery )
```

10.357.3.23 SendNSet() [2/2]

```
void gdcm::network::ULConnectionManager::SendNSet (
            const BaseQuery * inQuery,
            ULConnectionCallback * inCallback )
```

10.357.3.24 SendStore() [1/2]

```
std::vector< DataSet > gdcm::network::ULConnectionManager::SendStore (
            const File & file,
            std::istream * pStream = nullptr,
            std::streampos dataSetOffset = 0 )
```

10.357.3.25 SendStore() [2/2]

```
void gdcm::network::ULConnectionManager::SendStore (
            const File & file,
            ULConnectionCallback * inCallback,
            std::istream * pStream = nullptr,
            std::streampos dataSetOffset = 0 )
```

callback based API

10.357.4 Member Data Documentation

10.357.4.1 mConnection

```
ULConnection* gdcm::network::ULConnectionManager::mConnection  [protected]
```

10.357.4.2 mSecondaryConnection

```
ULConnection* gdcm::network::ULConnectionManager::mSecondaryConnection  [protected]
```

10.357.4.3 mTransitions

```
ULTransitionTable gdcm::network::ULConnectionManager::mTransitions  [protected]
```

The documentation for this class was generated from the following file:

- gdcmULConnectionManager.h

10.358 gdcm::network::ULEvent Class Reference

ULEvent.

```
#include <gdcmULEvent.h>
```

Public Member Functions

- ULEvent (const EEventID &inEventID, BasePDU *inBasePDU, std::istream *iStream=nullptr, std::streampos posDataSet=0)
- ULEvent (const EEventID &inEventID, std::vector< BasePDU * > inBasePDU, std::istream *iStream=nullptr, std::streampos posDataSet=0)
- ~ULEvent ()
- std::streampos GetDataSetPos () const
- EEventID GetEvent () const
- std::istream * GetIStream () const
- std::vector< BasePDU * > const & GetPDUs () const
- void SetEvent (const EEventID &inEvent)
- void SetPDU (std::vector< BasePDU * > const &inPDU)

10.358.1 Detailed Description

ULEvent.

base class for network events.

An event consists of the event ID and the data associated with that event.

Note that once a PDU is created, it is now the responsibility of the associated event to destroy it!

10.358.2 Constructor & Destructor Documentation

10.358.2.1 ULEvent() [1/2]

```
gdcm::network::ULEvent::ULEvent (
            const EEventID & inEventID,
            std::vector< BasePDU * > inBasePDU,
            std::istream * iStream = nullptr,
            std::streampos posDataSet = 0 )  [inline]
```

10.358.2.2 ULEvent() [2/2]

```
gdcm::network::ULEvent::ULEvent (
            const EEventID & inEventID,
            BasePDU * inBasePDU,
            std::istream * iStream = nullptr,
            std::streampos posDataSet = 0 )  [inline]
```

10.358.2.3 ~ULEvent()

```
gdcm::network::ULEvent::~ULEvent ( )  [inline]
```

10.358.3 Member Function Documentation

10.358.3.1 GetDataSetPos()

```
std::streampos gdcm::network::ULEvent::GetDataSetPos ( ) const  [inline]
```

10.358.3.2 GetEvent()

```
EEventID gdcm::network::ULEvent::GetEvent ( ) const  [inline]
```

10.358.3.3 GetIStream()

```
std::istream * gdcm::network::ULEvent::GetIStream ( ) const  [inline]
```

10.358.3.4 GetPDUs()

```
std::vector< BasePDU * > const & gdcm::network::ULEvent::GetPDUs ( ) const  [inline]
```

10.358.3.5 SetEvent()

```
void gdcm::network::ULEvent::SetEvent (
            const EEventID & inEvent )  [inline]
```

10.358.3.6 SetPDU()

```
void gdcm::network::ULEvent::SetPDU (
            std::vector< BasePDU * > const & inPDU )  [inline]
```

The documentation for this class was generated from the following file:

- gdcmULEvent.h

10.359 gdcm::network::ULTransitionTable Class Reference

ULTransitionTable The transition table of all the ULEvents, new ULActions, and ULStates.

`#include <gdcmULTransitionTable.h>`

Public Member Functions

- ULTransitionTable ()
- void HandleEvent (Subject *s, ULEvent &inEvent, ULConnection &inConnection, bool &outWaitingForEvent, EEventID &outRaisedEvent) const
- void PrintTable () const

10.359.1 Detailed Description

ULTransitionTable The transition table of all the ULEvents, new ULActions, and ULStates.

Based roughly on the solutions in player2.cpp in the boost examples and this so question: http↩
://stackoverflow.com/questions/1647631/c-state-machine-design

The transition table is constructed of TableRows. Each row is based on an event, and an event handler in the Transition↩
Table object takes a given event, and then finds the given row.

Then, given the current state of the connection, determines the appropriate action to take and then the state to transition to next.

10.359.2 Constructor & Destructor Documentation

10.359.2.1 ULTransitionTable()

```
gdcm::network::ULTransitionTable::ULTransitionTable ( )
```

10.359.3 Member Function Documentation

10.359.3.1 HandleEvent()

```
void gdcm::network::ULTransitionTable::HandleEvent (
            Subject * s,
            ULEvent & inEvent,
            ULConnection & inConnection,
            bool & outWaitingForEvent,
            EEventID & outRaisedEvent ) const
```

10.359.3.2 PrintTable()

```
void gdcm::network::ULTransitionTable::PrintTable ( ) const
```

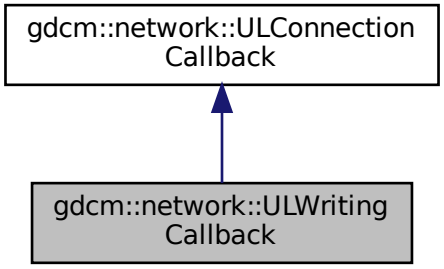
The documentation for this class was generated from the following file:

- gdcmULTransitionTable.h

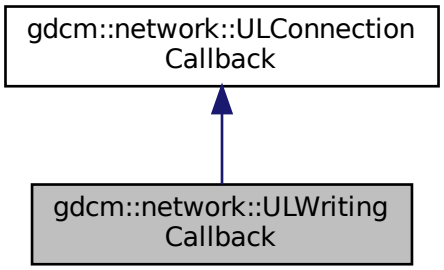
10.360 gdcm::network::ULWritingCallback Class Reference

`#include <gdcmULWritingCallback.h>`

Inheritance diagram for gdcm::network::ULWritingCallback:



Collaboration diagram for gdcm::network::ULWritingCallback:



Public Member Functions

- ULWritingCallback ()=default
- ~ULWritingCallback () override=default
- void HandleDataSet (const DataSet &inDataSet) override
- void HandleResponse (const DataSet &inDataSet) override
- void SetDirectory (const std::string &inDirectoryName)

  provide the directory into which all files are written.

Additional Inherited Members

10.360.1 Constructor & Destructor Documentation

10.360.1.1 ULWritingCallback()

```
gdcm::network::ULWritingCallback::ULWritingCallback ( )  [default]
```

10.360.1.2 ~ULWritingCallback()

```
gdcm::network::ULWritingCallback::~ULWritingCallback ( )  [override], [default]
```

10.360.2 Member Function Documentation

10.360.2.1 HandleDataSet()

```
void gdcm::network::ULWritingCallback::HandleDataSet (
            const DataSet & inDataSet )  [override], [virtual]
```

Implements gdcm::network::ULConnectionCallback.

10.360.2.2 HandleResponse()

```
void gdcm::network::ULWritingCallback::HandleResponse (
            const DataSet & inDataSet )  [override], [virtual]
```

Implements gdcm::network::ULConnectionCallback.

10.360.2.3 SetDirectory()

```
void gdcm::network::ULWritingCallback::SetDirectory (
            const std::string & inDirectoryName )  [inline]
```

provide the directory into which all files are written.

The documentation for this class was generated from the following file:

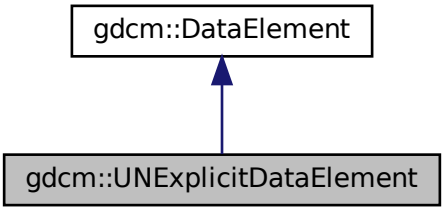
- gdcmULWritingCallback.h

10.361 gdcm::UNExplicitDataElement Class Reference

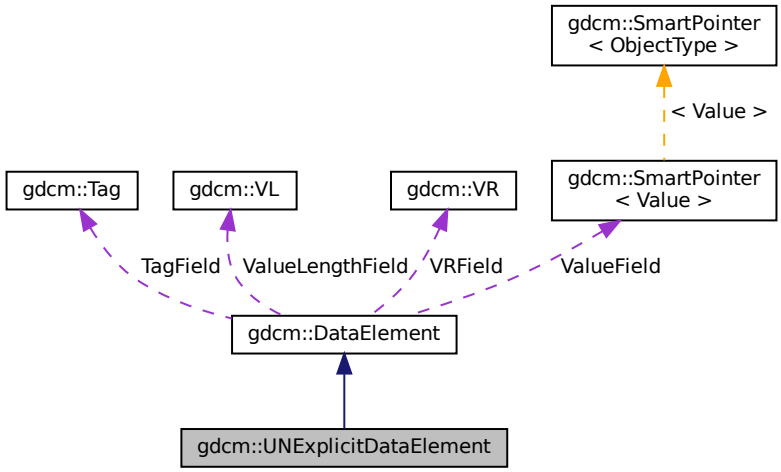
Class to read/write a DataElement as UNExplicit Data Element.

```
#include <gdcmUNExplicitDataElement.h>
```

Inheritance diagram for gdcm::UNExplicitDataElement:



Collaboration diagram for gdcm::UNExplicitDataElement:



Public Member Functions

- VL GetLength () const
- template<typename TSwap >
  std::istream & Read (std::istream &is)
- template<typename TSwap >
  std::istream & ReadPreValue (std::istream &is)
- template<typename TSwap >
  std::istream & ReadValue (std::istream &is, bool readvalues=true)
- template<typename TSwap >
  std::istream & ReadWithLength (std::istream &is, VL &length)

Additional Inherited Members

10.361.1 Detailed Description

Class to read/write a DataElement as UNExplicit Data Element.

Note

bla

10.361.2 Member Function Documentation

10.361.2.1 GetLength()

```
VL gdcm::UNExplicitDataElement::GetLength ( ) const
```

10.361.2.2 Read()

```
template<typename TSwap >
std::istream & gdcm::UNExplicitDataElement::Read (
             std::istream & is )
```

10.361.2.3 ReadPreValue()

```
template<typename TSwap >
std::istream & gdcm::UNExplicitDataElement::ReadPreValue (
             std::istream & is )
```

10.361.2.4 ReadValue()

```
template<typename TSwap >
std::istream & gdcm::UNExplicitDataElement::ReadValue (
             std::istream & is,
             bool readvalues = true )
```

10.361.2.5 ReadWithLength()

```
template<typename TSwap >
std::istream & gdcm::UNExplicitDataElement::ReadWithLength (
             std::istream & is,
             VL & length )
```

The documentation for this class was generated from the following file:

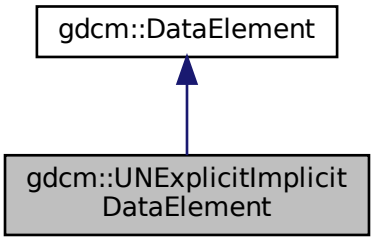
- gdcmUNExplicitDataElement.h

10.362 gdcm::UNExplicitImplicitDataElement Class Reference

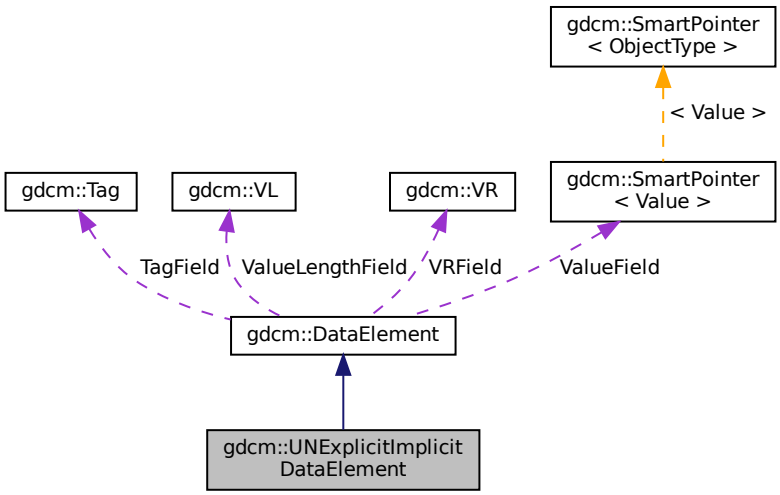
Class to read/write a DataElement as ExplicitImplicit Data Element.

`#include <gdcmUNExplicitImplicitDataElement.h>`

Inheritance diagram for gdcm::UNExplicitImplicitDataElement:



Collaboration diagram for gdcm::UNExplicitImplicitDataElement:



Public Member Functions

- VL GetLength () const

- template<typename TSwap >
  std::istream & Read (std::istream &is)
- template<typename TSwap >
  std::istream & ReadPreValue (std::istream &is)
- template<typename TSwap >
  std::istream & ReadValue (std::istream &is)

Additional Inherited Members

10.362.1 Detailed Description

Class to read/write a DataElement as ExplicitImplicit Data Element.

This class gather two known bugs:

1. GDCM 1.2.0 would rewrite VR=UN Value Length on 2 bytes instead of 4 bytes
2. GDCM 1.2.0 would also rewrite DataElement as Implicit when the VR would not be known this would only happen in some very rare cases. gdcm 2.X design could handle bug #1 or #2 exclusively, this class can now handle file which have both issues. See: gdcmData/TheralysGDCM120Bug.dcm

10.362.2 Member Function Documentation

10.362.2.1 GetLength()

```
VL gdcm::UNExplicitImplicitDataElement::GetLength ( ) const
```

10.362.2.2 Read()

```
template<typename TSwap >
std::istream & gdcm::UNExplicitImplicitDataElement::Read (
            std::istream & is )
```

10.362.2.3 ReadPreValue()

```
template<typename TSwap >
std::istream & gdcm::UNExplicitImplicitDataElement::ReadPreValue (
            std::istream & is )
```

10.362.2.4 ReadValue()

```
template<typename TSwap >
std::istream & gdcm::UNExplicitImplicitDataElement::ReadValue (
             std::istream & is )
```

The documentation for this class was generated from the following file:

- gdcmUNExplicitImplicitDataElement.h

10.363 gdcm::Unpacker12Bits Class Reference

Pack/Unpack 12 bits pixel into 16bits.

```
#include <gdcmUnpacker12Bits.h>
```

Static Public Member Functions

- static bool Pack (char *out, const char *in, size_t n)
- static bool Unpack (char *out, const char *in, size_t n)

10.363.1 Detailed Description

Pack/Unpack 12 bits pixel into 16bits.

- You can only pack an even number of 16bits, which means a multiple of 4 (expressed in bytes)
- You can only unpack a multiple of 3 bytes

This class has no purpose in general purpose DICOM implementation. However to be able to cope with some early ACR-NEMA file generated by a well-known private vendor, one would need to unpack 12bits Stored Pixel Value into a more standard 16bits Stored Pixel Value.

See also

Rescaler

10.363.2 Member Function Documentation

10.363.2.1 Pack()

```
static bool gdcm::Unpacker12Bits::Pack (
            char * out,
            const char * in,
            size_t n )  [static]
```

Pack an array of 16bits where all values are 12bits into a pack form. n is the length in bytes of array in, out will be a fake 8bits array of size (n / 2) ∗ 3

10.363.2.2 Unpack()

```
static bool gdcm::Unpacker12Bits::Unpack (
            char * out,
            const char * in,
            size_t n )  [static]
```

Unpack an array of 'packed' 12bits data into a more conventional 16bits array. n is the length in bytes of array in, out will be a 16bits array of size (n / 3) ∗ 2

The documentation for this class was generated from the following file:

- gdcmUnpacker12Bits.h

10.364 gdcm::Usage Class Reference

Usage.

```
#include <gdcmUsage.h>
```

Public Types

- enum UsageType {
  Mandatory ,
  Conditional ,
  UserOption ,
  Invalid }

Public Member Functions

- Usage (UsageType type=Invalid)
- operator UsageType () const

Static Public Member Functions

- static const char * GetUsageString (UsageType type)
- static UsageType GetUsageType (const char *type)

Friends

- std::ostream & operator<< (std::ostream &os, const Usage &vr)

10.364.1 Detailed Description

Usage.

Note

A.1.3 IOD Module Table and Functional Group Macro Table This Section of each IOD defines in a tabular form the Modules comprising the IOD. The following information must be specified for each Module in the table:

- The name of the Module or Functional Group
- A reference to the Section in Annex C which defines the Module or Functional Group
- The usage of the Module or Functional Group; whether it is:
- Mandatory (see A.1.3.1) , abbreviated M
- Conditional (see A.1.3.2) , abbreviated C
- User Option (see A.1.3.3) , abbreviated U The Modules referenced are defined in Annex C. A.1.3.1 MANDATORY MODULES For each IOD, Mandatory Modules shall be supported per the definitions, semantics and requirements defined in Annex C.

A.1.3.2 CONDITIONAL MODULES Conditional Modules are Mandatory Modules if specific conditions are met. If the specified conditions are not met, this Module shall not be supported; that is, no information defined in that Module shall be sent. A.1.3.3 USER OPTION MODULES User Option Modules may or may not be supported. If an optional Module is supported, the Attribute Types specified in the Modules in Annex C shall be supported.

10.364.2 Member Enumeration Documentation

10.364.2.1 UsageType

`enum gdcm::Usage::UsageType`

Enumerator

Mandatory	
Conditional	
UserOption	
Invalid	

10.364.3 Constructor & Destructor Documentation

10.364.3.1 Usage()

```
gdcm::Usage::Usage (
            UsageType type = Invalid )  [inline]
```

10.364.4 Member Function Documentation

10.364.4.1 GetUsageString()

```
static const char * gdcm::Usage::GetUsageString (
            UsageType type )  [static]
```

10.364.4.2 GetUsageType()

```
static UsageType gdcm::Usage::GetUsageType (
            const char * type )  [static]
```

10.364.4.3 operator UsageType()

```
gdcm::Usage::operator UsageType ( ) const  [inline]
```

10.364.5 Friends And Related Function Documentation

10.364.5.1 operator<<

```
std::ostream & operator<< (
            std::ostream & os,
            const Usage & vr )  [friend]
```

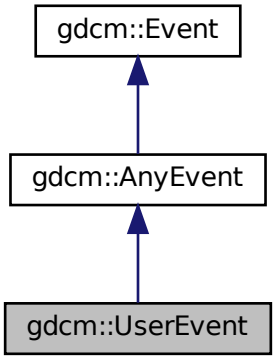
The documentation for this class was generated from the following file:

- gdcmUsage.h

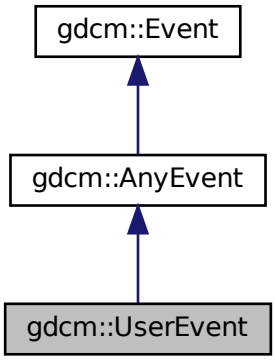
10.365 gdcm::UserEvent Class Reference

`#include <gdcmEvent.h>`

Inheritance diagram for gdcm::UserEvent:



Collaboration diagram for gdcm::UserEvent:



Additional Inherited Members

The documentation for this class was generated from the following file:

- gdcmEvent.h

10.366 gdcm::network::UserInformation Class Reference

UserInformation.

`#include <gdcmUserInformation.h>`

Public Member Functions

- UserInformation ()
- UserInformation (const UserInformation &)=delete
- ~UserInformation ()
- void AddRoleSelectionSub (RoleSelectionSub const &r)
- void AddSOPClassExtendedNegociationSub (SOPClassExtendedNegociationSub const &s)
- MaximumLengthSub & GetMaximumLengthSub ()
- const MaximumLengthSub & GetMaximumLengthSub () const
- UserInformation & operator= (const UserInformation &)
- void Print (std::ostream &os) const
- std::istream & Read (std::istream &is)
- size_t Size () const
- const std::ostream & Write (std::ostream &os) const

10.366.1 Detailed Description

UserInformation.

Table 9-16 USER INFORMATION ITEM FIELDS

TODO what is the goal of :

Table 9-20 USER INFORMATION ITEM FIELDS

10.366.2 Constructor & Destructor Documentation

10.366.2.1 UserInformation() [1/2]

`gdcm::network::UserInformation::UserInformation ( )`

10.366.2.2 ~UserInformation()

`gdcm::network::UserInformation::~UserInformation ( )`

10.366.2.3 UserInformation() [2/2]

```
gdcm::network::UserInformation::UserInformation (
            const UserInformation &  )  [delete]
```

10.366.3 Member Function Documentation

10.366.3.1 AddRoleSelectionSub()

```
void gdcm::network::UserInformation::AddRoleSelectionSub (
            RoleSelectionSub const & r )
```

10.366.3.2 AddSOPClassExtendedNegociationSub()

```
void gdcm::network::UserInformation::AddSOPClassExtendedNegociationSub (
            SOPClassExtendedNegociationSub const & s )
```

10.366.3.3 GetMaximumLengthSub() [1/2]

```
MaximumLengthSub & gdcm::network::UserInformation::GetMaximumLengthSub ( )  [inline]
```

10.366.3.4 GetMaximumLengthSub() [2/2]

```
const MaximumLengthSub & gdcm::network::UserInformation::GetMaximumLengthSub ( ) const  [inline]
```

10.366.3.5 operator=()

```
UserInformation & gdcm::network::UserInformation::operator= (
            const UserInformation &  )
```

10.366.3.6 Print()

```
void gdcm::network::UserInformation::Print (
            std::ostream & os ) const
```

10.366.3.7 Read()

```
std::istream & gdcm::network::UserInformation::Read (
            std::istream & is )
```

10.366.3.8 Size()

```
size_t gdcm::network::UserInformation::Size ( ) const
```

10.366.3.9 Write()

```
const std::ostream & gdcm::network::UserInformation::Write (
            std::ostream & os ) const
```

The documentation for this class was generated from the following file:

- gdcmUserInformation.h

10.367 gdcm::UUIDGenerator Class Reference

Class for generating unique UUID.

`#include <gdcmUUIDGenerator.h>`

Public Member Functions

- const char * Generate ()

Static Public Member Functions

- static bool IsValid (const char *uid)

  Find out if the string is a valid UUID or not.

10.367.1 Detailed Description

Class for generating unique UUID.

generate DCE 1.1 uid

10.367.2 Member Function Documentation

10.367.2.1 Generate()

```
const char * gdcm::UUIDGenerator::Generate ( )
```

Return the generated uuid NOT THREAD SAFE

10.367.2.2 IsValid()

```
static bool gdcm::UUIDGenerator::IsValid (
            const char * uid )  [static]
```

Find out if the string is a valid UUID or not.

The documentation for this class was generated from the following file:

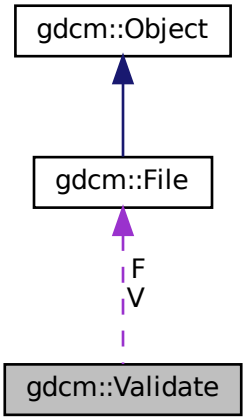
- gdcmUUIDGenerator.h

10.368 gdcm::Validate Class Reference

Validate class.

`#include <gdcmValidate.h>`

Collaboration diagram for gdcm::Validate:



Public Member Functions

- Validate ()
- ~Validate ()
- const File & GetValidatedFile ()
- void SetFile (File const &f)
- void Validation ()

Protected Attributes

- const File * F
- File V

10.368.1 Detailed Description

Validate class.

10.368.2 Constructor & Destructor Documentation

10.368.2.1 Validate()

```
gdcm::Validate::Validate ( )
```

10.368.2.2 ~Validate()

```
gdcm::Validate::~Validate ( )
```

10.368.3 Member Function Documentation

10.368.3.1 GetValidatedFile()

```
const File & gdcm::Validate::GetValidatedFile ( )  [inline]
```

10.368.3.2 SetFile()

```
void gdcm::Validate::SetFile (
            File const & f )  [inline]
```

10.368.3.3 Validation()

```
void gdcm::Validate::Validation ( )
```

10.368.4 Member Data Documentation

10.368.4.1 F

```
const File* gdcm::Validate::F  [protected]
```

10.368.4.2 V

```
File gdcm::Validate::V  [protected]
```

The documentation for this class was generated from the following file:

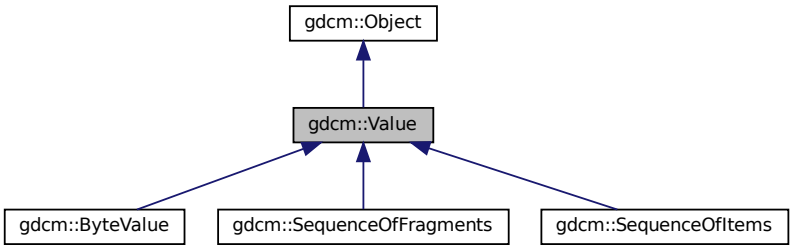
- gdcmValidate.h

10.369 gdcm::Value Class Reference

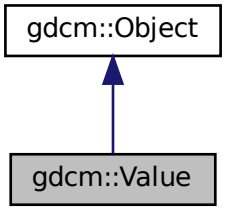
Class to represent the value of a Data Element.

`#include <gdcmValue.h>`

Inheritance diagram for gdcm::Value:



Collaboration diagram for gdcm::Value:



Public Member Functions

- Value ()=default
- ~Value () override=default
- virtual void Clear ()=0
- virtual VL GetLength () const =0
- virtual bool operator== (const Value &val) const =0
- virtual void SetLength (VL l)=0

Protected Member Functions

- virtual void SetLengthOnly (VL l)

Friends

- class DataElement

10.369.1 Detailed Description

Class to represent the value of a Data Element.

Note

VALUE: A component of a Value Field. A Value Field may consist of one or more of these components.

10.369.2 Constructor & Destructor Documentation

10.369.2.1 Value()

```
gdcm::Value::Value ( )  [default]
```

10.369.2.2 ~Value()

```
gdcm::Value::~Value ( )  [override], [default]
```

10.369.3 Member Function Documentation

10.369.3.1 Clear()

```
virtual void gdcm::Value::Clear ( )  [pure virtual]
```

Implemented in gdcm::ByteValue, gdcm::SequenceOfFragments, and gdcm::SequenceOfItems.

10.369.3.2 GetLength()

```
virtual VL gdcm::Value::GetLength ( ) const  [pure virtual]
```

Implemented in gdcm::ByteValue, gdcm::SequenceOfFragments, and gdcm::SequenceOfItems.

Referenced by gdcm::DataSet::InsertDataElement(), and gdcm::DataElement::SetValue().

10.369.3.3 operator==()

```
virtual bool gdcm::Value::operator== (
            const Value & val ) const  [pure virtual]
```

Implemented in gdcm::ByteValue, gdcm::SequenceOfFragments, and gdcm::SequenceOfItems.

10.369.3.4 SetLength()

```
virtual void gdcm::Value::SetLength (
            VL l )  [pure virtual]
```

Implemented in gdcm::SequenceOfFragments, gdcm::SequenceOfItems, and gdcm::ByteValue.

10.369.3.5 SetLengthOnly()

```
virtual void gdcm::Value::SetLengthOnly (
            VL l )  [protected], [virtual]
```

Reimplemented in gdcm::ByteValue.

10.369.4 Friends And Related Function Documentation

10.369.4.1 DataElement

```
friend class DataElement  [friend]
```

The documentation for this class was generated from the following file:

- gdcmValue.h

10.370 gdcm::ValueIO< TDE, TSwap, TType > Class Template Reference

Class to dispatch template calls.

`#include <gdcmValueIO.h>`

Static Public Member Functions

- static std::istream & Read (std::istream &is, Value &v, bool readvalues)
- static const std::ostream & Write (std::ostream &os, const Value &v)

10.370.1 Detailed Description

template<typename TDE, typename TSwap, typename TType = uint8_t>
class gdcm::ValueIO< TDE, TSwap, TType >

Class to dispatch template calls.

10.370.2 Member Function Documentation

10.370.2.1 Read()

```
template<typename TDE , typename TSwap , typename TType = uint8_t>
static std::istream & gdcm::ValueIO< TDE, TSwap, TType >::Read (
            std::istream & is,
            Value & v,
            bool readvalues )  [static]
```

10.370.2.2 Write()

```
template<typename TDE , typename TSwap , typename TType = uint8_t>
static const std::ostream & gdcm::ValueIO< TDE, TSwap, TType >::Write (
            std::ostream & os,
            const Value & v )  [static]
```

The documentation for this class was generated from the following file:

- gdcmValueIO.h

10.371 gdcm::MrProtocol::Vector3 Struct Reference

`#include <gdcmMrProtocol.h>`

Public Attributes

- double dCor
- double dSag
- double dTra

10.371.1 Member Data Documentation

10.371.1.1 dCor

`double gdcm::MrProtocol::Vector3::dCor`

10.371.1.2 dSag

`double gdcm::MrProtocol::Vector3::dSag`

10.371.1.3 dTra

`double gdcm::MrProtocol::Vector3::dTra`

The documentation for this struct was generated from the following file:

- gdcmMrProtocol.h

10.372 gdcm::Version Class Reference

major/minor and build version

`#include <gdcmVersion.h>`

Public Member Functions

- Version ()=default
- ~Version ()=default
- void Print (std::ostream &os=std::cout) const

Static Public Member Functions

- static int GetBuildVersion ()
- static int GetMajorVersion ()
- static int GetMinorVersion ()
- static const char * GetVersion ()

Friends

- std::ostream & operator<< (std::ostream &_os, const Version &v)

10.372.1 Detailed Description

major/minor and build version

10.372.2 Constructor & Destructor Documentation

10.372.2.1 Version()

```
gdcm::Version::Version ( )  [default]
```

10.372.2.2 ~Version()

```
gdcm::Version::~Version ( )  [default]
```

10.372.3 Member Function Documentation

10.372.3.1 GetBuildVersion()

```
static int gdcm::Version::GetBuildVersion ( )  [static]
```

10.372.3.2 GetMajorVersion()

```
static int gdcm::Version::GetMajorVersion ( )  [static]
```

10.372.3.3 GetMinorVersion()

```
static int gdcm::Version::GetMinorVersion ( )  [static]
```

10.372.3.4 GetVersion()

```
static const char * gdcm::Version::GetVersion ( )  [static]
```

10.372.3.5 Print()

```
void gdcm::Version::Print (
            std::ostream & os = std::cout ) const
```

10.372.4 Friends And Related Function Documentation

10.372.4.1 operator<<

```
std::ostream & operator<< (
            std::ostream & _os,
            const Version & v )  [friend]
```

The documentation for this class was generated from the following file:

- gdcmVersion.h

10.373 gdcm::VL Class Reference

Value Length.

`#include <gdcmVL.h>`

Public Types

- typedef uint32_t Type

Public Member Functions

- VL (uint32_t vl=0)
- VL GetLength () const
- bool IsOdd () const

  Return whether or not the VL is odd or not.
- bool IsUndefined () const
- operator uint32_t () const
- VL & operator++ ()
- VL operator++ (int)
- VL & operator+= (VL const &vl)

  += operator
- template<typename TSwap >
  std::istream & Read (std::istream &is)
- template<typename TSwap >
  std::istream & Read16 (std::istream &is)
- void SetToUndefined ()
- template<typename TSwap >
  const std::ostream & Write (std::ostream &os) const
- template<typename TSwap >
  const std::ostream & Write16 (std::ostream &os) const

Static Public Member Functions

- static uint16_t GetVL16Max ()
- static uint32_t GetVL32Max ()

Friends

- std::ostream & operator<< (std::ostream &os, const VL &vl)

10.373.1 Detailed Description

Value Length.

Warning

this is a 4bytes value ! Do not try to use it for 2bytes value length

Examples

BasicImageAnonymizer.cs, DecompressImage.cs, ReadAndDumpDICOMDIR2.cxx, and rle2img.cxx.

10.373.2 Member Typedef Documentation

10.373.2.1 Type

```
typedef uint32_t gdcm::VL::Type
```

10.373.3 Constructor & Destructor Documentation

10.373.3.1 VL()

```
gdcm::VL::VL (
             uint32_t vl = 0 )  [inline]
```

10.373.4 Member Function Documentation

10.373.4.1 GetLength()

```
VL gdcm::VL::GetLength ( ) const  [inline]
```

Examples

ReadAndDumpDICOMDIR2.cxx.

Referenced by gdcm::FileMetaInformation::GetFullLength(), gdcm::DataSet::GetLength(), and gdcm::Item::Write().

10.373.4.2 GetVL16Max()

```
static uint16_t gdcm::VL::GetVL16Max ( )  [inline], [static]
```

10.373.4.3 GetVL32Max()

```
static uint32_t gdcm::VL::GetVL32Max ( )  [inline], [static]
```

10.373.4.4 IsOdd()

```
bool gdcm::VL::IsOdd ( ) const  [inline]
```

Return whether or not the VL is odd or not.

10.373.4.5 IsUndefined()

```
bool gdcm::VL::IsUndefined ( ) const  [inline]
```

10.373.4.6 operator uint32_t()

```
gdcm::VL::operator uint32_t ( ) const  [inline]
```

10.373.4.7 operator++() [1/2]

```
VL & gdcm::VL::operator++ ( )  [inline]
```

10.373.4.8 operator++() [2/2]

```
VL gdcm::VL::operator++ (
             int  )  [inline]
```

10.373.4.9 operator+=()

```
VL & gdcm::VL::operator+= (
             VL const & vl )  [inline]
```

+= operator

10.373.4.10 Read()

```
template<typename TSwap >
std::istream & gdcm::VL::Read (
             std::istream & is )  [inline]
```

10.373.4.11 Read16()

```
template<typename TSwap >
std::istream & gdcm::VL::Read16 (
             std::istream & is )  [inline]
```

10.373.4.12 SetToUndefined()

```
void gdcm::VL::SetToUndefined ( )  [inline]
```

10.373.4.13 Write()

```
template<typename TSwap >
const std::ostream & gdcm::VL::Write (
             std::ostream & os ) const  [inline]
```

Referenced by gdcm::Fragment::Write(), gdcm::Item::Write(), gdcm::SequenceOfFragments::Write(), and gdcm::SequenceOfItems::Write(

10.373.4.14 Write16()

```
template<typename TSwap >
const std::ostream & gdcm::VL::Write16 (
             std::ostream & os ) const  [inline]
```

10.373.5 Friends And Related Function Documentation

10.373.5.1 operator<<

```
std::ostream & operator<< (
            std::ostream & os,
            const VL & vl )  [friend]
```

The documentation for this class was generated from the following file:

- gdcmVL.h

10.374 gdcm::VM Class Reference

Value Multiplicity Looking at the DICOMV3 dict only there is very few cases: 1 2 3 4 5 6 8 16 24 1-2 1-3 1-8 1-32 1-99 1-n 2-2n 2-n 3-3n 3-n.

```
#include <gdcmVM.h>
```

Public Types

- enum VMType {
  VM0 = 0 ,
  VM1 = 1 ,
  VM2 = 2 ,
  VM3 = 4 ,
  VM4 = 8 ,
  VM5 = 16 ,
  VM6 = 32 ,
  VM8 = 64 ,
  VM9 = 128 ,
  VM10 = 256 ,
  VM12 = 512 ,
  VM16 = 1024 ,
  VM18 = 2048 ,
  VM24 = 4096 ,
  VM28 = 8192 ,
  VM32 = 16384 ,
  VM35 = 32768 ,
  VM99 = 65536 ,
  VM256 = 131072 ,
  VM1_2 = VM1 | VM2 ,
  VM1_3 = VM1 | VM2 | VM3 ,
  VM1_4 = VM1 | VM2 | VM3 | VM4 ,
  VM1_5 = VM1 | VM2 | VM3 | VM4 | VM5 ,

VM1_8 = VM1 | VM2 | VM3 | VM4 | VM5 | VM6 | VM8 ,
VM1_32 = VM1 | VM2 | VM3 | VM4 | VM5 | VM6 | VM8 | VM9 | VM16 | VM24 | VM32 ,
VM1_99 = VM1 | VM2 | VM3 | VM4 | VM5 | VM6 | VM8 | VM9 | VM16 | VM24 | VM32 | VM99 ,
VM1_n = VM1 | VM2 | VM3 | VM4 | VM5 | VM6 | VM8 | VM9 | VM16 | VM24 | VM32 | VM99 | VM256 ,
VM2_2n = VM2 | VM4 | VM6 | VM8 | VM16 | VM24 | VM32 | VM256 ,
VM2_n = VM2 | VM3 | VM4 | VM5 | VM6 | VM8 | VM9 | VM16 | VM24 | VM32 | VM99 | VM256 ,
VM3_4 = VM3 | VM4 ,
VM3_3n = VM3 | VM6 | VM9 | VM24 | VM99 | VM256 ,
VM3_n = VM3 | VM4 | VM5 | VM6 | VM8 | VM9 | VM16 | VM24 | VM32 | VM99 | VM256 ,
VM4_4n = VM4 | VM16 | VM24 | VM32 | VM256 ,
VM6_6n = VM6 | VM12 | VM18 | VM24 ,
VM6_n = VM6 | VM8 | VM9 | VM16 | VM24 | VM32 | VM99 | VM256 ,
VM7_7n ,
VM30_30n ,
VM47_47n ,
VM_END = VM1_n + 1 }

Public Member Functions

- VM (VMType type=VM0)
- bool Compatible (VM const &vm) const
- unsigned int GetLength () const
- operator VMType () const

Static Public Member Functions

- static size_t GetNumberOfElementsFromArray (const char *array, size_t length)
- static const char * GetVMString (VMType vm)
- static VMType GetVMType (const char *vm)
- static VMType GetVMTypeFromLength (size_t length, unsigned int size)
- static bool IsValid (int vm1, VMType vm2)

Static Protected Member Functions

- static unsigned int GetIndex (VMType vm)

Friends

- std::ostream & operator<< (std::ostream &os, const VM &vm)

10.374.1 Detailed Description

Value Multiplicity Looking at the DICOMV3 dict only there is very few cases: 1 2 3 4 5 6 8 16 24 1-2 1-3 1-8 1-32 1-99 1-n 2-2n 2-n 3-3n 3-n.

Some private dict define some more: 4-4n 1-4 1-5 256 9 3-4

even more:

7-7n 10 18 12 35 47_47n 30_30n 28

6-6n

10.374.2 Member Enumeration Documentation

10.374.2.1 VMType

enum gdcm::VM::VMType

Enumerator

VM0	
VM1	
VM2	
VM3	
VM4	
VM5	
VM6	
VM8	
VM9	
VM10	
VM12	
VM16	
VM18	
VM24	
VM28	
VM32	
VM35	
VM99	
VM256	
VM1_2	
VM1_3	
VM1_4	
VM1_5	
VM1_8	
VM1_32	
VM1_99	
VM1_n	
VM2_2n	
VM2_n	
VM3_4	
VM3_3n	
VM3_n	
VM4_4n	
VM6_6n	
VM6_n	
VM7_7n	
VM30_30n	
VM47_47n	
VM_END	

10.374.3 Constructor & Destructor Documentation

10.374.3.1 VM()

```
gdcm::VM::VM (
            VMType type = VM0 )  [inline]
```

10.374.4 Member Function Documentation

10.374.4.1 Compatible()

```
bool gdcm::VM::Compatible (
            VM const & vm ) const
```

WARNING: Implementation deficiency The Compatible function is poorly implemented, the reference vm should be coming from the dictionary, while the passed in value is the value guess from the file.

10.374.4.2 GetIndex()

```
static unsigned int gdcm::VM::GetIndex (
            VMType vm )  [static], [protected]
```

10.374.4.3 GetLength()

```
unsigned int gdcm::VM::GetLength ( ) const
```

10.374.4.4 GetNumberOfElementsFromArray()

```
static size_t gdcm::VM::GetNumberOfElementsFromArray (
            const char * array,
            size_t length )  [static]
```

10.374.4.5 GetVMString()

```
static const char * gdcm::VM::GetVMString (
            VMType vm )  [static]
```

Return the string as written in the official DICOM dict from a custom enum type

10.374.4.6 GetVMType()

```
static VMType gdcm::VM::GetVMType (
            const char * vm )  [static]
```

10.374.4.7 GetVMTypeFromLength()

```
static VMType gdcm::VM::GetVMTypeFromLength (
            size_t length,
            unsigned int size )  [static]
```

10.374.4.8 IsValid()

```
static bool gdcm::VM::IsValid (
            int vm1,
            VMType vm2 )  [static]
```

Check if vm1 is valid compare to vm2, i.e vm1 is element of vm2 vm1 is typically deduce from counting in a ValueField

10.374.4.9 operator VMType()

```
gdcm::VM::operator VMType ( ) const  [inline]
```

10.374.5 Friends And Related Function Documentation

10.374.5.1 operator<<

```
std::ostream & operator<< (
            std::ostream & os,
            const VM & vm )  [friend]
```

The documentation for this class was generated from the following file:

- gdcmVM.h

10.375 gdcm::VMToLength< T > Struct Template Reference

The documentation for this struct was generated from the following file:

- gdcmVM.h

10.376 gdcm::VR Class Reference

VR class.

`#include <gdcmVR.h>`

Public Types

- enum VRType : long long {
  INVALID = 0 ,
  AE = 1 ,
  AS = 2 ,
  AT = 4 ,
  CS = 8 ,
  DA = 16 ,
  DS = 32 ,
  DT = 64 ,
  FD = 128 ,
  FL = 256 ,
  IS = 512 ,
  LO = 1024 ,
  LT = 2048 ,
  OB = 4096 ,
  OD = 134217728 ,
  OF = 8192 ,
  OL = 268435456 ,
  OV = 2147483648 ,
  OW = 16384 ,
  PN = 32768 ,
  SH = 65536 ,
  SL = 131072 ,
  SQ = 262144 ,
  SS = 524288 ,
  ST = 1048576 ,
  SV = 4294967296 ,
  TM = 2097152 ,
  UC = 536870912 ,
  UI = 4194304 ,
  UL = 8388608 ,
  UN = 16777216 ,
  UR = 1073741824 ,
  US = 33554432 ,
  UT = 67108864 ,

UV = 8589934592 ,
OB_OW = OB | OW ,
US_SS = US | SS ,
US_SS_OW = US | SS | OW ,
US_OW = US | OW ,
VL16 = AE | AS | AT | CS | DA | DS | DT | FD | FL | IS | LO | LT | PN | SH | SL | SS | ST | TM | UI | UL | US ,
VL32 = OB | OW | OD | OF | OL | OV | SQ | SV | UC | UN | UR | UT | UV ,
VRASCII = AE | AS | CS | DA | DS | DT | IS | LO | LT | PN | SH | ST | TM | UC | UI | UR | UT ,
VRBINARY = AT | FL | FD | OB | OD | OF | OL | OV | OW | SL | SQ | SS | SV | UL | UN | US | UV ,
VR_VM1 = AS | LT | ST | UT | SQ | OF | OL | OV | OD | OW | OB | UN ,
VRALL = VRASCII | VRBINARY ,
VR_END = UV+1 }

Public Member Functions

- VR (VRType vr=INVALID)
- bool Compatible (VR const &vr) const
- int GetLength () const
- unsigned int GetSize () const
- unsigned int GetSizeof () const
- bool IsDual () const
- bool IsVRFile () const
- operator VRType () const
- std::istream & Read (std::istream &is)
- const std::ostream & Write (std::ostream &os) const

Static Public Member Functions

- static bool CanDisplay (VRType vr)
- static uint32_t GetLength (VRType vr)
- static const char * GetVRString (VRType vr)
- static const char * GetVRStringFromFile (VRType vr)
- static VRType GetVRType (const char *vr)
- static VRType GetVRTypeFromFile (const char *vr)
- static bool IsASCII (VRType vr)
- static bool IsASCII2 (VRType vr)
- static bool IsBinary (VRType vr)
- static bool IsBinary2 (VRType vr)
- static bool IsSwap (const char *vr)
- static bool IsValid (const char *vr)
- static bool IsValid (const char *vr1, VRType vr2)

Friends

- std::ostream & operator<< (std::ostream &os, const VR &vr)

10.376.1 Detailed Description

VR class.

This is adapted from DICOM standard The biggest difference is the INVALID VR and the composite one that differ from standard (more like an addition) This allow us to represent all the possible case express in the DICOMV3 dict

Note

VALUE REPRESENTATION (VR) Specifies the data type and format of the Value(s) contained in the Value Field of a Data Element. VALUE REPRESENTATION FIELD: The field where the Value Representation of a Data Element is stored in the encoding of a Data Element structure with explicit VR.

Examples

GenAllVR.cxx, GenFakeIdentifyFile.cxx, and SimplePrint.cs.

10.376.2 Member Enumeration Documentation

10.376.2.1 VRType

```
enum gdcm::VR::VRType :  long long
```

Enumerator

INVALID	
AE	
AS	
AT	
CS	
DA	
DS	
DT	
FD	
FL	
IS	
LO	
LT	
OB	
OD	
OF	
OL	
OV	
OW	
PN	
SH	

Enumerator

SL	
SQ	
SS	
ST	
SV	
TM	
UC	
UI	
UL	
UN	
UR	
US	
UT	
UV	
OB_OW	
US_SS	
US_SS_OW	
US_OW	
VL16	
VL32	
VRASCII	
VRBINARY	
VR_VM1	
VRALL	
VR_END	

Examples

NewSequence.cs, and SimplePrint.cs.

10.376.3 Constructor & Destructor Documentation

10.376.3.1 VR()

```
gdcm::VR::VR (
            VRType vr = INVALID )  [inline]
```

10.376.4 Member Function Documentation

10.376.4.1 CanDisplay()

```
static bool gdcm::VR::CanDisplay (
            VRType vr )  [static]
```

10.376.4.2 Compatible()

```
bool gdcm::VR::Compatible (
            VR const & vr ) const
```

Examples

SimplePrint.cs.

10.376.4.3 GetLength() [1/2]

```
int gdcm::VR::GetLength ( ) const  [inline]
```

References GetLength().

Referenced by GetLength().

10.376.4.4 GetLength() [2/2]

```
static uint32_t gdcm::VR::GetLength (
            VRType vr )  [inline], [static]
```

10.376.4.5 GetSize()

```
unsigned int gdcm::VR::GetSize ( ) const  [inline]
```

References AE, AS, AT, CS, DA, DS, DT, FD, FL, INVALID, IS, LT, OB, OB_OW, OD, OF, OL, OV, OW, PN, SH, SL, SQ, SS, ST, SV, TM, UC, UL, UN, UR, US, US_OW, US_SS, US_SS_OW, UT, UV, VL16, VL32, VR_END, VR_VM1, VRALL, VRASCII, VRBINARY, and VRTypeTemplateCase.

10.376.4.6 GetSizeof()

```
unsigned int gdcm::VR::GetSizeof ( ) const
```

10.376.4.7 GetVRString()

```
static const char * gdcm::VR::GetVRString (
            VRType vr )  [static]
```

10.376.4.8 GetVRStringFromFile()

```
static const char * gdcm::VR::GetVRStringFromFile (
            VRType vr )  [static]
```

10.376.4.9 GetVRType()

```
static VRType gdcm::VR::GetVRType (
            const char * vr )  [static]
```

10.376.4.10 GetVRTypeFromFile()

```
static VRType gdcm::VR::GetVRTypeFromFile (
            const char * vr )  [static]
```

10.376.4.11 IsASCII()

```
static bool gdcm::VR::IsASCII (
            VRType vr )  [static]
```

10.376.4.12 IsASCII2()

```
static bool gdcm::VR::IsASCII2 (
            VRType vr )  [static]
```

10.376.4.13 IsBinary()

```
static bool gdcm::VR::IsBinary (
            VRType vr )  [static]
```

10.376.4.14 IsBinary2()

```
static bool gdcm::VR::IsBinary2 (
            VRType vr )  [static]
```

10.376.4.15 IsDual()

```
bool gdcm::VR::IsDual ( ) const
```

10.376.4.16 IsSwap()

```
static bool gdcm::VR::IsSwap (
            const char * vr )  [static]
```

10.376.4.17 IsValid() [1/2]

```
static bool gdcm::VR::IsValid (
            const char * vr )  [static]
```

10.376.4.18 IsValid() [2/2]

```
static bool gdcm::VR::IsValid (
            const char * vr1,
            VRType vr2 )  [static]
```

10.376.4.19 IsVRFile()

```
bool gdcm::VR::IsVRFile ( ) const
```

Referenced by gdcm::DataElement::SetVR().

10.376.4.20 operator VRType()

```
gdcm::VR::operator VRType ( ) const  [inline]
```

10.376.4.21 Read()

```
std::istream & gdcm::VR::Read (
            std::istream & is )  [inline]
```

References gdcmDebugMacro, INVALID, and VR_END.

10.376.4.22 Write()

```
const std::ostream & gdcm::VR::Write (
            std::ostream & os ) const  [inline]
```

References gdcmAssertAlwaysMacro, and INVALID.

10.376.5 Friends And Related Function Documentation

10.376.5.1 operator<<

```
std::ostream & operator<< (
             std::ostream & os,
             const VR & vr )  [friend]
```

The documentation for this class was generated from the following file:

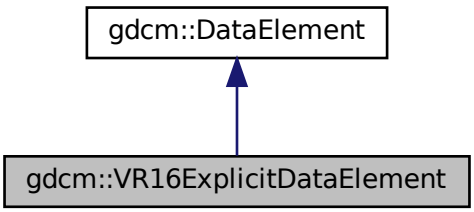
- gdcmVR.h

10.377 gdcm::VR16ExplicitDataElement Class Reference

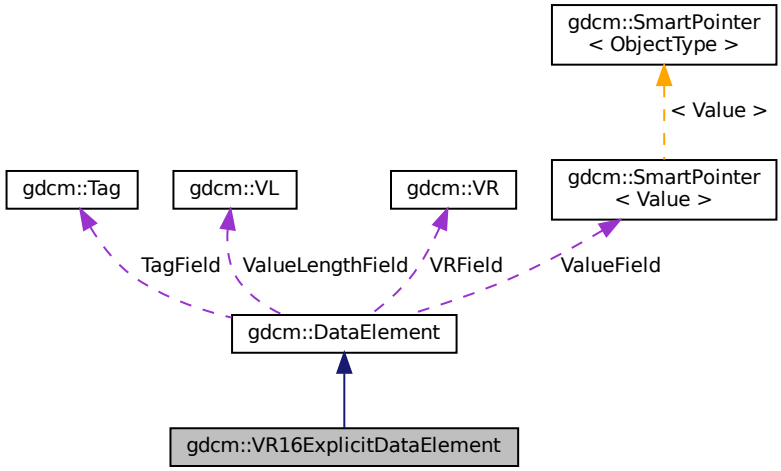
Class to read/write a DataElement as Explicit Data Element.

`#include <gdcmVR16ExplicitDataElement.h>`

Inheritance diagram for gdcm::VR16ExplicitDataElement:



Collaboration diagram for gdcm::VR16ExplicitDataElement:



Public Member Functions

- VL GetLength () const
- template<typename TSwap >
  std::istream & Read (std::istream &is)
- template<typename TSwap >
  std::istream & ReadPreValue (std::istream &is)
- template<typename TSwap >
  std::istream & ReadValue (std::istream &is, bool readvalues=true)
- template<typename TSwap >
  std::istream & ReadWithLength (std::istream &is, VL &length)

Additional Inherited Members

10.377.1 Detailed Description

Class to read/write a DataElement as Explicit Data Element.

Note

This class support 16 bits when finding an unknown VR: For instance: Siemens_CT_Sensation64_has_VR_RT.↩ dcm

10.377.2 Member Function Documentation

10.377.2.1 GetLength()

```
VL gdcm::VR16ExplicitDataElement::GetLength ( ) const
```

10.377.2.2 Read()

```
template<typename TSwap >
std::istream & gdcm::VR16ExplicitDataElement::Read (
             std::istream & is )
```

10.377.2.3 ReadPreValue()

```
template<typename TSwap >
std::istream & gdcm::VR16ExplicitDataElement::ReadPreValue (
             std::istream & is )
```

10.377.2.4 ReadValue()

```
template<typename TSwap >
std::istream & gdcm::VR16ExplicitDataElement::ReadValue (
             std::istream & is,
             bool readvalues = true )
```

10.377.2.5 ReadWithLength()

```
template<typename TSwap >
std::istream & gdcm::VR16ExplicitDataElement::ReadWithLength (
             std::istream & is,
             VL & length )
```

The documentation for this class was generated from the following file:

- gdcmVR16ExplicitDataElement.h

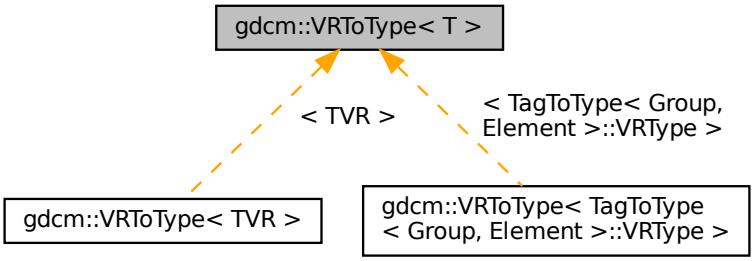
10.378 gdcm::VRToEncoding< T > Struct Template Reference

The documentation for this struct was generated from the following file:

- gdcmVR.h

10.379 gdcm::VRToType< T > Struct Template Reference

Inheritance diagram for gdcm::VRToType< T >:



10.379.1 Detailed Description

template<long long T>
struct gdcm::VRToType< T >

Examples

DumpGEMSMovieGroup.cxx.

The documentation for this struct was generated from the following file:

- gdcmVR.h

10.380 gdcm::VRVLSize< T > Class Template Reference

The documentation for this class was generated from the following file:

- gdcmAttribute.h

10.381 gdcm::VRVLSize< 0 > Class Reference

`#include <gdcmAttribute.h>`

Static Public Member Functions

- static uint16_t Read (std::istream &_is)
- static void Write (std::ostream &os)

10.381.1 Member Function Documentation

10.381.1.1 Read()

```
static uint16_t gdcm::VRVLSize< 0 >::Read (
            std::istream & _is )  [inline], [static]
```

10.381.1.2 Write()

```
static void gdcm::VRVLSize< 0 >::Write (
            std::ostream & os )  [inline], [static]
```

The documentation for this class was generated from the following file:

- gdcmAttribute.h

10.382 gdcm::VRVLSize< 1 > Class Reference

`#include <gdcmAttribute.h>`

Static Public Member Functions

- static uint32_t Read (std::istream &_is)
- static void Write (std::ostream &os)

10.382.1 Member Function Documentation

10.382.1.1 Read()

```
static uint32_t gdcm::VRVLSize< 1 >::Read (
             std::istream & _is )  [inline], [static]
```

10.382.1.2 Write()

```
static void gdcm::VRVLSize< 1 >::Write (
             std::ostream & os )  [inline], [static]
```

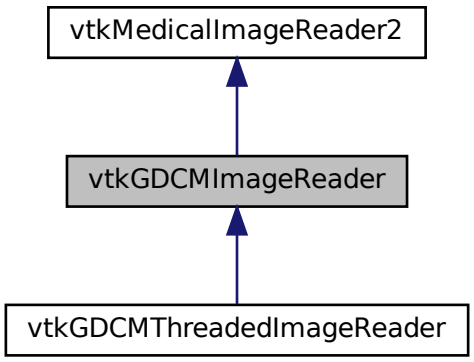
The documentation for this class was generated from the following file:

- gdcmAttribute.h

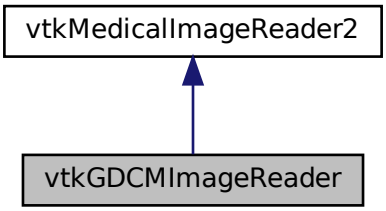
10.383 vtkGDCMImageReader Class Reference

```
#include <vtkGDCMImageReader.h>
```

Inheritance diagram for vtkGDCMImageReader:



Collaboration diagram for vtkGDCMImageReader:



Public Member Functions

- virtual int CanReadFile (const char *fname)
- virtual const char * GetDescriptiveName ()
- virtual const char * GetFileExtensions ()
- vtkImageData * GetIconImage ()
- vtkImageData * GetOverlay (int i)
- virtual void PrintSelf (ostream &os, vtkIndent indent)
- virtual void SetCurve (vtkPolyData *pd)
- virtual void SetFileNames (vtkStringArray *)
- virtual void SetMedicalImageProperties (vtkMedicalImageProperties *pd)
- vtkBooleanMacro (ApplyLookupTable, int)
- int vtkBooleanMacro (ApplyYBRToRGB, int)
- vtkBooleanMacro (LoadIconImage, int)
- vtkBooleanMacro (LoadOverlays, int)
- vtkBooleanMacro (LossyFlag, int)
- vtkGetMacro (ApplyLookupTable, int)
- vtkGetMacro (ApplyYBRToRGB, int) vtkSetMacro(ApplyYBRToRGB
- vtkGetMacro (ImageFormat, int)
- vtkGetMacro (LoadIconImage, int)
- vtkGetMacro (LoadOverlays, int)
- vtkGetMacro (LossyFlag, int)
- vtkGetMacro (NumberOfIconImages, int)
- vtkGetMacro (NumberOfOverlays, int)
- vtkGetMacro (PlanarConfiguration, int)
- vtkGetMacro (Scale, double)
- vtkGetMacro (Shift, double)
- vtkGetObjectMacro (Curve, vtkPolyData)
- vtkGetObjectMacro (DirectionCosines, vtkMatrix4x4)
- vtkGetObjectMacro (FileNames, vtkStringArray)
- vtkGetObjectMacro (MedicalImageProperties, vtkMedicalImageProperties)
- vtkGetVector3Macro (ImagePositionPatient, double)
- vtkGetVector6Macro (ImageOrientationPatient, double)
- vtkSetMacro (ApplyLookupTable, int)

- vtkSetMacro (LoadIconImage, int)
- vtkSetMacro (LoadOverlays, int)
- vtkSetMacro (LossyFlag, int)
- vtkTypeMacro (vtkGDCMImageReader, vtkMedicalImageReader2)

Static Public Member Functions

- static vtkGDCMImageReader * New ()

Protected Member Functions

- vtkGDCMImageReader ()
- ~vtkGDCMImageReader ()
- void ExecuteData (vtkDataObject *out)
- void ExecuteInformation ()
- void FillMedicalImageInformation (const gdcm::ImageReader &reader)
- int LoadSingleFile (const char *filename, char *pointer, unsigned long &outlen)
- int RequestDataCompat ()
- int RequestInformationCompat ()
- void SetFilePattern (const char *)
- void SetFilePrefix (const char *)
- vtkGetStringMacro (FilePattern)
- vtkGetStringMacro (FilePrefix)
- vtkSetVector6Macro (ImageOrientationPatient, double)

Protected Attributes

- int ApplyInverseVideo
- int ApplyLookupTable
- int ApplyPlanarConfiguration
- int ApplyShiftScale
- int ApplyYBRToRGB
- vtkPolyData * Curve
- vtkMatrix4x4 * DirectionCosines
- vtkStringArray * FileNames
- int ForceRescale
- int IconDataScalarType
- int IconImageDataExtent [6]
- int IconNumberOfScalarComponents
- int ImageFormat
- double ImageOrientationPatient [6]
- double ImagePositionPatient [3]
- int LoadIconImage
- int LoadOverlays
- int LossyFlag
- vtkMedicalImageProperties * MedicalImageProperties
- int NumberOfIconImages
- int NumberOfOverlays
- int PlanarConfiguration
- double Scale
- double Shift

10.383.1 Detailed Description

Examples

AWTMedical3.java, Convert16BitsTo8Bits.cxx, ConvertMultiFrameToSingleFrame.cxx, ConvertRGBToLuminance.cxx, ConvertSingleBitTo8Bits.cxx, HelloActiviz.cs, HelloActiviz2.cs, HelloActiviz3.cs, HelloActiviz4.cs, HelloActiviz5.cs, HelloVTKWorld.cs, HelloVTKWorld.java, MIPViewer.java, MPRViewer.java, MPRViewer2.java, MagnifyFile.cxx, MetaImageMD5Activiz.cs, ReadSeriesIntoVTK.java, RefCounting.cs, gdcmorthoplanes.cxx, gdcmreslice.cxx, gdcmtexture.cxx, gdcmvolume.cxx, offscreenimage.cxx, and reslicesphere.cxx.

10.383.2 Constructor & Destructor Documentation

10.383.2.1 vtkGDCMImageReader()

```
vtkGDCMImageReader::vtkGDCMImageReader ( )  [protected]
```

Examples

HelloActiviz2.cs.

10.383.2.2 ~vtkGDCMImageReader()

```
vtkGDCMImageReader::~vtkGDCMImageReader ( )  [protected]
```

10.383.3 Member Function Documentation

10.383.3.1 CanReadFile()

```
virtual int vtkGDCMImageReader::CanReadFile (
            const char * fname )  [virtual]
```

Examples

AWTMedical3.java, and MetaImageMD5Activiz.cs.

10.383.3.2 ExecuteData()

```
void vtkGDCMImageReader::ExecuteData (
            vtkDataObject * out )  [protected]
```

10.383.3.3 ExecuteInformation()

```
void vtkGDCMImageReader::ExecuteInformation ( )  [protected]
```

10.383.3.4 FillMedicalImageInformation()

```
void vtkGDCMImageReader::FillMedicalImageInformation (
            const gdcm::ImageReader & reader )  [protected]
```

10.383.3.5 GetDescriptiveName()

```
virtual const char * vtkGDCMImageReader::GetDescriptiveName ( )  [inline], [virtual]
```

10.383.3.6 GetFileExtensions()

```
virtual const char * vtkGDCMImageReader::GetFileExtensions ( )  [inline], [virtual]
```

10.383.3.7 GetIconImage()

```
vtkImageData * vtkGDCMImageReader::GetIconImage ( )
```

10.383.3.8 GetOverlay()

```
vtkImageData * vtkGDCMImageReader::GetOverlay (
            int i )
```

10.383.3.9 LoadSingleFile()

```
int vtkGDCMImageReader::LoadSingleFile (
            const char * filename,
            char * pointer,
            unsigned long & outlen )  [protected]
```

10.383.3.10 New()

```
static vtkGDCMImageReader * vtkGDCMImageReader::New ( )  [static]
```

Examples

Convert16BitsTo8Bits.cxx, ConvertMultiFrameToSingleFrame.cxx, ConvertRGBToLuminance.cxx, ConvertSingleBitTo8Bits.cxx, HelloActiviz.cs, HelloActiviz3.cs, HelloActiviz4.cs, HelloActiviz5.cs, HelloVTKWorld.cs, MagnifyFile.cxx, MetaImageMD5Activiz.cs, RefCounting.cs, gdcmorthoplanes.cxx, gdcmreslice.cxx, gdcmtexture.cxx, gdcmvolume.cxx, offscreenimage.cxx, and reslicesphere.cxx.

10.383.3.11 PrintSelf()

```
virtual void vtkGDCMImageReader::PrintSelf (
            ostream & os,
            vtkIndent indent )  [virtual]
```

Reimplemented in vtkGDCMThreadedImageReader.

10.383.3.12 RequestDataCompat()

```
int vtkGDCMImageReader::RequestDataCompat ( )  [protected]
```

10.383.3.13 RequestInformationCompat()

```
int vtkGDCMImageReader::RequestInformationCompat ( )  [protected]
```

10.383.3.14 SetCurve()

```
virtual void vtkGDCMImageReader::SetCurve (
            vtkPolyData * pd )  [virtual]
```

10.383.3.15 SetFileNames()

```
virtual void vtkGDCMImageReader::SetFileNames (
            vtkStringArray *  )  [virtual]
```

Examples

AWTMedical3.java, HelloActiviz3.cs, HelloActiviz4.cs, HelloActiviz5.cs, MIPViewer.java, MPRViewer.java, MPRViewer2.java, ReadSeriesIntoVTK.java, and gdcmorthoplanes.cxx.

10.383.3.16 SetFilePattern()

```
void vtkGDCMImageReader::SetFilePattern (
            const char *  )  [inline], [protected]
```

10.383.3.17 SetFilePrefix()

```
void vtkGDCMImageReader::SetFilePrefix (
            const char *  )  [inline], [protected]
```

10.383.3.18 SetMedicalImageProperties()

```
virtual void vtkGDCMImageReader::SetMedicalImageProperties (
            vtkMedicalImageProperties * pd )  [virtual]
```

10.383.3.19 vtkBooleanMacro() [1/5]

```
vtkGDCMImageReader::vtkBooleanMacro (
            ApplyLookupTable ,
            int  )
```

10.383.3.20 vtkBooleanMacro() [2/5]

```
int vtkGDCMImageReader::vtkBooleanMacro (
            ApplyYBRToRGB ,
            int  )
```

10.383.3.21 vtkBooleanMacro() [3/5]

```
vtkGDCMImageReader::vtkBooleanMacro (
            LoadIconImage ,
            int  )
```

10.383.3.22 vtkBooleanMacro() [4/5]

```
vtkGDCMImageReader::vtkBooleanMacro (
            LoadOverlays ,
            int  )
```

10.383.3.23 vtkBooleanMacro() [5/5]

```
vtkGDCMImageReader::vtkBooleanMacro (
            LossyFlag ,
            int  )
```

10.383.3.24 vtkGetMacro() [1/11]

```
vtkGDCMImageReader::vtkGetMacro (
            ApplyLookupTable ,
            int  )
```

10.383.3.25 vtkGetMacro() [2/11]

```
vtkGDCMImageReader::vtkGetMacro (
            ApplyYBRToRGB ,
            int  )
```

10.383.3.26 vtkGetMacro() [3/11]

```
vtkGDCMImageReader::vtkGetMacro (
            ImageFormat ,
            int  )
```

10.383.3.27 vtkGetMacro() [4/11]

```
vtkGDCMImageReader::vtkGetMacro (
            LoadIconImage ,
            int  )
```

10.383.3.28 vtkGetMacro() [5/11]

```
vtkGDCMImageReader::vtkGetMacro (
            LoadOverlays ,
            int  )
```

10.383.3.29 vtkGetMacro() [6/11]

```
vtkGDCMImageReader::vtkGetMacro (
            LossyFlag ,
            int  )
```

10.383.3.30 vtkGetMacro() [7/11]

```
vtkGDCMImageReader::vtkGetMacro (
            NumberOfIconImages ,
            int  )
```

10.383.3.31 vtkGetMacro() [8/11]

```
vtkGDCMImageReader::vtkGetMacro (
            NumberOfOverlays ,
            int  )
```

10.383.3.32 vtkGetMacro() [9/11]

```
vtkGDCMImageReader::vtkGetMacro (
            PlanarConfiguration ,
            int  )
```

10.383.3.33 vtkGetMacro() [10/11]

```
vtkGDCMImageReader::vtkGetMacro (
            Scale ,
            double  )
```

10.383.3.34 vtkGetMacro() [11/11]

```
vtkGDCMImageReader::vtkGetMacro (
            Shift ,
            double  )
```

10.383.3.35 vtkGetObjectMacro() [1/4]

```
vtkGDCMImageReader::vtkGetObjectMacro (
            Curve ,
            vtkPolyData  )
```

10.383.3.36 vtkGetObjectMacro() [2/4]

```
vtkGDCMImageReader::vtkGetObjectMacro (
            DirectionCosines ,
            vtkMatrix4x4  )
```

10.383.3.37 vtkGetObjectMacro() [3/4]

```
vtkGDCMImageReader::vtkGetObjectMacro (
            FileNames ,
            vtkStringArray  )
```

10.383.3.38 vtkGetObjectMacro() [4/4]

```
vtkGDCMImageReader::vtkGetObjectMacro (
            MedicalImageProperties ,
            vtkMedicalImageProperties  )
```

10.383.3.39 vtkGetStringMacro() [1/2]

```
vtkGDCMImageReader::vtkGetStringMacro (
            FilePattern  )  [protected]
```

10.383.3.40 vtkGetStringMacro() [2/2]

```
vtkGDCMImageReader::vtkGetStringMacro (
            FilePrefix  )  [protected]
```

10.383.3.41 vtkGetVector3Macro()

```
vtkGDCMImageReader::vtkGetVector3Macro (
            ImagePositionPatient ,
            double  )
```

10.383.3.42 vtkGetVector6Macro()

```
vtkGDCMImageReader::vtkGetVector6Macro (
            ImageOrientationPatient ,
            double  )
```

10.383.3.43 vtkSetMacro() [1/4]

```
vtkGDCMImageReader::vtkSetMacro (
            ApplyLookupTable ,
            int  )
```

10.383.3.44 vtkSetMacro() [2/4]

```
vtkGDCMImageReader::vtkSetMacro (
            LoadIconImage ,
            int  )
```

10.383.3.45 vtkSetMacro() [3/4]

```
vtkGDCMImageReader::vtkSetMacro (
            LoadOverlays ,
            int  )
```

10.383.3.46 vtkSetMacro() [4/4]

```
vtkGDCMImageReader::vtkSetMacro (
            LossyFlag ,
            int  )
```

10.383.3.47 vtkSetVector6Macro()

```
vtkGDCMImageReader::vtkSetVector6Macro (
            ImageOrientationPatient ,
            double  )  [protected]
```

10.383.3.48 vtkTypeMacro()

```
vtkGDCMImageReader::vtkTypeMacro (
            vtkGDCMImageReader ,
            vtkMedicalImageReader2  )
```

10.383.4 Member Data Documentation

10.383.4.1 ApplyInverseVideo

```
int vtkGDCMImageReader::ApplyInverseVideo  [protected]
```

10.383.4.2 ApplyLookupTable

```
int vtkGDCMImageReader::ApplyLookupTable  [protected]
```

10.383.4.3 ApplyPlanarConfiguration

```
int vtkGDCMImageReader::ApplyPlanarConfiguration  [protected]
```

10.383.4.4 ApplyShiftScale

```
int vtkGDCMImageReader::ApplyShiftScale  [protected]
```

10.383.4.5 ApplyYBRToRGB

```
int vtkGDCMImageReader::ApplyYBRToRGB  [protected]
```

10.383.4.6 Curve

```
vtkPolyData* vtkGDCMImageReader::Curve  [protected]
```

10.383.4.7 DirectionCosines

```
vtkMatrix4x4* vtkGDCMImageReader::DirectionCosines  [protected]
```

10.383.4.8 FileNames

```
vtkStringArray* vtkGDCMImageReader::FileNames  [protected]
```

10.383.4.9 ForceRescale

```
int vtkGDCMImageReader::ForceRescale  [protected]
```

10.383.4.10 IconDataScalarType

```
int vtkGDCMImageReader::IconDataScalarType  [protected]
```

10.383.4.11 IconImageDataExtent

```
int vtkGDCMImageReader::IconImageDataExtent[6]  [protected]
```

10.383.4.12 IconNumberOfScalarComponents

```
int vtkGDCMImageReader::IconNumberOfScalarComponents  [protected]
```

10.383.4.13 ImageFormat

```
int vtkGDCMImageReader::ImageFormat  [protected]
```

10.383.4.14 ImageOrientationPatient

```
double vtkGDCMImageReader::ImageOrientationPatient[6]  [protected]
```

10.383.4.15 ImagePositionPatient

```
double vtkGDCMImageReader::ImagePositionPatient[3]  [protected]
```

10.383.4.16 LoadIconImage

```
int vtkGDCMImageReader::LoadIconImage  [protected]
```

10.383.4.17 LoadOverlays

```
int vtkGDCMImageReader::LoadOverlays  [protected]
```

10.383.4.18 LossyFlag

```
int vtkGDCMImageReader::LossyFlag  [protected]
```

10.383.4.19 MedicalImageProperties

```
vtkMedicalImageProperties* vtkGDCMImageReader::MedicalImageProperties  [protected]
```

10.383.4.20 NumberOfIconImages

```
int vtkGDCMImageReader::NumberOfIconImages  [protected]
```

10.383.4.21 NumberOfOverlays

```
int vtkGDCMImageReader::NumberOfOverlays  [protected]
```

10.383.4.22 PlanarConfiguration

```
int vtkGDCMImageReader::PlanarConfiguration  [protected]
```

10.383.4.23 Scale

```
double vtkGDCMImageReader::Scale  [protected]
```

10.383.4.24 Shift

```
double vtkGDCMImageReader::Shift  [protected]
```

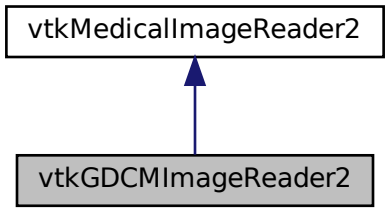
The documentation for this class was generated from the following file:

- vtkGDCMImageReader.h

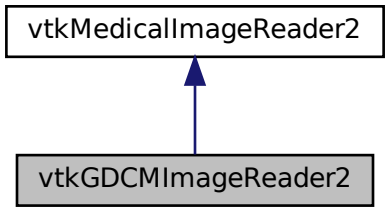
10.384 vtkGDCMImageReader2 Class Reference

```
#include <vtkGDCMImageReader2.h>
```

Inheritance diagram for vtkGDCMImageReader2:



Collaboration diagram for vtkGDCMImageReader2:



Public Member Functions

- virtual int CanReadFile (const char ∗fname)
- virtual const char ∗ GetDescriptiveName ()
- virtual const char ∗ GetFileExtensions ()
- vtkImageData ∗ GetIconImage ()
- vtkAlgorithmOutput ∗ GetIconImagePort ()
- vtkImageData ∗ GetOverlay (int i)
- vtkAlgorithmOutput ∗ GetOverlayPort (int index)
- virtual void PrintSelf (ostream &os, vtkIndent indent)
- virtual void SetCurve (vtkPolyData ∗pd)
- virtual void SetMedicalImageProperties (vtkMedicalImageProperties ∗pd)
- vtkBooleanMacro (ApplyLookupTable, int)
- int vtkBooleanMacro (ApplyYBRToRGB, int)
- vtkBooleanMacro (LoadIconImage, int)
- vtkBooleanMacro (LoadOverlays, int)
- vtkBooleanMacro (LossyFlag, int)
- vtkGetMacro (ApplyLookupTable, int)
- vtkGetMacro (ApplyYBRToRGB, int) vtkSetMacro(ApplyYBRToRGB
- vtkGetMacro (ImageFormat, int)
- vtkGetMacro (LoadIconImage, int)
- vtkGetMacro (LoadOverlays, int)
- vtkGetMacro (LossyFlag, int)
- vtkGetMacro (NumberOfIconImages, int)
- vtkGetMacro (NumberOfOverlays, int)
- vtkGetMacro (PlanarConfiguration, int)
- vtkGetMacro (Scale, double)
- vtkGetMacro (Shift, double)
- vtkGetObjectMacro (Curve, vtkPolyData)
- vtkGetObjectMacro (DirectionCosines, vtkMatrix4x4)
- vtkGetVector3Macro (ImagePositionPatient, double)
- vtkGetVector6Macro (ImageOrientationPatient, double)
- vtkSetMacro (ApplyLookupTable, int)
- vtkSetMacro (LoadIconImage, int)
- vtkSetMacro (LoadOverlays, int)
- vtkSetMacro (LossyFlag, int)
- vtkTypeMacro (vtkGDCMImageReader2, vtkMedicalImageReader2)

Static Public Member Functions

- static vtkGDCMImageReader2 * New ()

Protected Member Functions

- vtkGDCMImageReader2 ()
- ~vtkGDCMImageReader2 ()
- void FillMedicalImageInformation (const gdcm::ImageReader &reader)
- int LoadSingleFile (const char *filename, char *pointer, unsigned long &outlen)
- int ProcessRequest (vtkInformation *request, vtkInformationVector **inputVector, vtkInformationVector *output↩ Vector)
- int RequestData (vtkInformation *request, vtkInformationVector **inputVector, vtkInformationVector *output↩ Vector)
- int RequestDataCompat ()
- int RequestInformation (vtkInformation *request, vtkInformationVector **inputVector, vtkInformationVector *outputVector)
- int RequestInformationCompat ()
- void SetFilePattern (const char *)
- void SetFilePrefix (const char *)
- vtkGetStringMacro (FilePattern)
- vtkGetStringMacro (FilePrefix)
- vtkSetVector6Macro (ImageOrientationPatient, double)

Protected Attributes

- int ApplyInverseVideo
- int ApplyLookupTable
- int ApplyPlanarConfiguration
- int ApplyShiftScale
- int ApplyYBRToRGB
- vtkPolyData * Curve
- vtkMatrix4x4 * DirectionCosines
- int ForceRescale
- int IconDataScalarType
- int IconImageDataExtent [6]
- int IconNumberOfScalarComponents
- int ImageFormat
- double ImageOrientationPatient [6]
- double ImagePositionPatient [3]
- int LoadIconImage
- int LoadOverlays
- int LossyFlag
- int NumberOfIconImages
- int NumberOfOverlays
- int PlanarConfiguration
- double Scale
- double Shift

10.384.1 Detailed Description

Examples

Compute3DSpacing.cxx.

10.384.2 Constructor & Destructor Documentation

10.384.2.1 vtkGDCMImageReader2()

```
vtkGDCMImageReader2::vtkGDCMImageReader2 ( )  [protected]
```

10.384.2.2 ~vtkGDCMImageReader2()

```
vtkGDCMImageReader2::~vtkGDCMImageReader2 ( )  [protected]
```

10.384.3 Member Function Documentation

10.384.3.1 CanReadFile()

```
virtual int vtkGDCMImageReader2::CanReadFile (
            const char * fname )  [virtual]
```

10.384.3.2 FillMedicalImageInformation()

```
void vtkGDCMImageReader2::FillMedicalImageInformation (
            const gdcm::ImageReader & reader )  [protected]
```

10.384.3.3 GetDescriptiveName()

```
virtual const char * vtkGDCMImageReader2::GetDescriptiveName ( )  [inline], [virtual]
```

10.384.3.4 GetFileExtensions()

```
virtual const char * vtkGDCMImageReader2::GetFileExtensions ( )  [inline], [virtual]
```

10.384.3.5 GetIconImage()

```
vtkImageData * vtkGDCMImageReader2::GetIconImage ( )
```

10.384.3.6 GetIconImagePort()

```
vtkAlgorithmOutput * vtkGDCMImageReader2::GetIconImagePort ( )
```

10.384.3.7 GetOverlay()

```
vtkImageData * vtkGDCMImageReader2::GetOverlay (
            int i )
```

10.384.3.8 GetOverlayPort()

```
vtkAlgorithmOutput * vtkGDCMImageReader2::GetOverlayPort (
            int index )
```

10.384.3.9 LoadSingleFile()

```
int vtkGDCMImageReader2::LoadSingleFile (
            const char * filename,
            char * pointer,
            unsigned long & outlen )  [protected]
```

10.384.3.10 New()

```
static vtkGDCMImageReader2 * vtkGDCMImageReader2::New ( )  [static]
```

Examples

Compute3DSpacing.cxx.

10.384.3.11 PrintSelf()

```
virtual void vtkGDCMImageReader2::PrintSelf (
            ostream & os,
            vtkIndent indent )  [virtual]
```

10.384.3.12 ProcessRequest()

```
int vtkGDCMImageReader2::ProcessRequest (
            vtkInformation * request,
            vtkInformationVector ** inputVector,
            vtkInformationVector * outputVector )  [protected]
```

10.384.3.13 RequestData()

```
int vtkGDCMImageReader2::RequestData (
            vtkInformation * request,
            vtkInformationVector ** inputVector,
            vtkInformationVector * outputVector )  [protected]
```

10.384.3.14 RequestDataCompat()

```
int vtkGDCMImageReader2::RequestDataCompat ( )  [protected]
```

10.384.3.15 RequestInformation()

```
int vtkGDCMImageReader2::RequestInformation (
            vtkInformation * request,
            vtkInformationVector ** inputVector,
            vtkInformationVector * outputVector )  [protected]
```

10.384.3.16 RequestInformationCompat()

```
int vtkGDCMImageReader2::RequestInformationCompat ( )  [protected]
```

10.384.3.17 SetCurve()

```
virtual void vtkGDCMImageReader2::SetCurve (
            vtkPolyData * pd )  [virtual]
```

10.384.3.18 SetFilePattern()

```
void vtkGDCMImageReader2::SetFilePattern (
            const char *  )  [inline], [protected]
```

10.384.3.19 SetFilePrefix()

```
void vtkGDCMImageReader2::SetFilePrefix (
            const char *  )  [inline], [protected]
```

10.384.3.20 SetMedicalImageProperties()

```
virtual void vtkGDCMImageReader2::SetMedicalImageProperties (
            vtkMedicalImageProperties * pd )  [virtual]
```

10.384.3.21 vtkBooleanMacro() [1/5]

```
vtkGDCMImageReader2::vtkBooleanMacro (
            ApplyLookupTable ,
            int  )
```

10.384.3.22 vtkBooleanMacro() [2/5]

```
int vtkGDCMImageReader2::vtkBooleanMacro (
            ApplyYBRToRGB ,
            int  )
```

10.384.3.23 vtkBooleanMacro() [3/5]

```
vtkGDCMImageReader2::vtkBooleanMacro (
            LoadIconImage ,
            int  )
```

10.384.3.24 vtkBooleanMacro() [4/5]

```
vtkGDCMImageReader2::vtkBooleanMacro (
            LoadOverlays ,
            int  )
```

10.384.3.25 vtkBooleanMacro() [5/5]

```
vtkGDCMImageReader2::vtkBooleanMacro (
            LossyFlag ,
            int  )
```

10.384.3.26 vtkGetMacro() [1/11]

```
vtkGDCMImageReader2::vtkGetMacro (
            ApplyLookupTable ,
            int  )
```

10.384.3.27 vtkGetMacro() [2/11]

```
vtkGDCMImageReader2::vtkGetMacro (
            ApplyYBRToRGB ,
            int  )
```

10.384.3.28 vtkGetMacro() [3/11]

```
vtkGDCMImageReader2::vtkGetMacro (
            ImageFormat ,
            int  )
```

10.384.3.29 vtkGetMacro() [4/11]

```
vtkGDCMImageReader2::vtkGetMacro (
            LoadIconImage ,
            int  )
```

10.384.3.30 vtkGetMacro() [5/11]

```
vtkGDCMImageReader2::vtkGetMacro (
            LoadOverlays ,
            int  )
```

10.384.3.31 vtkGetMacro() [6/11]

```
vtkGDCMImageReader2::vtkGetMacro (
            LossyFlag ,
            int  )
```

10.384.3.32 vtkGetMacro() [7/11]

```
vtkGDCMImageReader2::vtkGetMacro (
            NumberOfIconImages ,
            int  )
```

10.384.3.33 vtkGetMacro() [8/11]

```
vtkGDCMImageReader2::vtkGetMacro (
            NumberOfOverlays ,
            int  )
```

10.384.3.34 vtkGetMacro() [9/11]

```
vtkGDCMImageReader2::vtkGetMacro (
            PlanarConfiguration ,
            int  )
```

10.384.3.35 vtkGetMacro() [10/11]

```
vtkGDCMImageReader2::vtkGetMacro (
            Scale ,
            double  )
```

10.384.3.36 vtkGetMacro() [11/11]

```
vtkGDCMImageReader2::vtkGetMacro (
            Shift ,
            double  )
```

10.384.3.37 vtkGetObjectMacro() [1/2]

```
vtkGDCMImageReader2::vtkGetObjectMacro (
            Curve ,
            vtkPolyData  )
```

10.384.3.38 vtkGetObjectMacro() [2/2]

```
vtkGDCMImageReader2::vtkGetObjectMacro (
            DirectionCosines ,
            vtkMatrix4x4  )
```

10.384.3.39 vtkGetStringMacro() [1/2]

```
vtkGDCMImageReader2::vtkGetStringMacro (
            FilePattern  )  [protected]
```

10.384.3.40 vtkGetStringMacro() [2/2]

```
vtkGDCMImageReader2::vtkGetStringMacro (
            FilePrefix  )  [protected]
```

10.384.3.41 vtkGetVector3Macro()

```
vtkGDCMImageReader2::vtkGetVector3Macro (
            ImagePositionPatient ,
            double  )
```

10.384.3.42 vtkGetVector6Macro()

```
vtkGDCMImageReader2::vtkGetVector6Macro (
            ImageOrientationPatient ,
            double  )
```

10.384.3.43 vtkSetMacro() [1/4]

```
vtkGDCMImageReader2::vtkSetMacro (
            ApplyLookupTable ,
            int  )
```

10.384.3.44 vtkSetMacro() [2/4]

```
vtkGDCMImageReader2::vtkSetMacro (
            LoadIconImage ,
            int  )
```

10.384.3.45 vtkSetMacro() [3/4]

```
vtkGDCMImageReader2::vtkSetMacro (
            LoadOverlays ,
            int  )
```

10.384.3.46 vtkSetMacro() [4/4]

```
vtkGDCMImageReader2::vtkSetMacro (
            LossyFlag ,
            int  )
```

10.384.3.47 vtkSetVector6Macro()

```
vtkGDCMImageReader2::vtkSetVector6Macro (
            ImageOrientationPatient ,
            double  )  [protected]
```

10.384.3.48 vtkTypeMacro()

```
vtkGDCMImageReader2::vtkTypeMacro (
            vtkGDCMImageReader2 ,
            vtkMedicalImageReader2  )
```

10.384.4 Member Data Documentation

10.384.4.1 ApplyInverseVideo

```
int vtkGDCMImageReader2::ApplyInverseVideo  [protected]
```

10.384.4.2 ApplyLookupTable

```
int vtkGDCMImageReader2::ApplyLookupTable  [protected]
```

10.384.4.3 ApplyPlanarConfiguration

```
int vtkGDCMImageReader2::ApplyPlanarConfiguration  [protected]
```

10.384.4.4 ApplyShiftScale

```
int vtkGDCMImageReader2::ApplyShiftScale  [protected]
```

10.384.4.5 ApplyYBRToRGB

```
int vtkGDCMImageReader2::ApplyYBRToRGB  [protected]
```

10.384.4.6 Curve

```
vtkPolyData* vtkGDCMImageReader2::Curve  [protected]
```

10.384.4.7 DirectionCosines

```
vtkMatrix4x4* vtkGDCMImageReader2::DirectionCosines  [protected]
```

10.384.4.8 ForceRescale

```
int vtkGDCMImageReader2::ForceRescale  [protected]
```

10.384.4.9 IconDataScalarType

```
int vtkGDCMImageReader2::IconDataScalarType  [protected]
```

10.384.4.10 IconImageDataExtent

```
int vtkGDCMImageReader2::IconImageDataExtent[6]  [protected]
```

10.384.4.11 IconNumberOfScalarComponents

```
int vtkGDCMImageReader2::IconNumberOfScalarComponents  [protected]
```

10.384.4.12 ImageFormat

```
int vtkGDCMImageReader2::ImageFormat  [protected]
```

10.384.4.13 ImageOrientationPatient

```
double vtkGDCMImageReader2::ImageOrientationPatient[6]  [protected]
```

10.384.4.14 ImagePositionPatient

```
double vtkGDCMImageReader2::ImagePositionPatient[3]  [protected]
```

10.384.4.15 LoadIconImage

```
int vtkGDCMImageReader2::LoadIconImage  [protected]
```

10.384.4.16 LoadOverlays

```
int vtkGDCMImageReader2::LoadOverlays  [protected]
```

10.384.4.17 LossyFlag

```
int vtkGDCMImageReader2::LossyFlag  [protected]
```

10.384.4.18 NumberOfIconImages

```
int vtkGDCMImageReader2::NumberOfIconImages  [protected]
```

10.384.4.19 NumberOfOverlays

```
int vtkGDCMImageReader2::NumberOfOverlays  [protected]
```

10.384.4.20 PlanarConfiguration

```
int vtkGDCMImageReader2::PlanarConfiguration  [protected]
```

10.384.4.21 Scale

```
double vtkGDCMImageReader2::Scale  [protected]
```

10.384.4.22 Shift

```
double vtkGDCMImageReader2::Shift  [protected]
```

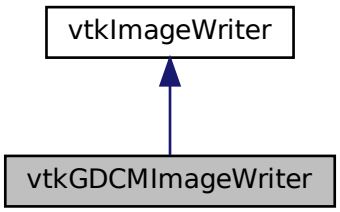
The documentation for this class was generated from the following file:

- vtkGDCMImageReader2.h

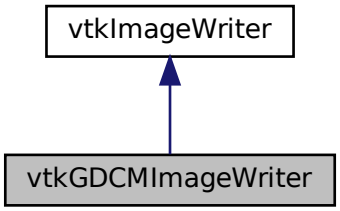
10.385 vtkGDCMImageWriter Class Reference

`#include <vtkGDCMImageWriter.h>`

Inheritance diagram for vtkGDCMImageWriter:



Collaboration diagram for vtkGDCMImageWriter:



Public Types

- enum CompressionTypes {
  NO_COMPRESSION = 0 ,
  JPEG_COMPRESSION ,
  JPEG2000_COMPRESSION ,
  JPEGLS_COMPRESSION ,
  RLE_COMPRESSION }

Public Member Functions

- virtual const char * GetDescriptiveName ()
- virtual const char * GetFileExtensions ()
- virtual void PrintSelf (ostream &os, vtkIndent indent)
- virtual void SetDirectionCosines (vtkMatrix4x4 *matrix)
- virtual void SetDirectionCosinesFromImageOrientationPatient (const double dircos[6])
- virtual void SetFileNames (vtkStringArray *)
- virtual void SetMedicalImageProperties (vtkMedicalImageProperties *)
- vtkBooleanMacro (FileLowerLeft, int)
- vtkBooleanMacro (LossyFlag, int)
- vtkGetMacro (CompressionType, int)
- vtkGetMacro (FileLowerLeft, int)
- vtkGetMacro (ImageFormat, int)
- vtkGetMacro (LossyFlag, int)
- vtkGetMacro (PlanarConfiguration, int)
- vtkGetMacro (Scale, double)
- vtkGetMacro (Shift, double)
- vtkGetObjectMacro (DirectionCosines, vtkMatrix4x4)
- vtkGetObjectMacro (FileNames, vtkStringArray)
- vtkGetObjectMacro (MedicalImageProperties, vtkMedicalImageProperties)
- vtkGetStringMacro (SeriesUID)
- vtkGetStringMacro (StudyUID)
- vtkSetMacro (CompressionType, int)
- vtkSetMacro (FileLowerLeft, int)
- vtkSetMacro (ImageFormat, int)
- vtkSetMacro (LossyFlag, int)
- vtkSetMacro (PlanarConfiguration, int)
- vtkSetMacro (Scale, double)
- vtkSetMacro (Shift, double)
- vtkSetStringMacro (SeriesUID)
- vtkSetStringMacro (StudyUID)
- vtkTypeMacro (vtkGDCMImageWriter, vtkImageWriter)
- virtual void Write ()

Static Public Member Functions

- static vtkGDCMImageWriter * New ()

Protected Member Functions

- vtkGDCMImageWriter ()
- ~vtkGDCMImageWriter ()
- virtual char * GetFileName ()
- int WriteGDCMData (vtkImageData *data, int timeStep)
- void WriteSlice (vtkImageData *data)

10.385.1 Detailed Description

Examples

Convert16BitsTo8Bits.cxx, ConvertMultiFrameToSingleFrame.cxx, ConvertRGBToLuminance.cxx, ConvertSingleBitTo8Bits.cxx, CreateFakePET.cxx, CreateFakeRTDOSE.cxx, HelloActiviz.cs, HelloActiviz2.cs, HelloVTKWorld.cs, HelloVTKWorld.java, HelloVTKWorld2.cs, MagnifyFile.cxx, RefCounting.cs, and gdcmorthoplanes.cxx.

10.385.2 Member Enumeration Documentation

10.385.2.1 CompressionTypes

```
enum vtkGDCMImageWriter::CompressionTypes
```

Enumerator

NO_COMPRESSION	
JPEG_COMPRESSION	
JPEG2000_COMPRESSION	
JPEGLS_COMPRESSION	
RLE_COMPRESSION	

10.385.3 Constructor & Destructor Documentation

10.385.3.1 vtkGDCMImageWriter()

```
vtkGDCMImageWriter::vtkGDCMImageWriter ( )  [protected]
```

10.385.3.2 ~vtkGDCMImageWriter()

```
vtkGDCMImageWriter::~vtkGDCMImageWriter ( )  [protected]
```

10.385.4 Member Function Documentation

10.385.4.1 GetDescriptiveName()

```
virtual const char * vtkGDCMImageWriter::GetDescriptiveName ( )  [inline], [virtual]
```

10.385.4.2 GetFileExtensions()

```
virtual const char * vtkGDCMImageWriter::GetFileExtensions ( )  [inline], [virtual]
```

10.385.4.3 GetFileName()

```
virtual char * vtkGDCMImageWriter::GetFileName ( )  [protected], [virtual]
```

10.385.4.4 New()

```
static vtkGDCMImageWriter * vtkGDCMImageWriter::New ( )  [static]
```

Examples

Convert16BitsTo8Bits.cxx, ConvertMultiFrameToSingleFrame.cxx, ConvertRGBToLuminance.cxx, ConvertSingleBitTo8Bits.cxx, CreateFakePET.cxx, CreateFakeRTDOSE.cxx, HelloActiviz.cs, HelloVTKWorld.cs, HelloVTKWorld2.cs, MagnifyFile.cxx, RefCounting.cs, and gdcmorthoplanes.cxx.

10.385.4.5 PrintSelf()

```
virtual void vtkGDCMImageWriter::PrintSelf (
            ostream & os,
            vtkIndent indent )  [virtual]
```

10.385.4.6 SetDirectionCosines()

```
virtual void vtkGDCMImageWriter::SetDirectionCosines (
            vtkMatrix4x4 * matrix )  [virtual]
```

Examples

Convert16BitsTo8Bits.cxx, ConvertRGBToLuminance.cxx, ConvertSingleBitTo8Bits.cxx, HelloActiviz2.cs, HelloVTKWorld.cs, HelloVTKWorld.java, MagnifyFile.cxx, and gdcmorthoplanes.cxx.

10.385.4.7 SetDirectionCosinesFromImageOrientationPatient()

```
virtual void vtkGDCMImageWriter::SetDirectionCosinesFromImageOrientationPatient (
            const double dircos[6] )  [virtual]
```

10.385.4.8 SetFileNames()

```
virtual void vtkGDCMImageWriter::SetFileNames (
            vtkStringArray *  )  [virtual]
```

Examples

ConvertMultiFrameToSingleFrame.cxx, and CreateFakePET.cxx.

10.385.4.9 SetMedicalImageProperties()

```
virtual void vtkGDCMImageWriter::SetMedicalImageProperties (
            vtkMedicalImageProperties *  )  [virtual]
```

Examples

Convert16BitsTo8Bits.cxx, ConvertRGBToLuminance.cxx, ConvertSingleBitTo8Bits.cxx, HelloActiviz.cs, HelloActiviz2.cs, HelloVTKWorld.cs, HelloVTKWorld.java, MagnifyFile.cxx, and gdcmorthoplanes.cxx.

10.385.4.10 vtkBooleanMacro() [1/2]

```
vtkGDCMImageWriter::vtkBooleanMacro (
            FileLowerLeft ,
            int  )
```

10.385.4.11 vtkBooleanMacro() [2/2]

```
vtkGDCMImageWriter::vtkBooleanMacro (
            LossyFlag ,
            int  )
```

10.385.4.12 vtkGetMacro() [1/7]

```
vtkGDCMImageWriter::vtkGetMacro (
            CompressionType ,
            int  )
```

10.385.4.13 vtkGetMacro() [2/7]

```
vtkGDCMImageWriter::vtkGetMacro (
            FileLowerLeft ,
            int  )
```

10.385.4.14 vtkGetMacro() [3/7]

```
vtkGDCMImageWriter::vtkGetMacro (
            ImageFormat ,
            int  )
```

10.385.4.15 vtkGetMacro() [4/7]

```
vtkGDCMImageWriter::vtkGetMacro (
            LossyFlag ,
            int  )
```

10.385.4.16 vtkGetMacro() [5/7]

```
vtkGDCMImageWriter::vtkGetMacro (
            PlanarConfiguration ,
            int  )
```

10.385.4.17 vtkGetMacro() [6/7]

```
vtkGDCMImageWriter::vtkGetMacro (
            Scale ,
            double  )
```

10.385.4.18 vtkGetMacro() [7/7]

```
vtkGDCMImageWriter::vtkGetMacro (
            Shift ,
            double  )
```

10.385.4.19 vtkGetObjectMacro() [1/3]

```
vtkGDCMImageWriter::vtkGetObjectMacro (
            DirectionCosines ,
            vtkMatrix4x4  )
```

10.385.4.20 vtkGetObjectMacro() [2/3]

```
vtkGDCMImageWriter::vtkGetObjectMacro (
            FileNames ,
            vtkStringArray  )
```

10.385.4.21 vtkGetObjectMacro() [3/3]

```
vtkGDCMImageWriter::vtkGetObjectMacro (
            MedicalImageProperties ,
            vtkMedicalImageProperties  )
```

10.385.4.22 vtkGetStringMacro() [1/2]

```
vtkGDCMImageWriter::vtkGetStringMacro (
            SeriesUID  )
```

10.385.4.23 vtkGetStringMacro() [2/2]

```
vtkGDCMImageWriter::vtkGetStringMacro (
            StudyUID  )
```

10.385.4.24 vtkSetMacro() [1/7]

```
vtkGDCMImageWriter::vtkSetMacro (
            CompressionType ,
            int  )
```

10.385.4.25 vtkSetMacro() [2/7]

```
vtkGDCMImageWriter::vtkSetMacro (
            FileLowerLeft ,
            int  )
```

10.385.4.26 vtkSetMacro() [3/7]

```
vtkGDCMImageWriter::vtkSetMacro (
            ImageFormat ,
            int  )
```

10.385.4.27 vtkSetMacro() [4/7]

```
vtkGDCMImageWriter::vtkSetMacro (
            LossyFlag ,
            int  )
```

10.385.4.28 vtkSetMacro() [5/7]

```
vtkGDCMImageWriter::vtkSetMacro (
            PlanarConfiguration ,
            int  )
```

10.385.4.29 vtkSetMacro() [6/7]

```
vtkGDCMImageWriter::vtkSetMacro (
            Scale ,
            double  )
```

10.385.4.30 vtkSetMacro() [7/7]

```
vtkGDCMImageWriter::vtkSetMacro (
            Shift ,
            double  )
```

10.385.4.31 vtkSetStringMacro() [1/2]

```
vtkGDCMImageWriter::vtkSetStringMacro (
            SeriesUID  )
```

10.385.4.32 vtkSetStringMacro() [2/2]

```
vtkGDCMImageWriter::vtkSetStringMacro (
            StudyUID  )
```

10.385.4.33 vtkTypeMacro()

```
vtkGDCMImageWriter::vtkTypeMacro (
            vtkGDCMImageWriter ,
            vtkImageWriter  )
```

10.385.4.34 Write()

```
virtual void vtkGDCMImageWriter::Write ( )  [virtual]
```

Examples

Convert16BitsTo8Bits.cxx, ConvertMultiFrameToSingleFrame.cxx, ConvertRGBToLuminance.cxx, ConvertSingleBitTo8Bits.cxx, CreateFakePET.cxx, CreateFakeRTDOSE.cxx, HelloActiviz.cs, HelloActiviz2.cs, HelloVTKWorld.cs, HelloVTKWorld.java, HelloVTKWorld2.cs, MagnifyFile.cxx, and gdcmorthoplanes.cxx.

10.385.4.35 WriteGDCMData()

```
int vtkGDCMImageWriter::WriteGDCMData (
            vtkImageData * data,
            int timeStep )  [protected]
```

10.385.4.36 WriteSlice()

```
void vtkGDCMImageWriter::WriteSlice (
            vtkImageData * data )  [protected]
```

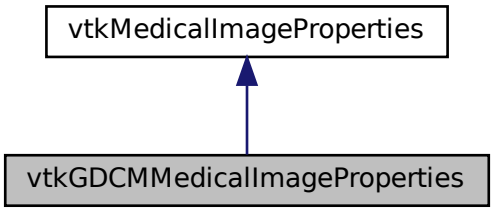
The documentation for this class was generated from the following file:

- vtkGDCMImageWriter.h

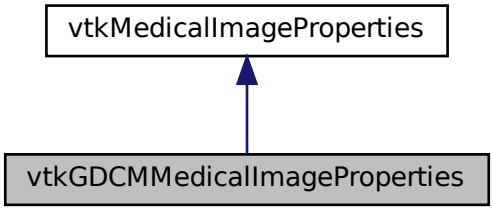
10.386 vtkGDCMMedicalImageProperties Class Reference

`#include <vtkGDCMMedicalImageProperties.h>`

Inheritance diagram for vtkGDCMMedicalImageProperties:



Collaboration diagram for vtkGDCMMedicalImageProperties:



Public Member Functions

- virtual void Clear ()
- void PrintSelf (ostream &os, vtkIndent indent)
- vtkTypeMacro (vtkGDCMMedicalImageProperties, vtkMedicalImageProperties)

Static Public Member Functions

- static vtkGDCMMedicalImageProperties * New ()

Protected Member Functions

- vtkGDCMMedicalImageProperties ()
- ~vtkGDCMMedicalImageProperties ()
- gdcm::File const & GetFile (unsigned int t)
- void PushBackFile (gdcm::File const &f)

Friends

- class vtkGDCMImageReader
- class vtkGDCMImageReader2
- class vtkGDCMImageWriter

10.386.1 Constructor & Destructor Documentation

10.386.1.1 vtkGDCMMedicalImageProperties()

```
vtkGDCMMedicalImageProperties::vtkGDCMMedicalImageProperties ( )  [protected]
```

10.386.1.2 ~vtkGDCMMedicalImageProperties()

```
vtkGDCMMedicalImageProperties::~vtkGDCMMedicalImageProperties ( )  [protected]
```

10.386.2 Member Function Documentation

10.386.2.1 Clear()

```
virtual void vtkGDCMMedicalImageProperties::Clear ( )  [virtual]
```

10.386.2.2 GetFile()

```
gdcm::File const & vtkGDCMMedicalImageProperties::GetFile (
            unsigned int t )  [protected]
```

10.386.2.3 New()

```
static vtkGDCMMedicalImageProperties * vtkGDCMMedicalImageProperties::New ( )  [static]
```

10.386.2.4 PrintSelf()

```
void vtkGDCMMedicalImageProperties::PrintSelf (
            ostream & os,
            vtkIndent indent )
```

10.386.2.5 PushBackFile()

```
void vtkGDCMMedicalImageProperties::PushBackFile (
            gdcm::File const & f )  [protected]
```

10.386.2.6 vtkTypeMacro()

```
vtkGDCMMedicalImageProperties::vtkTypeMacro (
            vtkGDCMMedicalImageProperties ,
            vtkMedicalImageProperties  )
```

10.386.3 Friends And Related Function Documentation

10.386.3.1 vtkGDCMImageReader

```
friend class vtkGDCMImageReader  [friend]
```

10.386.3.2 vtkGDCMImageReader2

```
friend class vtkGDCMImageReader2  [friend]
```

10.386.3.3 vtkGDCMImageWriter

```
friend class vtkGDCMImageWriter  [friend]
```

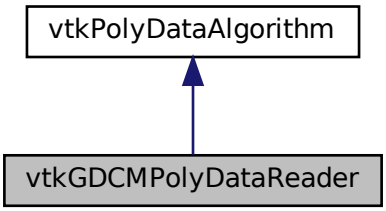
The documentation for this class was generated from the following file:

- vtkGDCMMedicalImageProperties.h

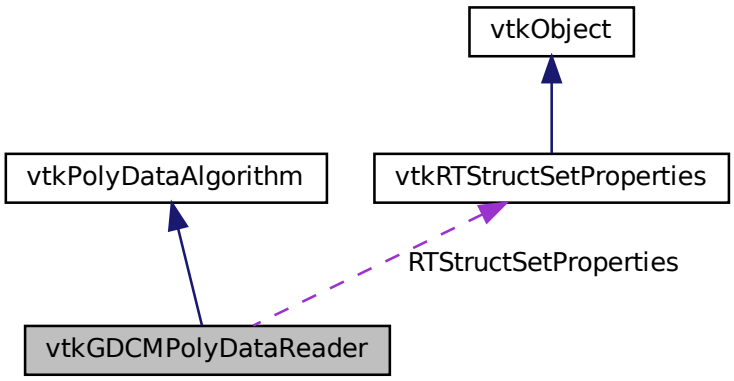
10.387 vtkGDCMPolyDataReader Class Reference

```
#include <vtkGDCMPolyDataReader.h>
```

Inheritance diagram for vtkGDCMPolyDataReader:



Collaboration diagram for vtkGDCMPolyDataReader:



Public Member Functions

- virtual void PrintSelf (ostream &os, vtkIndent indent)
- vtkGetObjectMacro (MedicalImageProperties, vtkMedicalImageProperties)
- vtkGetObjectMacro (RTStructSetProperties, vtkRTStructSetProperties)
- vtkGetStringMacro (FileName)
- vtkSetStringMacro (FileName)
- vtkTypeMacro (vtkGDCMPolyDataReader, vtkPolyDataAlgorithm)

Static Public Member Functions

- static vtkGDCMPolyDataReader * New ()

Protected Member Functions

- vtkGDCMPolyDataReader ()
- ~vtkGDCMPolyDataReader ()
- void FillMedicalImageInformation (const gdcm::Reader &reader)
- int RequestData (vtkInformation *, vtkInformationVector **, vtkInformationVector *)
- int RequestData_HemodynamicWaveformStorage (gdcm::Reader const &reader, vtkInformationVector *output↩ Vector)
- int RequestData_RTStructureSetStorage (gdcm::Reader const &reader, vtkInformationVector *outputVector)
- int RequestInformation (vtkInformation *vtkNotUsed(request), vtkInformationVector **vtkNotUsed(inputVector), vtkInformationVector *outputVector)
- int RequestInformation_HemodynamicWaveformStorage (gdcm::Reader const &reader)
- int RequestInformation_RTStructureSetStorage (gdcm::Reader const &reader)

Protected Attributes

- char * FileName
- vtkMedicalImageProperties * MedicalImageProperties
- vtkRTStructSetProperties * RTStructSetProperties

10.387.1 Detailed Description

Examples

GenerateRTSTRUCT.cxx, gdcmscene.cxx, and rtstructapp.cxx.

10.387.2 Constructor & Destructor Documentation

10.387.2.1 vtkGDCMPolyDataReader()

```
vtkGDCMPolyDataReader::vtkGDCMPolyDataReader ( )  [protected]
```

10.387.2.2 ~vtkGDCMPolyDataReader()

```
vtkGDCMPolyDataReader::~vtkGDCMPolyDataReader ( )  [protected]
```

10.387.3 Member Function Documentation

10.387.3.1 FillMedicalImageInformation()

```
void vtkGDCMPolyDataReader::FillMedicalImageInformation (
            const gdcm::Reader & reader )  [protected]
```

10.387.3.2 New()

```
static vtkGDCMPolyDataReader * vtkGDCMPolyDataReader::New ( )  [static]
```

Examples

GenerateRTSTRUCT.cxx, gdcmscene.cxx, and rtstructapp.cxx.

10.387.3.3 PrintSelf()

```
virtual void vtkGDCMPolyDataReader::PrintSelf (
            ostream & os,
            vtkIndent indent )  [virtual]
```

10.387.3.4 RequestData()

```
int vtkGDCMPolyDataReader::RequestData (
            vtkInformation * ,
            vtkInformationVector ** ,
            vtkInformationVector *  )  [protected]
```

10.387.3.5 RequestData_HemodynamicWaveformStorage()

```
int vtkGDCMPolyDataReader::RequestData_HemodynamicWaveformStorage (
            gdcm::Reader const & reader,
            vtkInformationVector * outputVector )  [protected]
```

10.387.3.6 RequestData_RTStructureSetStorage()

```
int vtkGDCMPolyDataReader::RequestData_RTStructureSetStorage (
            gdcm::Reader const & reader,
            vtkInformationVector * outputVector )  [protected]
```

10.387.3.7 RequestInformation()

```
int vtkGDCMPolyDataReader::RequestInformation (
            vtkInformation * vtkNotUsedrequest,
            vtkInformationVector ** vtkNotUsedinputVector,
            vtkInformationVector * outputVector )  [protected]
```

10.387.3.8 RequestInformation_HemodynamicWaveformStorage()

```
int vtkGDCMPolyDataReader::RequestInformation_HemodynamicWaveformStorage (
            gdcm::Reader const & reader )  [protected]
```

10.387.3.9 RequestInformation_RTStructureSetStorage()

```
int vtkGDCMPolyDataReader::RequestInformation_RTStructureSetStorage (
            gdcm::Reader const & reader )  [protected]
```

10.387.3.10 vtkGetObjectMacro() [1/2]

```
vtkGDCMPolyDataReader::vtkGetObjectMacro (
            MedicalImageProperties ,
            vtkMedicalImageProperties  )
```

10.387.3.11 vtkGetObjectMacro() [2/2]

```
vtkGDCMPolyDataReader::vtkGetObjectMacro (
            RTStructSetProperties ,
            vtkRTStructSetProperties  )
```

10.387.3.12 vtkGetStringMacro()

```
vtkGDCMPolyDataReader::vtkGetStringMacro (
            FileName  )
```

10.387.3.13 vtkSetStringMacro()

```
vtkGDCMPolyDataReader::vtkSetStringMacro (
            FileName  )
```

10.387.3.14 vtkTypeMacro()

```
vtkGDCMPolyDataReader::vtkTypeMacro (
            vtkGDCMPolyDataReader ,
            vtkPolyDataAlgorithm  )
```

10.387.4 Member Data Documentation

10.387.4.1 FileName

```
char* vtkGDCMPolyDataReader::FileName  [protected]
```

10.387.4.2 MedicalImageProperties

```
vtkMedicalImageProperties* vtkGDCMPolyDataReader::MedicalImageProperties  [protected]
```

10.387.4.3 RTStructSetProperties

```
vtkRTStructSetProperties* vtkGDCMPolyDataReader::RTStructSetProperties  [protected]
```

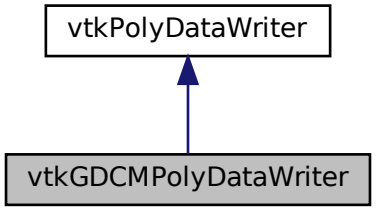
The documentation for this class was generated from the following file:

- vtkGDCMPolyDataReader.h

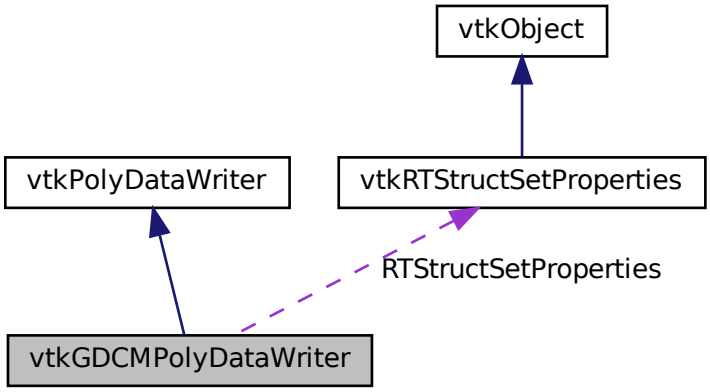
10.388 vtkGDCMPolyDataWriter Class Reference

```
#include <vtkGDCMPolyDataWriter.h>
```

Inheritance diagram for vtkGDCMPolyDataWriter:



Collaboration diagram for vtkGDCMPolyDataWriter:



Public Member Functions

- void InitializeRTStructSet (vtkStdString inDirectory, vtkStdString inStructLabel, vtkStdString inStructName, vtk↩ StringArray ∗inROINames, vtkStringArray ∗inROIAlgorithmName, vtkStringArray ∗inROIType)
- virtual void PrintSelf (ostream &os, vtkIndent indent)
- virtual void SetMedicalImageProperties (vtkMedicalImageProperties ∗pd)
- void SetNumberOfInputPorts (int n)
- virtual void SetRTStructSetProperties (vtkRTStructSetProperties ∗pd)
- vtkTypeMacro (vtkGDCMPolyDataWriter, vtkPolyDataWriter)

Static Public Member Functions

- static vtkGDCMPolyDataWriter ∗ New ()

Protected Member Functions

- vtkGDCMPolyDataWriter ()
- ∼vtkGDCMPolyDataWriter ()
- void WriteData ()
- void WriteRTSTRUCTData (gdcm::File &file, int num)
- void WriteRTSTRUCTInfo (gdcm::File &file)

Protected Attributes

- vtkMedicalImageProperties ∗ MedicalImageProperties
- vtkRTStructSetProperties ∗ RTStructSetProperties

10.388.1 Detailed Description

Examples

GenerateRTSTRUCT.cxx, and rtstructapp.cxx.

10.388.2 Constructor & Destructor Documentation

10.388.2.1 vtkGDCMPolyDataWriter()

```
vtkGDCMPolyDataWriter::vtkGDCMPolyDataWriter ( )  [protected]
```

10.388.2.2 ~vtkGDCMPolyDataWriter()

```
vtkGDCMPolyDataWriter::~vtkGDCMPolyDataWriter ( )  [protected]
```

10.388.3 Member Function Documentation

10.388.3.1 InitializeRTStructSet()

```
void vtkGDCMPolyDataWriter::InitializeRTStructSet (
            vtkStdString inDirectory,
            vtkStdString inStructLabel,
            vtkStdString inStructName,
            vtkStringArray * inROINames,
            vtkStringArray * inROIAlgorithmName,
            vtkStringArray * inROIType )
```

Examples

GenerateRTSTRUCT.cxx.

10.388.3.2 New()

```
static vtkGDCMPolyDataWriter * vtkGDCMPolyDataWriter::New ( )  [static]
```

Examples

GenerateRTSTRUCT.cxx, and rtstructapp.cxx.

10.388.3.3 PrintSelf()

```
virtual void vtkGDCMPolyDataWriter::PrintSelf (
            ostream & os,
            vtkIndent indent )  [virtual]
```

10.388.3.4 SetMedicalImageProperties()

```
virtual void vtkGDCMPolyDataWriter::SetMedicalImageProperties (
            vtkMedicalImageProperties * pd )  [virtual]
```

Examples

GenerateRTSTRUCT.cxx, and rtstructapp.cxx.

10.388.3.5 SetNumberOfInputPorts()

```
void vtkGDCMPolyDataWriter::SetNumberOfInputPorts (
            int n )
```

Examples

GenerateRTSTRUCT.cxx, and rtstructapp.cxx.

10.388.3.6 SetRTStructSetProperties()

```
virtual void vtkGDCMPolyDataWriter::SetRTStructSetProperties (
            vtkRTStructSetProperties * pd )  [virtual]
```

Examples

GenerateRTSTRUCT.cxx, and rtstructapp.cxx.

10.388.3.7 vtkTypeMacro()

```
vtkGDCMPolyDataWriter::vtkTypeMacro (
            vtkGDCMPolyDataWriter ,
            vtkPolyDataWriter  )
```

10.388.3.8 WriteData()

```
void vtkGDCMPolyDataWriter::WriteData ( )  [protected]
```

10.388.3.9 WriteRTSTRUCTData()

```
void vtkGDCMPolyDataWriter::WriteRTSTRUCTData (
            gdcm::File & file,
            int num )  [protected]
```

10.388.3.10 WriteRTSTRUCTInfo()

```
void vtkGDCMPolyDataWriter::WriteRTSTRUCTInfo (
            gdcm::File & file )  [protected]
```

10.388.4 Member Data Documentation

10.388.4.1 MedicalImageProperties

```
vtkMedicalImageProperties* vtkGDCMPolyDataWriter::MedicalImageProperties  [protected]
```

10.388.4.2 RTStructSetProperties

```
vtkRTStructSetProperties* vtkGDCMPolyDataWriter::RTStructSetProperties  [protected]
```

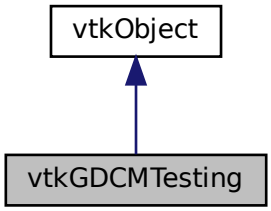
The documentation for this class was generated from the following file:

- vtkGDCMPolyDataWriter.h

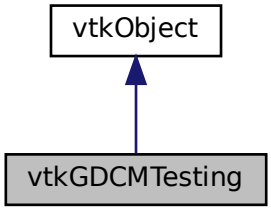
10.389 vtkGDCMTesting Class Reference

```
#include <vtkGDCMTesting.h>
```

Inheritance diagram for vtkGDCMTesting:



Collaboration diagram for vtkGDCMTesting:



Public Types

- typedef const char ∗const (∗ MD5MetaImagesType)[3]

Public Member Functions

- void PrintSelf (ostream &os, vtkIndent indent)
- vtkTypeMacro (vtkGDCMTesting, vtkObject)

Static Public Member Functions

- static const char ∗ GetGDCMDataRoot ()
- static const char ∗const ∗ GetMD5MetaImage (unsigned int file)
- static const char ∗ GetMHDMD5FromFile (const char ∗filepath)
- static unsigned int GetNumberOfMD5MetaImages ()
- static const char ∗ GetRAWMD5FromFile (const char ∗filepath)
- static const char ∗ GetVTKDataRoot ()
- static vtkGDCMTesting ∗ New ()

Protected Member Functions

- vtkGDCMTesting ()
- ∼vtkGDCMTesting ()

10.389.1 Detailed Description

Examples

HelloActiviz5.cs, HelloVTKWorld2.cs, MetaImageMD5Activiz.cs, ReadSeriesIntoVTK.java, and RefCounting.cs.

10.389.2 Member Typedef Documentation

10.389.2.1 MD5MetaImagesType

```
typedef const char* const(* vtkGDCMTesting::MD5MetaImagesType)[3]
```

10.389.3 Constructor & Destructor Documentation

10.389.3.1 vtkGDCMTesting()

```
vtkGDCMTesting::vtkGDCMTesting ( )  [protected]
```

10.389.3.2 ~vtkGDCMTesting()

```
vtkGDCMTesting::~vtkGDCMTesting ( )  [protected]
```

10.389.4 Member Function Documentation

10.389.4.1 GetGDCMDataRoot()

```
static const char * vtkGDCMTesting::GetGDCMDataRoot ( )  [static]
```

Examples

HelloActiviz5.cs, and ReadSeriesIntoVTK.java.

10.389.4.2 GetMD5MetaImage()

```
static const char *const * vtkGDCMTesting::GetMD5MetaImage (
            unsigned int file )  [static]
```

10.389.4.3 GetMHDMD5FromFile()

```
static const char * vtkGDCMTesting::GetMHDMD5FromFile (
            const char * filepath )  [static]
```

Examples

MetaImageMD5Activiz.cs.

10.389.4.4 GetNumberOfMD5MetaImages()

```
static unsigned int vtkGDCMTesting::GetNumberOfMD5MetaImages ( )  [static]
```

10.389.4.5 GetRAWMD5FromFile()

```
static const char * vtkGDCMTesting::GetRAWMD5FromFile (
            const char * filepath )  [static]
```

Examples

MetaImageMD5Activiz.cs.

10.389.4.6 GetVTKDataRoot()

```
static const char * vtkGDCMTesting::GetVTKDataRoot ( )  [static]
```

Examples

HelloActiviz5.cs, and HelloVTKWorld2.cs.

10.389.4.7 New()

```
static vtkGDCMTesting * vtkGDCMTesting::New ( )  [static]
```

Examples

RefCounting.cs.

10.389.4.8 PrintSelf()

```
void vtkGDCMTesting::PrintSelf (
            ostream & os,
            vtkIndent indent )
```

10.389.4.9 vtkTypeMacro()

```
vtkGDCMTesting::vtkTypeMacro (
             vtkGDCMTesting ,
             vtkObject  )
```

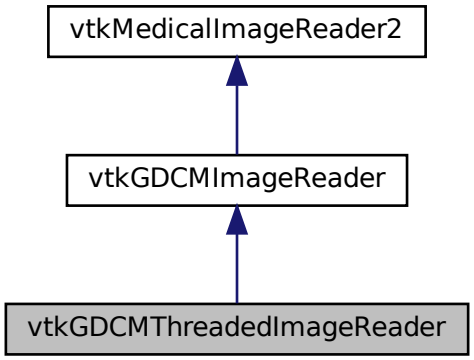
The documentation for this class was generated from the following file:

- vtkGDCMTesting.h

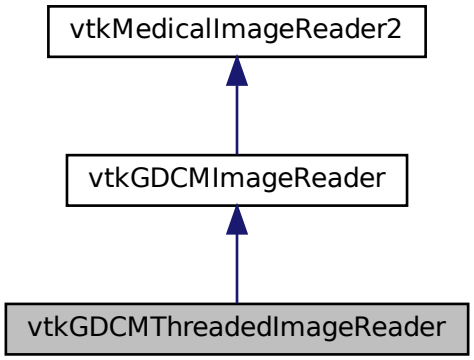
10.390 vtkGDCMThreadedImageReader Class Reference

```
#include <vtkGDCMThreadedImageReader.h>
```

Inheritance diagram for vtkGDCMThreadedImageReader:



Collaboration diagram for vtkGDCMThreadedImageReader:



Public Member Functions

- virtual void PrintSelf (ostream &os, vtkIndent indent)
- vtkBooleanMacro (UseShiftScale, int)
- vtkGetMacro (UseShiftScale, int)
- vtkSetMacro (Scale, double)
- vtkSetMacro (Shift, double)
- vtkSetMacro (UseShiftScale, int)
- vtkTypeMacro (vtkGDCMThreadedImageReader, vtkGDCMImageReader)

Static Public Member Functions

- static vtkGDCMThreadedImageReader * New ()

Protected Member Functions

- vtkGDCMThreadedImageReader ()
- ~vtkGDCMThreadedImageReader ()
- void ExecuteData (vtkDataObject *out)
- void ExecuteInformation ()
- void ReadFiles (unsigned int nfiles, const char *filenames[])
- void RequestDataCompat ()

Additional Inherited Members

10.390.1 Constructor & Destructor Documentation

10.390.1.1 vtkGDCMThreadedImageReader()

```
vtkGDCMThreadedImageReader::vtkGDCMThreadedImageReader ( )  [protected]
```

10.390.1.2 ~vtkGDCMThreadedImageReader()

```
vtkGDCMThreadedImageReader::~vtkGDCMThreadedImageReader ( )  [protected]
```

10.390.2 Member Function Documentation

10.390.2.1 ExecuteData()

```
void vtkGDCMThreadedImageReader::ExecuteData (
            vtkDataObject * out )  [protected]
```

10.390.2.2 ExecuteInformation()

```
void vtkGDCMThreadedImageReader::ExecuteInformation ( )  [protected]
```

10.390.2.3 New()

```
static vtkGDCMThreadedImageReader * vtkGDCMThreadedImageReader::New ( )  [static]
```

10.390.2.4 PrintSelf()

```
virtual void vtkGDCMThreadedImageReader::PrintSelf (
            ostream & os,
            vtkIndent indent )  [virtual]
```

Reimplemented from vtkGDCMImageReader.

10.390.2.5 ReadFiles()

```
void vtkGDCMThreadedImageReader::ReadFiles (
            unsigned int nfiles,
            const char * filenames[] )  [protected]
```

10.390.2.6 RequestDataCompat()

```
void vtkGDCMThreadedImageReader::RequestDataCompat ( )  [protected]
```

10.390.2.7 vtkBooleanMacro()

```
vtkGDCMThreadedImageReader::vtkBooleanMacro (
            UseShiftScale ,
            int  )
```

10.390.2.8 vtkGetMacro()

```
vtkGDCMThreadedImageReader::vtkGetMacro (
            UseShiftScale ,
            int  )
```

10.390.2.9 vtkSetMacro() [1/3]

```
vtkGDCMThreadedImageReader::vtkSetMacro (
            Scale ,
            double  )
```

10.390.2.10 vtkSetMacro() [2/3]

```
vtkGDCMThreadedImageReader::vtkSetMacro (
            Shift ,
            double  )
```

10.390.2.11 vtkSetMacro() [3/3]

```
vtkGDCMThreadedImageReader::vtkSetMacro (
            UseShiftScale ,
            int  )
```

10.390.2.12 vtkTypeMacro()

```
vtkGDCMThreadedImageReader::vtkTypeMacro (
            vtkGDCMThreadedImageReader ,
            vtkGDCMImageReader  )
```

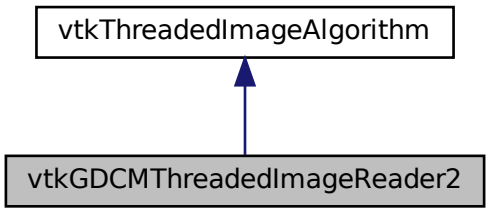
The documentation for this class was generated from the following file:

- vtkGDCMThreadedImageReader.h

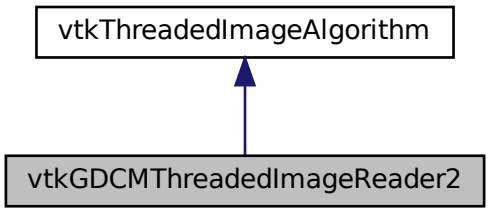
10.391 vtkGDCMThreadedImageReader2 Class Reference

```
#include <vtkGDCMThreadedImageReader2.h>
```

Inheritance diagram for vtkGDCMThreadedImageReader2:



Collaboration diagram for vtkGDCMThreadedImageReader2:



Public Member Functions

- virtual const char * GetFileName (int i=0)
- virtual void PrintSelf (ostream &os, vtkIndent indent)
- virtual void SetFileName (const char *filename)
- virtual void SetFileNames (vtkStringArray *)
- int SplitExtent (int splitExt[6], int startExt[6], int num, int total)
- vtkBooleanMacro (FileLowerLeft, int)
- vtkBooleanMacro (LoadOverlays, int)
- vtkBooleanMacro (UseShiftScale, int)
- vtkGetMacro (DataScalarType, int)
- vtkGetMacro (FileLowerLeft, int)
- vtkGetMacro (LoadOverlays, int)
- vtkGetMacro (NumberOfOverlays, int)
- vtkGetMacro (NumberOfScalarComponents, int)
- vtkGetMacro (Scale, double)
- vtkGetMacro (Shift, double)
- vtkGetMacro (UseShiftScale, int)
- vtkGetObjectMacro (FileNames, vtkStringArray)
- vtkGetVector3Macro (DataOrigin, double)
- vtkGetVector3Macro (DataSpacing, double)
- vtkGetVector6Macro (DataExtent, int)
- vtkSetMacro (DataScalarType, int)
- vtkSetMacro (FileLowerLeft, int)
- vtkSetMacro (LoadOverlays, int)
- vtkSetMacro (NumberOfScalarComponents, int)
- vtkSetMacro (Scale, double)
- vtkSetMacro (Shift, double)
- vtkSetMacro (UseShiftScale, int)
- vtkSetVector3Macro (DataOrigin, double)
- vtkSetVector3Macro (DataSpacing, double)
- vtkSetVector6Macro (DataExtent, int)
- vtkTypeMacro (vtkGDCMThreadedImageReader2, vtkThreadedImageAlgorithm)

Static Public Member Functions

- static vtkGDCMThreadedImageReader2 * New ()

Protected Member Functions

- vtkGDCMThreadedImageReader2 ()
- ~vtkGDCMThreadedImageReader2 ()
- int RequestInformation (vtkInformation *request, vtkInformationVector **inputVector, vtkInformationVector *outputVector)
- void ThreadedRequestData (vtkInformation *request, vtkInformationVector **inputVector, vtkInformationVector *outputVector, vtkImageData ***inData, vtkImageData **outData, int outExt[6], int id)

10.391.1 Constructor & Destructor Documentation

10.391.1.1 vtkGDCMThreadedImageReader2()

```
vtkGDCMThreadedImageReader2::vtkGDCMThreadedImageReader2 ( )  [protected]
```

10.391.1.2 ~vtkGDCMThreadedImageReader2()

```
vtkGDCMThreadedImageReader2::~vtkGDCMThreadedImageReader2 ( )  [protected]
```

10.391.2 Member Function Documentation

10.391.2.1 GetFileName()

```
virtual const char * vtkGDCMThreadedImageReader2::GetFileName (
            int i = 0 )  [virtual]
```

10.391.2.2 New()

```
static vtkGDCMThreadedImageReader2 * vtkGDCMThreadedImageReader2::New ( )  [static]
```

10.391.2.3 PrintSelf()

```
virtual void vtkGDCMThreadedImageReader2::PrintSelf (
            ostream & os,
            vtkIndent indent )  [virtual]
```

10.391.2.4 RequestInformation()

```
int vtkGDCMThreadedImageReader2::RequestInformation (
            vtkInformation * request,
            vtkInformationVector ** inputVector,
            vtkInformationVector * outputVector )  [protected]
```

10.391.2.5 SetFileName()

```
virtual void vtkGDCMThreadedImageReader2::SetFileName (
            const char * filename )  [virtual]
```

10.391.2.6 SetFileNames()

```
virtual void vtkGDCMThreadedImageReader2::SetFileNames (
            vtkStringArray *  )  [virtual]
```

10.391.2.7 SplitExtent()

```
int vtkGDCMThreadedImageReader2::SplitExtent (
            int splitExt[6],
            int startExt[6],
            int num,
            int total )
```

10.391.2.8 ThreadedRequestData()

```
void vtkGDCMThreadedImageReader2::ThreadedRequestData (
            vtkInformation * request,
            vtkInformationVector ** inputVector,
            vtkInformationVector * outputVector,
            vtkImageData *** inData,
            vtkImageData ** outData,
            int outExt[6],
            int id )  [protected]
```

10.391.2.9 vtkBooleanMacro() [1/3]

```
vtkGDCMThreadedImageReader2::vtkBooleanMacro (
            FileLowerLeft ,
            int  )
```

10.391.2.10 vtkBooleanMacro() [2/3]

```
vtkGDCMThreadedImageReader2::vtkBooleanMacro (
            LoadOverlays ,
            int  )
```

10.391.2.11 vtkBooleanMacro() [3/3]

```
vtkGDCMThreadedImageReader2::vtkBooleanMacro (
            UseShiftScale ,
            int  )
```

10.391.2.12 vtkGetMacro() [1/8]

```
vtkGDCMThreadedImageReader2::vtkGetMacro (
            DataScalarType ,
            int  )
```

10.391.2.13 vtkGetMacro() [2/8]

```
vtkGDCMThreadedImageReader2::vtkGetMacro (
            FileLowerLeft ,
            int  )
```

10.391.2.14 vtkGetMacro() [3/8]

```
vtkGDCMThreadedImageReader2::vtkGetMacro (
            LoadOverlays ,
            int  )
```

10.391.2.15 vtkGetMacro() [4/8]

```
vtkGDCMThreadedImageReader2::vtkGetMacro (
            NumberOfOverlays ,
            int  )
```

10.391.2.16 vtkGetMacro() [5/8]

```
vtkGDCMThreadedImageReader2::vtkGetMacro (
            NumberOfScalarComponents ,
            int  )
```

10.391.2.17 vtkGetMacro() [6/8]

```
vtkGDCMThreadedImageReader2::vtkGetMacro (
            Scale ,
            double  )
```

10.391.2.18 vtkGetMacro() [7/8]

```
vtkGDCMThreadedImageReader2::vtkGetMacro (
            Shift ,
            double  )
```

10.391.2.19 vtkGetMacro() [8/8]

```
vtkGDCMThreadedImageReader2::vtkGetMacro (
            UseShiftScale ,
            int  )
```

10.391.2.20 vtkGetObjectMacro()

```
vtkGDCMThreadedImageReader2::vtkGetObjectMacro (
            FileNames ,
            vtkStringArray  )
```

10.391.2.21 vtkGetVector3Macro() [1/2]

```
vtkGDCMThreadedImageReader2::vtkGetVector3Macro (
            DataOrigin ,
            double  )
```

10.391.2.22 vtkGetVector3Macro() [2/2]

```
vtkGDCMThreadedImageReader2::vtkGetVector3Macro (
            DataSpacing ,
            double  )
```

10.391.2.23 vtkGetVector6Macro()

```
vtkGDCMThreadedImageReader2::vtkGetVector6Macro (
            DataExtent ,
            int  )
```

10.391.2.24 vtkSetMacro() [1/7]

```
vtkGDCMThreadedImageReader2::vtkSetMacro (
            DataScalarType ,
            int  )
```

10.391.2.25 vtkSetMacro() [2/7]

```
vtkGDCMThreadedImageReader2::vtkSetMacro (
            FileLowerLeft ,
            int  )
```

10.391.2.26 vtkSetMacro() [3/7]

```
vtkGDCMThreadedImageReader2::vtkSetMacro (
            LoadOverlays ,
            int  )
```

10.391.2.27 vtkSetMacro() [4/7]

```
vtkGDCMThreadedImageReader2::vtkSetMacro (
            NumberOfScalarComponents ,
            int  )
```

10.391.2.28 vtkSetMacro() [5/7]

```
vtkGDCMThreadedImageReader2::vtkSetMacro (
            Scale ,
            double  )
```

10.391.2.29 vtkSetMacro() [6/7]

```
vtkGDCMThreadedImageReader2::vtkSetMacro (
            Shift ,
            double  )
```

10.391.2.30 vtkSetMacro() [7/7]

```
vtkGDCMThreadedImageReader2::vtkSetMacro (
            UseShiftScale ,
            int  )
```

10.391.2.31 vtkSetVector3Macro() [1/2]

```
vtkGDCMThreadedImageReader2::vtkSetVector3Macro (
            DataOrigin ,
            double  )
```

10.391.2.32 vtkSetVector3Macro() [2/2]

```
vtkGDCMThreadedImageReader2::vtkSetVector3Macro (
            DataSpacing ,
            double  )
```

10.391.2.33 vtkSetVector6Macro()

```
vtkGDCMThreadedImageReader2::vtkSetVector6Macro (
            DataExtent ,
            int  )
```

10.391.2.34 vtkTypeMacro()

```
vtkGDCMThreadedImageReader2::vtkTypeMacro (
            vtkGDCMThreadedImageReader2 ,
            vtkThreadedImageAlgorithm  )
```

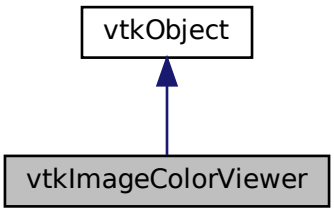
The documentation for this class was generated from the following file:

- vtkGDCMThreadedImageReader2.h

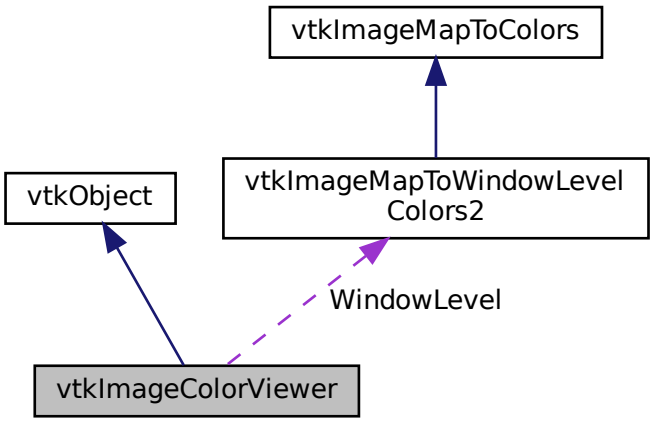
10.392 vtkImageColorViewer Class Reference

`#include <vtkImageColorViewer.h>`

Inheritance diagram for vtkImageColorViewer:



Collaboration diagram for vtkImageColorViewer:



Public Types

- enum {
  SLICE_ORIENTATION_YZ = 0 ,
  SLICE_ORIENTATION_XZ = 1 ,
  SLICE_ORIENTATION_XY = 2 }

Public Member Functions

- virtual void AddInput (vtkImageData *input)
- virtual void AddInputConnection (vtkAlgorithmOutput *input)
- virtual double GetColorLevel ()
- virtual double GetColorWindow ()
- virtual vtkImageData * GetInput ()
- virtual int GetOffScreenRendering ()
- double GetOverlayVisibility ()
- virtual int * GetPosition ()
- virtual int * GetSize ()
- virtual int GetSliceMax ()
- virtual int GetSliceMin ()
- virtual int * GetSliceRange ()
- virtual void GetSliceRange (int &min, int &max)
- virtual void GetSliceRange (int range[2])
- virtual const char * GetWindowName ()
- void PrintSelf (ostream &os, vtkIndent indent)
- virtual void Render (void)
- virtual void SetColorLevel (double s)
- virtual void SetColorWindow (double s)
- virtual void SetDisplayId (void *a)
- virtual void SetInput (vtkImageData *in)
- virtual void SetInputConnection (vtkAlgorithmOutput *input)
- virtual void SetOffScreenRendering (int)
- void SetOverlayVisibility (double vis)
- virtual void SetParentId (void *a)
- virtual void SetPosition (int a, int b)
- virtual void SetPosition (int a[2])
- virtual void SetRenderer (vtkRenderer *arg)
- virtual void SetRenderWindow (vtkRenderWindow *arg)
- virtual void SetSize (int a, int b)
- virtual void SetSize (int a[2])
- virtual void SetSlice (int s)
- virtual void SetSliceOrientation (int orientation)
- virtual void SetSliceOrientationToXY ()
- virtual void SetSliceOrientationToXZ ()
- virtual void SetSliceOrientationToYZ ()
- virtual void SetupInteractor (vtkRenderWindowInteractor *)
- virtual void SetWindowId (void *a)
- virtual void UpdateDisplayExtent ()
- VTK_LEGACY (int GetWholeZMax())
- VTK_LEGACY (int GetWholeZMin())
- VTK_LEGACY (int GetZSlice())
- VTK_LEGACY (void SetZSlice(int))
- vtkBooleanMacro (OffScreenRendering, int)
- vtkGetMacro (Slice, int)
- vtkGetMacro (SliceOrientation, int)
- vtkGetObjectMacro (ImageActor, vtkImageActor)
- vtkGetObjectMacro (InteractorStyle, vtkInteractorStyleImage)
- vtkGetObjectMacro (Renderer, vtkRenderer)
- vtkGetObjectMacro (RenderWindow, vtkRenderWindow)
- vtkGetObjectMacro (WindowLevel, vtkImageMapToWindowLevelColors2)
- vtkTypeMacro (vtkImageColorViewer, vtkObject)

Static Public Member Functions

- static vtkImageColorViewer * New ()

Protected Member Functions

- vtkImageColorViewer ()
- ~vtkImageColorViewer ()
- virtual void InstallPipeline ()
- virtual void UnInstallPipeline ()
- virtual void UpdateOrientation ()

Protected Attributes

- int FirstRender
- vtkImageActor * ImageActor
- vtkRenderWindowInteractor * Interactor
- vtkInteractorStyleImage * InteractorStyle
- vtkImageActor * OverlayImageActor
- vtkRenderer * Renderer
- vtkRenderWindow * RenderWindow
- int Slice
- int SliceOrientation
- vtkImageMapToWindowLevelColors2 * WindowLevel

Friends

- class vtkImageColorViewerCallback

10.392.1 Detailed Description

Examples

gdcmrtionplan.cxx, and gdcmrtplan.cxx.

10.392.2 Member Enumeration Documentation

10.392.2.1 anonymous enum

```
anonymous enum
```

Enumerator

SLICE_ORIENTATION_YZ	
SLICE_ORIENTATION_XZ	
SLICE_ORIENTATION_XY	

10.392.3 Constructor & Destructor Documentation

10.392.3.1 vtkImageColorViewer()

```
vtkImageColorViewer::vtkImageColorViewer ( )  [protected]
```

10.392.3.2 ∼vtkImageColorViewer()

```
vtkImageColorViewer::∼vtkImageColorViewer ( )  [protected]
```

10.392.4 Member Function Documentation

10.392.4.1 AddInput()

```
virtual void vtkImageColorViewer::AddInput (
            vtkImageData * input )  [virtual]
```

10.392.4.2 AddInputConnection()

```
virtual void vtkImageColorViewer::AddInputConnection (
            vtkAlgorithmOutput * input )  [virtual]
```

10.392.4.3 GetColorLevel()

```
virtual double vtkImageColorViewer::GetColorLevel ( )  [virtual]
```

10.392.4.4 GetColorWindow()

```
virtual double vtkImageColorViewer::GetColorWindow ( )  [virtual]
```

10.392.4.5 GetInput()

```
virtual vtkImageData * vtkImageColorViewer::GetInput ( )  [virtual]
```

10.392.4.6 GetOffScreenRendering()

```
virtual int vtkImageColorViewer::GetOffScreenRendering ( )  [virtual]
```

10.392.4.7 GetOverlayVisibility()

```
double vtkImageColorViewer::GetOverlayVisibility ( )
```

10.392.4.8 GetPosition()

```
virtual int * vtkImageColorViewer::GetPosition ( )  [virtual]
```

10.392.4.9 GetSize()

```
virtual int * vtkImageColorViewer::GetSize ( )  [virtual]
```

10.392.4.10 GetSliceMax()

```
virtual int vtkImageColorViewer::GetSliceMax ( )  [virtual]
```

10.392.4.11 GetSliceMin()

```
virtual int vtkImageColorViewer::GetSliceMin ( )  [virtual]
```

10.392.4.12 GetSliceRange() [1/3]

```
virtual int * vtkImageColorViewer::GetSliceRange ( )  [virtual]
```

10.392.4.13 GetSliceRange() [2/3]

```
virtual void vtkImageColorViewer::GetSliceRange (
            int & min,
            int & max )  [virtual]
```

10.392.4.14 GetSliceRange() [3/3]

```
virtual void vtkImageColorViewer::GetSliceRange (
            int range[2] )  [inline], [virtual]
```

10.392.4.15 GetWindowName()

```
virtual const char * vtkImageColorViewer::GetWindowName ( )  [virtual]
```

10.392.4.16 InstallPipeline()

```
virtual void vtkImageColorViewer::InstallPipeline ( )  [protected], [virtual]
```

10.392.4.17 New()

```
static vtkImageColorViewer * vtkImageColorViewer::New ( )  [static]
```

Examples

gdcmrtionplan.cxx, and gdcmrtplan.cxx.

10.392.4.18 PrintSelf()

```
void vtkImageColorViewer::PrintSelf (
            ostream & os,
            vtkIndent indent )
```

10.392.4.19 Render()

```
virtual void vtkImageColorViewer::Render (
            void  )  [virtual]
```

Examples

gdcmrtionplan.cxx, and gdcmrtplan.cxx.

10.392.4.20 SetColorLevel()

```
virtual void vtkImageColorViewer::SetColorLevel (
            double s )  [virtual]
```

10.392.4.21 SetColorWindow()

```
virtual void vtkImageColorViewer::SetColorWindow (
            double s )  [virtual]
```

10.392.4.22 SetDisplayId()

```
virtual void vtkImageColorViewer::SetDisplayId (
            void * a )  [virtual]
```

10.392.4.23 SetInput()

```
virtual void vtkImageColorViewer::SetInput (
            vtkImageData * in )  [virtual]
```

Examples

gdcmrtionplan.cxx, and gdcmrtplan.cxx.

10.392.4.24 SetInputConnection()

```
virtual void vtkImageColorViewer::SetInputConnection (
            vtkAlgorithmOutput * input )  [virtual]
```

10.392.4.25 SetOffScreenRendering()

```
virtual void vtkImageColorViewer::SetOffScreenRendering (
            int  )  [virtual]
```

10.392.4.26 SetOverlayVisibility()

```
void vtkImageColorViewer::SetOverlayVisibility (
            double vis )
```

10.392.4.27 SetParentId()

```
virtual void vtkImageColorViewer::SetParentId (
            void * a )  [virtual]
```

10.392.4.28 SetPosition() [1/2]

```
virtual void vtkImageColorViewer::SetPosition (
            int a,
            int b )  [virtual]
```

10.392.4.29 SetPosition() [2/2]

```
virtual void vtkImageColorViewer::SetPosition (
            int a[2] )  [inline], [virtual]
```

References SetPosition().

Referenced by SetPosition().

10.392.4.30 SetRenderer()

```
virtual void vtkImageColorViewer::SetRenderer (
            vtkRenderer * arg )  [virtual]
```

10.392.4.31 SetRenderWindow()

```
virtual void vtkImageColorViewer::SetRenderWindow (
            vtkRenderWindow * arg )  [virtual]
```

10.392.4.32 SetSize() [1/2]

```
virtual void vtkImageColorViewer::SetSize (
            int a,
            int b )  [virtual]
```

Examples

gdcmrtionplan.cxx, and gdcmrtplan.cxx.

10.392.4.33 SetSize() [2/2]

```
virtual void vtkImageColorViewer::SetSize (
            int a[2] )  [inline], [virtual]
```

References SetSize().

Referenced by SetSize().

10.392.4.34 SetSlice()

```
virtual void vtkImageColorViewer::SetSlice (
            int s )  [virtual]
```

10.392.4.35 SetSliceOrientation()

```
virtual void vtkImageColorViewer::SetSliceOrientation (
            int orientation )  [virtual]
```

10.392.4.36 SetSliceOrientationToXY()

```
virtual void vtkImageColorViewer::SetSliceOrientationToXY ( )  [inline], [virtual]
```

References SLICE_ORIENTATION_XY.

10.392.4.37 SetSliceOrientationToXZ()

```
virtual void vtkImageColorViewer::SetSliceOrientationToXZ ( )  [inline], [virtual]
```

References SLICE_ORIENTATION_XZ.

10.392.4.38 SetSliceOrientationToYZ()

```
virtual void vtkImageColorViewer::SetSliceOrientationToYZ ( )  [inline], [virtual]
```

References SLICE_ORIENTATION_YZ.

10.392.4.39 SetupInteractor()

```
virtual void vtkImageColorViewer::SetupInteractor (
            vtkRenderWindowInteractor *  )  [virtual]
```

Examples

gdcmrtionplan.cxx, and gdcmrtplan.cxx.

10.392.4.40 SetWindowId()

```
virtual void vtkImageColorViewer::SetWindowId (
            void * a )  [virtual]
```

10.392.4.41 UnInstallPipeline()

```
virtual void vtkImageColorViewer::UnInstallPipeline ( )  [protected], [virtual]
```

10.392.4.42 UpdateDisplayExtent()

```
virtual void vtkImageColorViewer::UpdateDisplayExtent ( )  [virtual]
```

10.392.4.43 UpdateOrientation()

```
virtual void vtkImageColorViewer::UpdateOrientation ( )  [protected], [virtual]
```

10.392.4.44 VTK_LEGACY() [1/4]

```
vtkImageColorViewer::VTK_LEGACY (
            int  GetWholeZMax() )
```

10.392.4.45 VTK_LEGACY() [2/4]

```
vtkImageColorViewer::VTK_LEGACY (
            int  GetWholeZMin() )
```

10.392.4.46 VTK_LEGACY() [3/4]

```
vtkImageColorViewer::VTK_LEGACY (
            int  GetZSlice() )
```

10.392.4.47 VTK_LEGACY() [4/4]

```
vtkImageColorViewer::VTK_LEGACY (
            void  SetZSliceint )
```

10.392.4.48 vtkBooleanMacro()

```
vtkImageColorViewer::vtkBooleanMacro (
            OffScreenRendering ,
            int  )
```

10.392.4.49 vtkGetMacro() [1/2]

```
vtkImageColorViewer::vtkGetMacro (
            Slice ,
            int  )
```

10.392.4.50 vtkGetMacro() [2/2]

```
vtkImageColorViewer::vtkGetMacro (
            SliceOrientation ,
            int  )
```

10.392.4.51 vtkGetObjectMacro() [1/5]

```
vtkImageColorViewer::vtkGetObjectMacro (
            ImageActor ,
            vtkImageActor  )
```

10.392.4.52 vtkGetObjectMacro() [2/5]

```
vtkImageColorViewer::vtkGetObjectMacro (
            InteractorStyle ,
            vtkInteractorStyleImage  )
```

10.392.4.53 vtkGetObjectMacro() [3/5]

```
vtkImageColorViewer::vtkGetObjectMacro (
            Renderer ,
            vtkRenderer  )
```

10.392.4.54 vtkGetObjectMacro() [4/5]

```
vtkImageColorViewer::vtkGetObjectMacro (
            RenderWindow ,
            vtkRenderWindow  )
```

10.392.4.55 vtkGetObjectMacro() [5/5]

```
vtkImageColorViewer::vtkGetObjectMacro (
            WindowLevel ,
            vtkImageMapToWindowLevelColors2  )
```

10.392.4.56 vtkTypeMacro()

```
vtkImageColorViewer::vtkTypeMacro (
            vtkImageColorViewer ,
            vtkObject  )
```

10.392.5 Friends And Related Function Documentation

10.392.5.1 vtkImageColorViewerCallback

```
friend class vtkImageColorViewerCallback  [friend]
```

10.392.6 Member Data Documentation

10.392.6.1 FirstRender

```
int vtkImageColorViewer::FirstRender  [protected]
```

10.392.6.2 ImageActor

```
vtkImageActor* vtkImageColorViewer::ImageActor  [protected]
```

10.392.6.3 Interactor

```
vtkRenderWindowInteractor* vtkImageColorViewer::Interactor  [protected]
```

10.392.6.4 InteractorStyle

```
vtkInteractorStyleImage* vtkImageColorViewer::InteractorStyle  [protected]
```

10.392.6.5 OverlayImageActor

```
vtkImageActor* vtkImageColorViewer::OverlayImageActor  [protected]
```

10.392.6.6 Renderer

```
vtkRenderer* vtkImageColorViewer::Renderer  [protected]
```

10.392.6.7 RenderWindow

```
vtkRenderWindow* vtkImageColorViewer::RenderWindow  [protected]
```

10.392.6.8 Slice

```
int vtkImageColorViewer::Slice  [protected]
```

10.392.6.9 SliceOrientation

```
int vtkImageColorViewer::SliceOrientation  [protected]
```

10.392.6.10 WindowLevel

```
vtkImageMapToWindowLevelColors2* vtkImageColorViewer::WindowLevel  [protected]
```

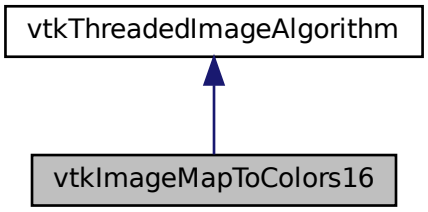
The documentation for this class was generated from the following file:

- vtkImageColorViewer.h

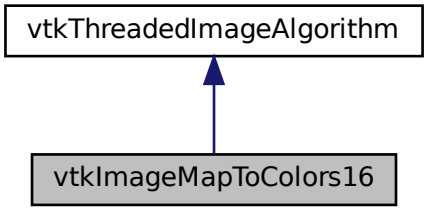
10.393 vtkImageMapToColors16 Class Reference

`#include <vtkImageMapToColors16.h>`

Inheritance diagram for vtkImageMapToColors16:



Collaboration diagram for vtkImageMapToColors16:



Public Member Functions

- virtual unsigned long GetMTime ()
- void PrintSelf (ostream &os, vtkIndent indent)
- virtual void SetLookupTable (vtkScalarsToColors ∗)
- void SetOutputFormatToLuminance ()
- void SetOutputFormatToLuminanceAlpha ()
- void SetOutputFormatToRGB ()
- void SetOutputFormatToRGBA ()
- vtkBooleanMacro (PassAlphaToOutput, int)
- vtkGetMacro (ActiveComponent, int)
- vtkGetMacro (OutputFormat, int)

- vtkGetMacro (PassAlphaToOutput, int)
- vtkGetObjectMacro (LookupTable, vtkScalarsToColors)
- vtkSetMacro (ActiveComponent, int)
- vtkSetMacro (OutputFormat, int)
- vtkSetMacro (PassAlphaToOutput, int)
- vtkTypeMacro (vtkImageMapToColors16, vtkThreadedImageAlgorithm)

Static Public Member Functions

- static vtkImageMapToColors16 * New ()

Protected Member Functions

- vtkImageMapToColors16 ()
- ~vtkImageMapToColors16 ()
- virtual int RequestData (vtkInformation *request, vtkInformationVector **inputVector, vtkInformationVector *outputVector)
- virtual int RequestInformation (vtkInformation *, vtkInformationVector **, vtkInformationVector *)
- void ThreadedRequestData (vtkInformation *request, vtkInformationVector **inputVector, vtkInformationVector *outputVector, vtkImageData ***inData, vtkImageData **outData, int extent[6], int id)

Protected Attributes

- int ActiveComponent
- int DataWasPassed
- vtkScalarsToColors * LookupTable
- int OutputFormat
- int PassAlphaToOutput

10.393.1 Constructor & Destructor Documentation

10.393.1.1 vtkImageMapToColors16()

```
vtkImageMapToColors16::vtkImageMapToColors16 ( )  [protected]
```

10.393.1.2 ~vtkImageMapToColors16()

```
vtkImageMapToColors16::~vtkImageMapToColors16 ( )  [protected]
```

10.393.2 Member Function Documentation

10.393.2.1 GetMTime()

```
virtual unsigned long vtkImageMapToColors16::GetMTime ( )  [virtual]
```

10.393.2.2 New()

```
static vtkImageMapToColors16 * vtkImageMapToColors16::New ( )  [static]
```

10.393.2.3 PrintSelf()

```
void vtkImageMapToColors16::PrintSelf (
            ostream & os,
            vtkIndent indent )
```

10.393.2.4 RequestData()

```
virtual int vtkImageMapToColors16::RequestData (
            vtkInformation * request,
            vtkInformationVector ** inputVector,
            vtkInformationVector * outputVector )  [protected], [virtual]
```

10.393.2.5 RequestInformation()

```
virtual int vtkImageMapToColors16::RequestInformation (
            vtkInformation * ,
            vtkInformationVector ** ,
            vtkInformationVector *  )  [protected], [virtual]
```

10.393.2.6 SetLookupTable()

```
virtual void vtkImageMapToColors16::SetLookupTable (
            vtkScalarsToColors *  )  [virtual]
```

10.393.2.7 SetOutputFormatToLuminance()

```
void vtkImageMapToColors16::SetOutputFormatToLuminance ( )  [inline]
```

10.393.2.8 SetOutputFormatToLuminanceAlpha()

```
void vtkImageMapToColors16::SetOutputFormatToLuminanceAlpha ( )  [inline]
```

10.393.2.9 SetOutputFormatToRGB()

```
void vtkImageMapToColors16::SetOutputFormatToRGB ( )  [inline]
```

10.393.2.10 SetOutputFormatToRGBA()

```
void vtkImageMapToColors16::SetOutputFormatToRGBA ( )  [inline]
```

10.393.2.11 ThreadedRequestData()

```
void vtkImageMapToColors16::ThreadedRequestData (
            vtkInformation * request,
            vtkInformationVector ** inputVector,
            vtkInformationVector * outputVector,
            vtkImageData *** inData,
            vtkImageData ** outData,
            int extent[6],
            int id )  [protected]
```

10.393.2.12 vtkBooleanMacro()

```
vtkImageMapToColors16::vtkBooleanMacro (
            PassAlphaToOutput ,
            int  )
```

10.393.2.13 vtkGetMacro() [1/3]

```
vtkImageMapToColors16::vtkGetMacro (
            ActiveComponent ,
            int  )
```

10.393.2.14 vtkGetMacro() [2/3]

```
vtkImageMapToColors16::vtkGetMacro (
            OutputFormat ,
            int  )
```

10.393.2.15 vtkGetMacro() [3/3]

```
vtkImageMapToColors16::vtkGetMacro (
            PassAlphaToOutput ,
            int  )
```

10.393.2.16 vtkGetObjectMacro()

```
vtkImageMapToColors16::vtkGetObjectMacro (
            LookupTable ,
            vtkScalarsToColors  )
```

10.393.2.17 vtkSetMacro() [1/3]

```
vtkImageMapToColors16::vtkSetMacro (
            ActiveComponent ,
            int  )
```

10.393.2.18 vtkSetMacro() [2/3]

```
vtkImageMapToColors16::vtkSetMacro (
            OutputFormat ,
            int  )
```

10.393.2.19 vtkSetMacro() [3/3]

```
vtkImageMapToColors16::vtkSetMacro (
            PassAlphaToOutput ,
            int  )
```

10.393.2.20 vtkTypeMacro()

```
vtkImageMapToColors16::vtkTypeMacro (
            vtkImageMapToColors16 ,
            vtkThreadedImageAlgorithm  )
```

10.393.3 Member Data Documentation

10.393.3.1 ActiveComponent

```
int vtkImageMapToColors16::ActiveComponent  [protected]
```

10.393.3.2 DataWasPassed

```
int vtkImageMapToColors16::DataWasPassed  [protected]
```

10.393.3.3 LookupTable

```
vtkScalarsToColors* vtkImageMapToColors16::LookupTable  [protected]
```

10.393.3.4 OutputFormat

```
int vtkImageMapToColors16::OutputFormat  [protected]
```

10.393.3.5 PassAlphaToOutput

```
int vtkImageMapToColors16::PassAlphaToOutput  [protected]
```

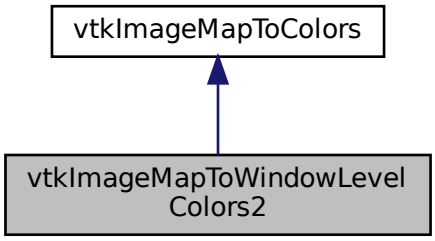
The documentation for this class was generated from the following file:

- vtkImageMapToColors16.h

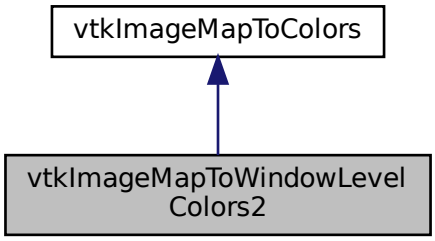
10.394 vtkImageMapToWindowLevelColors2 Class Reference

```
#include <vtkImageMapToWindowLevelColors2.h>
```

Inheritance diagram for vtkImageMapToWindowLevelColors2:



Collaboration diagram for vtkImageMapToWindowLevelColors2:



Public Member Functions

- void PrintSelf (ostream &os, vtkIndent indent)
- vtkGetMacro (Level, double)
- vtkGetMacro (Window, double)
- vtkSetMacro (Level, double)
- vtkSetMacro (Window, double)
- vtkTypeMacro (vtkImageMapToWindowLevelColors2, vtkImageMapToColors)

Static Public Member Functions

- static vtkImageMapToWindowLevelColors2 * New ()

Protected Member Functions

- vtkImageMapToWindowLevelColors2 ()
- ~vtkImageMapToWindowLevelColors2 ()
- virtual int RequestData (vtkInformation *request, vtkInformationVector **inputVector, vtkInformationVector *outputVector)
- virtual int RequestInformation (vtkInformation *, vtkInformationVector **, vtkInformationVector *)
- void ThreadedRequestData (vtkInformation *request, vtkInformationVector **inputVector, vtkInformationVector *outputVector, vtkImageData ***inData, vtkImageData **outData, int extent[6], int id)

Protected Attributes

- double Level
- double Window

10.394.1 Constructor & Destructor Documentation

10.394.1.1 vtkImageMapToWindowLevelColors2()

```
vtkImageMapToWindowLevelColors2::vtkImageMapToWindowLevelColors2 ( )  [protected]
```

10.394.1.2 ~vtkImageMapToWindowLevelColors2()

```
vtkImageMapToWindowLevelColors2::~vtkImageMapToWindowLevelColors2 ( )  [protected]
```

10.394.2 Member Function Documentation

10.394.2.1 New()

```
static vtkImageMapToWindowLevelColors2 * vtkImageMapToWindowLevelColors2::New ( )  [static]
```

10.394.2.2 PrintSelf()

```
void vtkImageMapToWindowLevelColors2::PrintSelf (
            ostream & os,
            vtkIndent indent )
```

10.394.2.3 RequestData()

```
virtual int vtkImageMapToWindowLevelColors2::RequestData (
            vtkInformation * request,
            vtkInformationVector ** inputVector,
            vtkInformationVector * outputVector )  [protected], [virtual]
```

10.394.2.4 RequestInformation()

```
virtual int vtkImageMapToWindowLevelColors2::RequestInformation (
            vtkInformation * ,
            vtkInformationVector ** ,
            vtkInformationVector *  )  [protected], [virtual]
```

10.394.2.5 ThreadedRequestData()

```
void vtkImageMapToWindowLevelColors2::ThreadedRequestData (
            vtkInformation * request,
            vtkInformationVector ** inputVector,
            vtkInformationVector * outputVector,
            vtkImageData *** inData,
            vtkImageData ** outData,
            int extent[6],
            int id )  [protected]
```

10.394.2.6 vtkGetMacro() [1/2]

```
vtkImageMapToWindowLevelColors2::vtkGetMacro (
            Level ,
            double  )
```

10.394.2.7 vtkGetMacro() [2/2]

```
vtkImageMapToWindowLevelColors2::vtkGetMacro (
            Window ,
            double  )
```

10.394.2.8 vtkSetMacro() [1/2]

```
vtkImageMapToWindowLevelColors2::vtkSetMacro (
            Level ,
            double  )
```

10.394.2.9 vtkSetMacro() [2/2]

```
vtkImageMapToWindowLevelColors2::vtkSetMacro (
            Window ,
            double  )
```

10.394.2.10 vtkTypeMacro()

```
vtkImageMapToWindowLevelColors2::vtkTypeMacro (
            vtkImageMapToWindowLevelColors2 ,
            vtkImageMapToColors  )
```

10.394.3 Member Data Documentation

10.394.3.1 Level

```
double vtkImageMapToWindowLevelColors2::Level  [protected]
```

10.394.3.2 Window

```
double vtkImageMapToWindowLevelColors2::Window  [protected]
```

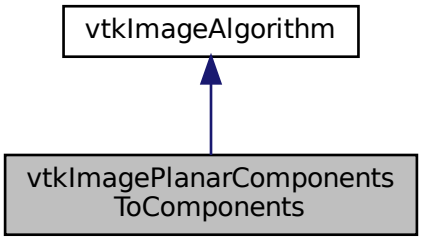
The documentation for this class was generated from the following file:

- vtkImageMapToWindowLevelColors2.h

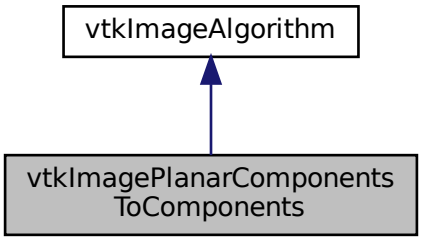
10.395 vtkImagePlanarComponentsToComponents Class Reference

```
#include <vtkImagePlanarComponentsToComponents.h>
```

Inheritance diagram for vtkImagePlanarComponentsToComponents:



Collaboration diagram for vtkImagePlanarComponentsToComponents:



Public Member Functions

- void PrintSelf (ostream &os, vtkIndent indent)
- vtkTypeMacro (vtkImagePlanarComponentsToComponents, vtkImageAlgorithm)

Static Public Member Functions

- static vtkImagePlanarComponentsToComponents * New ()

Protected Member Functions

- vtkImagePlanarComponentsToComponents ()
- ~vtkImagePlanarComponentsToComponents ()
- virtual int RequestData (vtkInformation *, vtkInformationVector **, vtkInformationVector *)

10.395.1 Constructor & Destructor Documentation

10.395.1.1 vtkImagePlanarComponentsToComponents()

```
vtkImagePlanarComponentsToComponents::vtkImagePlanarComponentsToComponents ( )  [protected]
```

10.395.1.2 ~vtkImagePlanarComponentsToComponents()

```
vtkImagePlanarComponentsToComponents::~vtkImagePlanarComponentsToComponents ( )  [inline], [protected]
```

10.395.2 Member Function Documentation

10.395.2.1 New()

```
static vtkImagePlanarComponentsToComponents * vtkImagePlanarComponentsToComponents::New ( )  [static]
```

10.395.2.2 PrintSelf()

```
void vtkImagePlanarComponentsToComponents::PrintSelf (
            ostream & os,
            vtkIndent indent )
```

10.395.2.3 RequestData()

```
virtual int vtkImagePlanarComponentsToComponents::RequestData (
            vtkInformation * ,
            vtkInformationVector ** ,
            vtkInformationVector *  )  [protected], [virtual]
```

10.395.2.4 vtkTypeMacro()

```
vtkImagePlanarComponentsToComponents::vtkTypeMacro (
            vtkImagePlanarComponentsToComponents ,
            vtkImageAlgorithm  )
```

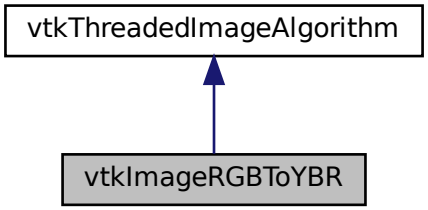
The documentation for this class was generated from the following file:

- vtkImagePlanarComponentsToComponents.h

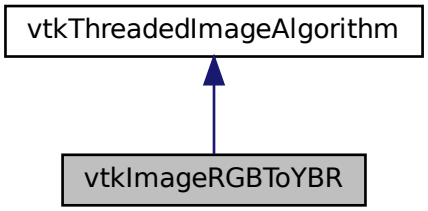
10.396 vtkImageRGBToYBR Class Reference

```
#include <vtkImageRGBToYBR.h>
```

Inheritance diagram for vtkImageRGBToYBR:



Collaboration diagram for vtkImageRGBToYBR:



Public Member Functions

- void PrintSelf (ostream &os, vtkIndent indent)
- vtkTypeMacro (vtkImageRGBToYBR, vtkThreadedImageAlgorithm)

Static Public Member Functions

- static vtkImageRGBToYBR * New ()

Protected Member Functions

- vtkImageRGBToYBR ()
- ~vtkImageRGBToYBR ()
- void ThreadedExecute (vtkImageData *inData, vtkImageData *outData, int ext[6], int id)

10.396.1 Constructor & Destructor Documentation

10.396.1.1 vtkImageRGBToYBR()

```
vtkImageRGBToYBR::vtkImageRGBToYBR ( )  [protected]
```

10.396.1.2 ~vtkImageRGBToYBR()

```
vtkImageRGBToYBR::~vtkImageRGBToYBR ( )  [inline], [protected]
```

10.396.2 Member Function Documentation

10.396.2.1 New()

```
static vtkImageRGBToYBR * vtkImageRGBToYBR::New ( )  [static]
```

10.396.2.2 PrintSelf()

```
void vtkImageRGBToYBR::PrintSelf (
             ostream & os,
             vtkIndent indent )
```

10.396.2.3 ThreadedExecute()

```
void vtkImageRGBToYBR::ThreadedExecute (
             vtkImageData * inData,
             vtkImageData * outData,
             int ext[6],
             int id )  [protected]
```

10.396.2.4 vtkTypeMacro()

```
vtkImageRGBToYBR::vtkTypeMacro (
             vtkImageRGBToYBR ,
             vtkThreadedImageAlgorithm  )
```

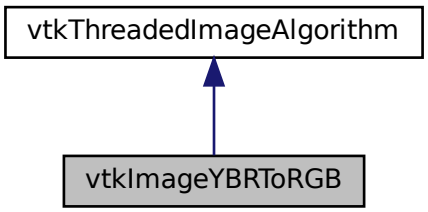
The documentation for this class was generated from the following file:

- vtkImageRGBToYBR.h

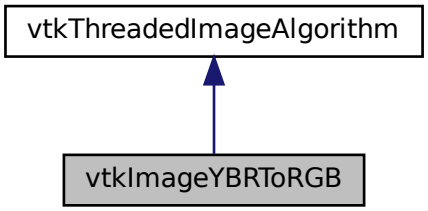
10.397 vtkImageYBRToRGB Class Reference

`#include <vtkImageYBRToRGB.h>`

Inheritance diagram for vtkImageYBRToRGB:



Collaboration diagram for vtkImageYBRToRGB:



Public Member Functions

- void PrintSelf (ostream &os, vtkIndent indent)
- vtkTypeMacro (vtkImageYBRToRGB, vtkThreadedImageAlgorithm)

Static Public Member Functions

- static vtkImageYBRToRGB * New ()

Protected Member Functions

- vtkImageYBRToRGB ()
- ∼vtkImageYBRToRGB ()
- void ThreadedExecute (vtkImageData ∗inData, vtkImageData ∗outData, int ext[6], int id)

10.397.1 Constructor & Destructor Documentation

10.397.1.1 vtkImageYBRToRGB()

```
vtkImageYBRToRGB::vtkImageYBRToRGB ( )  [protected]
```

10.397.1.2 ∼vtkImageYBRToRGB()

```
vtkImageYBRToRGB::∼vtkImageYBRToRGB ( )  [inline], [protected]
```

10.397.2 Member Function Documentation

10.397.2.1 New()

```
static vtkImageYBRToRGB * vtkImageYBRToRGB::New ( )  [static]
```

10.397.2.2 PrintSelf()

```
void vtkImageYBRToRGB::PrintSelf (
            ostream & os,
            vtkIndent indent )
```

10.397.2.3 ThreadedExecute()

```
void vtkImageYBRToRGB::ThreadedExecute (
            vtkImageData * inData,
            vtkImageData * outData,
            int ext[6],
            int id )  [protected]
```

10.397.2.4 vtkTypeMacro()

```
vtkImageYBRToRGB::vtkTypeMacro (
            vtkImageYBRToRGB ,
            vtkThreadedImageAlgorithm  )
```

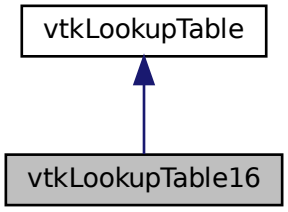
The documentation for this class was generated from the following file:

- vtkImageYBRToRGB.h

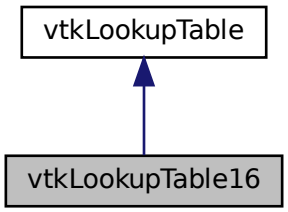
10.398 vtkLookupTable16 Class Reference

```
#include <vtkLookupTable16.h>
```

Inheritance diagram for vtkLookupTable16:



Collaboration diagram for vtkLookupTable16:



Public Member Functions

- void Build ()
- unsigned short * GetPointer (const vtkIdType id)
- void PrintSelf (ostream &os, vtkIndent indent)
- void SetNumberOfTableValues (vtkIdType number)
- vtkTypeMacro (vtkLookupTable16, vtkLookupTable)
- unsigned char * WritePointer (const vtkIdType id, const int number)

Static Public Member Functions

- static vtkLookupTable16 * New ()

Protected Member Functions

- vtkLookupTable16 (int sze=256, int ext=256)
- ~vtkLookupTable16 ()
- void MapScalarsThroughTable2 (void *input, unsigned char *output, int inputDataType, int numberOfValues, int inputIncrement, int outputFormat)

Protected Attributes

- vtkUnsignedShortArray * Table16

10.398.1 Constructor & Destructor Documentation

10.398.1.1 vtkLookupTable16()

```
vtkLookupTable16::vtkLookupTable16 (
            int sze = 256,
            int ext = 256 )  [protected]
```

10.398.1.2 ~vtkLookupTable16()

```
vtkLookupTable16::~vtkLookupTable16 ( )  [protected]
```

10.398.2 Member Function Documentation

10.398.2.1 Build()

```
void vtkLookupTable16::Build ( )
```

10.398.2.2 GetPointer()

```
unsigned short * vtkLookupTable16::GetPointer (
            const vtkIdType id )  [inline]
```

10.398.2.3 MapScalarsThroughTable2()

```
void vtkLookupTable16::MapScalarsThroughTable2 (
            void * input,
            unsigned char * output,
            int inputDataType,
            int numberOfValues,
            int inputIncrement,
            int outputFormat )  [protected]
```

10.398.2.4 New()

```
static vtkLookupTable16 * vtkLookupTable16::New ( )  [static]
```

10.398.2.5 PrintSelf()

```
void vtkLookupTable16::PrintSelf (
            ostream & os,
            vtkIndent indent )
```

10.398.2.6 SetNumberOfTableValues()

```
void vtkLookupTable16::SetNumberOfTableValues (
            vtkIdType number )
```

10.398.2.7 vtkTypeMacro()

```
vtkLookupTable16::vtkTypeMacro (
            vtkLookupTable16 ,
            vtkLookupTable  )
```

10.398.2.8 WritePointer()

```
unsigned char * vtkLookupTable16::WritePointer (
            const vtkIdType id,
            const int number )  [inline]
```

References Table16.

10.398.3 Member Data Documentation

10.398.3.1 Table16

```
vtkUnsignedShortArray* vtkLookupTable16::Table16  [protected]
```

Referenced by WritePointer().

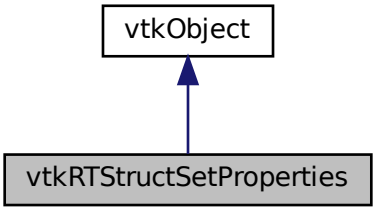
The documentation for this class was generated from the following file:

- vtkLookupTable16.h

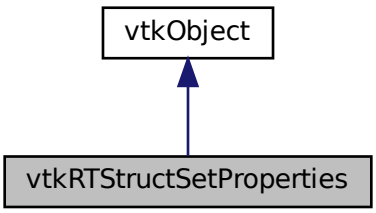
10.399 vtkRTStructSetProperties Class Reference

```
#include <vtkRTStructSetProperties.h>
```

Inheritance diagram for vtkRTStructSetProperties:



Collaboration diagram for vtkRTStructSetProperties:



Public Member Functions

- void AddContourReferencedFrameOfReference (vtkIdType pdnum, const char *classuid, const char *instanceuid)
- void AddReferencedFrameOfReference (const char *classuid, const char *instanceuid)
- void AddStructureSetROI (int roinumber, const char *reframerefuid, const char *roiname, const char *ROIGenerationAlgorithm, const char *ROIDescription=0)
- void AddStructureSetROIObservation (int refnumber, int observationnumber, const char *rtroiinterpretedtype, const char *roiinterpreter, const char *roiobservationlabel=0)
- virtual void Clear ()
- virtual void DeepCopy (vtkRTStructSetProperties *p)
- const char * GetContourReferencedFrameOfReferenceClassUID (vtkIdType pdnum, vtkIdType id)
- const char * GetContourReferencedFrameOfReferenceInstanceUID (vtkIdType pdnum, vtkIdType id)
- vtkIdType GetNumberOfContourReferencedFrameOfReferences ()
- vtkIdType GetNumberOfContourReferencedFrameOfReferences (vtkIdType pdnum)
- vtkIdType GetNumberOfReferencedFrameOfReferences ()
- vtkIdType GetNumberOfStructureSetROIs ()
- const char * GetReferencedFrameOfReferenceClassUID (vtkIdType id)
- const char * GetReferencedFrameOfReferenceInstanceUID (vtkIdType id)
- int GetStructureSetObservationNumber (vtkIdType id)
- const char * GetStructureSetROIDescription (vtkIdType id)
- const char * GetStructureSetROIGenerationAlgorithm (vtkIdType)
- const char * GetStructureSetROIName (vtkIdType)
- int GetStructureSetROINumber (vtkIdType id)
- const char * GetStructureSetROIObservationLabel (vtkIdType id)
- const char * GetStructureSetROIRefFrameRefUID (vtkIdType)
- const char * GetStructureSetRTROIInterpretedType (vtkIdType id)
- void PrintSelf (ostream &os, vtkIndent indent)
- vtkGetStringMacro (ReferenceFrameOfReferenceUID)
- vtkGetStringMacro (ReferenceSeriesInstanceUID)
- vtkGetStringMacro (SeriesInstanceUID)
- vtkGetStringMacro (SOPInstanceUID)
- vtkGetStringMacro (StructureSetDate)
- vtkGetStringMacro (StructureSetLabel)
- vtkGetStringMacro (StructureSetName)
- vtkGetStringMacro (StructureSetTime)
- vtkGetStringMacro (StudyInstanceUID)
- vtkSetStringMacro (ReferenceFrameOfReferenceUID)
- vtkSetStringMacro (ReferenceSeriesInstanceUID)
- vtkSetStringMacro (SeriesInstanceUID)
- vtkSetStringMacro (SOPInstanceUID)
- vtkSetStringMacro (StructureSetDate)
- vtkSetStringMacro (StructureSetLabel)
- vtkSetStringMacro (StructureSetName)
- vtkSetStringMacro (StructureSetTime)
- vtkSetStringMacro (StudyInstanceUID)
- vtkTypeMacro (vtkRTStructSetProperties, vtkObject)

Static Public Member Functions

- static vtkRTStructSetProperties * New ()

Protected Member Functions

- vtkRTStructSetProperties ()
- ~vtkRTStructSetProperties ()

Protected Attributes

- vtkRTStructSetPropertiesInternals * Internals
- char * ReferenceFrameOfReferenceUID
- char * ReferenceSeriesInstanceUID
- char * SeriesInstanceUID
- char * SOPInstanceUID
- char * StructureSetDate
- char * StructureSetLabel
- char * StructureSetName
- char * StructureSetTime
- char * StudyInstanceUID

10.399.1 Detailed Description

Examples

GenerateRTSTRUCT.cxx.

10.399.2 Constructor & Destructor Documentation

10.399.2.1 vtkRTStructSetProperties()

```
vtkRTStructSetProperties::vtkRTStructSetProperties ( )  [protected]
```

10.399.2.2 ~vtkRTStructSetProperties()

```
vtkRTStructSetProperties::~vtkRTStructSetProperties ( )  [protected]
```

10.399.3 Member Function Documentation

10.399.3.1 AddContourReferencedFrameOfReference()

```
void vtkRTStructSetProperties::AddContourReferencedFrameOfReference (
            vtkIdType pdnum,
            const char * classuid,
            const char * instanceuid )
```

10.399.3.2 AddReferencedFrameOfReference()

```
void vtkRTStructSetProperties::AddReferencedFrameOfReference (
            const char * classuid,
            const char * instanceuid )
```

10.399.3.3 AddStructureSetROI()

```
void vtkRTStructSetProperties::AddStructureSetROI (
            int roinumber,
            const char * refframerefuid,
            const char * roiname,
            const char * ROIGenerationAlgorithm,
            const char * ROIDescription = 0 )
```

10.399.3.4 AddStructureSetROIObservation()

```
void vtkRTStructSetProperties::AddStructureSetROIObservation (
            int refnumber,
            int observationnumber,
            const char * rtroiinterpretedtype,
            const char * roiinterpreter,
            const char * roiobservationlabel = 0 )
```

10.399.3.5 Clear()

```
virtual void vtkRTStructSetProperties::Clear ( )  [virtual]
```

10.399.3.6 DeepCopy()

```
virtual void vtkRTStructSetProperties::DeepCopy (
            vtkRTStructSetProperties * p )  [virtual]
```

10.399.3.7 GetContourReferencedFrameOfReferenceClassUID()

```
const char * vtkRTStructSetProperties::GetContourReferencedFrameOfReferenceClassUID (
            vtkIdType pdnum,
            vtkIdType id )
```

10.399.3.8 GetContourReferencedFrameOfReferenceInstanceUID()

```
const char * vtkRTStructSetProperties::GetContourReferencedFrameOfReferenceInstanceUID (
            vtkIdType pdnum,
            vtkIdType id )
```

10.399.3.9 GetNumberOfContourReferencedFrameOfReferences() [1/2]

```
vtkIdType vtkRTStructSetProperties::GetNumberOfContourReferencedFrameOfReferences ( )
```

10.399.3.10 GetNumberOfContourReferencedFrameOfReferences() [2/2]

```
vtkIdType vtkRTStructSetProperties::GetNumberOfContourReferencedFrameOfReferences (
            vtkIdType pdnum )
```

10.399.3.11 GetNumberOfReferencedFrameOfReferences()

```
vtkIdType vtkRTStructSetProperties::GetNumberOfReferencedFrameOfReferences ( )
```

10.399.3.12 GetNumberOfStructureSetROIs()

```
vtkIdType vtkRTStructSetProperties::GetNumberOfStructureSetROIs ( )
```

10.399.3.13 GetReferencedFrameOfReferenceClassUID()

```
const char * vtkRTStructSetProperties::GetReferencedFrameOfReferenceClassUID (
            vtkIdType id )
```

10.399.3.14 GetReferencedFrameOfReferenceInstanceUID()

```
const char * vtkRTStructSetProperties::GetReferencedFrameOfReferenceInstanceUID (
            vtkIdType id )
```

10.399.3.15 GetStructureSetObservationNumber()

```
int vtkRTStructSetProperties::GetStructureSetObservationNumber (
            vtkIdType id )
```

10.399.3.16 GetStructureSetROIDescription()

```
const char * vtkRTStructSetProperties::GetStructureSetROIDescription (
            vtkIdType id )
```

10.399.3.17 GetStructureSetROIGenerationAlgorithm()

```
const char * vtkRTStructSetProperties::GetStructureSetROIGenerationAlgorithm (
            vtkIdType  )
```

10.399.3.18 GetStructureSetROIName()

```
const char * vtkRTStructSetProperties::GetStructureSetROIName (
            vtkIdType  )
```

10.399.3.19 GetStructureSetROINumber()

```
int vtkRTStructSetProperties::GetStructureSetROINumber (
            vtkIdType id )
```

10.399.3.20 GetStructureSetROIObservationLabel()

```
const char * vtkRTStructSetProperties::GetStructureSetROIObservationLabel (
            vtkIdType id )
```

10.399.3.21 GetStructureSetROIRefFrameRefUID()

```
const char * vtkRTStructSetProperties::GetStructureSetROIRefFrameRefUID (
            vtkIdType  )
```

10.399.3.22 GetStructureSetRTROIInterpretedType()

```
const char * vtkRTStructSetProperties::GetStructureSetRTROIInterpretedType (
            vtkIdType id )
```

10.399.3.23 New()

```
static vtkRTStructSetProperties * vtkRTStructSetProperties::New ( )  [static]
```

Examples

GenerateRTSTRUCT.cxx.

10.399.3.24 PrintSelf()

```
void vtkRTStructSetProperties::PrintSelf (
            ostream & os,
            vtkIndent indent )
```

10.399.3.25 vtkGetStringMacro() [1/9]

```
vtkRTStructSetProperties::vtkGetStringMacro (
            ReferenceFrameOfReferenceUID  )
```

10.399.3.26 vtkGetStringMacro() [2/9]

```
vtkRTStructSetProperties::vtkGetStringMacro (
            ReferenceSeriesInstanceUID  )
```

10.399.3.27 vtkGetStringMacro() [3/9]

```
vtkRTStructSetProperties::vtkGetStringMacro (
            SeriesInstanceUID  )
```

10.399.3.28 vtkGetStringMacro() [4/9]

```
vtkRTStructSetProperties::vtkGetStringMacro (
            SOPInstanceUID  )
```

10.399.3.29 vtkGetStringMacro() [5/9]

```
vtkRTStructSetProperties::vtkGetStringMacro (
            StructureSetDate  )
```

10.399.3.30 vtkGetStringMacro() [6/9]

```
vtkRTStructSetProperties::vtkGetStringMacro (
            StructureSetLabel  )
```

10.399.3.31 vtkGetStringMacro() [7/9]

```
vtkRTStructSetProperties::vtkGetStringMacro (
            StructureSetName  )
```

10.399.3.32 vtkGetStringMacro() [8/9]

```
vtkRTStructSetProperties::vtkGetStringMacro (
            StructureSetTime  )
```

10.399.3.33 vtkGetStringMacro() [9/9]

```
vtkRTStructSetProperties::vtkGetStringMacro (
            StudyInstanceUID  )
```

10.399.3.34 vtkSetStringMacro() [1/9]

```
vtkRTStructSetProperties::vtkSetStringMacro (
            ReferenceFrameOfReferenceUID  )
```

10.399.3.35 vtkSetStringMacro() [2/9]

```
vtkRTStructSetProperties::vtkSetStringMacro (
            ReferenceSeriesInstanceUID  )
```

10.399.3.36 vtkSetStringMacro() [3/9]

```
vtkRTStructSetProperties::vtkSetStringMacro (
            SeriesInstanceUID  )
```

10.399.3.37 vtkSetStringMacro() [4/9]

```
vtkRTStructSetProperties::vtkSetStringMacro (
            SOPInstanceUID  )
```

10.399.3.38 vtkSetStringMacro() [5/9]

```
vtkRTStructSetProperties::vtkSetStringMacro (
            StructureSetDate  )
```

10.399.3.39 vtkSetStringMacro() [6/9]

```
vtkRTStructSetProperties::vtkSetStringMacro (
            StructureSetLabel  )
```

10.399.3.40 vtkSetStringMacro() [7/9]

```
vtkRTStructSetProperties::vtkSetStringMacro (
            StructureSetName  )
```

10.399.3.41 vtkSetStringMacro() [8/9]

```
vtkRTStructSetProperties::vtkSetStringMacro (
            StructureSetTime  )
```

10.399.3.42 vtkSetStringMacro() [9/9]

```
vtkRTStructSetProperties::vtkSetStringMacro (
            StudyInstanceUID  )
```

10.399.3.43 vtkTypeMacro()

```
vtkRTStructSetProperties::vtkTypeMacro (
            vtkRTStructSetProperties ,
            vtkObject  )
```

10.399.4 Member Data Documentation

10.399.4.1 Internals

```
vtkRTStructSetPropertiesInternals* vtkRTStructSetProperties::Internals  [protected]
```

10.399.4.2 ReferenceFrameOfReferenceUID

```
char* vtkRTStructSetProperties::ReferenceFrameOfReferenceUID  [protected]
```

10.399.4.3 ReferenceSeriesInstanceUID

```
char* vtkRTStructSetProperties::ReferenceSeriesInstanceUID  [protected]
```

10.399.4.4 SeriesInstanceUID

```
char* vtkRTStructSetProperties::SeriesInstanceUID  [protected]
```

10.399.4.5 SOPInstanceUID

```
char* vtkRTStructSetProperties::SOPInstanceUID  [protected]
```

10.399.4.6 StructureSetDate

```
char* vtkRTStructSetProperties::StructureSetDate  [protected]
```

10.399.4.7 StructureSetLabel

```
char* vtkRTStructSetProperties::StructureSetLabel  [protected]
```

10.399.4.8 StructureSetName

```
char* vtkRTStructSetProperties::StructureSetName  [protected]
```

10.399.4.9 StructureSetTime

```
char* vtkRTStructSetProperties::StructureSetTime  [protected]
```

10.399.4.10 StudyInstanceUID

```
char* vtkRTStructSetProperties::StudyInstanceUID  [protected]
```

The documentation for this class was generated from the following file:

- vtkRTStructSetProperties.h

10.400 gdcm::Waveform Class Reference

Waveform class.

```
#include <gdcmWaveform.h>
```

Public Member Functions

- Waveform ()=default

10.400.1 Detailed Description

Waveform class.

10.400.2 Constructor & Destructor Documentation

10.400.2.1 Waveform()

```
gdcm::Waveform::Waveform ( )  [default]
```

The documentation for this class was generated from the following file:

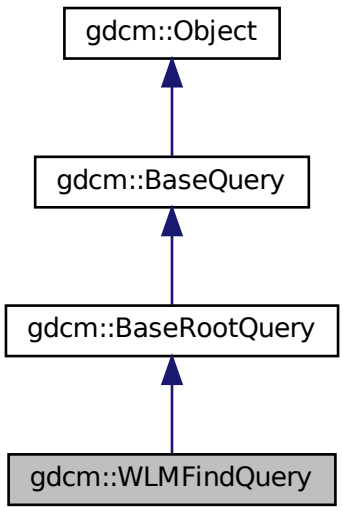
- gdcmWaveform.h

10.401 gdcm::WLMFindQuery Class Reference

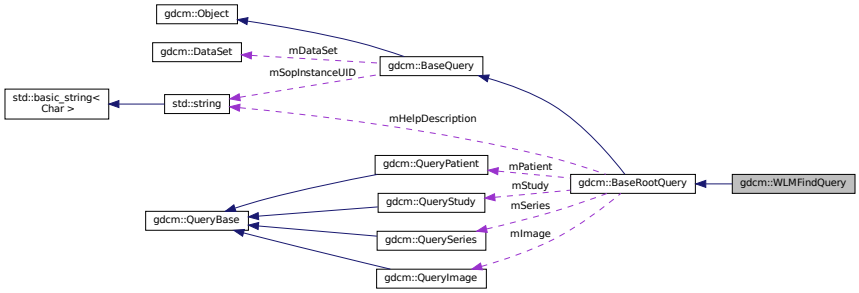
PatientRootQuery.

```
#include <gdcmWLMFindQuery.h>
```

Inheritance diagram for gdcm::WLMFindQuery:



Collaboration diagram for gdcm::WLMFindQuery:



Public Member Functions

- WLMFindQuery ()
- UIDs::TSName GetAbstractSyntaxUID () const override
- std::vector< Tag > GetTagListByLevel (const EQueryLevel &inQueryLevel) override
- void InitializeDataSet (const EQueryLevel &inQueryLevel) override
- bool ValidateQuery (bool inStrict=true) const override

Protected Member Functions

- DataSet GetValidDataSet () const

Friends

- class QueryFactory

Additional Inherited Members

10.401.1 Detailed Description

PatientRootQuery.

contains: the class which will produce a dataset for c-find with patient root

10.401.2 Constructor & Destructor Documentation

10.401.2.1 WLMFindQuery()

```
gdcm::WLMFindQuery::WLMFindQuery ( )
```

10.401.3 Member Function Documentation

10.401.3.1 GetAbstractSyntaxUID()

```
UIDs::TSName gdcm::WLMFindQuery::GetAbstractSyntaxUID ( ) const  [override], [virtual]
```

Implements gdcm::BaseQuery.

10.401.3.2 GetTagListByLevel()

```
std::vector< Tag > gdcm::WLMFindQuery::GetTagListByLevel (
            const EQueryLevel & inQueryLevel )  [override], [virtual]
```

this function will return all tags at a given query level, so that they maybe selected for searching. The boolean forFind is true if the query is a find query, or false for a move query.

Implements gdcm::BaseRootQuery.

10.401.3.3 GetValidDataSet()

```
DataSet gdcm::WLMFindQuery::GetValidDataSet ( ) const  [protected]
```

10.401.3.4 InitializeDataSet()

```
void gdcm::WLMFindQuery::InitializeDataSet (
            const EQueryLevel & inQueryLevel )  [override], [virtual]
```

this function sets tag 8,52 to the appropriate value based on query level also fills in the right unique tags, as per the standard's requirements should allow for connection with dcmtk

Implements gdcm::BaseRootQuery.

10.401.3.5 ValidateQuery()

```
bool gdcm::WLMFindQuery::ValidateQuery (
            bool inStrict = true ) const  [override], [virtual]
```

have to be able to ensure that 0x8,0x52 is set (which will be true if InitializeDataSet is called...) that the level is appropriate (ie, not setting PATIENT for a study query that the tags in the query match the right level (either required, unique, optional) by default, this function checks to see if the query is for finding, which is more permissive than for moving. For moving, only the unique tags are allowed. 10 Jan 2011: adding in the 'strict' mode. according to the standard (at least, how I've read it), only tags for a particular level should be allowed in a particular query (ie, just series level tags in a series level query). However, it seems that dcm4chee doesn't share that interpretation. So, if 'inStrict' is false, then tags from the current level and all higher levels are now considered valid. So, if you're doing a non-strict series-level query, tags from the patient and study level can be passed along as well.

Implements gdcm::BaseRootQuery.

10.401.4 Friends And Related Function Documentation

10.401.4.1 QueryFactory

```
friend class QueryFactory  [friend]
```

The documentation for this class was generated from the following file:

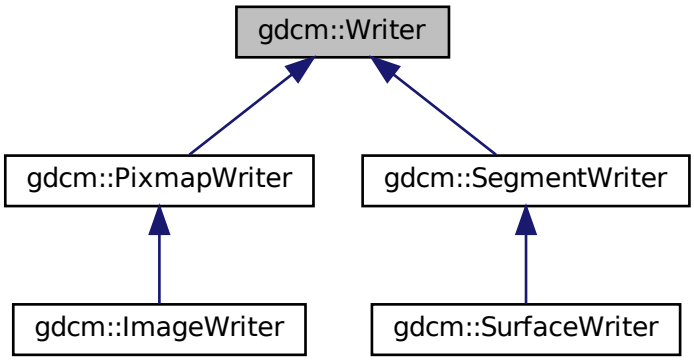
- gdcmWLMFindQuery.h

10.402 gdcm::Writer Class Reference

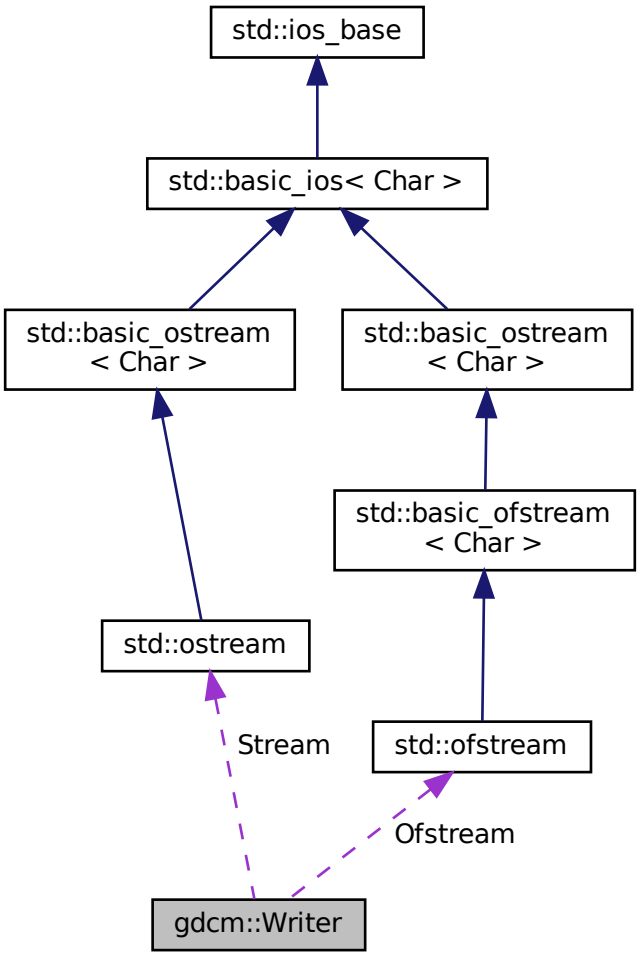
Writer ala DOM (Document Object Model)

```
#include <gdcmWriter.h>
```

Inheritance diagram for gdcm::Writer:



Collaboration diagram for gdcm::Writer:



Public Member Functions

- Writer ()
- virtual ~Writer ()
- void CheckFileMetaInformationOff ()
- void CheckFileMetaInformationOn ()
- File & GetFile ()
- void SetCheckFileMetaInformation (bool b)

  Undocumented function, do not use (= leave default)
- void SetFile (const File &f)

  Set/Get the DICOM file (DataSet + Header)

- void SetFileName (const char *filename_native)

  Set the filename of DICOM file to write:
- void SetStream (std::ostream &output_stream)

  Set user ostream buffer.
- virtual bool Write ()

  Main function to tell the writer to write.

Protected Member Functions

- bool GetCheckFileMetaInformation () const
- std::ostream * GetStreamPtr () const
- void SetWriteDataSetOnly (bool b)

Protected Attributes

- std::ofstream * Ofstream
- std::ostream * Stream

Friends

- class StreamImageWriter

10.402.1 Detailed Description

Writer ala DOM (Document Object Model)

This class is a non-validating writer, it will only performs well- formedness check only.

Detailed description here To avoid GDCM being yet another broken DICOM lib we try to be user level and avoid writing illegal stuff (odd length, non-zero value for Item start/end length ...) Therefore you cannot (well unless you are really smart) write DICOM with even length tag. All the checks are consider basics:

- Correct Meta Information Header (see gdcm::FileMetaInformation)
- Zero value for Item Length (0xfffe, 0xe00d/0xe0dd)
- Even length for any elements
- Alphabetical order for elements (guaranteed by design of internals)
- 32bits VR will be rewritten with 00

Warning

gdcm::Writer cannot write a DataSet if no SOP Instance UID (0008,0018) is found, unless a DICOMDIR is being written out

See also

Reader DataSet File

Examples

BasicAnonymizer.cs, ChangePrivateTags.cxx, ChangeSequenceUltrasound.cxx, ClinicalTrialAnnotate.cxx, ClinicalTrialIdentificationWorkflow.cs, CreateFakeRTDOSE.cxx, CreateJPIPDataSet.cxx, DeriveSeries.cxx, DuplicatePCDE.cxx, EncapsulateFileInRawData.cxx, Extracting_All_Resolution.cxx, Fake_Image_Using_Stream_Image_Writer.cxx FixBrokenJ2K.cxx, FixJAIBugJPEGLS.cxx, FixOrientation.cxx, GenAllVR.cxx, GenFakeIdentifyFile.cxx, GenLongSeqs.cxx, GenSeqs.cxx, GenerateDICOMDIR.cs, HelloWorld.cxx, LargeVRDSExplicit.cxx, MakeTemplate.cxx, ManipulateFile.cs, NewSequence.cs, PatchFile.cxx, QIDO-RS.cxx, ReformatFile.cs, StreamImageReaderTest.cxx, pmsct_rgb1.cxx, and rle2img.cxx.

10.402.2 Constructor & Destructor Documentation

10.402.2.1 Writer()

```
gdcm::Writer::Writer ( )
```

10.402.2.2 ~Writer()

```
virtual gdcm::Writer::~Writer ( )  [virtual]
```

10.402.3 Member Function Documentation

10.402.3.1 CheckFileMetaInformationOff()

```
void gdcm::Writer::CheckFileMetaInformationOff ( )  [inline]
```

Examples

CreateFakeRTDOSE.cxx, FixBrokenJ2K.cxx, and HelloWorld.cxx.

10.402.3.2 CheckFileMetaInformationOn()

```
void gdcm::Writer::CheckFileMetaInformationOn ( )  [inline]
```

10.402.3.3 GetCheckFileMetaInformation()

```
bool gdcm::Writer::GetCheckFileMetaInformation ( ) const  [inline], [protected]
```

10.402.3.4 GetFile()

```
File & gdcm::Writer::GetFile ( )  [inline]
```

Examples

CreateJPIPDataSet.cxx, EncapsulateFileInRawData.cxx, Extracting_All_Resolution.cxx, Fake_Image_Using_Stream_Image_Writer GenAllVR.cxx, GenFakeIdentifyFile.cxx, GetSubSequenceData.cxx, MpegVideoInfo.cs, QIDO-RS.cxx, StreamImageReaderTest.cxx, TemplateEmptyImage.cxx, iU22tomultisc.cxx, pmsct_rgb1.cxx, and rle2img.cxx.

10.402.3.5 GetStreamPtr()

```
std::ostream * gdcm::Writer::GetStreamPtr ( ) const  [inline], [protected]
```

10.402.3.6 SetCheckFileMetaInformation()

```
void gdcm::Writer::SetCheckFileMetaInformation (
            bool b )  [inline]
```

Undocumented function, do not use (= leave default)

Examples

GenAllVR.cxx, GenFakeIdentifyFile.cxx, and PatchFile.cxx.

10.402.3.7 SetFile()

```
void gdcm::Writer::SetFile (
            const File & f )  [inline]
```

Set/Get the DICOM file (DataSet + Header)

Examples

BasicAnonymizer.cs, BasicImageAnonymizer.cs, ChangePrivateTags.cxx, ChangeSequenceUltrasound.cxx, ClinicalTrialAnnotate.cxx, ClinicalTrialIdentificationWorkflow.cs, CompressImage.cxx, CompressLossyJPEG.cs, CreateFakeRTDOSE.cxx, DecompressImage.cs, DeriveSeries.cxx, DuplicatePCDE.cxx, FixBrokenJ2K.cxx, FixJAIBugJPEGLS.cxx, FixOrientation.cxx, GenFakeImage.cxx, GenLongSeqs.cxx, GenSeqs.cxx, GenerateDICOMDIR.cs, HelloWorld.cxx, LargeVRDSExplicit.cxx, MakeTemplate.cxx, ManipulateFile.cs, MergeTwoFiles.cxx, NewSequence.cs, PatchFile.cxx, ReformatFile.cs, StandardizeFiles.cs, pmsct_rgb1.cxx, and rle2img.cxx.

10.402.3.8 SetFileName()

```
void gdcm::Writer::SetFileName (
            const char * filename_native )
```

Set the filename of DICOM file to write:

Examples

BasicAnonymizer.cs, BasicImageAnonymizer.cs, ChangePrivateTags.cxx, ChangeSequenceUltrasound.cxx, ClinicalTrialAnnotate.cxx, ClinicalTrialIdentificationWorkflow.cs, CompressImage.cxx, CompressLossyJPEG.cs, CreateARGBImage.cxx, CreateCMYKImage.cxx, CreateFakeRTDOSE.cxx, CreateJPIPDataSet.cxx, DecompressImage.cs, DeriveSeries.cxx, DuplicatePCDE.cxx, EncapsulateFileInRawData.cxx, FixBrokenJ2K.cxx, FixJAIBugJPEGLS.cxx, FixOrientation.cxx, GenAllVR.cxx, GenFakeIdentifyFile.cxx, GenFakeImage.cxx, GenLongSeqs.cxx, GenSeqs.cxx, GenerateDICOMDIR.cs, GetSubSequenceData.cxx, HelloVizWorld.cxx, HelloWorld.cxx, LargeVRDSExplicit.cxx, MakeTemplate.cxx, ManipulateFile.cs, MergeTwoFiles.cxx, MpegVideoInfo.cs, NewSequence.cs, PatchFile.cxx, QIDO-RS.cxx, ReformatFile.cs, StandardizeFiles.cs, TemplateEmptyImage.cxx, csa2img.cxx, iU22tomultisc.cxx, pmsct_rgb1.cxx, and rle2img.cxx.

10.402.3.9 SetStream()

```
void gdcm::Writer::SetStream (
            std::ostream & output_stream )  [inline]
```

Set user ostream buffer.

10.402.3.10 SetWriteDataSetOnly()

```
void gdcm::Writer::SetWriteDataSetOnly (
            bool b )  [inline], [protected]
```

10.402.3.11 Write()

```
virtual bool gdcm::Writer::Write ( )  [virtual]
```

Main function to tell the writer to write.

Reimplemented in gdcm::ImageWriter, gdcm::PixmapWriter, gdcm::SegmentWriter, and gdcm::SurfaceWriter.

Examples

BasicAnonymizer.cs, ChangePrivateTags.cxx, ChangeSequenceUltrasound.cxx, ClinicalTrialAnnotate.cxx, ClinicalTrialIdentificationWorkflow.cs, CreateFakeRTDOSE.cxx, CreateJPIPDataSet.cxx, DeriveSeries.cxx, DuplicatePCDE.cxx, EncapsulateFileInRawData.cxx, FixBrokenJ2K.cxx, FixJAIBugJPEGLS.cxx, FixOrientation.cxx, GenAllVR.cxx, GenFakeIdentifyFile.cxx, GenLongSeqs.cxx, GenSeqs.cxx, GenerateDICOMDIR.cs, HelloWorld.cxx, LargeVRDSExplicit.cxx, MakeTemplate.cxx, ManipulateFile.cs, NewSequence.cs, PatchFile.cxx, QIDO-RS.cxx, ReformatFile.cs, pmsct_rgb1.cxx, and rle2img.cxx.

10.402.4 Friends And Related Function Documentation

10.402.4.1 StreamImageWriter

```
friend class StreamImageWriter  [friend]
```

10.402.5 Member Data Documentation

10.402.5.1 Ofstream

```
std::ofstream* gdcm::Writer::Ofstream  [protected]
```

10.402.5.2 Stream

```
std::ostream* gdcm::Writer::Stream  [protected]
```

The documentation for this class was generated from the following file:

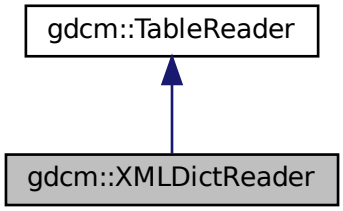
- gdcmWriter.h

10.403 gdcm::XMLDictReader Class Reference

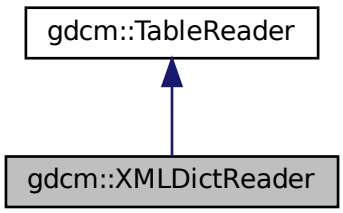
Class for representing a XMLDictReader.

```
#include <gdcmXMLDictReader.h>
```

Inheritance diagram for gdcm::XMLDictReader:



Collaboration diagram for gdcm::XMLDictReader:



Public Member Functions

- XMLDictReader ()
- ~XMLDictReader ()
- void CharacterDataHandler (const char *data, int length)
- void EndElement (const char *name)
- const Dict & GetDict ()
- void StartElement (const char *name, const char **atts)

Protected Member Functions

- void HandleDescription (const char **atts)
- void HandleEntry (const char **atts)

10.403.1 Detailed Description

Class for representing a XMLDictReader.

Note

bla Will read the DICOMV3.xml file

10.403.2 Constructor & Destructor Documentation

10.403.2.1 XMLDictReader()

```
gdcm::XMLDictReader::XMLDictReader ( )
```

10.403.2.2 ~XMLDictReader()

```
gdcm::XMLDictReader::~XMLDictReader ( )  [inline]
```

10.403.3 Member Function Documentation

10.403.3.1 CharacterDataHandler()

```
void gdcm::XMLDictReader::CharacterDataHandler (
            const char * data,
            int length )  [virtual]
```

Reimplemented from gdcm::TableReader.

10.403.3.2 EndElement()

```
void gdcm::XMLDictReader::EndElement (
            const char * name )  [virtual]
```

Reimplemented from gdcm::TableReader.

10.403.3.3 GetDict()

```
const Dict & gdcm::XMLDictReader::GetDict ( )  [inline]
```

10.403.3.4 HandleDescription()

```
void gdcm::XMLDictReader::HandleDescription (
            const char ** atts )  [protected]
```

10.403.3.5 HandleEntry()

```
void gdcm::XMLDictReader::HandleEntry (
            const char ** atts )  [protected]
```

10.403.3.6 StartElement()

```
void gdcm::XMLDictReader::StartElement (
            const char * name,
            const char ** atts )  [virtual]
```

Reimplemented from gdcm::TableReader.

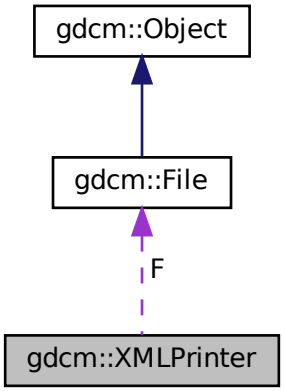
The documentation for this class was generated from the following file:

- gdcmXMLDictReader.h

10.404 gdcm::XMLPrinter Class Reference

`#include <gdcmXMLPrinter.h>`

Collaboration diagram for gdcm::XMLPrinter:



Public Types

- enum PrintStyles {
  OnlyUUID = 0 ,
  LOADBULKDATA = 1 }

Public Member Functions

- XMLPrinter ()
- virtual ~XMLPrinter ()
- PrintStyles GetPrintStyle () const
- virtual void HandleBulkData (const char *uuid, const TransferSyntax &ts, const char *bulkdata, size_t bulklen)
- void Print (std::ostream &os)
- void PrintDataSet (const DataSet &ds, const TransferSyntax &ts, std::ostream &os)
- void SetFile (File const &f)
- void SetStyle (PrintStyles ps)

Protected Member Functions

- VR PrintDataElement (std::ostream &os, const Dicts &dicts, const DataSet &ds, const DataElement &de, const TransferSyntax &ts)
- void PrintSQ (const SequenceOfItems *sqi, const TransferSyntax &ts, std::ostream &os)

Protected Attributes

- const File * F
- PrintStyles PrintStyle

10.404.1 Member Enumeration Documentation

10.404.1.1 PrintStyles

```
enum gdcm::XMLPrinter::PrintStyles
```

Enumerator

OnlyUUID	
LOADBULKDATA	

10.404.2 Constructor & Destructor Documentation

10.404.2.1 XMLPrinter()

```
gdcm::XMLPrinter::XMLPrinter ( )
```

10.404.2.2 ~XMLPrinter()

```
virtual gdcm::XMLPrinter::~XMLPrinter ( )  [virtual]
```

10.404.3 Member Function Documentation

10.404.3.1 GetPrintStyle()

```
PrintStyles gdcm::XMLPrinter::GetPrintStyle ( ) const  [inline]
```

10.404.3.2 HandleBulkData()

```
virtual void gdcm::XMLPrinter::HandleBulkData (
            const char * uuid,
            const TransferSyntax & ts,
            const char * bulkdata,
            size_t bulklen )  [virtual]
```

Virtual function mechanism to allow application programmer to override the default mechanism for BulkData handling. By default GDCM will simply discard the BulkData and only write the UUID

10.404.3.3 Print()

```
void gdcm::XMLPrinter::Print (
            std::ostream & os )
```

10.404.3.4 PrintDataElement()

```
VR gdcm::XMLPrinter::PrintDataElement (
            std::ostream & os,
            const Dicts & dicts,
            const DataSet & ds,
            const DataElement & de,
            const TransferSyntax & ts )  [protected]
```

10.404.3.5 PrintDataSet()

```
void gdcm::XMLPrinter::PrintDataSet (
            const DataSet & ds,
            const TransferSyntax & ts,
            std::ostream & os )
```

10.404.3.6 PrintSQ()

```
void gdcm::XMLPrinter::PrintSQ (
            const SequenceOfItems * sqi,
            const TransferSyntax & ts,
            std::ostream & os )  [protected]
```

10.404.3.7 SetFile()

```
void gdcm::XMLPrinter::SetFile (
            File const & f )  [inline]
```

10.404.3.8 SetStyle()

```
void gdcm::XMLPrinter::SetStyle (
            PrintStyles ps )  [inline]
```

10.404.4 Member Data Documentation

10.404.4.1 F

```
const File* gdcm::XMLPrinter::F  [protected]
```

10.404.4.2 PrintStyle

```
PrintStyles gdcm::XMLPrinter::PrintStyle  [protected]
```

The documentation for this class was generated from the following file:

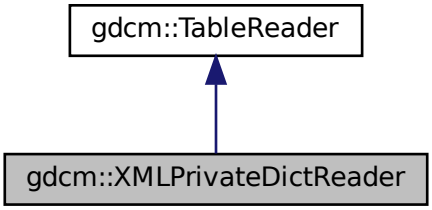
- gdcmXMLPrinter.h

10.405 gdcm::XMLPrivateDictReader Class Reference

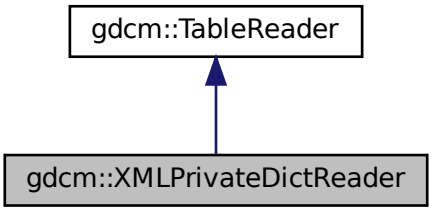
Class for representing a XMLPrivateDictReader.

`#include <gdcmXMLPrivateDictReader.h>`

Inheritance diagram for gdcm::XMLPrivateDictReader:



Collaboration diagram for gdcm::XMLPrivateDictReader:



Public Member Functions

- XMLPrivateDictReader ()
- ~XMLPrivateDictReader ()
- void CharacterDataHandler (const char *data, int length)
- void EndElement (const char *name)
- const PrivateDict & GetPrivateDict ()
- void StartElement (const char *name, const char **atts)

Protected Member Functions

- void HandleDescription (const char **atts)
- void HandleEntry (const char **atts)

10.405.1 Detailed Description

Class for representing a XMLPrivateDictReader.

Note

bla Will read the Private.xml file

10.405.2 Constructor & Destructor Documentation

10.405.2.1 XMLPrivateDictReader()

```
gdcm::XMLPrivateDictReader::XMLPrivateDictReader ( )
```

10.405.2.2 ~XMLPrivateDictReader()

```
gdcm::XMLPrivateDictReader::~XMLPrivateDictReader ( )  [inline]
```

10.405.3 Member Function Documentation

10.405.3.1 CharacterDataHandler()

```
void gdcm::XMLPrivateDictReader::CharacterDataHandler (
            const char * data,
            int length )  [virtual]
```

Reimplemented from gdcm::TableReader.

10.405.3.2 EndElement()

```
void gdcm::XMLPrivateDictReader::EndElement (
            const char * name )  [virtual]
```

Reimplemented from gdcm::TableReader.

10.405.3.3 GetPrivateDict()

```
const PrivateDict & gdcm::XMLPrivateDictReader::GetPrivateDict ( )  [inline]
```

10.405.3.4 HandleDescription()

```
void gdcm::XMLPrivateDictReader::HandleDescription (
            const char ** atts )  [protected]
```

10.405.3.5 HandleEntry()

```
void gdcm::XMLPrivateDictReader::HandleEntry (
            const char ** atts )  [protected]
```

10.405.3.6 StartElement()

```
void gdcm::XMLPrivateDictReader::StartElement (
            const char * name,
            const char ** atts )  [virtual]
```

Reimplemented from gdcm::TableReader.

The documentation for this class was generated from the following file:

- gdcmXMLPrivateDictReader.h

Chapter 11

File Documentation

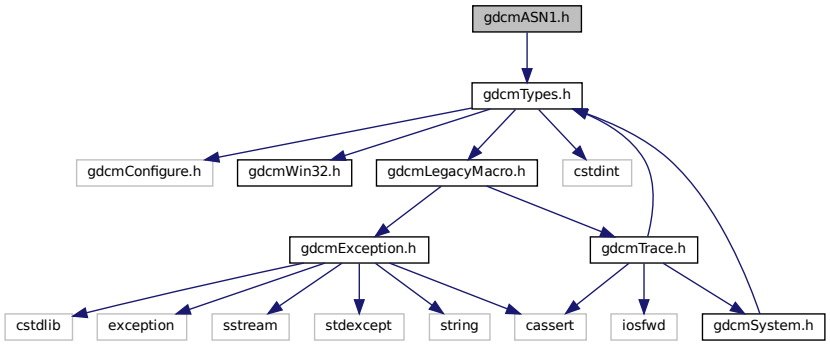
11.1 README.txt File Reference

11.2 TestsList.txt File Reference

11.3 gdcmASN1.h File Reference

```
#include "gdcmTypes.h"
```

Include dependency graph for gdcmASN1.h:



Classes

- class gdcm::ASN1

  Class for ASN1.

Namespaces

- namespace gdcm

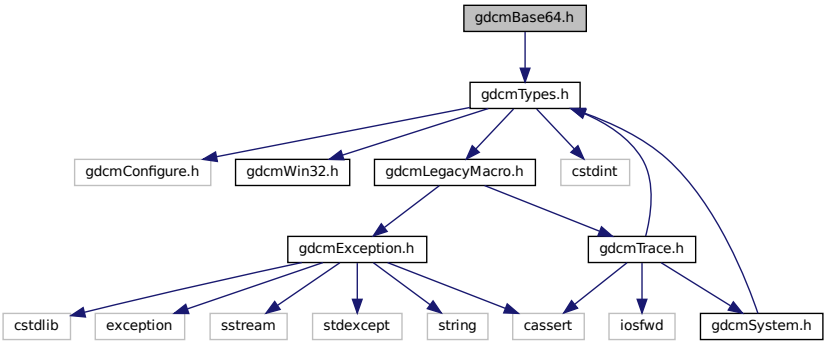
11.4 gdcmASN1.h

Go to the documentation of this file.

```
1 /*=========================================================================
2
3 Program:  GDCM (Grassroots DICOM). A DICOM library
4
5 Copyright (c) 2006-2011 Mathieu Malaterre
6 All rights reserved.
7 See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.
8
9 This software is distributed WITHOUT ANY WARRANTY; without even
10 the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
11 PURPOSE.  See the above copyright notice for more information.
12
13 =========================================================================*/
14 #ifndef GDCMASN1_H
15 #define GDCMASN1_H
16
17 #include "gdcmTypes.h"
18
19
20 namespace gdcm
21 {
22 //-----------------------------------------------------------------------------
23 class ASN1Internals;
28 class GDCM_EXPORT ASN1
29 {
30 public :
31   ASN1();
32   ~ASN1();
33
34   static bool ParseDumpFile(const char *filename);
35
36   static bool ParseDump(const char *array, size_t length);
37
38   ASN1(const ASN1&) = delete;
39   void operator=(const ASN1&) = delete;
40 protected:
41   int TestPBKDF2();
42
43 private:
44   ASN1Internals *Internals;
45 };
46 } // end namespace gdcm
47 //-----------------------------------------------------------------------------
48 #endif //GDCMASN1_H
```

11.5 gdcmBase64.h File Reference

`#include "gdcmTypes.h"`
Include dependency graph for gdcmBase64.h:



Classes

- class gdcm::Base64
  Class for Base64.

Namespaces

- namespace gdcm

11.6 gdcmBase64.h

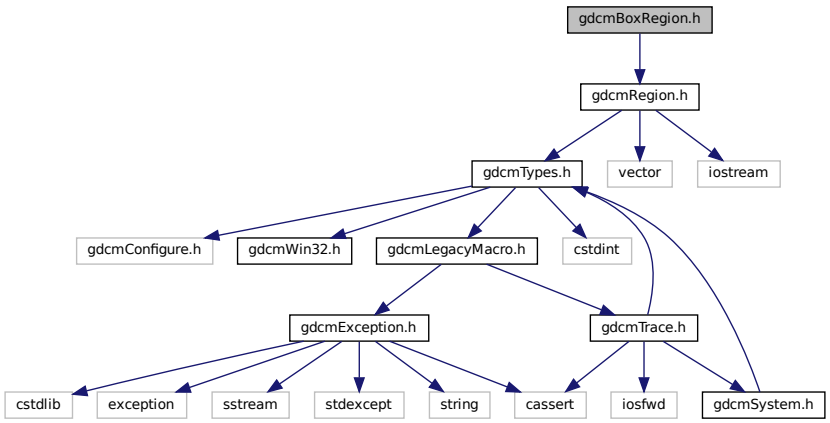
Go to the documentation of this file.

```
1 /*=========================================================================
2
3 Program:  GDCM (Grassroots DICOM). A DICOM library
4
5 Copyright (c) 2006-2011 Mathieu Malaterre
6 All rights reserved.
7 See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.
8
9 This software is distributed WITHOUT ANY WARRANTY; without even
10 the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
11 PURPOSE.  See the above copyright notice for more information.
12
13 =========================================================================*/
14 #ifndef GDCMBASE64_H
15 #define GDCMBASE64_H
16
17 #include "gdcmTypes.h"
18
19 namespace gdcm
20 {
25 class GDCM_EXPORT Base64
26 {
```

```
27 public:
28
32   static size_t GetEncodeLength(const char *src, size_t srclen );
33
45   static size_t Encode( char *dst, size_t dlen, const char *src, size_t slen );
46
50   static size_t GetDecodeLength( const char *src, size_t len );
51
62   static size_t Decode( char *dst, size_t dlen, const char *src, size_t slen );
63
64   Base64(const Base64&) = delete;
65   void operator=(const Base64&) = delete;
66 };
67
68 } // end namespace gdcm
69
70 #endif // GDCMBASE64_H
```

11.7 gdcmBoxRegion.h File Reference

`#include "gdcmRegion.h"`
Include dependency graph for gdcmBoxRegion.h:



Classes

- class gdcm::BoxRegion

  Class for manipulation box region.

Namespaces

- namespace gdcm

11.8 gdcmBoxRegion.h

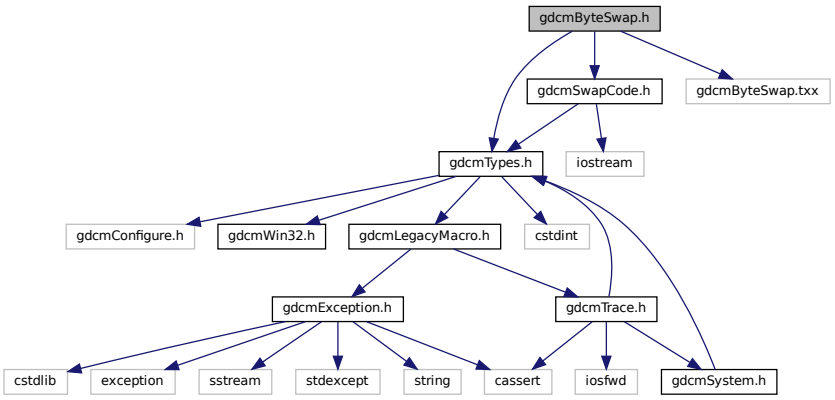
Go to the documentation of this file.

```
1 /*=========================================================================
2
3 Program:  GDCM (Grassroots DICOM). A DICOM library
4
5 Copyright (c) 2006-2011 Mathieu Malaterre
6 All rights reserved.
7 See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.
8
9 This software is distributed WITHOUT ANY WARRANTY; without even
10 the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
11 PURPOSE.  See the above copyright notice for more information.
12
13 =========================================================================*/
14 #ifndef GDCMBOXREGION_H
15 #define GDCMBOXREGION_H
16
17 #include "gdcmRegion.h"
18
19 namespace gdcm
20 {
21 class BoxRegionInternals;
29 //-----------------------------------------------------------------------------
30 class GDCM_EXPORT BoxRegion :  public Region
31 {
32 public :
33   BoxRegion();
34   ~BoxRegion() override;
35
37   void SetDomain(unsigned int xmin, unsigned int xmax,
38     unsigned int ymin, unsigned int ymax,
39     unsigned int zmin, unsigned int zmax);
40
42   unsigned int GetXMin() const;
43   unsigned int GetXMax() const;
44   unsigned int GetYMin() const;
45   unsigned int GetYMax() const;
46   unsigned int GetZMin() const;
47   unsigned int GetZMax() const;
48
49   // Satisfy pure virtual parent class
50   Region *Clone() const override;
51   bool Empty() const override;
52   bool IsValid() const override;
53   size_t Area() const override;
54   BoxRegion ComputeBoundingBox() override;
55
56   void Print(std::ostream &os = std::cout) const override;
57
59   static BoxRegion BoundingBox(BoxRegion const & b1, BoxRegion const & b2 );
60
62   BoxRegion(const BoxRegion&);
63   void operator=(const BoxRegion&);
64 private:
65   BoxRegionInternals *Internals;
66 };
67
68 } // end namespace gdcm
69 //-----------------------------------------------------------------------------
70 #endif //GDCMREGION_H
```

11.9 gdcmByteSwap.h File Reference

```
#include "gdcmTypes.h"
#include "gdcmSwapCode.h"
```

`#include "gdcmByteSwap.txx"`
Include dependency graph for gdcmByteSwap.h:



Classes

- class gdcm::ByteSwap< T >

  ByteSwap.

Namespaces

- namespace gdcm

11.10 gdcmByteSwap.h

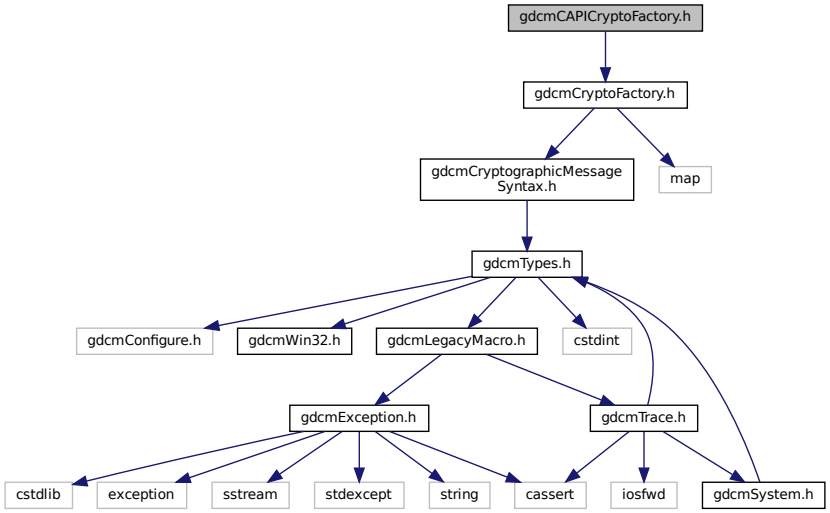
Go to the documentation of this file.

```
1 /*=========================================================================
2
3 Program:  GDCM (Grassroots DICOM). A DICOM library
4
5 Copyright (c) 2006-2011 Mathieu Malaterre
6 All rights reserved.
7 See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.
8
9 This software is distributed WITHOUT ANY WARRANTY; without even
10 the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
11 PURPOSE.  See the above copyright notice for more information.
12
13 =========================================================================*/
14 #ifndef GDCMBYTESWAP_H
15 #define GDCMBYTESWAP_H
16
17 #include "gdcmTypes.h"
18 #include "gdcmSwapCode.h"
19
20 namespace gdcm
21 {
22
29 template<class T>
30 class ByteSwap
```

```
31 {
32 public:
34   static bool SystemIsBigEndian ();
35   static bool SystemIsLittleEndian ();
36
37   static void Swap(T &p);
38   static void SwapFromSwapCodeIntoSystem(T &p, SwapCode const &sc);
39   static void SwapRange(T *p, unsigned int num);
40   static void SwapRangeFromSwapCodeIntoSystem(T *p, SwapCode const &sc,
41     std::streamoff num);
42
43 protected:
44 //  ByteSwap() {}
45 //  ~ByteSwap() {}
46
47 private:
48
49 };
50
56 } // end namespace gdcm
57
58 #include "gdcmByteSwap.txx"
59
60 #endif //GDCMBYTESWAP_H
```

11.11 gdcmCAPICryptoFactory.h File Reference

`#include "gdcmCryptoFactory.h"`
Include dependency graph for gdcmCAPICryptoFactory.h:



Classes

- class gdcm::CAPICryptoFactory

Namespaces

- namespace gdcm

11.12 gdcmCAPICryptoFactory.h

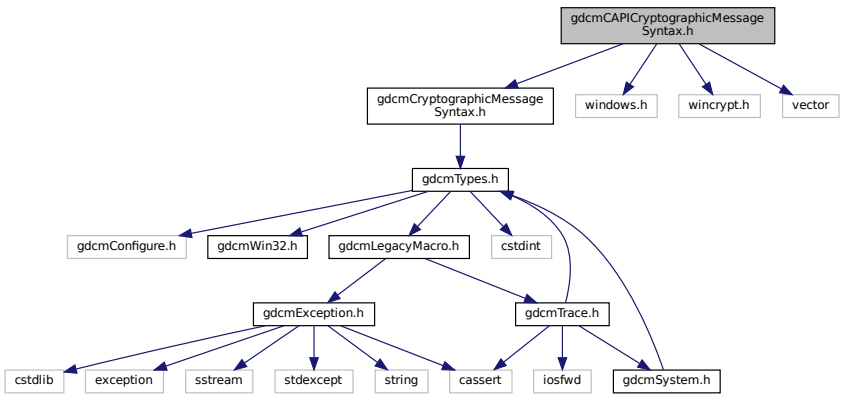
Go to the documentation of this file.

```
1 /*=========================================================================
2
3 Program:  GDCM (Grassroots DICOM). A DICOM library
4
5 Copyright (c) 2006-2011 Mathieu Malaterre
6 All rights reserved.
7 See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.
8
9 This software is distributed WITHOUT ANY WARRANTY; without even
10 the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
11 PURPOSE.  See the above copyright notice for more information.
12
13 =========================================================================*/
14 #ifndef GDCMCAPICRYPTOFACTORY_H
15 #define GDCMCAPICRYPTOFACTORY_H
16
17 #include "gdcmCryptoFactory.h"
18
19 namespace gdcm
20 {
21
22 class GDCM_EXPORT CAPICryptoFactory :  public CryptoFactory
23 {
24 public:
25   CAPICryptoFactory(CryptoLib id);
26   CryptographicMessageSyntax* CreateCMSProvider();
27
28 private:
29   CAPICryptoFactory() {}
30 };
31
32 } // end namespace gdcm
33
34 #endif //GDCMCAPICRYPTOFACTORY_H
```

11.13 gdcmCAPICryptographicMessageSyntax.h File Reference

```
#include "gdcmCryptographicMessageSyntax.h"
#include <windows.h>
#include <wincrypt.h>
#include <vector>
```

Include dependency graph for gdcmCAPICryptographicMessageSyntax.h:



Classes

- class gdcm::CAPICryptographicMessageSyntax

Namespaces

- namespace gdcm

11.14 gdcmCAPICryptographicMessageSyntax.h

Go to the documentation of this file.

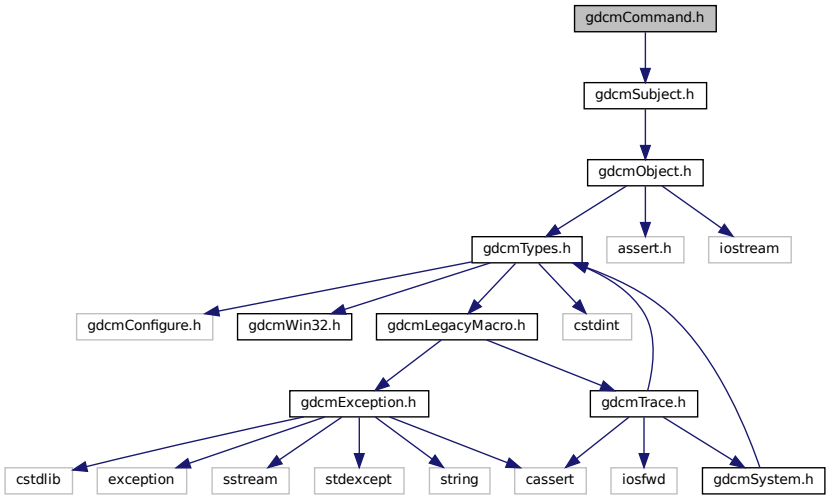
```
1 /*=========================================================================
2
3 Program:  GDCM (Grassroots DICOM). A DICOM library
4
5 Copyright (c) 2006-2011 Mathieu Malaterre
6 All rights reserved.
7 See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.
8
9 This software is distributed WITHOUT ANY WARRANTY; without even
10 the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
11 PURPOSE.  See the above copyright notice for more information.
12
13 =========================================================================*/
14 #ifndef GDCMCAPICRYPTOGRAPHICMESSAGESYNTAX_H
15 #define GDCMCAPICRYPTOGRAPHICMESSAGESYNTAX_H
16
17 #include "gdcmCryptographicMessageSyntax.h"
18 #include <windows.h>
19 #include <wincrypt.h>
20 #include <vector>
21
22 namespace gdcm
23 {
24
25 class GDCM_EXPORT CAPICryptographicMessageSyntax :  public CryptographicMessageSyntax
26 {
27 public:
28   CAPICryptographicMessageSyntax();
29   ~CAPICryptographicMessageSyntax();
30
31   // X.509
32   bool ParseCertificateFile( const char *filename );
33   bool ParseKeyFile( const char *filename );
34
35   // PBE
36   bool SetPassword(const char * pass, size_t passLen);
37
38   void SetCipherType(CipherTypes type);
39
40   CipherTypes GetCipherType() const;
41
43   bool Encrypt(char *output, size_t &outlen, const char *array, size_t len) const;
45   bool Decrypt(char *output, size_t &outlen, const char *array, size_t len) const;
46
47   bool GetInitialized()const
48 {
49     return initialized;
50   }
51
52 private:
53   bool Initialize();
54   static ALG_ID GetAlgIdByObjId(const char * pszObjId);
55   const char *GetCipherObjId() const;
56   static void ReverseBytes(unsigned char* data, DWORD len);
57   static bool LoadFile(const char * filename, unsigned char* & buffer, DWORD & bufLen);
58
59 private:
60   bool initialized;
61   HCRYPTPROV hProv;
```

```
62   std::vector<PCCERT_CONTEXT> certifList;
63   HCRYPTKEY hRsaPrivK;
64   CipherTypes cipherType;
65 };
66
67 } // end namespace gdcm
68
69 #endif //GDCMCAPICRYPTOGRAPHICMESSAGESYNTAX_H
```

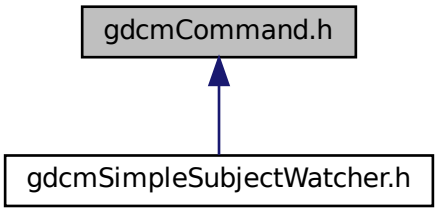
11.15 gdcmCommand.h File Reference

`#include "gdcmSubject.h"`

Include dependency graph for gdcmCommand.h:



This graph shows which files directly or indirectly include this file:



Classes

- class gdcm::Command

  Command superclass for callback/observer methods.
- class gdcm::MemberCommand< T >

  Command subclass that calls a pointer to a member function.
- class gdcm::SimpleMemberCommand< T >

  Command subclass that calls a pointer to a member function.

Namespaces

- namespace gdcm

11.16 gdcmCommand.h

Go to the documentation of this file.

```
1 /*=========================================================================
2
3 Program:  GDCM (Grassroots DICOM). A DICOM library
4
5 Copyright (c) 2006-2011 Mathieu Malaterre
6 All rights reserved.
7 See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.
8
9 This software is distributed WITHOUT ANY WARRANTY; without even
10 the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
11 PURPOSE.  See the above copyright notice for more information.
12
13 =========================================================================*/
14 #ifndef GDCMCOMMAND_H
15 #define GDCMCOMMAND_H
16
17 #include "gdcmSubject.h"
18
19 namespace gdcm
20 {
21 class Event;
22
27 class GDCM_EXPORT Command :  public Subject
28 {
29 public :
30   Command(const Command&) = delete;
31   void operator=(const Command&) = delete;
32
34   virtual void Execute(Subject *caller, const Event & event ) = 0;
35
40   virtual void Execute(const Subject *caller, const Event & event ) = 0;
41
42 protected:
43   Command();
44   ~Command() override;
45 };
46
54 template <class T>
55 class MemberCommand :  public Command
56 {
57 public:
58
60   typedef  void (T::*TMemberFunctionPointer)(Subject*, const Event &);
61   typedef  void (T::*TConstMemberFunctionPointer)(const Subject*,
62                                                   const Event &);
63
65   typedef MemberCommand       Self;
66   //typedef SmartPointer<Self>  Pointer;
67
68   MemberCommand(const Self&) = delete;
```

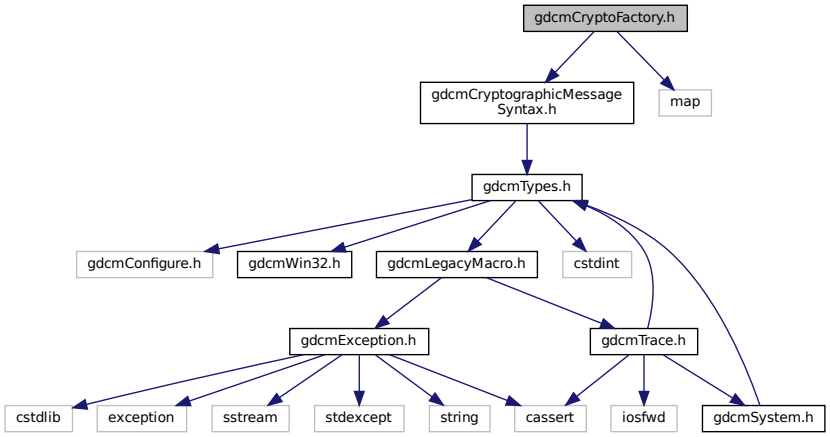
```
69   void operator=(const Self&) = delete;
70
72   static SmartPointer<MemberCommand> New()
73     {
74     return new MemberCommand;
75     }
76
78   //gdcmTypeMacro(MemberCommand,Command);
79
82   void SetCallbackFunction(T* object,
83                            TMemberFunctionPointer memberFunction)
84     {
85     m_This = object;
86     m_MemberFunction = memberFunction;
87     }
88   void SetCallbackFunction(T* object,
89                            TConstMemberFunctionPointer memberFunction)
90     {
91     m_This = object;
92     m_ConstMemberFunction = memberFunction;
93     }
94
96   void Execute(Subject *caller, const Event & event )override
97 {
98     if( m_MemberFunction )
99       {
100      ((*m_This).*(m_MemberFunction))(caller, event);
101      }
102    }
103
105   void Execute( const Subject *caller, const Event & event )override
106 {
107    if( m_ConstMemberFunction )
108      {
109      ((*m_This).*(m_ConstMemberFunction))(caller, event);
110      }
111    }
112
113 protected:
114
115   T* m_This;
116   TMemberFunctionPointer m_MemberFunction;
117   TConstMemberFunctionPointer m_ConstMemberFunction;
118   MemberCommand():m_MemberFunction(nullptr),m_ConstMemberFunction(nullptr) {}
119   ~MemberCommand() override= default;
120
121 };
122
129 template <typename T>
130 class SimpleMemberCommand :  public Command
131 {
132 public:
133
135   typedef  void (T::*TMemberFunctionPointer)();
136
138   typedef SimpleMemberCommand   Self;
139   //typedef SmartPointer<Self>    Pointer;
140
141   SimpleMemberCommand(const Self&) = delete;
142   void operator=(const Self&) = delete;
143
145   //gdcmTypeMacro(SimpleMemberCommand,Command);
146
148   static SmartPointer<SimpleMemberCommand> New()
149     {
150     return new SimpleMemberCommand;
151     }
152
154   void SetCallbackFunction(T* object,
155                            TMemberFunctionPointer memberFunction)
156     {
157     m_This = object;
158     m_MemberFunction = memberFunction;
159     }
160
162   void Execute(Subject *,const Event & )override
163 {
164    if( m_MemberFunction )
165      {
166      ((*m_This).*(m_MemberFunction))();
167      }
```

```
168     }
169   void Execute(const Subject *,const Event & )override
170 {
171     if( m_MemberFunction )
172       {
173       ((*m_This).*(m_MemberFunction))();
174       }
175     }
176
177 protected:
178   T* m_This;
179   TMemberFunctionPointer m_MemberFunction;
180   SimpleMemberCommand():m_This(nullptr),m_MemberFunction(nullptr) {}
181   ~SimpleMemberCommand() override = default;
182 };
183
184 } // end namespace gdcm
185 //----------------------------------------------------------------------------
186 #endif //GDCMCOMMAND_H
```

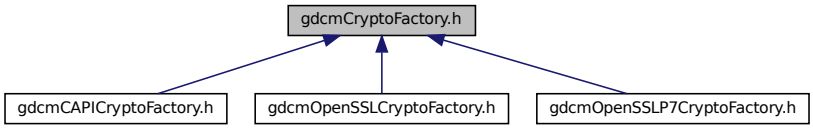
11.17 gdcmCryptoFactory.h File Reference

```
#include "gdcmCryptographicMessageSyntax.h"
#include <map>
```

Include dependency graph for gdcmCryptoFactory.h:



This graph shows which files directly or indirectly include this file:



Classes

- class gdcm::CryptoFactory

  Class to do handle the crypto factory.

Namespaces

- namespace gdcm

11.18 gdcmCryptoFactory.h

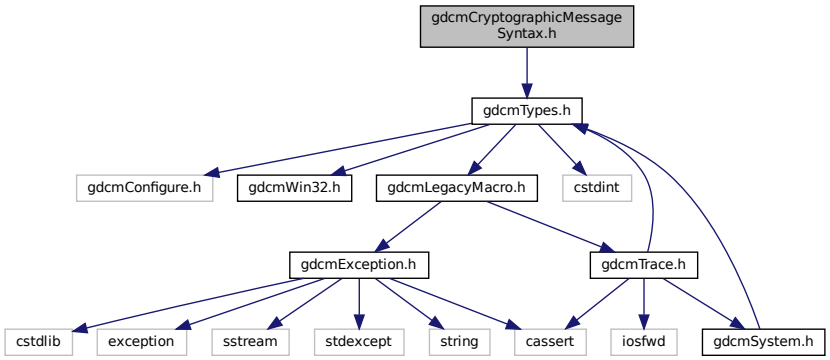
Go to the documentation of this file.

```
1 /*=========================================================================
2
3 Program:  GDCM (Grassroots DICOM). A DICOM library
4
5 Copyright (c) 2006-2011 Mathieu Malaterre
6 All rights reserved.
7 See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.
8
9 This software is distributed WITHOUT ANY WARRANTY; without even
10 the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
11 PURPOSE.  See the above copyright notice for more information.
12
13 =========================================================================*/
14 #ifndef GDCMCRYPTOFACTORY_H
15 #define GDCMCRYPTOFACTORY_H
16
17 #include "gdcmCryptographicMessageSyntax.h"
18 #include <map>
19
20 namespace gdcm
21 {
22
35 class GDCM_EXPORT CryptoFactory
36 {
37 public:
38   enum CryptoLib {DEFAULT = 0, OPENSSL = 1, CAPI = 2, OPENSSLP7 = 3};
39
40   virtual CryptographicMessageSyntax* CreateCMSProvider() = 0;
41   static CryptoFactory* GetFactoryInstance(CryptoLib id = DEFAULT);
42
43 protected:
44   CryptoFactory(CryptoLib id)
45   {
46     AddLib(id, this);
47   }
48
49 private:
50   static std::map<CryptoLib, CryptoFactory*>& getInstanceMap()
51   {
52     static std::map<CryptoLib, CryptoFactory*> libs;
53     return libs;
54   }
55
56   static void AddLib(CryptoLib id, CryptoFactory* f)
57   {
58     if (getInstanceMap().insert(std::pair<CryptoLib, CryptoFactory*>(id, f)).second == false)
59       {
60       gdcmErrorMacro( "Library already registered under id " « (int)id );
61       }
62   }
63
64 protected:
65   CryptoFactory()= default;
66   ~CryptoFactory()= default;
67 };
68
69 } // end namespace gdcm
70
71 #endif // GDCMCRYPTOFACTORY_H
```

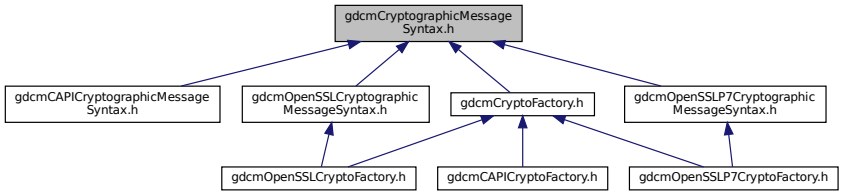
11.19 gdcmCryptographicMessageSyntax.h File Reference

```
#include "gdcmTypes.h"
```

Include dependency graph for gdcmCryptographicMessageSyntax.h:



This graph shows which files directly or indirectly include this file:



Classes

- class gdcm::CryptographicMessageSyntax

Namespaces

- namespace gdcm

11.20 gdcmCryptographicMessageSyntax.h

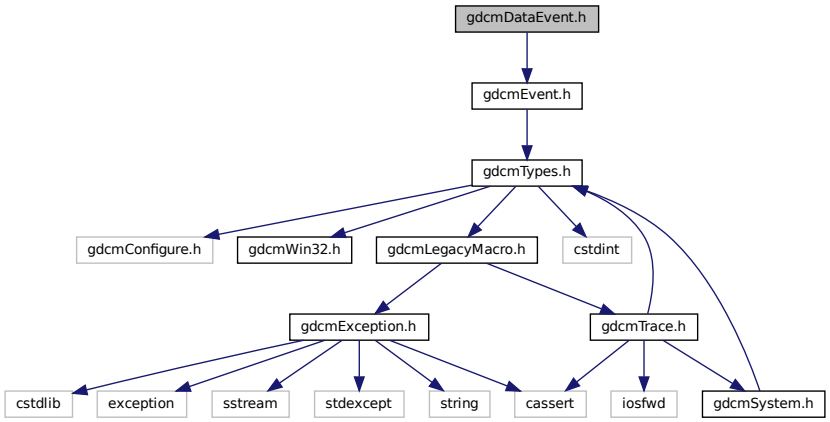
Go to the documentation of this file.

```
1 /*=========================================================================
2
3 Program:  GDCM (Grassroots DICOM). A DICOM library
4
5 Copyright (c) 2006-2011 Mathieu Malaterre
6 All rights reserved.
7 See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.
8
9 This software is distributed WITHOUT ANY WARRANTY; without even
10 the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
11 PURPOSE.  See the above copyright notice for more information.
12
13 =========================================================================*/
14 #ifndef GDCMCRYPTOGRAPHICMESSAGESYNTAX_H
15 #define GDCMCRYPTOGRAPHICMESSAGESYNTAX_H
16
17 #include "gdcmTypes.h"
18
19 namespace gdcm
20 {
21
22 class GDCM_EXPORT CryptographicMessageSyntax
23 {
24 public:
25   CryptographicMessageSyntax() = default;
26
27   virtual ~CryptographicMessageSyntax() = default;
28   CryptographicMessageSyntax(const CryptographicMessageSyntax&) = delete;
29   void operator=(const CryptographicMessageSyntax&) = delete;
30
31   typedef enum {
32     DES3_CIPHER,   // Triple DES
33     AES128_CIPHER, // CBC AES
34     AES192_CIPHER, // '   '
35     AES256_CIPHER  // '   '
36   } CipherTypes;
37
38     // X.509
39   virtual bool ParseCertificateFile( const char *filename ) = 0;
40   virtual bool ParseKeyFile( const char *filename ) = 0;
41
42   // PBE
43   virtual bool SetPassword(const char * pass, size_t passLen) = 0;
44
46   virtual bool Encrypt(char *output, size_t &outlen, const char *array, size_t len) const = 0;
48   virtual bool Decrypt(char *output, size_t &outlen, const char *array, size_t len) const = 0;
49
50   virtual void SetCipherType(CipherTypes type) = 0;
51
52   virtual CipherTypes GetCipherType() const = 0;
53 };
54
55 } // end namespace gdcm
56
57 #endif //GDCMCRYPTOGRAPHICMESSAGESYNTAX_H
```

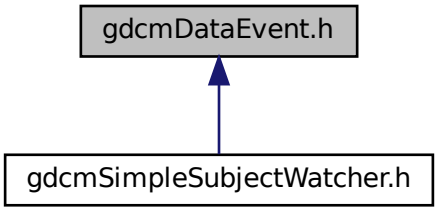
11.21 gdcmDataEvent.h File Reference

```
#include "gdcmEvent.h"
```

Include dependency graph for gdcmDataEvent.h:



This graph shows which files directly or indirectly include this file:



Classes

- class gdcm::DataEvent

  DataEvent.

Namespaces

- namespace gdcm

11.22 gdcmDataEvent.h

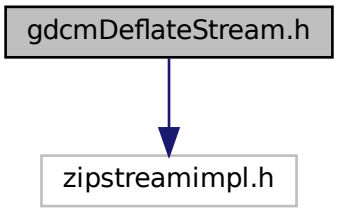
Go to the documentation of this file.

```
1 /*=========================================================================
2
3 Program:  GDCM (Grassroots DICOM). A DICOM library
4
5 Copyright (c) 2006-2011 Mathieu Malaterre
6 All rights reserved.
7 See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.
8
9 This software is distributed WITHOUT ANY WARRANTY; without even
10 the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
11 PURPOSE.  See the above copyright notice for more information.
12
13 =========================================================================*/
14 #ifndef GDCMDATAEVENT_H
15 #define GDCMDATAEVENT_H
16
17 #include "gdcmEvent.h"
18
19 namespace gdcm
20 {
21
25 class DataEvent :  public AnyEvent
26 {
27 public:
28   typedef DataEvent Self;
29   typedef AnyEvent Superclass;
30   DataEvent(const char *bytes = nullptr, size_t len = 0):Bytes(bytes),Length(len) {}
31   ~DataEvent() override = default;
32   DataEvent(const Self&s) :  AnyEvent(s){};
33   void operator=(const Self&) = delete;
34
35   const char * GetEventName()const override { return "DataEvent"; }
36   bool CheckEvent(const ::gdcm::Event* e)const override
37 { return (dynamic_cast<const Self*>(e) == nullptr ?  false :  true) ; }
38   ::gdcm::Event* MakeObject()const override
39 { return new Self; }
40
41   void SetData(const char *bytes, size_t len) {
42     Bytes = bytes;
43     Length = len;
44   }
45   size_t GetDataLength()const { return Length; }
46   const char *GetData()const { return Bytes; }
47
48   //std::string GetValueAsString() const { return; }
49
50 private:
51   const char *Bytes;
52   size_t Length;
53 };
54
55
56 } // end namespace gdcm
57
58 #endif //GDCMDATAEVENT_H
```

11.23 gdcmDeflateStream.h File Reference

```
#include "zipstreamimpl.h"
```

Include dependency graph for gdcmDeflateStream.h:



11.24 gdcmDeflateStream.h

Go to the documentation of this file.

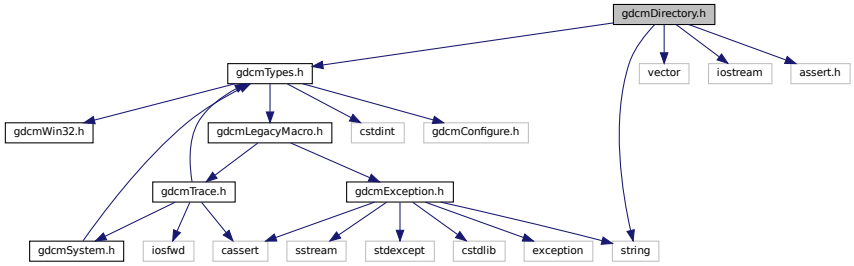
```
1 /*=========================================================================
2
3 Program:  GDCM (Grassroots DICOM). A DICOM library
4
5 Copyright (c) 2006-2011 Mathieu Malaterre
6 All rights reserved.
7 See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.
8
9 This software is distributed WITHOUT ANY WARRANTY; without even
10 the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
11 PURPOSE.  See the above copyright notice for more information.
12
13 =========================================================================*/
14 #ifndef GDCMDEFLATESTREAM_H
15 #define GDCMDEFLATESTREAM_H
16
17 #include "zipstreamimpl.h"
18
19 #endif //GDCMDEFLATESTREAM_H
```

11.25 gdcmDirectory.h File Reference

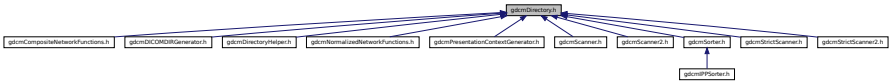
```
#include "gdcmTypes.h"
#include <string>
#include <vector>
#include <iostream>
```

```
#include <assert.h>
```

Include dependency graph for gdcmDirectory.h:



This graph shows which files directly or indirectly include this file:



Classes

- class gdcm::Directory

  Class for manipulation directories.

Namespaces

- namespace gdcm

Functions

- std::ostream & gdcm::operator<< (std::ostream &os, const Directory &d)

11.26 gdcmDirectory.h

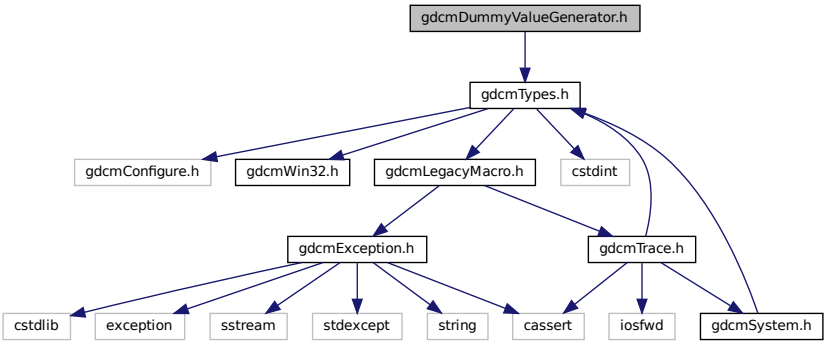
Go to the documentation of this file.

```
1 /*=========================================================================
2
3 Program:  GDCM (Grassroots DICOM). A DICOM library
4
5 Copyright (c) 2006-2011 Mathieu Malaterre
6 All rights reserved.
7 See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.
8
9 This software is distributed WITHOUT ANY WARRANTY; without even
10 the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
```

```
11 PURPOSE.  See the above copyright notice for more information.
12
13 =========================================================================*/
14 #ifndef GDCMDIRECTORY_H
15 #define GDCMDIRECTORY_H
16
17 #include "gdcmTypes.h"
18
19 #include <string>
20 #include <vector>
21 #include <iostream>
22 #include <assert.h>
23
24 namespace gdcm
25 {
41 //-----------------------------------------------------------------------------
42 class GDCM_EXPORT Directory
43 {
44   friend std::ostream& operator<<(std::ostream &_os, const Directory &d);
45 public :
46   Directory() = default;
47   ~Directory() = default;
48   typedef std::string FilenameType;
49   typedef std::vector<FilenameType> FilenamesType;
50
52   void Print(std::ostream &os = std::cout) const;
53
55   FilenameType const &GetToplevel()const { return Toplevel; }
56
58   FilenamesType const &GetFilenames()const {
59     assert( !(Toplevel.empty()) && "Need to call Explore first" );
60     return Filenames; }
61
63   FilenamesType const &GetDirectories()const { return Directories; }
64
67   unsigned int Load(FilenameType const &name, bool recursive = false);
68
69   // \todo later:  GLOB
70   // The glob() function searches for all the pathnames matching pattern according to
71   // the rules used by the shell (see glob(7)).    No tilde expansion or parameter
72   // substitution is done; if you want these, use wordexp(3).
73   // int Glob(...);
74
75 protected:
77   unsigned int Explore(FilenameType const &name, bool recursive);
78
79 private :
81   FilenamesType Filenames;
82   FilenamesType Directories;
83
85   FilenameType Toplevel;
86 };
87 //-----------------------------------------------------------------------------
88 inline std::ostream& operator<<(std::ostream &os, const Directory &d)
89 {
90   d.Print( os );
91   return os;
92 }
93
94 } // end namespace gdcm
95 //-----------------------------------------------------------------------------
96 #endif //GDCMDIRECTORY_H
```

11.27 gdcmDummyValueGenerator.h File Reference

`#include "gdcmTypes.h"`
Include dependency graph for gdcmDummyValueGenerator.h:



Classes

- class gdcm::DummyValueGenerator

  Class for generating dummy value.

Namespaces

- namespace gdcm

11.28 gdcmDummyValueGenerator.h

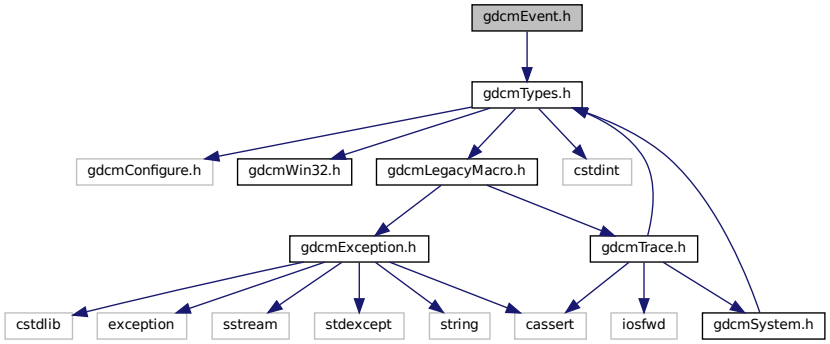
Go to the documentation of this file.

```
1 /*=========================================================================
2
3 Program:  GDCM (Grassroots DICOM). A DICOM library
4
5 Copyright (c) 2006-2011 Mathieu Malaterre
6 All rights reserved.
7 See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.
8
9 This software is distributed WITHOUT ANY WARRANTY; without even
10 the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
11 PURPOSE.  See the above copyright notice for more information.
12
13 =========================================================================*/
14 #ifndef GDCMDUMMYVALUEGENERATOR_H
15 #define GDCMDUMMYVALUEGENERATOR_H
16
17 #include "gdcmTypes.h"
18
19 namespace gdcm
20 {
21
26 class GDCM_EXPORT DummyValueGenerator
```

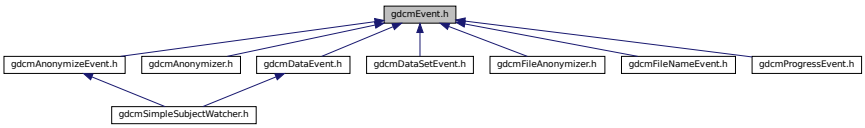
```
27 {
28 public:
29
35   static const char* Generate(const char *input);
36
37 private:
38 };
39
40
41 } // end namespace gdcm
42
43 #endif //GDCMDUMMYVALUEGENERATOR_H
```

11.29 gdcmEvent.h File Reference

`#include "gdcmTypes.h"`
Include dependency graph for gdcmEvent.h:



This graph shows which files directly or indirectly include this file:



Classes

- class gdcm::AbortEvent
- class gdcm::AnyEvent
- class gdcm::EndEvent
- class gdcm::Event

  superclass for callback/observer methods
- class gdcm::ExitEvent

- class gdcm::InitializeEvent
- class gdcm::IterationEvent
- class gdcm::ModifiedEvent
- class gdcm::NoEvent
- class gdcm::StartEvent
- class gdcm::UserEvent

Namespaces

- namespace gdcm

Macros

- #define gdcmEventMacro(classname, super)

Functions

- std::ostream & gdcm::operator<< (std::ostream &os, Event &e)

  Generic inserter operator for Event and its subclasses.

11.29.1 Macro Definition Documentation

11.29.1.1 gdcmEventMacro

```
#define gdcmEventMacro(
              classname,
              super )
```

Value:

```
  \
 class  classname :  public super { \
   public:  \
     typedef classname Self; \
     typedef super Superclass; \
     classname() {} \
     virtual ~classname() override = default; \
     virtual const char * GetEventName()const override { return #classname; } \
     virtual bool CheckEvent(const ::gdcm::Event* e) const override \
       { return dynamic_cast<const Self*>(e) ?  true :  false; } \
     virtual ::gdcm::Event* MakeObject() const override \
       { return new Self; } \
     classname(const Self&s) :  super(s){}; \
   private:  \
     void operator=(const Self&); \
 }
```

11.30 gdcmEvent.h

Go to the documentation of this file.

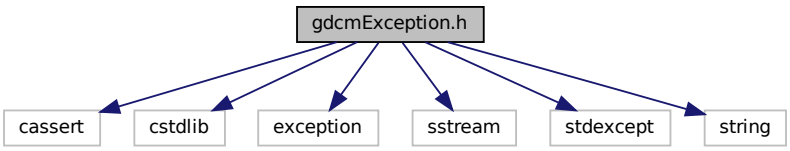
```
1 /*=========================================================================
2
3 Program:  GDCM (Grassroots DICOM). A DICOM library
4
5 Copyright (c) 2006-2011 Mathieu Malaterre
6 All rights reserved.
7 See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.
8
9 This software is distributed WITHOUT ANY WARRANTY; without even
10 the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
11 PURPOSE.  See the above copyright notice for more information.
12
13 =========================================================================*/
14 #ifndef GDCMEVENT_H
15 #define GDCMEVENT_H
16
17 #include "gdcmTypes.h"
18
19 namespace gdcm
20 {
21 //-----------------------------------------------------------------------------
26 class GDCM_EXPORT Event
27 {
28 public :
29   Event();
30   virtual ~Event();
31   Event(const Event&);
32   void operator=(const Event&) = delete;
33
36   virtual Event* MakeObject() const = 0;
37
41   virtual void Print(std::ostream& os) const;
42
44   virtual const char * GetEventName() const = 0;
45
47   virtual bool CheckEvent(const Event*) const = 0;
48
49 };
50
52 inline std::ostream& operator<<(std::ostream& os, Event &e)
53 {
54   e.Print(os);
55   return os;
56 }
57
58 /*
59 *  Macro for creating new Events
60 */
61 #define gdcmEventMacro( classname , super ) \
62 \
63 class  classname :  public super { \
64 public:  \
65 typedef classname Self; \
66 typedef super Superclass; \
67 classname() {} \
68 virtual ~classname() override = default; \
69 virtual const char * GetEventName() const override { return #classname; } \
70 virtual bool CheckEvent(const ::gdcm::Event* e) const override \
71 { return dynamic_cast<const Self*>(e) ?  true :  false; } \
72 virtual ::gdcm::Event* MakeObject() const override \
73 { return new Self; } \
74 classname(const Self&s) :  super(s){}; \
75 private:  \
76 void operator=(const Self&); \
77 }
78
82 gdcmEventMacro( NoEvent            , Event );
83 gdcmEventMacro( AnyEvent           , Event );
84 gdcmEventMacro( StartEvent         , AnyEvent );
85 gdcmEventMacro( EndEvent           , AnyEvent );
86 //gdcmEventMacro( ProgressEvent      , AnyEvent );
87 gdcmEventMacro( ExitEvent          , AnyEvent );
88 gdcmEventMacro( AbortEvent         , AnyEvent );
89 gdcmEventMacro( ModifiedEvent      , AnyEvent );
90 gdcmEventMacro( InitializeEvent    , AnyEvent );
91 gdcmEventMacro( IterationEvent     , AnyEvent );
```

```
92 //gdcmEventMacro( AnonymizeEvent     , AnyEvent );
93 gdcmEventMacro( UserEvent          , AnyEvent );
94
95
96 } // end namespace gdcm
97 //----------------------------------------------------------------------------
98 #endif //GDCMEVENT_H
```

11.31 gdcmException.h File Reference

```
#include <cassert>
#include <cstdlib>
#include <exception>
#include <sstream>
#include <stdexcept>
#include <string>
```

Include dependency graph for gdcmException.h:



This graph shows which files directly or indirectly include this file:

Classes

- class gdcm::Exception

  Exception.

Namespaces

- namespace gdcm

11.32 gdcmException.h

Go to the documentation of this file.

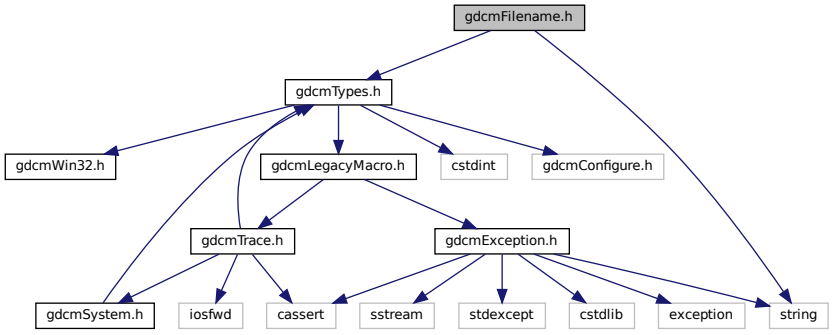
```
1 /*=========================================================================
2
3 Program:  GDCM (Grassroots DICOM). A DICOM library
4
5 Copyright (c) 2006-2011 Mathieu Malaterre
6 All rights reserved.
7 See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.
8
9 This software is distributed WITHOUT ANY WARRANTY; without even
10 the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
11 PURPOSE.  See the above copyright notice for more information.
12
13 =========================================================================*/
14 #ifndef GDCMEXCEPTION_H
15 #define GDCMEXCEPTION_H
16
17 #include <cassert>
18 #include <cstdlib> // NULL
19 #include <exception>
20 #include <sstream> // ostringstream
21 #include <stdexcept> // logic_error
22 #include <string>
23
24 // Disable clang warning "dynamic exception specifications are deprecated".
25 // We need to be C++03 and C++11 compatible, and if we remove the 'throw()'
26 // specifier we'll get an error in C++03 by not matching the superclass.
27 #if defined(__clang__) && defined(__has_warning)
28 # if __has_warning("-Wdeprecated")
29 #  pragma clang diagnostic push
30 #  pragma clang diagnostic ignored "-Wdeprecated"
31 # endif
32 #endif
33
34 namespace gdcm
35 {
36
43 class Exception :  public std::exception
44 {
49   typedef std::logic_error StringHolder;
50
52   static StringHolder CreateWhat(const char* const desc,
53                                  const char* const file,
54                                  const unsigned int lineNumber,
55                                  const char* const func)
56   {
57     assert(desc != nullptr);
58     assert(file != nullptr);
59     assert(func != nullptr);
60     std::ostringstream oswhat;
61     oswhat « file « ":" « lineNumber « " (" « func « "):\n";
62     oswhat « desc;
63     return StringHolder( oswhat.str() );
64   }
65
66 public:
72   explicit Exception(const char *desc = "None",
73              const char *file = __FILE__,
74              unsigned int lineNumber = __LINE__,
75              // FIXME:  __PRETTY_FUNCTION__ is the non-mangled version for __GNUC__ compiler
76              const char *func = "" /*__FUNCTION__*/)
77   :
78   What( CreateWhat(desc, file, lineNumber, func) ),
79   Description(desc)
80   {
81   }
82
83   ~Exception() throw()override {}
84
86   const char* what() const throw()override
87 {
88     return What.what();
89     }
90
92   const char * GetDescription()const { return Description.what(); }
93
94 private:
```

```
95   StringHolder  What;
96   StringHolder  Description;
97 };
98
99 } // end namespace gdcm
100
101 // Undo warning suppression.
102 #if defined(__clang__) && defined(__has_warning)
103 # if __has_warning("-Wdeprecated")
104 #  pragma clang diagnostic pop
105 # endif
106 #endif
107
108 #endif
```

11.33 gdcmFilename.h File Reference

```
#include "gdcmTypes.h"
#include <string>
```

Include dependency graph for gdcmFilename.h:



Classes

- class gdcm::Filename

  Class to manipulate file name's.

Namespaces

- namespace gdcm

11.34 gdcmFilename.h

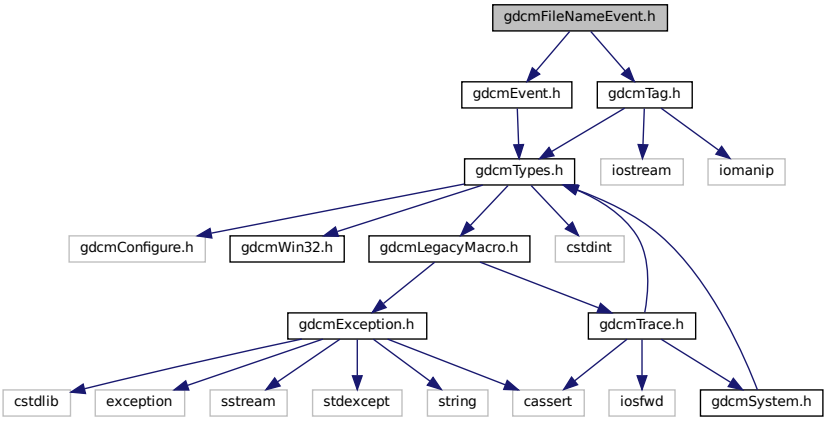
Go to the documentation of this file.

```
1 /*=========================================================================
2
3 Program:  GDCM (Grassroots DICOM). A DICOM library
4
5 Copyright (c) 2006-2011 Mathieu Malaterre
6 All rights reserved.
7 See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.
8
9 This software is distributed WITHOUT ANY WARRANTY; without even
10 the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
11 PURPOSE.  See the above copyright notice for more information.
12
13 =========================================================================*/
14 #ifndef GDCMFILENAME_H
15 #define GDCMFILENAME_H
16
17 #include "gdcmTypes.h"
18
19 #include <string>
20
21 namespace gdcm
22 {
27 class GDCM_EXPORT Filename
28 {
29 public:
30   Filename(const char* filename = ""):FileName(filename ?  filename :  ""),Path(),Conversion() {}
31
33   const char *GetFileName()const { return FileName.c_str(); }
35   const char *GetPath();
37   const char *GetName();
39   const char *GetExtension();
41   const char *ToUnixSlashes();
43   const char *ToWindowsSlashes();
44
47   static const char *Join(const char *path, const char *filename);
48
50   bool IsEmpty()const { return FileName.empty(); }
51
55   operator const char * () const { return GetFileName(); }
56
57   // FIXME: I don't like this function
58   // It hides the realpath call (maybe useful)
59   // and it forces file to exist on the disk whereas Filename
60   // should be independent from file existence.
61   bool IsIdentical(Filename const &fn) const;
62
64   bool EndWith(const char ending[]) const;
65
66 private:
67   std::string FileName;
68   std::string Path;
69   std::string Conversion;
70 };
71
72 } // end namespace gdcm
73
74 #endif //GDCMFILENAME_H
```

11.35 gdcmFileNameEvent.h File Reference

```
#include "gdcmEvent.h"
#include "gdcmTag.h"
```

Include dependency graph for gdcmFileNameEvent.h:



Classes

- class gdcm::FileNameEvent

  FileNameEvent.

Namespaces

- namespace gdcm

11.36 gdcmFileNameEvent.h

Go to the documentation of this file.

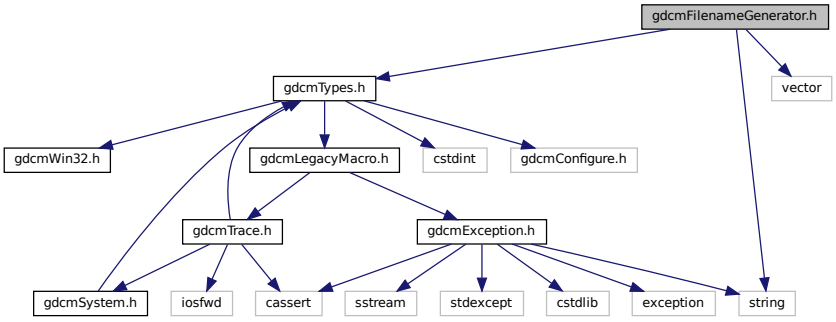
```
1 /*=========================================================================
2
3 Program:  GDCM (Grassroots DICOM). A DICOM library
4
5 Copyright (c) 2006-2011 Mathieu Malaterre
6 All rights reserved.
7 See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.
8
9 This software is distributed WITHOUT ANY WARRANTY; without even
10 the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
11 PURPOSE.  See the above copyright notice for more information.
12
13 =========================================================================*/
14 #ifndef GDCMFILENAMEEVENT_H
15 #define GDCMFILENAMEEVENT_H
16
17 #include "gdcmEvent.h"
18 #include "gdcmTag.h"
19
20 namespace gdcm
21 {
22
29 class FileNameEvent :  public AnyEvent
30 {
```

```
31 public:
32   typedef FileNameEvent Self;
33   typedef AnyEvent Superclass;
34   FileNameEvent(const char *s = ""):m_FileName(s) {}
35   ~FileNameEvent() override = default;
36
37   FileNameEvent(const Self&s) :  AnyEvent(s){};
38   void operator=(const Self&) = delete;
39
40
41   const char * GetEventName()const override { return "FileNameEvent"; }
42   bool CheckEvent(const ::gdcm::Event* e)const override
43 { return dynamic_cast<const Self*>(e) ?  true :  false; }
44   ::gdcm::Event* MakeObject()const override
45 { return new Self; }
46
47   void SetFileName(const char *f) { m_FileName = f; }
48   const char *GetFileName()const { return m_FileName.c_str(); }
49 private:
50   std::string m_FileName;
51 };
52
53
54 } // end namespace gdcm
55
56 #endif //GDCMFILENAMEEVENT_H
```

11.37 gdcmFilenameGenerator.h File Reference

```
#include "gdcmTypes.h"
#include <string>
#include <vector>
```

Include dependency graph for gdcmFilenameGenerator.h:



Classes

- class gdcm::FilenameGenerator

  FilenameGenerator.

Namespaces

- namespace gdcm

11.38 gdcmFilenameGenerator.h

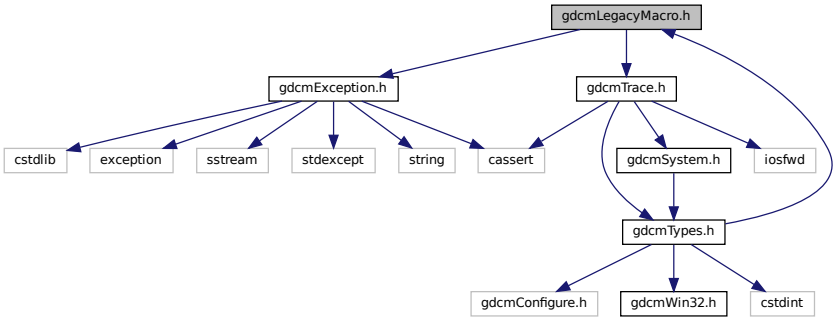
Go to the documentation of this file.

```
1 /*=========================================================================
2
3 Program:  GDCM (Grassroots DICOM). A DICOM library
4
5 Copyright (c) 2006-2011 Mathieu Malaterre
6 All rights reserved.
7 See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.
8
9 This software is distributed WITHOUT ANY WARRANTY; without even
10 the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
11 PURPOSE.  See the above copyright notice for more information.
12
13 =========================================================================*/
14 #ifndef GDCMFILENAMEGENERATOR_H
15 #define GDCMFILENAMEGENERATOR_H
16
17 #include "gdcmTypes.h"
18 #include <string>
19 #include <vector>
20
21
22 namespace gdcm
23 {
36 class GDCM_EXPORT FilenameGenerator
37 {
38 public:
39   FilenameGenerator():Pattern(),Prefix(),Filenames() {}
40   ~FilenameGenerator() = default;
41   // FIXME: already defines in gdcm::Directory
42   typedef std::string FilenameType;
43   typedef std::vector<FilenameType> FilenamesType;
44   typedef FilenamesType::size_type  SizeType;
45
47   void SetPattern(const char *pattern) { Pattern = pattern; }
48   const char *GetPattern()const { return Pattern.c_str(); }
49
51   void SetPrefix(const char *prefix) { Prefix = prefix; }
52   const char *GetPrefix()const { return Prefix.c_str(); }
53
55   bool Generate();
56
58   void SetNumberOfFilenames(SizeType nfiles);
59   SizeType GetNumberOfFilenames() const;
60
62   const char * GetFilename(SizeType n) const;
63
65   FilenamesType const & GetFilenames()const { assert( !Pattern.empty() ); return Filenames; }
66
67 private:
68   FilenameType Pattern;
69   FilenameType Prefix;
70   FilenamesType Filenames;
71 };
72
73 } // end namespace gdcm
74
75 #endif //GDCMFILENAMEGENERATOR_H
```

11.39 gdcmLegacyMacro.h File Reference

```
#include "gdcmException.h"
#include "gdcmTrace.h"
```

Include dependency graph for gdcmLegacyMacro.h:



This graph shows which files directly or indirectly include this file:

Macros

- #define GDCM_LEGACY(method) method;
- #define GDCM_LEGACY_BODY(method, version) gdcmWarningMacro(#method " was deprecated for " version " and will be removed in a future version.")
- #define GDCM_LEGACY_REPLACED_BODY(method, version, replace) gdcmWarningMacro(#method " was deprecated for " version " and will be removed in a future version. Use " #replace " instead.")
- #define GDCM_NOOP_STATEMENT static_assert(true, "")

11.39.1 Macro Definition Documentation

11.39.1.1 GDCM_LEGACY

```
#define GDCM_LEGACY(
              method ) method;
```

11.39.1.2 GDCM_LEGACY_BODY

```
#define GDCM_LEGACY_BODY(
              method,
              version )  gdcmWarningMacro(#method " was deprecated for " version " and will be
removed in a future version.")
```

11.39.1.3 GDCM_LEGACY_REPLACED_BODY

```
#define GDCM_LEGACY_REPLACED_BODY(
              method,
              version,
              replace )  gdcmWarningMacro(#method " was deprecated for " version " and will be
removed in a future version.  Use " #replace " instead.")
```

11.39.1.4 GDCM_NOOP_STATEMENT

```
#define GDCM_NOOP_STATEMENT static_assert(true, "")
```

The `static_assert(true, "")` idiom is commonly employed for C++11 or greater to ensure that it is compile-time only check that can not be part of the binary file. This allows a macro to be used anywhere that a statement is expected, and to enforce consistent use of ; after a macro. The static_assert is a constexpr that can be used in places where raw statements (i.e. 'do{} while(0)') are not allowed (i.e. after class member function definitions).

11.40 gdcmLegacyMacro.h

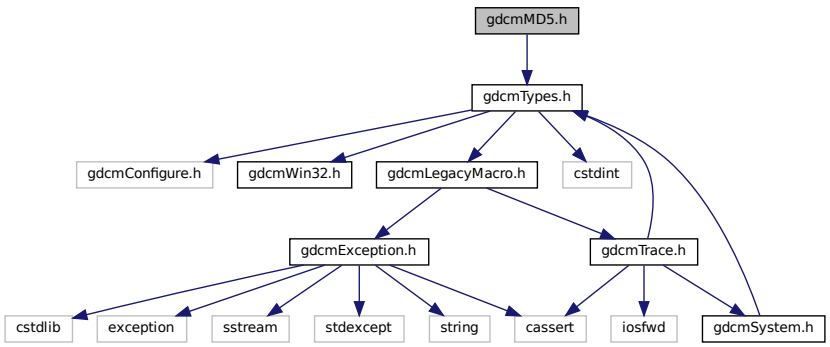
Go to the documentation of this file.

```
1 /*=========================================================================
2
3 Program:  GDCM (Grassroots DICOM). A DICOM library
4
5 Copyright (c) 2006-2011 Mathieu Malaterre
6 All rights reserved.
7 See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.
8
9 This software is distributed WITHOUT ANY WARRANTY; without even
10 the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
11 PURPOSE.  See the above copyright notice for more information.
12
13 =========================================================================*/
14 #ifndef GDCMLEGACYMACRO_H
15 #define GDCMLEGACYMACRO_H
16
17 #if !defined(GDCMTYPES_H) && !defined(SWIG)
18 #error you need to include gdcmTypes.h instead
19 #endif
20
21 #include "gdcmException.h"
22
23 //-----------------------------------------------------------------------------
24 // Setup legacy code policy.
25
26 // Define GDCM_LEGACY macro to mark legacy methods where they are
27 // declared in their class.    Example usage:
28 //
29 //   // @deprecated Replaced by MyOtherMethod() as of GDCM 2.0.
30 //   GDCM_LEGACY(void MyMethod());
31 #if defined(GDCM_LEGACY_REMOVE)
32 # define GDCM_LEGACY(method)
33 #elif defined(GDCM_LEGACY_SILENT) || defined(SWIG)
34   // Provide legacy methods with no warnings.
35 # define GDCM_LEGACY(method) method;
36 #else
37   // Setup compile-time warnings for uses of deprecated methods if
38   // possible on this compiler.
39 # if defined(__GNUC__) && !defined(__INTEL_COMPILER) && (__GNUC__ > 3 || (__GNUC__ == 3 && __GNUC_MINOR__ >=
      1))
40 #  define GDCM_LEGACY(method) method __attribute__((deprecated));
```

```
41 # elif defined(_MSC_VER) && _MSC_VER >= 1300
42 #  define GDCM_LEGACY(method) __declspec(deprecated) method;
43 # else
44 #  define GDCM_LEGACY(method) method;
45 # endif
46 #endif
47
57 #  define GDCM_NOOP_STATEMENT static_assert(true, "")
58
59 // Macros to create runtime deprecation warning messages in function
60 // bodies.    Example usage:
61 //
62 //   #if !defined(GDCM_LEGACY_REMOVE)
63 //   void gdcm::MyClass::MyOldMethod()
64 //   {
65 //     GDCM_LEGACY_BODY(gdcm::MyClass::MyOldMethod, "GDCM 2.0");
66 //   }
67 //   #endif
68 //
69 //   #if !defined(GDCM_LEGACY_REMOVE)
70 //   void gdcm::MyClass::MyMethod()
71 //   {
72 //     GDCM_LEGACY_REPLACED_BODY(gdcm::MyClass::MyMethod, "GDCM 2.0",
73 //                               gdcm::MyClass::MyOtherMethod);
74 //   }
75 //   #endif
76 #if defined(GDCM_LEGACY_REMOVE) || defined(GDCM_LEGACY_SILENT)
77 # define GDCM_LEGACY_BODY(method, version)
78 # define GDCM_LEGACY_REPLACED_BODY(method, version, replace)
79 #else
80 # define GDCM_LEGACY_BODY(method, version) \
81 gdcmWarningMacro(#method " was deprecated for " version " and will be removed in a future version.")
82 # define GDCM_LEGACY_REPLACED_BODY(method, version, replace) \
83 gdcmWarningMacro(#method " was deprecated for " version " and will be removed in a future version.    Use "
      #replace " instead.")
84 #endif
85
86 #include "gdcmTrace.h"
87
88 #endif // GDCMLEGACYMACRO_H
```

11.41 gdcmMD5.h File Reference

`#include "gdcmTypes.h"`
Include dependency graph for gdcmMD5.h:



Classes

- class gdcm::MD5

Class for MD5.

Namespaces

- namespace gdcm

11.42 gdcmMD5.h

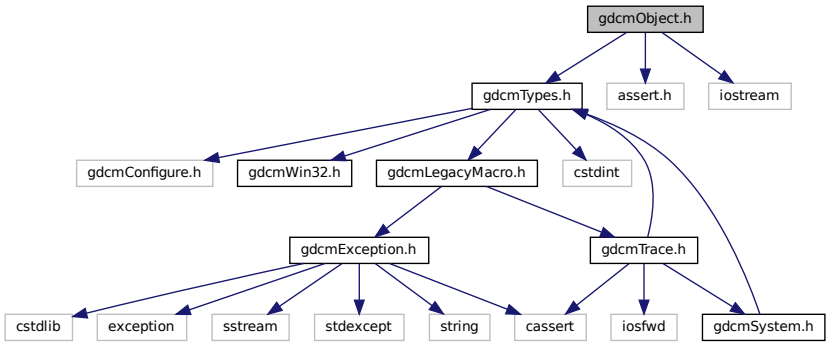
Go to the documentation of this file.

```
1 /*=========================================================================
2
3 Program:  GDCM (Grassroots DICOM). A DICOM library
4
5 Copyright (c) 2006-2011 Mathieu Malaterre
6 All rights reserved.
7 See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.
8
9 This software is distributed WITHOUT ANY WARRANTY; without even
10 the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
11 PURPOSE.  See the above copyright notice for more information.
12
13 =========================================================================*/
14 #ifndef GDCMMD5_H
15 #define GDCMMD5_H
16
17 #include "gdcmTypes.h"
18
19 namespace gdcm
20 {
21 //-----------------------------------------------------------------------------
32 class GDCM_EXPORT MD5
33 {
34 public :
35   // Compute md5 from memory pointed by `pointer` of size `buf_len`
36   static bool Compute(const char *buffer, size_t buf_len, char digest_str[33]);
37
39   static bool ComputeFile(const char *filename, char digest_str[33]);
40 };
41 } // end namespace gdcm
42 //-----------------------------------------------------------------------------
43 #endif //GDCMMD5_H
```

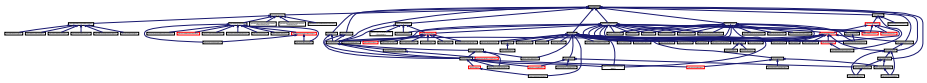
11.43 gdcmObject.h File Reference

```
#include "gdcmTypes.h"
#include <assert.h>
#include <iostream>
```

Include dependency graph for gdcmObject.h:



This graph shows which files directly or indirectly include this file:



Classes

- class gdcm::Object

  Object.

Namespaces

- namespace gdcm

Functions

- std::ostream & gdcm::operator<< (std::ostream &os, const Object &obj)

11.44 gdcmObject.h

Go to the documentation of this file.

```
1 /*=========================================================================
2
3 Program:  GDCM (Grassroots DICOM). A DICOM library
4
5 Copyright (c) 2006-2011 Mathieu Malaterre
6 All rights reserved.
7 See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.
8
9 This software is distributed WITHOUT ANY WARRANTY; without even
```

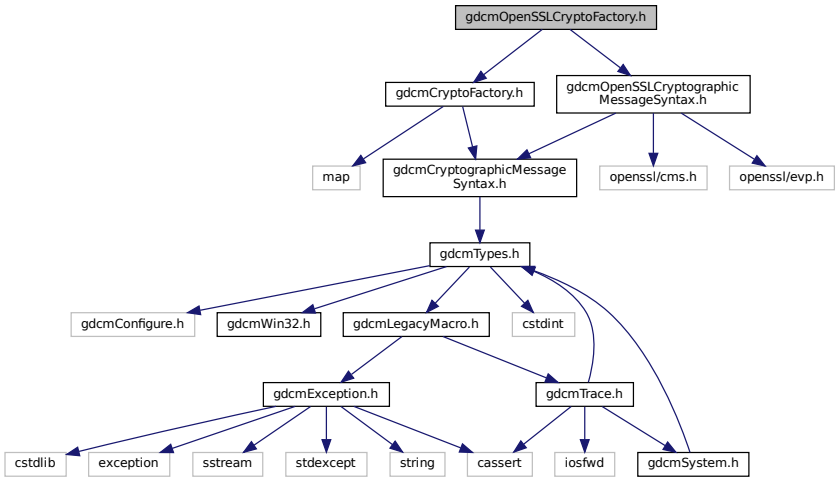
```
10 the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
11 PURPOSE.  See the above copyright notice for more information.
12
13 =========================================================================*/
14 #ifndef GDCMOBJECT_H
15 #define GDCMOBJECT_H
16
17 #include "gdcmTypes.h"
18
19 #include <assert.h>
20 #include <iostream> // grrrr
21
22 //namespace std { class ostream; }
23 namespace gdcm
24 {
25
26 template<class ObjectType> class SmartPointer;
27
36 class GDCM_EXPORT Object
37 {
38   template <class ObjectType> friend class SmartPointer;
39   friend std::ostream& operator<<(std::ostream &os, const Object &obj);
40
41 public:
42   Object():ReferenceCount(0) {}
43
44   // Implementation note:
45   // If I move ~Object in the protected section I can prevent people
46   // from writing:
47   // SmartPointer<Object> p = new Object;
48   // delete p; // due to SmartPointer::operator ObjectType * () const
49   // but on the other hand one could not define an Object on the stack
50   // Object obj;
51   // Furthermore it would not prevent anyone from doing:
52   // class MyObject :  public Object {};
53   // SmartPointer<MyObject> o = new MyObject;
54   // delete o; // grrrrrr
55   virtual ~Object() {
56     // If your debugger reach here it means you are doing something silly
57     // like using SmartPointer on object allocated on the stack (vs heap)
58     assert(ReferenceCount == 0);
59     }
60
61   // http://www.parashift.com/c++-faq-lite/freestore-mgmt.html#faq-16.24
62   // Set the ref count to 0
63   // Do NOT copy the reference count !
64   Object(const Object&):ReferenceCount(0){}
65   void operator=(const Object&){}
66
67   //static Object* New() { return new Object; }
68
69 protected:
70   // For the purpose of the invasive SmartPointer implementation
71   void Register() {
72     ReferenceCount++;
73     assert( ReferenceCount > 0 );
74   }
75   void UnRegister() {
76     assert( ReferenceCount > 0 );
77     ReferenceCount--;
78     if(!ReferenceCount)
79       {
80       delete this;
81       }
82     }
83
84 public:
85   // For dealing with printing of object and polymorphism
86   virtual void Print(std::ostream &)const {}
87
88 private:
89   long ReferenceCount;
90 };
91
92 //-----------------------------------------------------------------------------
93 // function do not carry vtable.  Thus define in the base class the operator
94 // and use the member function ->Print() to call the appropriate function
95 // NOTE: All subclass of Object needs to implement the Print function
96 inline std::ostream& operator<<(std::ostream &os, const Object &obj)
97 {
98   obj.Print(os);
99
```

```
100   return os;
101 }
102
103 } // end namespace gdcm
104
105 #endif //GDCMOBJECT_H
```

11.45 gdcmOpenSSLCryptoFactory.h File Reference

```
#include "gdcmCryptoFactory.h"
#include "gdcmOpenSSLCryptographicMessageSyntax.h"
```

Include dependency graph for gdcmOpenSSLCryptoFactory.h:



Classes

- class gdcm::OpenSSLCryptoFactory

Namespaces

- namespace gdcm

11.46 gdcmOpenSSLCryptoFactory.h

Go to the documentation of this file.

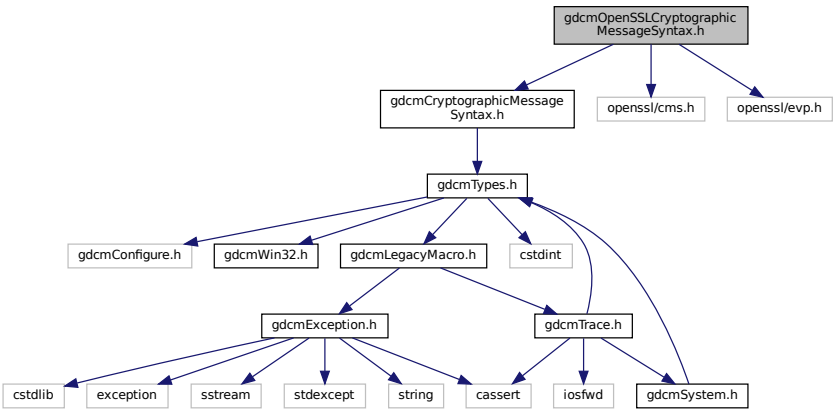
```
1 /*=========================================================================
2
3 Program:  GDCM (Grassroots DICOM). A DICOM library
4
5 Copyright (c) 2006-2011 Mathieu Malaterre
6 All rights reserved.
```

```
7 See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.
8
9 This software is distributed WITHOUT ANY WARRANTY; without even
10 the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
11 PURPOSE.  See the above copyright notice for more information.
12
13 =========================================================================*/
14 #ifndef GDCMOPENSSLCRYPTOFACTORY_H
15 #define GDCMOPENSSLCRYPTOFACTORY_H
16
17 #include "gdcmCryptoFactory.h"
18 #include "gdcmOpenSSLCryptographicMessageSyntax.h"
19
20 namespace gdcm
21 {
22
23 class GDCM_EXPORT OpenSSLCryptoFactory :  public CryptoFactory
24 {
25 public:
26   OpenSSLCryptoFactory(CryptoLib id) :  CryptoFactory(id)
27   {
28     gdcmDebugMacro( "OpenSSL Factory registered." );
29   }
30
31 public:
32   CryptographicMessageSyntax* CreateCMSProvider()
33   {
34     InitOpenSSL();
35     return new OpenSSLCryptographicMessageSyntax();
36   }
37
38 protected:
39   void InitOpenSSL();
40
41 private:
42   OpenSSLCryptoFactory(){}
43 };
44
45 } // end namespace gdcm
46
47 #endif //GDCMOPENSSLCRYPTOFACTORY_H
```

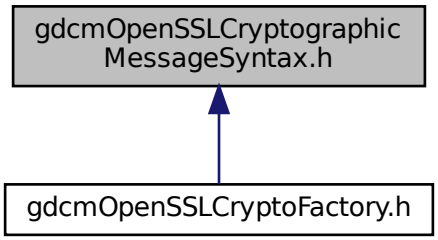
11.47 gdcmOpenSSLCryptographicMessageSyntax.h File Reference

```
#include "gdcmCryptographicMessageSyntax.h"
#include <openssl/cms.h>
#include <openssl/evp.h>
```

Include dependency graph for gdcmOpenSSLCryptographicMessageSyntax.h:



This graph shows which files directly or indirectly include this file:



Classes

- class gdcm::OpenSSLCryptographicMessageSyntax

Namespaces

- namespace gdcm

11.48 gdcmOpenSSLCryptographicMessageSyntax.h

Go to the documentation of this file.

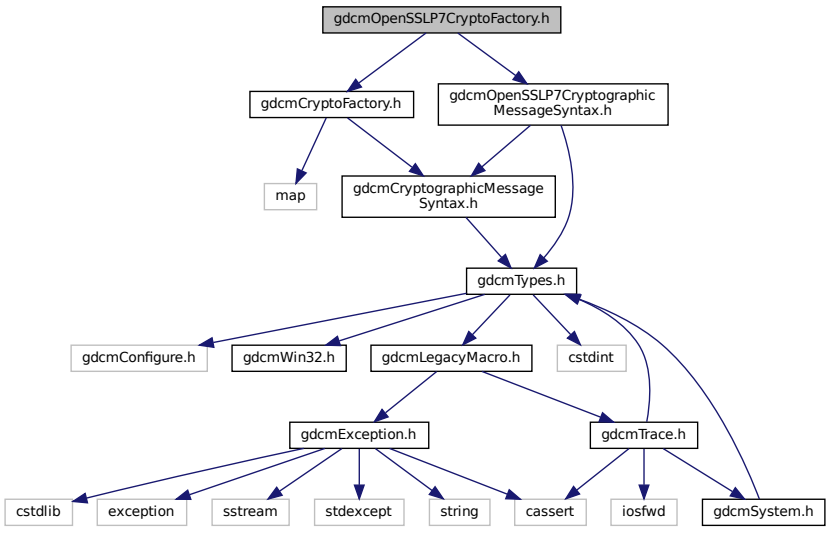
```
1 /*=========================================================================
2
3 Program:  GDCM (Grassroots DICOM). A DICOM library
4
5 Copyright (c) 2006-2011 Mathieu Malaterre
6 All rights reserved.
7 See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.
8
9 This software is distributed WITHOUT ANY WARRANTY; without even
10 the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
11 PURPOSE.  See the above copyright notice for more information.
12
13 =========================================================================*/
14 #ifndef GDCMOPENSSLCRYPTOGRAPHICMESSAGESYNTAX_H
15 #define GDCMOPENSSLCRYPTOGRAPHICMESSAGESYNTAX_H
16
17 #include "gdcmCryptographicMessageSyntax.h"
18 #include <openssl/cms.h>
19 #include <openssl/evp.h>
20
21 namespace gdcm
22 {
23
24 class GDCM_EXPORT OpenSSLCryptographicMessageSyntax :  public CryptographicMessageSyntax
25 {
26 public:
27   OpenSSLCryptographicMessageSyntax();
28   ~OpenSSLCryptographicMessageSyntax();
29
30   // X.509
```

```
31   bool ParseCertificateFile( const char *filename );
32   bool ParseKeyFile( const char *filename );
33
34   // PBE
35   bool SetPassword(const char * pass, size_t passLen);
36
39   void SetCipherType(CipherTypes type);
40   CipherTypes GetCipherType() const;
42   bool Encrypt(char *output, size_t &outlen, const char *array, size_t len) const;
44   bool Decrypt(char *output, size_t &outlen, const char *array, size_t len) const;
45
46 private:
47 //#ifdef GDCM_HAVE_CMS_RECIPIENT_PASSWORD
48 //  ::stack_st_X509 *recips;
49 //#else
50   STACK_OF(X509) *recips;
51 //#endif
52   ::EVP_PKEY *pkey;
53   const EVP_CIPHER *internalCipherType;
54   char * password;
55   size_t passwordLength;
56   CipherTypes cipherType;
57
58 private:
59   OpenSSLCryptographicMessageSyntax(const OpenSSLCryptographicMessageSyntax&);  // Not implemented.
60   void operator=(const OpenSSLCryptographicMessageSyntax&);  // Not implemented.
61   const EVP_CIPHER *CreateCipher( CryptographicMessageSyntax::CipherTypes ciphertype);
62
63 };
64
65 } // end namespace gdcm
66
67 #endif //GDCMOPENSSLCRYPTOGRAPHICMESSAGESYNTAX_H
```

11.49 gdcmOpenSSLP7CryptoFactory.h File Reference

```
#include "gdcmCryptoFactory.h"
#include "gdcmOpenSSLP7CryptographicMessageSyntax.h"
```

Include dependency graph for gdcmOpenSSLP7CryptoFactory.h:



Classes

- class gdcm::OpenSSLP7CryptoFactory

Namespaces

- namespace gdcm

11.50 gdcmOpenSSLP7CryptoFactory.h

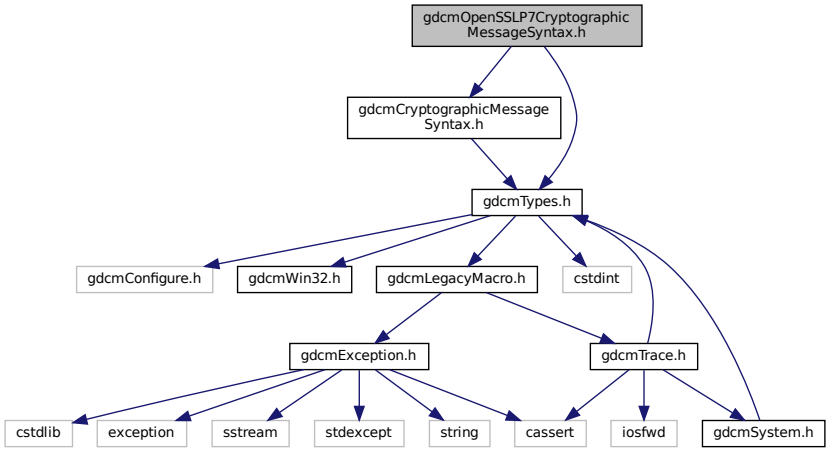
Go to the documentation of this file.

```
1 /*=========================================================================
2
3 Program:  GDCM (Grassroots DICOM). A DICOM library
4
5 Copyright (c) 2006-2011 Mathieu Malaterre
6 All rights reserved.
7 See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.
8
9 This software is distributed WITHOUT ANY WARRANTY; without even
10 the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
11 PURPOSE.  See the above copyright notice for more information.
12
13 =========================================================================*/
14 #ifndef GDCMOPENSSLP7CRYPTOFACTORY_H
15 #define GDCMOPENSSLP7CRYPTOFACTORY_H
16
17 #include "gdcmCryptoFactory.h"
18 #include "gdcmOpenSSLP7CryptographicMessageSyntax.h"
19
20 namespace gdcm
21 {
22 class GDCM_EXPORT OpenSSLP7CryptoFactory :  public CryptoFactory
23 {
24 public:
25   OpenSSLP7CryptoFactory(CryptoLib id) :  CryptoFactory(id)
26   {
27     gdcmDebugMacro( "OpenSSL (PKCS7) Factory registered." );
28   }
29
30 public:
31   CryptographicMessageSyntax* CreateCMSProvider()
32   {
33     return new OpenSSLP7CryptographicMessageSyntax();
34   }
35
36 private:
37   OpenSSLP7CryptoFactory(){}
38 };
39 }
40
41 #endif //GDCMOPENSSLP7CRYPTOFACTORY_H
```

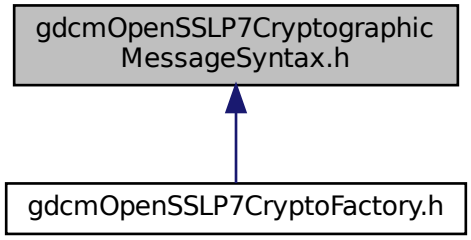
11.51 gdcmOpenSSLP7CryptographicMessageSyntax.h File Reference

```
#include "gdcmCryptographicMessageSyntax.h"
#include "gdcmTypes.h"
```

Include dependency graph for gdcmOpenSSLP7CryptographicMessageSyntax.h:



This graph shows which files directly or indirectly include this file:



Classes

- class gdcm::OpenSSLP7CryptographicMessageSyntax

Namespaces

- namespace gdcm

11.52 gdcmOpenSSLP7CryptographicMessageSyntax.h

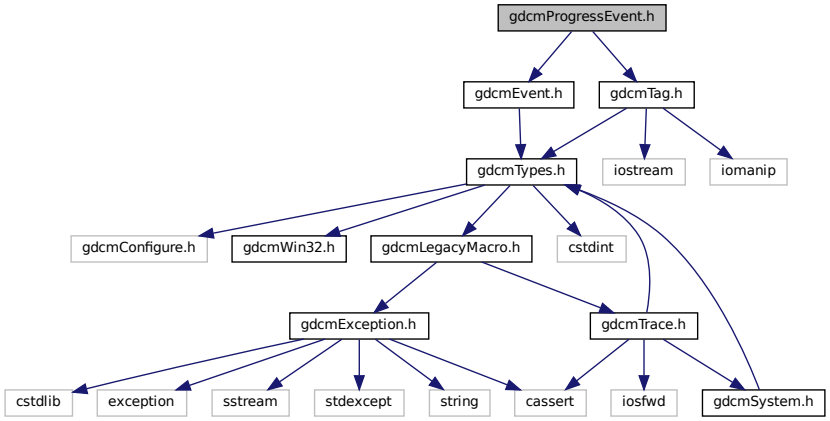
Go to the documentation of this file.

```
1 /*=========================================================================
2
3 Program:  GDCM (Grassroots DICOM). A DICOM library
4
5 Copyright (c) 2006-2011 Mathieu Malaterre
6 All rights reserved.
7 See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.
8
9 This software is distributed WITHOUT ANY WARRANTY; without even
10 the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
11 PURPOSE.  See the above copyright notice for more information.
12
13 =========================================================================*/
14 #ifndef GDCMOPENSSLP7CRYPTOGRAPHICMESSAGESYNTAX_H
15 #define GDCMOPENSSLP7CRYPTOGRAPHICMESSAGESYNTAX_H
16
17 #include "gdcmCryptographicMessageSyntax.h"
18 #include "gdcmTypes.h"
19
20 namespace gdcm
21 {
22 class CryptographicMessageSyntaxInternals;
23 //-----------------------------------------------------------------------------
24
34 class GDCM_EXPORT OpenSSLP7CryptographicMessageSyntax :  public CryptographicMessageSyntax
35 {
36 public:
37   OpenSSLP7CryptographicMessageSyntax();
38   ~OpenSSLP7CryptographicMessageSyntax();
39
40     // X.509
41   bool ParseCertificateFile( const char *filename );
42   bool ParseKeyFile( const char *filename );
43
44   // PBE
45   bool SetPassword(const char * /*pass*/, size_t /*passLen*/)
46   {
47     gdcmWarningMacro( "Openssl using PKCS7 does not support Password Based Encryption." );
48     return false;
49   }
50
53   void SetCipherType(CipherTypes type);
54   CipherTypes GetCipherType() const;
55
57   bool Encrypt(char *output, size_t &outlen, const char *array, size_t len) const;
58
60   bool Decrypt(char *output, size_t &outlen, const char *array, size_t len) const;
61
62 private:
63   CryptographicMessageSyntaxInternals *Internals;
64 private:
65   OpenSSLP7CryptographicMessageSyntax(const OpenSSLP7CryptographicMessageSyntax&);  // Not implemented.
66   void operator=(const OpenSSLP7CryptographicMessageSyntax&);  // Not implemented.
67 };
68 } // end namespace gdcm
69 //-----------------------------------------------------------------------------
70 #endif //GDCMOPENSSLP7CRYPTOGRAPHICMESSAGESYNTAX_H
```

11.53 gdcmProgressEvent.h File Reference

```
#include "gdcmEvent.h"
#include "gdcmTag.h"
```

Include dependency graph for gdcmProgressEvent.h:



Classes

- class gdcm::ProgressEvent

  ProgressEvent.

Namespaces

- namespace gdcm

11.54 gdcmProgressEvent.h

Go to the documentation of this file.

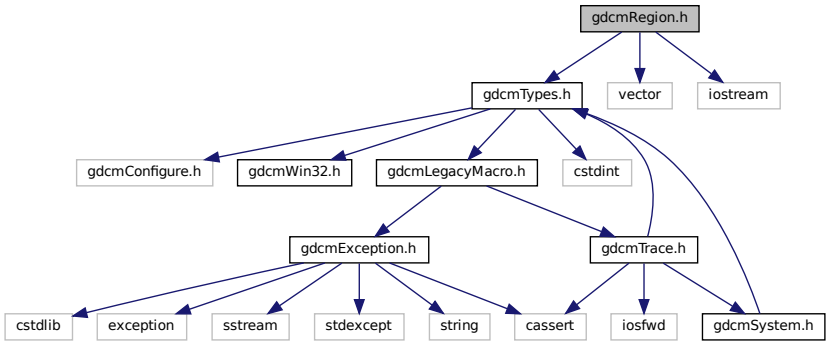
```
1 /*=========================================================================
2
3 Program:  GDCM (Grassroots DICOM). A DICOM library
4
5 Copyright (c) 2006-2011 Mathieu Malaterre
6 All rights reserved.
7 See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.
8
9 This software is distributed WITHOUT ANY WARRANTY; without even
10 the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
11 PURPOSE.  See the above copyright notice for more information.
12
13 =========================================================================*/
14 #ifndef GDCMPROGRESSEVENT_H
15 #define GDCMPROGRESSEVENT_H
16
17 #include "gdcmEvent.h"
18 #include "gdcmTag.h"
19
20 namespace gdcm
21 {
22
29 class ProgressEvent :  public AnyEvent
30 {
```

```
31 public:
32   typedef ProgressEvent Self;
33   typedef AnyEvent Superclass;
34   ProgressEvent(double p = 0):m_Progress(p) {}
35   ~ProgressEvent() override = default;
36
37   ProgressEvent(const Self&s) :  AnyEvent(s){};
38   void operator=(const Self&) = delete;
39
40   const char * GetEventName()const override { return "ProgressEvent"; }
41   bool CheckEvent(const ::gdcm::Event* e)const override
42 { return dynamic_cast<const Self*>(e) ?  true :  false; }
43   ::gdcm::Event* MakeObject()const override
44 { return new Self; }
45
46   void SetProgress(double p) { m_Progress = p; }
47   double GetProgress()const { return m_Progress; }
48 private:
49   double m_Progress;
50 };
51
52
53 } // end namespace gdcm
54
55 #endif //GDCMPROGRESSEVENT_H
```

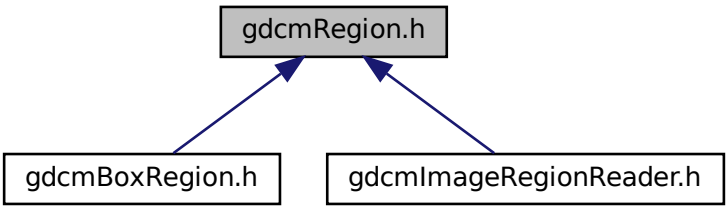
11.55 gdcmRegion.h File Reference

```
#include "gdcmTypes.h"
#include <vector>
#include <iostream>
```

Include dependency graph for gdcmRegion.h:



This graph shows which files directly or indirectly include this file:



Classes

- class gdcm::Region

  Class for manipulation region.

Namespaces

- namespace gdcm

Functions

- std::ostream & gdcm::operator<< (std::ostream &os, const Region &r)

11.56 gdcmRegion.h

Go to the documentation of this file.

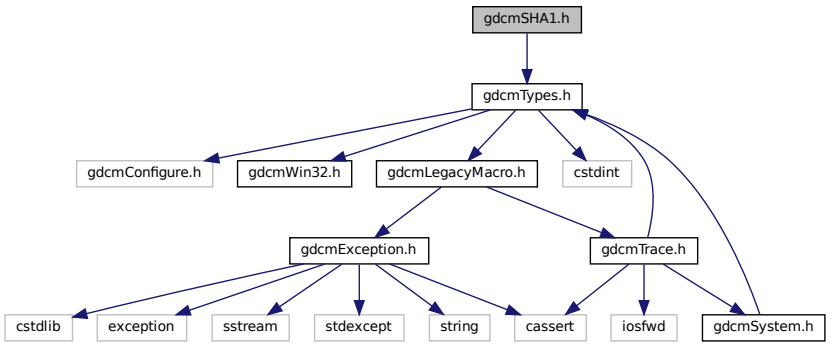
```
1 /*=========================================================================
2
3 Program:  GDCM (Grassroots DICOM). A DICOM library
4
5 Copyright (c) 2006-2011 Mathieu Malaterre
6 All rights reserved.
7 See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.
8
9 This software is distributed WITHOUT ANY WARRANTY; without even
10 the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
11 PURPOSE.  See the above copyright notice for more information.
12
13 =========================================================================*/
14 #ifndef GDCMREGION_H
15 #define GDCMREGION_H
16
17 #include "gdcmTypes.h"
18 #include <vector>
19 #include <iostream>
20
21 namespace gdcm
22 {
```

```
23 class BoxRegion;
27 //-----------------------------------------------------------------------------
28 class GDCM_EXPORT Region
29 {
30 public :
31   Region();
32   virtual ~Region();
33
35   virtual void Print(std::ostream &os = std::cout) const;
36
38   virtual bool Empty() const = 0;
39
41   virtual bool IsValid() const = 0;
42
44   virtual size_t Area() const = 0;
45
46   // implementation detail of heterogeneous container in C++
47   virtual Region *Clone() const = 0;
48
50   virtual BoxRegion ComputeBoundingBox() = 0;
51 private:
52 };
53 //-----------------------------------------------------------------------------
54 inline std::ostream& operator<<(std::ostream &os, const Region&r)
55 {
56   r.Print( os );
57   return os;
58 }
59
60 } // end namespace gdcm
61 //-----------------------------------------------------------------------------
62 #endif //GDCMREGION_H
```

11.57 gdcmSHA1.h File Reference

`#include "gdcmTypes.h"`

Include dependency graph for gdcmSHA1.h:



Classes

- class gdcm::SHA1

  Class for SHA1.

Namespaces

- namespace gdcm

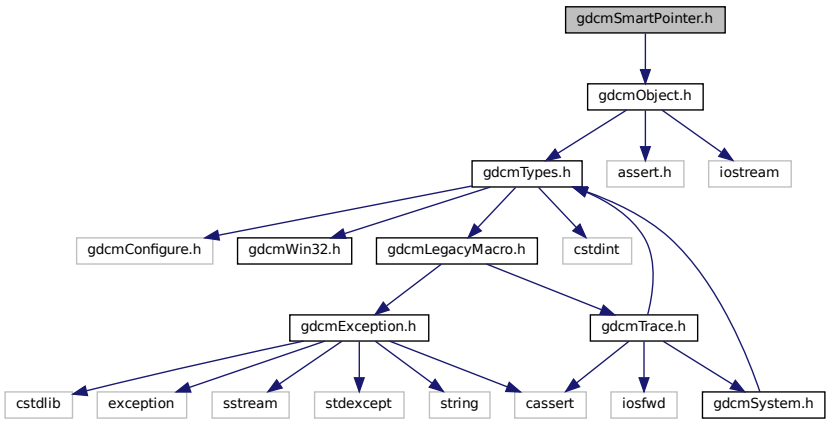
11.58 gdcmSHA1.h

Go to the documentation of this file.

```
1 /*=========================================================================
2
3 Program:  GDCM (Grassroots DICOM). A DICOM library
4
5 Copyright (c) 2006-2011 Mathieu Malaterre
6 All rights reserved.
7 See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.
8
9 This software is distributed WITHOUT ANY WARRANTY; without even
10 the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
11 PURPOSE.  See the above copyright notice for more information.
12
13 =========================================================================*/
14 #ifndef GDCMSHA1_H
15 #define GDCMSHA1_H
16
17 #include "gdcmTypes.h"
18
19 namespace gdcm
20 {
21 //-----------------------------------------------------------------------------
22 class SHA1Internals;
32 class GDCM_EXPORT SHA1
33 {
34 public :
35   SHA1();
36   ~SHA1();
37   SHA1(const SHA1&) = delete;
38   void operator=(const SHA1&) = delete;
39
40   static bool Compute(const char *buffer, unsigned long buf_len, char digest_str[20*2+1]);
41
42   static bool ComputeFile(const char *filename, char digest_str[20*2+1]);
43
44 private:
45   SHA1Internals *Internals;
46 };
47 } // end namespace gdcm
48 //-----------------------------------------------------------------------------
49 #endif //GDCMSHA1_H
```

11.59 gdcmSmartPointer.h File Reference

`#include "gdcmObject.h"`
Include dependency graph for gdcmSmartPointer.h:



This graph shows which files directly or indirectly include this file:

Classes

- class gdcm::SmartPointer< ObjectType >

  Class for Smart Pointer.

Namespaces

- namespace gdcm

11.60 gdcmSmartPointer.h

Go to the documentation of this file.

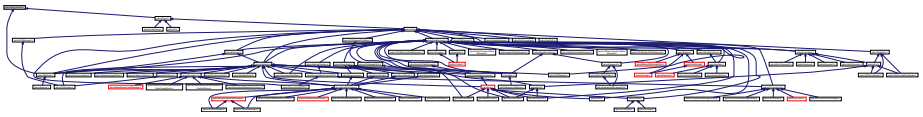
```
1 /*=========================================================================
2
3 Program:  GDCM (Grassroots DICOM). A DICOM library
4
5 Copyright (c) 2006-2011 Mathieu Malaterre
6 All rights reserved.
7 See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.
```

```
8
9 This software is distributed WITHOUT ANY WARRANTY; without even
10 the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
11 PURPOSE.  See the above copyright notice for more information.
12
13 =========================================================================*/
14 #ifndef GDCMSMARTPOINTER_H
15 #define GDCMSMARTPOINTER_H
16
17 #include "gdcmObject.h"
18
19 namespace gdcm
20 {
38 template<class ObjectType>
39 class SmartPointer
40 {
41 public:
42   SmartPointer():Pointer(nullptr) {}
43   SmartPointer(const SmartPointer<ObjectType>& p):Pointer(p.Pointer)
44     { Register(); }
45   SmartPointer(ObjectType* p):Pointer(p)
46     { Register(); }
47   SmartPointer(ObjectType const & p)
48     {
49     Pointer = const_cast<ObjectType*>(&p);
50     Register();
51     }
52   ~SmartPointer() {
53     UnRegister();
54     Pointer = nullptr;
55   }
56
58   ObjectType *operator -> ()const
59 { return Pointer; }
60
61   ObjectType& operator * ()const
62 {
63     assert( Pointer );
64     return *Pointer;
65     }
66
68   operator ObjectType * () const
69     { return Pointer; }
70
72   SmartPointer &operator = (SmartPointer const &r)
73     { return operator = (r.Pointer); }
74
76   SmartPointer &operator = (ObjectType *r)
77     {
78     // http://www.parashift.com/c++-faq-lite/freestore-mgmt.html#faq-16.22
79     // DO NOT CHANGE THE ORDER OF THESE STATEMENTS!
80     // (This order properly handles self-assignment)
81     // (This order also properly handles recursion, e.g., if a ObjectType contains SmartPointer<ObjectType>s)
82     if( Pointer != r )
83       {
84       ObjectType* old = Pointer;
85       Pointer = r;
86       Register();
87       if ( old ) { old->UnRegister(); }
88       }
89     return *this;
90     }
91
92   SmartPointer &operator = (ObjectType const &r)
93     {
94     ObjectType* tmp = const_cast<ObjectType*>(&r);
95     return operator = (tmp);
96     }
97
99   ObjectType *GetPointer()const
100 { return Pointer; }
101
102 private:
103   void Register()
104     {
105     if(Pointer) Pointer->Register();
106     }
107
108   void UnRegister()
109     {
110     if(Pointer) Pointer->UnRegister();
```

```
111     }
112
113   ObjectType* Pointer;
114 };
115
116 } // end namespace gdcm
117
118 #endif //GDCMSMARTPOINTER_H
```

11.61 gdcmStaticAssert.h File Reference

This graph shows which files directly or indirectly include this file:



Classes

- struct gdcm::static_assert_test< x >
- struct gdcm::STATIC_ASSERTION_FAILURE< true >

Namespaces

- namespace gdcm

Macros

- #define GDCM_DO_JOIN(X, Y) GDCM_DO_JOIN2(X,Y)
- #define GDCM_DO_JOIN2(X, Y) X##Y
- #define GDCM_JOIN(X, Y) GDCM_DO_JOIN(X, Y)
- #define GDCM_STATIC_ASSERT(B)

  *The GDCM_JOIN + **LINE** is needed to create a uniq identifier.*

11.61.1 Macro Definition Documentation

11.61.1.1 GDCM_DO_JOIN

```
#define GDCM_DO_JOIN(
              X,
              Y ) GDCM_DO_JOIN2(X,Y)
```

11.61.1.2 GDCM_DO_JOIN2

```
#define GDCM_DO_JOIN2(
              X,
              Y ) X##Y
```

11.61.1.3 GDCM_JOIN

```
#define GDCM_JOIN(
              X,
              Y ) GDCM_DO_JOIN( X, Y )
```

11.61.1.4 GDCM_STATIC_ASSERT

```
#define GDCM_STATIC_ASSERT(
              B )
```

Value:

```
  typedef ::gdcm::static_assert_test<\
    sizeof(::gdcm::STATIC_ASSERTION_FAILURE< (bool)( B ) >)>\
      GDCM_JOIN(gdcm_static_assert_typedef_, __LINE__)
```

The GDCM_JOIN + **LINE** is needed to create a uniq identifier.

11.62 gdcmStaticAssert.h

Go to the documentation of this file.

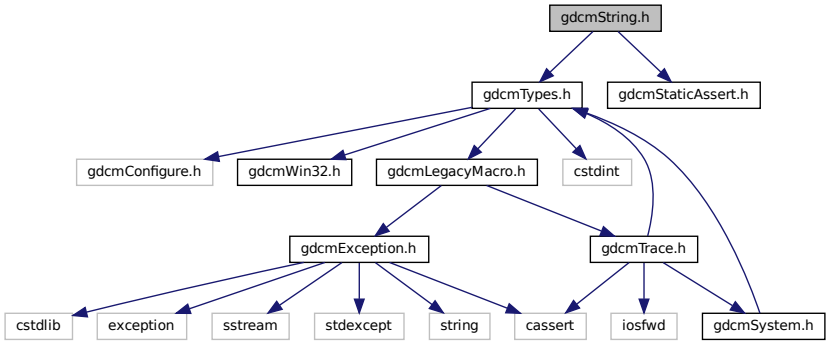
```
1 /*=========================================================================
2
3 Program:  GDCM (Grassroots DICOM). A DICOM library
4
5 Copyright (c) 2006-2011 Mathieu Malaterre
6 All rights reserved.
7 See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.
8
9 This software is distributed WITHOUT ANY WARRANTY; without even
10 the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
11 PURPOSE.  See the above copyright notice for more information.
12
13 =========================================================================*/
14 #ifndef GDCMSTATICASSERT_H
15 #define GDCMSTATICASSERT_H
16
17
18 // the following was shamelessly borowed from BOOST static assert:
19 namespace gdcm
20 {
21   template <bool x>
22   struct STATIC_ASSERTION_FAILURE;
23
24   template <>
25   struct STATIC_ASSERTION_FAILURE<true> { enum { value = 1 }; };
26
27   template <int x>
```

```
28   struct static_assert_test {};
29 }
30
31 #define GDCM_JOIN( X, Y ) GDCM_DO_JOIN( X, Y )
32 #define GDCM_DO_JOIN( X, Y ) GDCM_DO_JOIN2(X,Y)
33 #define GDCM_DO_JOIN2( X, Y ) X##Y
34
36 #define GDCM_STATIC_ASSERT( B ) \
37 typedef ::gdcm::static_assert_test<\
38 sizeof(::gdcm::STATIC_ASSERTION_FAILURE< (bool)( B ) >)>\
39 GDCM_JOIN(gdcm_static_assert_typedef_, __LINE__)
40
41
42 /* Example of use:
43 *
44 * template <class T>
45 * struct must_not_be_instantiated
46 * {
47 * // this will be triggered if this type is instantiated
48 * GDCM_STATIC_ASSERT(sizeof(T) == 0);
49 * };
50 *
51 */
52 #endif // GDCMSTATICASSERT_H
```

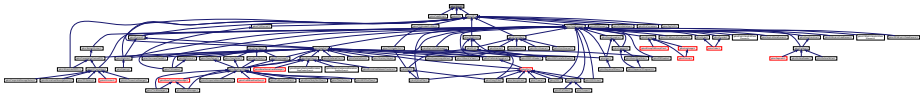
11.63 gdcmString.h File Reference

```
#include "gdcmTypes.h"
#include "gdcmStaticAssert.h"
```

Include dependency graph for gdcmString.h:



This graph shows which files directly or indirectly include this file:



Classes

- class gdcm::String< TDelimiter, TMaxLength, TPadChar >

  String.

Namespaces

- namespace gdcm

Functions

- template<char TDelimiter, unsigned int TMaxLength, char TPadChar>
  std::istream & gdcm::operator>> (std::istream &is, String< TDelimiter, TMaxLength, TPadChar > &ms)

11.64 gdcmString.h

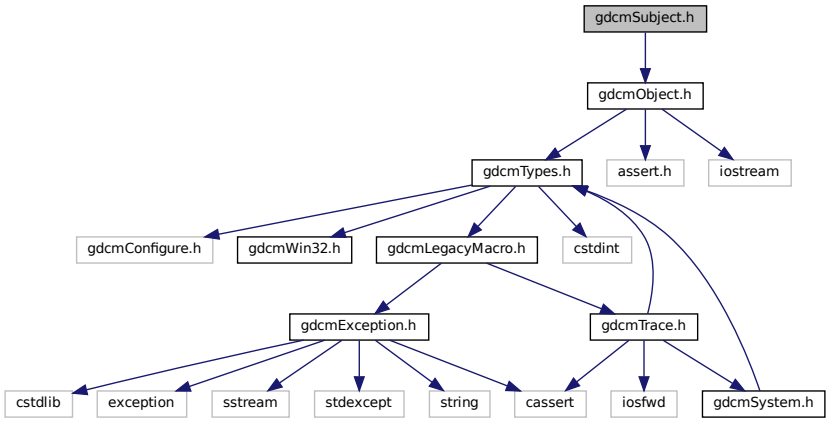
Go to the documentation of this file.

```
1 /*=========================================================================
2
3 Program:  GDCM (Grassroots DICOM). A DICOM library
4
5 Copyright (c) 2006-2011 Mathieu Malaterre
6 All rights reserved.
7 See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.
8
9 This software is distributed WITHOUT ANY WARRANTY; without even
10 the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
11 PURPOSE.  See the above copyright notice for more information.
12
13 =========================================================================*/
14 #ifndef GDCMSTRING_H
15 #define GDCMSTRING_H
16
17 #include "gdcmTypes.h"
18 #include "gdcmStaticAssert.h"
19
20 namespace gdcm
21 {
22
30 template <char TDelimiter = '\\', unsigned int TMaxLength = 64, char TPadChar = ' '>
31 class /*GDCM_EXPORT*/ String :  public std::string /* PLEASE do not export me */
32 {
33   // UI wants \0 for pad character, while ASCII ones wants space char...  do not allow anything else
34   GDCM_STATIC_ASSERT( TPadChar == ' ' || TPadChar == 0 );
35
36 public:
37   // typedef are not inherited:
38   typedef std::string::value_type             value_type;
39   typedef std::string::pointer                pointer;
40   typedef std::string::reference              reference;
41   typedef std::string::const_reference        const_reference;
42   typedef std::string::size_type              size_type;
43   typedef std::string::difference_type        difference_type;
44   typedef std::string::iterator               iterator;
45   typedef std::string::const_iterator         const_iterator;
46   typedef std::string::reverse_iterator       reverse_iterator;
47   typedef std::string::const_reverse_iterator const_reverse_iterator;
48
50   String():  std::string() {}
51   String(const value_type* s):  std::string(s)
52   {
53   if( size() % 2 )
54     {
55     push_back( TPadChar );
56     }
57   }
58   String(const value_type* s, size_type n):  std::string(s, n)
59   {
60   // We are being passed a const char* pointer, so s[n] == 0 (guaranteed!)
61   if( n % 2 )
62     {
63     push_back( TPadChar );
64     }
65   }
```

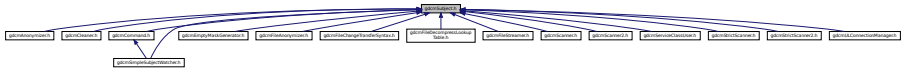
```
66   String(const std::string& s, size_type pos=0, size_type n=npos):
67     std::string(s, pos, n)
68   {
69   // FIXME: some users might already have padded the string 's' with a trailing \0...
70   if( size() % 2 )
71     {
72     push_back( TPadChar );
73     }
74   }
75
77   operator const char *() const { return this->c_str(); }
78
80   bool IsValid()const {
81     // Check Length:
82     size_type l = size();
83     if( l > TMaxLength ) return false;
84     return true;
85   }
86
87   gdcm::String<TDelimiter, TMaxLength, TPadChar> Truncate()const {
88     if( IsValid() ) return *this;
89     std::string str = *this; // copy
90     str.resize( TMaxLength );
91     return str;
92   }
93
96   std::string Trim()const {
97     std::string str = *this; // copy
98     std::string::size_type pos1 = str.find_first_not_of(' ');
99     std::string::size_type pos2 = str.find_last_not_of(' ');
100     str = str.substr( (pos1 == std::string::npos) ?  0 :  pos1,
101       (pos2 == std::string::npos) ?  (str.size() - 1) :  (pos2 - pos1 + 1));
102     return str;
103   }
104
105   static std::string Trim(const char *input) {
106     if( !input ) return "";
107     std::string str = input;
108     std::string::size_type pos1 = str.find_first_not_of(' ');
109     std::string::size_type pos2 = str.find_last_not_of(' ');
110     str = str.substr( (pos1 == std::string::npos) ?  0 :  pos1,
111       (pos2 == std::string::npos) ?  (str.size() - 1) :  (pos2 - pos1 + 1));
112     return str;
113   }
114 };
115 template <char TDelimiter, unsigned int TMaxLength, char TPadChar>
116 inline std::istream& operator»(std::istream &is, String<TDelimiter,TMaxLength,TPadChar> &ms)
117 {
118   if(is)
119     {
120     std::getline(is, ms, TDelimiter);
121     // no such thing as std::get where the delim char would be left, so I need to manually add it back...
122     // hopefully this is the right thing to do (no overhead)
123     if( !is.eof() ) is.putback( TDelimiter );
124     }
125   return is;
126 }
127 //template <char TDelimiter = EOF, unsigned int TMaxLength = 64, char TPadChar = ' '>
128 //String String::Trim() const
129 //{
130 //  String s;
131 //  return s;
132 //}
133
134 } // end namespace gdcm
135
136 #endif //GDCMSTRING_H
```

11.65 gdcmSubject.h File Reference

`#include "gdcmObject.h"`
Include dependency graph for gdcmSubject.h:



This graph shows which files directly or indirectly include this file:



Classes

- class gdcm::Subject
  Subject.

Namespaces

- namespace gdcm

11.66 gdcmSubject.h

Go to the documentation of this file.

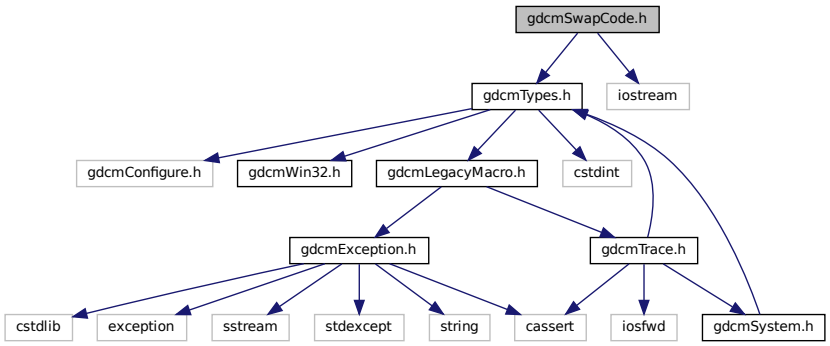
```
1 /*=========================================================================
2
3 Program:  GDCM (Grassroots DICOM). A DICOM library
4
5 Copyright (c) 2006-2011 Mathieu Malaterre
6 All rights reserved.
```

```
7 See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.
8
9 This software is distributed WITHOUT ANY WARRANTY; without even
10 the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
11 PURPOSE.  See the above copyright notice for more information.
12
13 =========================================================================*/
14 #ifndef GDCMSUBJECT_H
15 #define GDCMSUBJECT_H
16
17 #include "gdcmObject.h"
18
19 namespace gdcm
20 {
21 class Event;
22 class Command;
23 class SubjectInternals;
28 class GDCM_EXPORT Subject :  public Object
29 {
30 public:
31   Subject();
32   ~Subject() override;
33
42   unsigned long AddObserver(const Event & event, Command *);
43   unsigned long AddObserver(const Event & event, Command *) const;
44
50   Command* GetCommand(unsigned long tag);
51
53   void InvokeEvent( const Event & );
54
57   void InvokeEvent( const Event & ) const;
58
60   void RemoveObserver(unsigned long tag);
61
63   void RemoveAllObservers();
64
66   bool HasObserver( const Event & event ) const;
67
68 protected:
69
70 private:
71   SubjectInternals *Internals;
72 private:
73 };
74
75 } // end namespace gdcm
76
77 #endif //GDCMSUBJECT_H
```

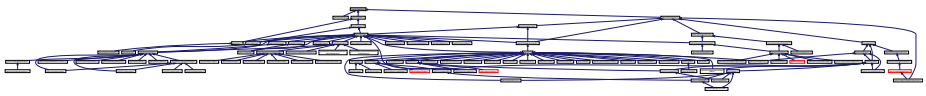
11.67 gdcmSwapCode.h File Reference

```
#include "gdcmTypes.h"
#include <iostream>
```

Include dependency graph for gdcmSwapCode.h:



This graph shows which files directly or indirectly include this file:



Classes

- class gdcm::SwapCode

  SwapCode representation.

Namespaces

- namespace gdcm

Functions

- std::ostream & gdcm::operator<< (std::ostream &os, const SwapCode &sc)

11.68 gdcmSwapCode.h

Go to the documentation of this file.

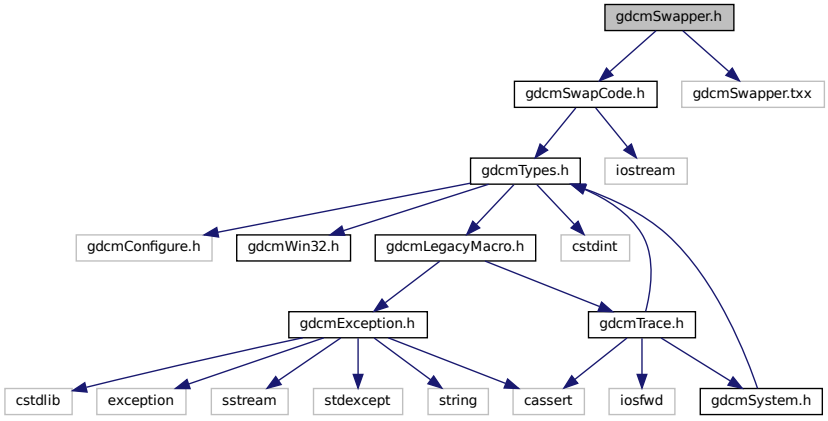
```
1 /*=========================================================================
2
3 Program:  GDCM (Grassroots DICOM). A DICOM library
4
5 Copyright (c) 2006-2011 Mathieu Malaterre
6 All rights reserved.
7 See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.
8
```

```
9 This software is distributed WITHOUT ANY WARRANTY; without even
10 the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
11 PURPOSE.  See the above copyright notice for more information.
12
13 =========================================================================*/
14 #ifndef GDCMSWAPCODE_H
15 #define GDCMSWAPCODE_H
16
17 #include "gdcmTypes.h"
18 #include <iostream>
19
20 namespace gdcm
21 {
22
27 class GDCM_EXPORT SwapCode
28 {
29 public:
30   typedef enum {
31     Unknown         = 0,
32     LittleEndian    = 1234,
33     BigEndian       = 4321,
34     BadLittleEndian = 3412,
35     BadBigEndian    = 2143
36   } SwapCodeType;
37
38   operator SwapCode::SwapCodeType()const { return SwapCodeValue; }
39   SwapCode(SwapCodeType sc = Unknown):SwapCodeValue(sc) { }
40   static const char* GetSwapCodeString(SwapCode const & sc);
41
42   friend std::ostream& operator<<(std::ostream& os, const SwapCode& sc);
43 protected:
44   static int GetIndex(SwapCode const & sc);
45
46 private:
47   SwapCodeType SwapCodeValue;
48 };
49 //-----------------------------------------------------------------------------
50 inline std::ostream& operator<<(std::ostream& os, const SwapCode& sc)
51 {
52   os << SwapCode::GetSwapCodeString(sc);
53   return os;
54 }
55
56 } // end namespace gdcm
57
58 #endif //GDCMSWAPCODE_H
```

11.69 gdcmSwapper.h File Reference

```
#include "gdcmSwapCode.h"
#include "gdcmSwapper.txx"
```

Include dependency graph for gdcmSwapper.h:



This graph shows which files directly or indirectly include this file:

Classes

- class gdcm::SwapperDoOp
- class gdcm::SwapperNoOp

Namespaces

- namespace gdcm

11.70 gdcmSwapper.h

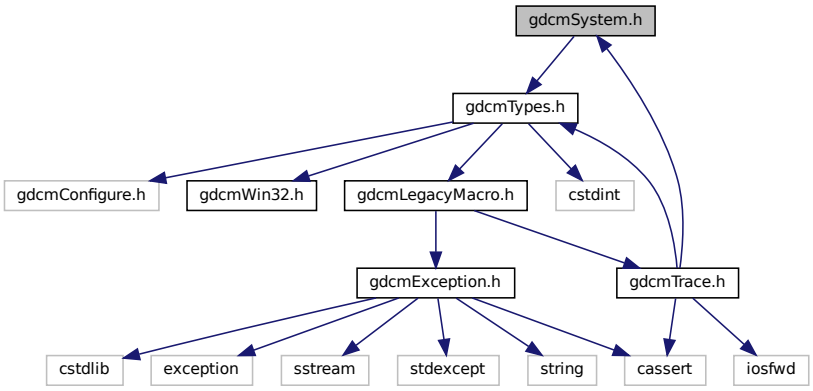
Go to the documentation of this file.

```
1 /*=========================================================================
2
3 Program:  GDCM (Grassroots DICOM). A DICOM library
4
5 Copyright (c) 2006-2011 Mathieu Malaterre
6 All rights reserved.
7 See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.
8
9 This software is distributed WITHOUT ANY WARRANTY; without even
10 the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
11 PURPOSE.  See the above copyright notice for more information.
12
13 =========================================================================*/
```

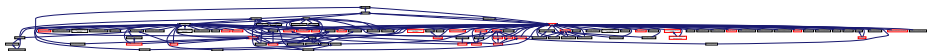
```
14 #ifndef GDCMSWAPPER_H
15 #define GDCMSWAPPER_H
16
17 #include "gdcmSwapCode.h"
18
19 namespace gdcm
20 {
21
22
23 #ifdef GDCM_WORDS_BIGENDIAN
24 class SwapperDoOp
25 {
26 public:
27   template <typename T> static T Swap(T val) {return val;}
28   template <typename T> static void SwapArray(T *, size_t ) {}
29 };
30
31 class SwapperNoOp
32 {
33 public:
34   template <typename T> static T Swap(T val);
35   template <typename T>
36   static void SwapArray(T *array, size_t n)
37   {
38     // TODO: need to unroll loop:
39     for(size_t i = 0; i < n; ++i)
40     {
41       array[i] = Swap<T>(array[i]);
42     }
43   }
44 };
45 #else
46 class SwapperNoOp
47 {
48 public:
49   template <typename T> static T Swap(T val) {return val;}
50   template <typename T> static void SwapArray(T *, size_t ) {}
51 };
52
53 class SwapperDoOp
54 {
55 public:
56   template <typename T> static T Swap(T val);
57   template <typename T>
58   static void SwapArray(T *array, size_t n)
59   {
60     // TODO: need to unroll loop:
61     for(size_t i = 0; i < n; ++i)
62     {
63       array[i] = Swap<T>(array[i]);
64     }
65   }
66 };
67 #endif
68
69
70 } // end namespace gdcm
71
72 #include "gdcmSwapper.txx"
73
74 #endif //GDCMSWAPPER_H
```

11.71 gdcmSystem.h File Reference

`#include "gdcmTypes.h"`
Include dependency graph for gdcmSystem.h:



This graph shows which files directly or indirectly include this file:



Classes

- class gdcm::System

  Class to do system operation.

Namespaces

- namespace gdcm

11.72 gdcmSystem.h

Go to the documentation of this file.

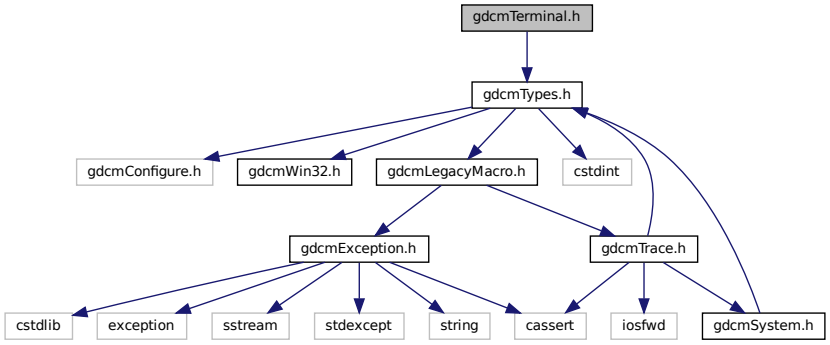
```
1 /*=========================================================================
2
3 Program:  GDCM (Grassroots DICOM). A DICOM library
4
5 Copyright (c) 2006-2011 Mathieu Malaterre
6 All rights reserved.
7 See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.
8
9 This software is distributed WITHOUT ANY WARRANTY; without even
```

```
10 the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
11 PURPOSE.  See the above copyright notice for more information.
12
13 =========================================================================*/
14 #ifndef GDCMSYSTEM_H
15 #define GDCMSYSTEM_H
16
17 #include "gdcmTypes.h"
18
19 namespace gdcm
20 {
21
26 class GDCM_EXPORT System
27 {
28 public:
30   static bool MakeDirectory(const char *path);
32   static bool FileExists(const char* filename);
34   static bool FileIsDirectory(const char* name);
36   static bool FileIsSymlink(const char* name);
38   static bool RemoveFile(const char* source);
40   static bool DeleteDirectory(const char *source);
41
45   static std::wstring ConvertToUNC(const char *utf8path);
46
48   static const char *GetLastSystemError();
49
54   static size_t FileSize(const char* filename);
55
58   static time_t FileTime(const char* filename);
59
62   static const char *GetCurrentProcessFileName();
63
66   static const char *GetCurrentModuleFileName();
67
70   static const char *GetCurrentResourcesDirectory();
71
72   // TODO some system calls
73   // Chdir
74   // copy a file
75
78   static bool GetHostName(char hostname[255]);
79
80   // In the following the size '22' is explicitly listed.  You need to pass in
81   // at least 22bytes of array.  If the string is an output it will be
82   // automatically padded ( array[21] == 0 ) for you.
83   // Those functions:  GetCurrentDateTime / FormatDateTime / ParseDateTime do
84   // not return the &YYZZ part of the DT structure as defined in DICOM PS 3.5 -
85   // 2008 In this case it is simple to split the date[22] into a DA and TM
86   // structure
87
92   static bool GetCurrentDateTime(char date[22]);
93
97   static bool FormatDateTime(char date[22], time_t t, long milliseconds = 0);
98
100   static bool ParseDateTime(time_t &timep, const char date[22]);
101
104   static bool ParseDateTime(time_t &timep, long &milliseconds, const char date[22]);
105
108   static const char *GetTimezoneOffsetFromUTC();
109
112   static size_t EncodeBytes(char *out, const unsigned char *data, int size);
113
115   static int StrCaseCmp(const char *s1, const char *s2);
117   static int StrNCaseCmp(const char *s1, const char *s2, size_t n);
118
123   static const char * GetCWD();
124
126   static char *StrTokR(char *ptr, const char *sep, char **end);
127
131   static char *StrSep(char **stringp, const char *delim);
132
134   static const char *GetLocaleCharset();
135
137 /*
138 static void SetArgv0(const char *);
139 static const char* GetArgv0();
140 */
141
142 protected:
143   static bool GetPermissions(const char* file, unsigned short& mode);
144   static bool SetPermissions(const char* file, unsigned short mode);
```

```
145
146 private:
147 };
148
149 } // end namespace gdcm
150
151 #endif //GDCMSYSTEM_H
```

11.73 gdcmTerminal.h File Reference

`#include "gdcmTypes.h"`
Include dependency graph for gdcmTerminal.h:



Namespaces

- namespace gdcm
- namespace gdcm::terminal

  Class for Terminal.

Enumerations

- enum gdcm::terminal::Attribute {
  gdcm::terminal::reset = 0 ,
  gdcm::terminal::bright = 1 ,
  gdcm::terminal::dim = 2 ,
  gdcm::terminal::underline = 3 ,
  gdcm::terminal::blink = 5 ,
  gdcm::terminal::reverse = 7 ,
  gdcm::terminal::hidden = 8 }
- enum gdcm::terminal::Color {
  gdcm::terminal::black = 0 ,
  gdcm::terminal::red ,
  gdcm::terminal::green ,
  gdcm::terminal::yellow ,

gdcm::terminal::blue ,
gdcm::terminal::magenta ,
gdcm::terminal::cyan ,
gdcm::terminal::white }

- enum gdcm::terminal::Mode {
  gdcm::terminal::CONSOLE = 0 ,
  gdcm::terminal::VT100 }

Functions

- GDCM_EXPORT std::string gdcm::terminal::setattribute (Attribute att)
- GDCM_EXPORT std::string gdcm::terminal::setbgcolor (Color c)
- GDCM_EXPORT std::string gdcm::terminal::setfgcolor (Color c)
- GDCM_EXPORT void gdcm::terminal::setmode (Mode m)

11.74 gdcmTerminal.h

Go to the documentation of this file.

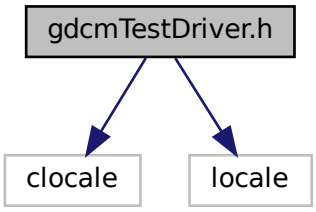
```
1 /*=========================================================================
2
3 Program:  GDCM (Grassroots DICOM). A DICOM library
4
5 Copyright (c) 2006-2011 Mathieu Malaterre
6 All rights reserved.
7 See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.
8
9 This software is distributed WITHOUT ANY WARRANTY; without even
10 the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
11 PURPOSE.  See the above copyright notice for more information.
12
13 =========================================================================*/
14 #ifndef GDCMTERMINAL_H
15 #define GDCMTERMINAL_H
16
17 #include "gdcmTypes.h"
18
19
20 namespace gdcm
21 {
28 //-----------------------------------------------------------------------------
29
30 namespace terminal
31 {
32   typedef enum
33     {
34     CONSOLE = 0,
35     VT100
36     } Mode;
37   typedef enum
38     {
39     black = 0,
40     red,
41     green,
42     yellow, // brown ??
43     blue,
44     magenta,
45     cyan,
46     white
47     } Color;
48   typedef enum
49     {
50     reset     = 0,
51     bright    = 1, // bold
52     dim       = 2,
53     underline = 3,
54     blink     = 5,
```

```
55     reverse    = 7,
56     hidden     = 8
57     } Attribute;
58   GDCM_EXPORT std::string setattribute( Attribute att );
59   GDCM_EXPORT std::string setfgcolor( Color c );
60   GDCM_EXPORT std::string setbgcolor( Color c );
61   GDCM_EXPORT void setmode( Mode m);
62 }
63
64 } // end namespace gdcm
65 //-----------------------------------------------------------------------------
66 #endif //GDCMTERMINAL_H
```

11.75 gdcmTestDriver.h File Reference

```
#include <clocale>
#include <locale>
```

Include dependency graph for gdcmTestDriver.h:



11.76 gdcmTestDriver.h

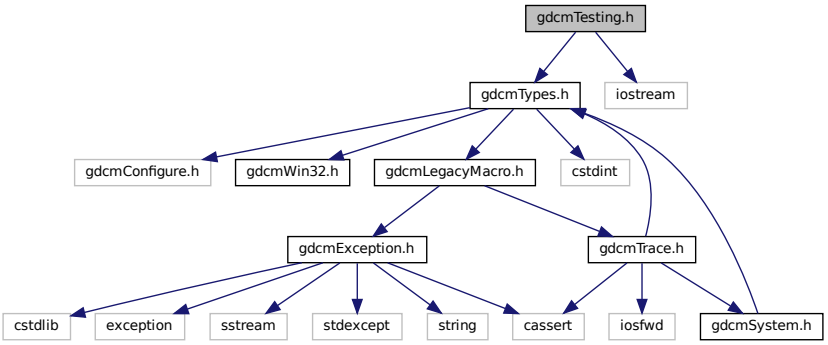
Go to the documentation of this file.

```
1 /*=========================================================================
2
3 Program:  GDCM (Grassroots DICOM). A DICOM library
4
5 Copyright (c) 2006-2011 Mathieu Malaterre
6 All rights reserved.
7 See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.
8
9 This software is distributed WITHOUT ANY WARRANTY; without even
10 the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
11 PURPOSE.  See the above copyright notice for more information.
12
13 =========================================================================*/
14 // This header is included by all the C++ test drivers in GDCM.
15 #ifndef GDCMTESTDRIVER_H
16 #define GDCMTESTDRIVER_H
17
18 // CREATE_TEST_SOURCELIST supports the flag EXTRA_INCLUDE but only one per call.
19 // So there is no way to specify we want to include two files...  instead
20 // gather the #include in a single file and include that one...
21 #include <clocale> // C setlocale()
22 #include <locale> // C++ locale
23
24 #endif // GDCMTESTDRIVER_H
```

11.77 gdcmTesting.h File Reference

```
#include "gdcmTypes.h"
#include <iostream>
```

Include dependency graph for gdcmTesting.h:



Classes

- class gdcm::Testing

  class for testing

Namespaces

- namespace gdcm

11.78 gdcmTesting.h

Go to the documentation of this file.

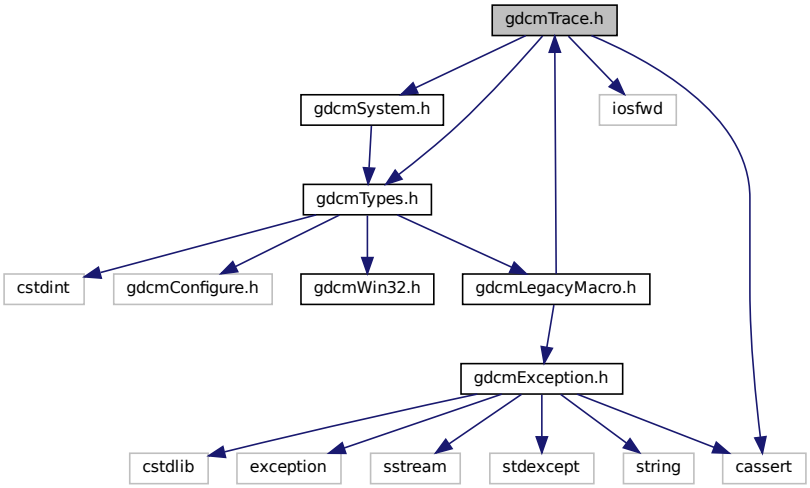
```
1 /*=========================================================================
2
3 Program:  GDCM (Grassroots DICOM). A DICOM library
4
5 Copyright (c) 2006-2011 Mathieu Malaterre
6 All rights reserved.
7 See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.
8
9 This software is distributed WITHOUT ANY WARRANTY; without even
10 the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
11 PURPOSE.  See the above copyright notice for more information.
12
13 =========================================================================*/
14 #ifndef GDCMTESTING_H
15 #define GDCMTESTING_H
16
17 #include "gdcmTypes.h"
18
19 #include <iostream>
20
21 namespace gdcm
```

```
22 {
30 //-----------------------------------------------------------------------------
31 class GDCM_EXPORT Testing
32 {
33 public :
34   Testing() = default;;
35   ~Testing() = default;;
36
42   static bool ComputeMD5(const char *buffer, size_t buf_len,
43     char digest_str[33]);
44   static bool ComputeFileMD5(const char *filename, char digest_str[33]);
45
47   void Print(std::ostream &os = std::cout);
48
50   static const char * const * GetFileNames();
51   static unsigned int GetNumberOfFileNames();
52   static const char * GetFileName(unsigned int file);
53
55   typedef const char* const (*MediaStorageDataFilesType)[2];
56   static MediaStorageDataFilesType GetMediaStorageDataFiles();
57   static unsigned int GetNumberOfMediaStorageDataFiles();
58   static const char * const * GetMediaStorageDataFile(unsigned int file);
59   static const char * GetMediaStorageFromFile(const char *filepath);
60
63   typedef const char* const (*MD5DataImagesType)[2];
64   static MD5DataImagesType GetMD5DataImages();
65   static unsigned int GetNumberOfMD5DataImages();
66   static const char * const * GetMD5DataImage(unsigned int file);
67   static const char * GetMD5FromFile(const char *filepath);
68
71   static const char * GetMD5FromBrokenFile(const char *filepath);
72
75   static std::streamoff GetStreamOffsetFromFile(const char *filepath);
76
80   static std::streamoff GetSelectedTagsOffsetFromFile(const char *filepath);
81
85   static std::streamoff GetSelectedPrivateGroupOffsetFromFile(const char *filepath);
86
91   static int GetLossyFlagFromFile(const char *filepath);
92
94   static const char * GetDataRoot();
95
97   static const char * GetDataExtraRoot();
98
100   static const char * GetPixelSpacingDataRoot();
101
104   static const char * GetTempDirectory(const char * subdir = nullptr);
105
107   static const wchar_t *GetTempDirectoryW(const wchar_t * subdir = nullptr);
108
110   static const char * GetTempFilename(const char *filename, const char * subdir = nullptr);
111
113   static const wchar_t* GetTempFilenameW(const wchar_t *filename, const wchar_t* subdir = nullptr);
114
115   static const char *GetSourceDirectory();
116 };
117 } // end namespace gdcm
118 //-----------------------------------------------------------------------------
119 #endif //GDCMTESTING_H
```

11.79 gdcmTrace.h File Reference

```
#include "gdcmTypes.h"
#include "gdcmSystem.h"
#include <iosfwd>
#include <cassert>
```

Include dependency graph for gdcmTrace.h:



This graph shows which files directly or indirectly include this file:

Classes

- class gdcm::Trace
  Trace.

Namespaces

- namespace gdcm

Macros

- #define GDCM_FUNCTION "<unknown>"
- #define gdcmAssertAlwaysMacro(arg) gdcmAssertMacro(arg)
  AssertAlways.
- #define gdcmAssertMacro(arg)
  Assert.
- #define gdcmDebugMacro(msg)
  Debug.
- #define gdcmErrorMacro(msg)
  Error this is pretty bad, more than just warning It could mean lost of data, something not handle...
- #define gdcmWarningMacro(msg)
  Warning.

11.79.1 Macro Definition Documentation

11.79.1.1 GDCM_FUNCTION

```
#define GDCM_FUNCTION "<unknown>"
```

11.79.1.2 gdcmAssertAlwaysMacro

```
#define gdcmAssertAlwaysMacro(
              arg ) gdcmAssertMacro(arg)
```

AssertAlways.

Parameters

arg	argument to test An easy solution to pass also a message is to do: gdcmAssertMacro("my message" && 2 < 3)

11.79.1.3 gdcmAssertMacro

```
#define gdcmAssertMacro(
              arg )
```

Value:

```
{                                                                    \
   if( !(arg) )                                                      \
   {                                                                 \
   std::ostringstream osmacro;                                       \
   osmacro « "Assert:  In " __FILE__ ", line " « __LINE__          \
           « ", function " « GDCM_FUNCTION                         \
           « "\n\n";                                                 \
   std::ostream &_os = gdcm::Trace::GetErrorStream();                \
   _os « osmacro.str() « std::endl;                                 \
   assert ( arg );                                                   \
   }                                                                 \
}                                                                    \
GDCM_NOOP_STATEMENT
```

Assert.

Parameters

arg	argument to test An easy solution to pass also a message is to do: gdcmAssertMacro("my message" && 2 < 3)

11.79.1.4 gdcmDebugMacro

```
#define gdcmDebugMacro(
              msg )
```

Value:

```
{                                                                    \
   if( gdcm::Trace::GetDebugFlag() )                                 \
   {                                                                 \
   std::ostringstream osmacro;                                       \
   osmacro « "Debug:  In " __FILE__ ", line " « __LINE__          \
           « ", function " « GDCM_FUNCTION « '\n'                \
           « "Last system error was:  "                              \
           « gdcm::System::GetLastSystemError() « '\n' « msg;  \
   std::ostream &_os = gdcm::Trace::GetDebugStream();                \
   _os « osmacro.str() « "\n\n" « std::endl;                      \
   }                                                                 \
}                                                                    \
GDCM_NOOP_STATEMENT
```

Debug.

Parameters

msg	message part

11.79.1.5 gdcmErrorMacro

```
#define gdcmErrorMacro(
              msg )
```

Value:

```
{                                                                    \
   if( gdcm::Trace::GetErrorFlag() )                                 \
   {                                                                 \
   std::ostringstream osmacro;                                       \
   osmacro « "Error:  In " __FILE__ ", line " « __LINE__          \
           « ", function " « GDCM_FUNCTION « '\n'                \
           « msg « "\n\n";                                           \
   std::ostream &_os = gdcm::Trace::GetErrorStream();                \
   _os « osmacro.str() « std::endl;                                \
   }                                                                 \
}                                                                    \
GDCM_NOOP_STATEMENT
```

Error this is pretty bad, more than just warning It could mean lost of data, something not handle...

Parameters

msg	second message part

11.79.1.6 gdcmWarningMacro

```
#define gdcmWarningMacro(
              msg )
```

Value:

```
{                                                                       \
   if( gdcm::Trace::GetWarningFlag() )                                  \
   {                                                                    \
   std::ostringstream osmacro;                                          \
   osmacro « "Warning:  In " __FILE__ ", line " « __LINE__       \
           « ", function " « GDCM_FUNCTION « "\n"                \
           « msg « "\n\n";                                       \
   std::ostream &_os = gdcm::Trace::GetWarningStream();           \
   _os « osmacro.str() « std::endl;                              \
   }                                                                    \
}                                                                       \
GDCM_NOOP_STATEMENT
```

Warning.

Parameters

msg	message part

11.80 gdcmTrace.h

Go to the documentation of this file.

```
1 /*=========================================================================
2
3 Program:  GDCM (Grassroots DICOM). A DICOM library
4
5 Copyright (c) 2006-2011 Mathieu Malaterre
6 All rights reserved.
7 See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.
8
9 This software is distributed WITHOUT ANY WARRANTY; without even
10 the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
11 PURPOSE.  See the above copyright notice for more information.
12
13 =========================================================================*/
14 #ifndef GDCMTRACE_H
15 #define GDCMTRACE_H
16
17 #include "gdcmTypes.h"
18 #include "gdcmSystem.h"
19
20 #include <iosfwd>
21 #include <cassert>
22
23 namespace gdcm
24 {
25
41 class GDCM_EXPORT Trace
42 {
43 public :
44   Trace();
45   ~Trace();
46
49   static void SetStream(std::ostream &os);
50   static std::ostream &GetStream();
51
53   static void SetDebugStream(std::ostream &os);
54   static std::ostream &GetDebugStream();
55
57   static void SetWarningStream(std::ostream &os);
```

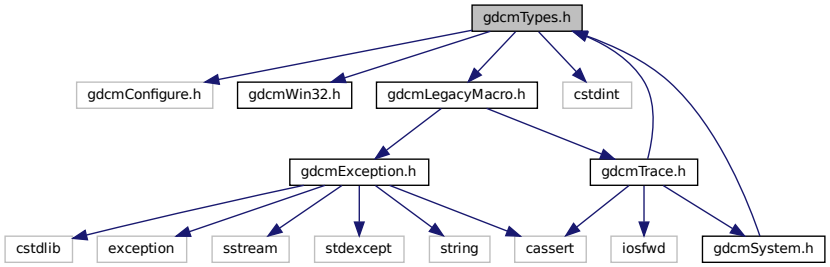
```
58   static std::ostream &GetWarningStream();
59
61   static void SetErrorStream(std::ostream &os);
62   static std::ostream &GetErrorStream();
63
66   static void SetStreamToFile( const char *filename );
67
69   static void SetDebug(bool debug);
70   static void DebugOn();
71   static void DebugOff();
72   static bool GetDebugFlag();
73
75   static void SetWarning(bool debug);
76   static void WarningOn();
77   static void WarningOff();
78   static bool GetWarningFlag();
79
81   static void SetError(bool debug);
82   static void ErrorOn();
83   static void ErrorOff();
84   static bool GetErrorFlag();
85
86 protected:
87 private:
88 };
89
90 // Here we define function this is the only way to be able to pass
91 // stuff with indirection like:
92 // gdcmDebug( "my message:" « i « '\t' );
93 // You cannot use function unless you use vnsprintf ...
94
95 // __FUNCTION is not always defined by preprocessor
96 // In c++ we should use __PRETTY_FUNCTION__ instead...
97 #ifdef GDCM_CXX_HAS_FUNCTION
98 // Handle particular case for GNU C++ which also defines __PRETTY_FUNCTION__
99 // which is a lot nice in C++
100 #ifdef __BORLANDC__
101 #  define __FUNCTION__ __FUNC__
102 #endif
103 #ifdef __GNUC__
104 #  define GDCM_FUNCTION __PRETTY_FUNCTION__
105 #else
106 #  define GDCM_FUNCTION __FUNCTION__
107 #endif //__GNUC__
108 #else
109 #  define GDCM_FUNCTION "<unknown>"
110 #endif //GDCM_CXX_HAS_FUNCTION
111
116 #if defined(NDEBUG) && !defined(GDCM_ALWAYS_TRACE_MACRO)
117 #define gdcmDebugMacro(msg) GDCM_NOOP_STATEMENT
118 #else
119 #define gdcmDebugMacro(msg)                                          \
120 {                                                                    \
121 if( gdcm::Trace::GetDebugFlag() )                                  \
122 {                                                                  \
123 std::ostringstream osmacro;                                        \
124 osmacro « "Debug:  In " __FILE__ ", line " « __LINE__           \
125           « ", function " « GDCM_FUNCTION « '\n'                   \
126           « "Last system error was:  "                             \
127           « gdcm::System::GetLastSystemError() « '\n' « msg;  \
128    std::ostream &_os = gdcm::Trace::GetDebugStream();               \
129    _os « osmacro.str() « "\n\n" « std::endl;                       \
130    }                                                                \
131 }                                                                   \
132 GDCM_NOOP_STATEMENT
133 #endif //NDEBUG
134
139 #if defined(NDEBUG) && !defined(GDCM_ALWAYS_TRACE_MACRO)
140 #define gdcmWarningMacro(msg) GDCM_NOOP_STATEMENT
141 #else
142 #define gdcmWarningMacro(msg)                                        \
143 {                                                                    \
144 if( gdcm::Trace::GetWarningFlag() )                                \
145 {                                                                  \
146 std::ostringstream osmacro;                                        \
147 osmacro « "Warning:  In " __FILE__ ", line " « __LINE__         \
148           « ", function " « GDCM_FUNCTION « "\n"                   \
149           « msg « "\n\n";                                          \
150    std::ostream &_os = gdcm::Trace::GetWarningStream();             \
151    _os « osmacro.str() « std::endl;                                \
152    }                                                                \
```

```
153 }                                                                    \
154 GDCM_NOOP_STATEMENT
155 #endif //NDEBUG
156
162 #if defined(NDEBUG) && !defined(GDCM_ALWAYS_TRACE_MACRO)
163 #define gdcmErrorMacro(msg) GDCM_NOOP_STATEMENT
164 #else
165 #define gdcmErrorMacro(msg)                                          \
166 {                                                                    \
167 if( gdcm::Trace::GetErrorFlag() )                                  \
168 {                                                                    \
169 std::ostringstream osmacro;                                        \
170 osmacro « "Error:  In " __FILE__ ", line " « __LINE__           \
171           « ", function " « GDCM_FUNCTION « '\n'                   \
172           « msg « "\n\n";                                           \
173    std::ostream &_os = gdcm::Trace::GetErrorStream();               \
174    _os « osmacro.str() « std::endl;                                \
175    }                                                                 \
176 }                                                                    \
177 GDCM_NOOP_STATEMENT
178 #endif //NDEBUG
179
186 #if defined(NDEBUG) && !defined(GDCM_ALWAYS_TRACE_MACRO)
187 #define gdcmAssertMacro(arg) GDCM_NOOP_STATEMENT
188 #else
189 #define gdcmAssertMacro(arg)                                         \
190 {                                                                    \
191 if( !(arg) )                                                       \
192 {                                                                    \
193 std::ostringstream osmacro;                                        \
194 osmacro « "Assert:  In " __FILE__ ", line " « __LINE__          \
195           « ", function " « GDCM_FUNCTION                          \
196           « "\n\n";                                                 \
197    std::ostream &_os = gdcm::Trace::GetErrorStream();               \
198    _os « osmacro.str() « std::endl;                                \
199    assert ( arg );                                                   \
200    }                                                                 \
201 }                                                                    \
202 GDCM_NOOP_STATEMENT
203 #endif //NDEBUG
204
211 #if defined(NDEBUG)
212 // User asked for release compilation, but still need to report
213 // if grave issue.
214 #define gdcmAssertAlwaysMacro(arg) \
215 {                                                                    \
216 if( !(arg) )                                                       \
217 {                                                                    \
218 std::ostringstream osmacro;                                        \
219 osmacro « "Assert:  In " __FILE__ ", line " « __LINE__          \
220           « ", function " « GDCM_FUNCTION                          \
221           « "\n\n";                                                 \
222    throw osmacro.str();                                              \
223    }                                                                 \
224 }                                                                    \
225 GDCM_NOOP_STATEMENT
226 #else
227 // Simply reproduce gdcmAssertMacro behavior:
228 #define gdcmAssertAlwaysMacro(arg) gdcmAssertMacro(arg)
229 #endif //NDEBUG
230
231 } // end namespace gdcm
232 //----------------------------------------------------------------------------
233 #endif //GDCMTRACE_H
```

11.81 gdcmTypes.h File Reference

```
#include "gdcmConfigure.h"
#include "gdcmWin32.h"
#include "gdcmLegacyMacro.h"
#include <cstdint>
```

Include dependency graph for gdcmTypes.h:



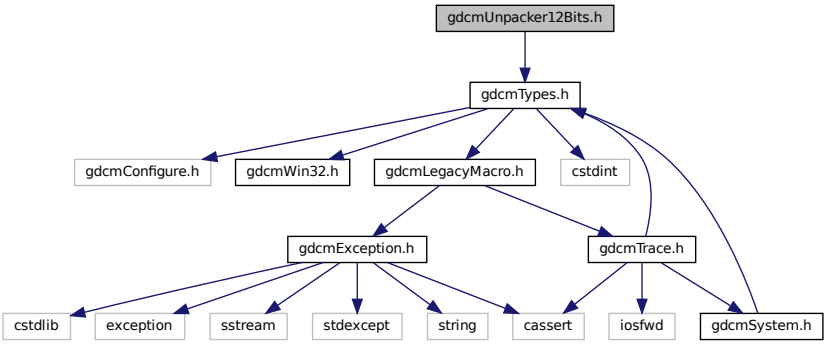
11.82 gdcmTypes.h

Go to the documentation of this file.

```
1 /*=========================================================================
2
3 Program:  GDCM (Grassroots DICOM). A DICOM library
4
5 Copyright (c) 2006-2011 Mathieu Malaterre
6 All rights reserved.
7 See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.
8
9 This software is distributed WITHOUT ANY WARRANTY; without even
10 the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
11 PURPOSE.  See the above copyright notice for more information.
12
13 =========================================================================*/
14 #ifndef GDCMTYPES_H
15 #define GDCMTYPES_H
16
17 #include "gdcmConfigure.h"
18 #include "gdcmWin32.h"
19 #include "gdcmLegacyMacro.h"
20
21 //-----------------------------------------------------------------------------
22 #include <cstdint>
23
24 //-----------------------------------------------------------------------------
25 #endif //GDCMTYPES_H
```

11.83 gdcmUnpacker12Bits.h File Reference

`#include "gdcmTypes.h"`
Include dependency graph for gdcmUnpacker12Bits.h:



Classes

- class gdcm::Unpacker12Bits

  Pack/Unpack 12 bits pixel into 16bits.

Namespaces

- namespace gdcm

11.84 gdcmUnpacker12Bits.h

Go to the documentation of this file.

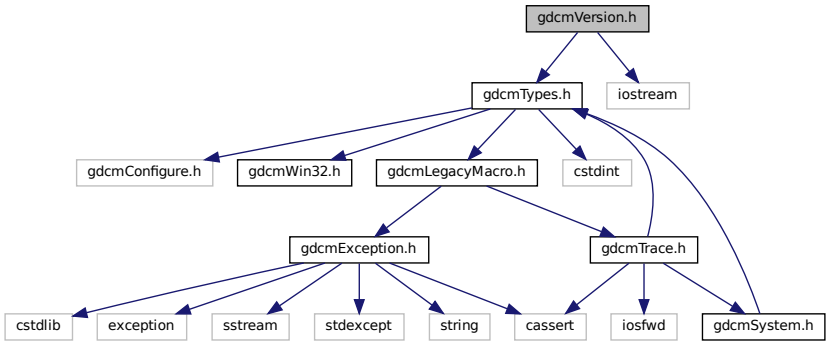
```
1 /*=========================================================================
2
3 Program:  GDCM (Grassroots DICOM). A DICOM library
4
5 Copyright (c) 2006-2011 Mathieu Malaterre
6 All rights reserved.
7 See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.
8
9 This software is distributed WITHOUT ANY WARRANTY; without even
10 the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
11 PURPOSE.  See the above copyright notice for more information.
12
13 =========================================================================*/
14 #ifndef GDCMUNPACKER12BITS_H
15 #define GDCMUNPACKER12BITS_H
16
17 #include "gdcmTypes.h"
18
19 namespace gdcm
20 {
34 class GDCM_EXPORT Unpacker12Bits
35 {
```

```
36 public:
40   static bool Pack(char *out, const char *in, size_t n);
41
45   static bool Unpack(char *out, const char *in, size_t n);
46 };
47
48 } // end namespace gdcm
49
50 #endif //GDCMUNPACKER12BITS_H
```

11.85 gdcmVersion.h File Reference

```
#include "gdcmTypes.h"
#include <iostream>
```

Include dependency graph for gdcmVersion.h:



Classes

- class gdcm::Version

  major/minor and build version

Namespaces

- namespace gdcm

Functions

- std::ostream & gdcm::operator<< (std::ostream &os, const Version &v)

11.86 gdcmVersion.h

Go to the documentation of this file.

```
1 /*=========================================================================
2
3 Program:  GDCM (Grassroots DICOM). A DICOM library
4
5 Copyright (c) 2006-2011 Mathieu Malaterre
6 All rights reserved.
7 See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.
8
9 This software is distributed WITHOUT ANY WARRANTY; without even
10 the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
11 PURPOSE.  See the above copyright notice for more information.
12
13 =========================================================================*/
14 #ifndef GDCMVERSION_H
15 #define GDCMVERSION_H
16
17 #include "gdcmTypes.h"
18 #include <iostream>
19
20 namespace gdcm
21 {
26 //-----------------------------------------------------------------------------
27 class GDCM_EXPORT Version
28 {
29   friend std::ostream& operator<<(std::ostream &_os, const Version &v);
30 public :
31   static const char *GetVersion();
32   static int GetMajorVersion();
33   static int GetMinorVersion();
34   static int GetBuildVersion();
35
36   void Print(std::ostream &os = std::cout) const;
37
38 //protected:
39   Version() = default;
40   ~Version() = default;
41 };
42 //-----------------------------------------------------------------------------
43 inline std::ostream& operator<<(std::ostream &os, const Version &v)
44 {
45   v.Print( os );
46   return os;
47 }
48
49 } // end namespace gdcm
50 //-----------------------------------------------------------------------------
51 #endif //GDCMVERSION_H
```

11.87 gdcmWin32.h File Reference

This graph shows which files directly or indirectly include this file:

Macros

- #define GDCM_EXPORT

11.87.1 Macro Definition Documentation

11.87.1.1 GDCM_EXPORT

```
#define GDCM_EXPORT
```

11.88 gdcmWin32.h

Go to the documentation of this file.

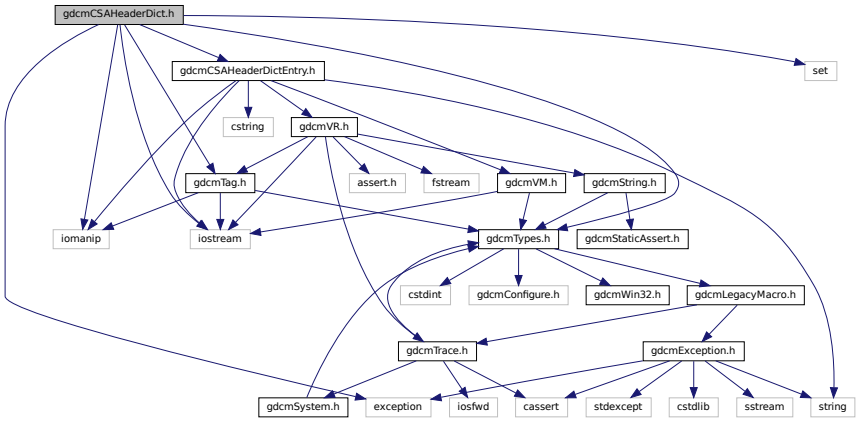
```
1 /*=========================================================================
2
3 Program:  GDCM (Grassroots DICOM). A DICOM library
4
5 Copyright (c) 2006-2011 Mathieu Malaterre
6 All rights reserved.
7 See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.
8
9 This software is distributed WITHOUT ANY WARRANTY; without even
10 the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
11 PURPOSE.  See the above copyright notice for more information.
12
13 =========================================================================*/
14
15 #ifndef GDCMWIN32_H
16 #define GDCMWIN32_H
17
18 #if !defined(GDCMTYPES_H)
19 #error you need to include gdcmTypes.h instead
20 #endif
21 //----------------------------------------------------------------------------
22 // http://gcc.gnu.org/wiki/Visibility
23 #if defined(_WIN32) && defined(GDCM_BUILD_SHARED_LIBS)
24 #if (defined(gdcmCommon_EXPORTS) || defined(gdcmDICT_EXPORTS) || defined(gdcmDSED_EXPORTS) ||
      defined(gdcmIOD_EXPORTS) || defined(gdcmMSFF_EXPORTS) || defined(gdcmMEXD_EXPORTS)||
      defined(_gdcmswig_EXPORTS)) || defined(vtkgdcm_EXPORTS)
25 #define GDCM_EXPORT __declspec( dllexport )
26 #else
27 #define GDCM_EXPORT __declspec( dllimport )
28 #endif
29 #else
30 #if __GNUC__ >= 4 && defined(GDCM_BUILD_SHARED_LIBS)
31 #define GDCM_EXPORT __attribute__ ((visibility ("default")))
32 #define GDCM_LOCAL  __attribute__ ((visibility ("hidden")))
33 #else
34 #define GDCM_EXPORT
35 #endif
36 #endif
37
38 #if defined(GDCM_OVERRIDE_BROKEN_IMPLEMENTATION) && !defined(GDCM_FORCE_EXPORT)
39 #undef GDCM_EXPORT
40 #define GDCM_EXPORT
41 #endif
42
43 // In VTK 4.2 vtkWrapPython does not like anything other than VTK_*EXPORT
44 // [ 86%] Generating vtkGDCMImageReaderPython.cxx
45 // syntax error
46 // *** SYNTAX ERROR found in parsing the header file
      /usr/local/src/gdcm2/tags/gdcm-2-0-11/Utilities/VTK/vtkGDCMImageReader.h before line 128***
47 // make[2]:  *** [Utilities/VTK/vtkGDCMImageReaderPython.cxx] Error 1
48 // make[1]:  *** [Utilities/VTK/CMakeFiles/vtkgdcmPythonD.dir/all] Error 2
49 // make:  *** [all] Error 2
50
51 #if defined(VTK_MAJOR_VERSION) && ( VTK_MAJOR_VERSION == 4 )
52 #undef VTK_EXPORT
53 #define VTK_EXPORT GDCM_EXPORT
54 #endif
55
56 //----------------------------------------------------------------------------
57 //This is needed when compiling in debug mode
58 #ifdef _MSC_VER
59 // to allow construct such as:  std::numeric_limits<int>::max() we need the following:
60 // warning C4003:  not enough actual parameters for macro 'max'
61 #ifndef NOMINMAX
62 #define NOMINMAX
63 #endif
```

```
64 # pragma warning ( default :  4263 ) /* no override, call convention differs */
65 // 'identifier' :  class 'type' needs to have dll-interface to be used by
66 // clients of class 'type2'
67 #pragma warning ( disable :  4251 )
68 // non dll-interface class 'type' used as base for dll-interface class 'type2'
69 #pragma warning ( disable :  4275 )
70 // 'identifier' :  identifier was truncated to 'number' characters in the
71 // debug information
72 #pragma warning ( disable :  4786 )
73 //'identifier' :  decorated name length exceeded, name was truncated
74 #pragma warning ( disable :  4503 )
75 #endif //_MSC_VER
76
77 //-----------------------------------------------------------------------------
78 #endif //GDCMWIN32_H
```

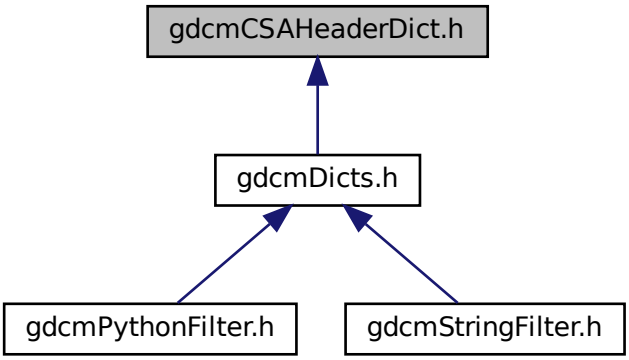
11.89 gdcmCSAHeaderDict.h File Reference

```
#include "gdcmTypes.h"
#include "gdcmTag.h"
#include "gdcmCSAHeaderDictEntry.h"
#include <iostream>
#include <iomanip>
#include <set>
#include <exception>
```

Include dependency graph for gdcmCSAHeaderDict.h:



This graph shows which files directly or indirectly include this file:



Classes

- class gdcm::CSAHeaderDict

  Class to represent a map of CSAHeaderDictEntry.
- class gdcm::CSAHeaderDictException

Namespaces

- namespace gdcm

Functions

- std::ostream & gdcm::operator<< (std::ostream &os, const CSAHeaderDict &val)

11.90 gdcmCSAHeaderDict.h

Go to the documentation of this file.

```
1 /*=========================================================================
2
3 Program:  GDCM (Grassroots DICOM). A DICOM library
4
5 Copyright (c) 2006-2011 Mathieu Malaterre
6 All rights reserved.
7 See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.
8
9 This software is distributed WITHOUT ANY WARRANTY; without even
10 the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
11 PURPOSE.  See the above copyright notice for more information.
12
13 =========================================================================*/
```

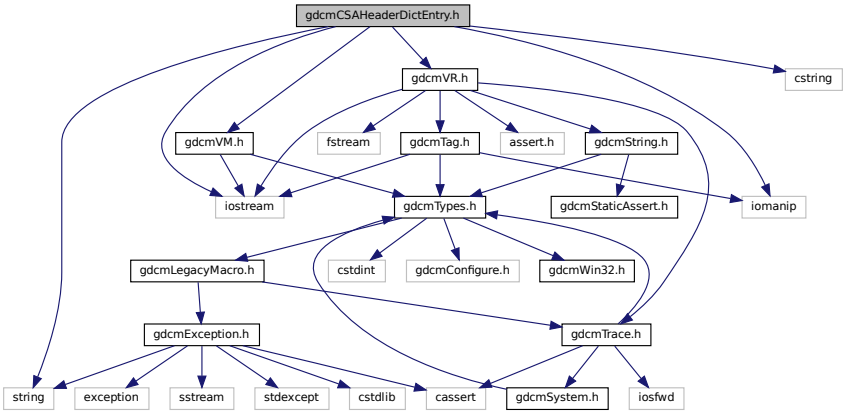
```
14 #ifndef GDCMCSAHEADERDICT_H
15 #define GDCMCSAHEADERDICT_H
16
17 #include "gdcmTypes.h"
18 #include "gdcmTag.h"
19 #include "gdcmCSAHeaderDictEntry.h"
20
21 #include <iostream>
22 #include <iomanip>
23 #include <set>
24 #include <exception>
25
26 namespace gdcm
27 {
28
29 class GDCM_EXPORT CSAHeaderDictException :  public std::exception {};
30
34 class GDCM_EXPORT CSAHeaderDict
35 {
36 public:
37   typedef std::set<CSAHeaderDictEntry> MapCSAHeaderDictEntry;
38   typedef MapCSAHeaderDictEntry::iterator Iterator;
39   typedef MapCSAHeaderDictEntry::const_iterator ConstIterator;
40   //static CSAHeaderDictEntry GroupLengthCSAHeaderDictEntry; // = CSAHeaderDictEntry("Group
      Length",VR::UL,VM::VM1);
41
42   CSAHeaderDict():CSAHeaderDictInternal() {
43     assert( CSAHeaderDictInternal.empty() );
44   }
45   CSAHeaderDict &operator=(const CSAHeaderDict &_val) = delete;
46   CSAHeaderDict(const CSAHeaderDict &_val) = delete;
47
48   friend std::ostream& operator«(std::ostream& _os, const CSAHeaderDict &_val);
49
50   ConstIterator Begin()const { return CSAHeaderDictInternal.begin(); }
51   ConstIterator End()const { return CSAHeaderDictInternal.end(); }
52
53   bool IsEmpty()const { return CSAHeaderDictInternal.empty(); }
54   void AddCSAHeaderDictEntry(const CSAHeaderDictEntry &de)
55     {
56 #ifndef NDEBUG
57     MapCSAHeaderDictEntry::size_type s = CSAHeaderDictInternal.size();
58 #endif
59     CSAHeaderDictInternal.insert( de );
60     assert( s < CSAHeaderDictInternal.size() );
61     }
62
63   const CSAHeaderDictEntry &GetCSAHeaderDictEntry(const char *name)const
64 {
65     MapCSAHeaderDictEntry::const_iterator it = CSAHeaderDictInternal.find( name );
66     if( it != CSAHeaderDictInternal.end() )
67       {
68       return *it;
69       }
70     throw CSAHeaderDictException();
71     }
72
73 protected:
74   friend class Dicts;
75   void LoadDefault();
76
77 private:
78
79   MapCSAHeaderDictEntry CSAHeaderDictInternal;
80 };
81 //-----------------------------------------------------------------------------
82 inline std::ostream& operator«(std::ostream& os, const CSAHeaderDict &val)
83 {
84   CSAHeaderDict::MapCSAHeaderDictEntry::const_iterator it = val.CSAHeaderDictInternal.begin();
85   for(;it != val.CSAHeaderDictInternal.end(); ++it)
86     {
87     const CSAHeaderDictEntry &de = *it;
88     os « de « '\n';
89     }
90
91
92   return os;
93 }
94
95
96 } // end namespace gdcm
```

```
97
98 #endif //GDCMCSAHEADERDICT_H
```

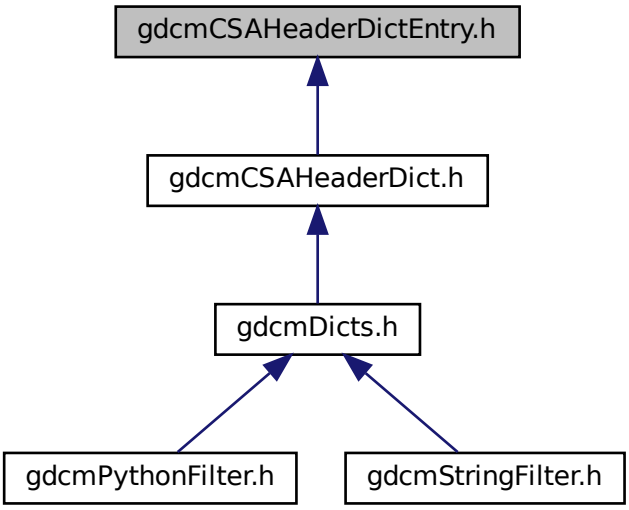
11.91 gdcmCSAHeaderDictEntry.h File Reference

```
#include "gdcmVR.h"
#include "gdcmVM.h"
#include <string>
#include <iostream>
#include <iomanip>
#include <cstring>
```

Include dependency graph for gdcmCSAHeaderDictEntry.h:



This graph shows which files directly or indirectly include this file:



Classes

- class gdcm::CSAHeaderDictEntry

  Class to represent an Entry in the Dict.

Namespaces

- namespace gdcm

Functions

- std::ostream & gdcm::operator<< (std::ostream &os, const CSAHeaderDictEntry &val)

11.92 gdcmCSAHeaderDictEntry.h

Go to the documentation of this file.

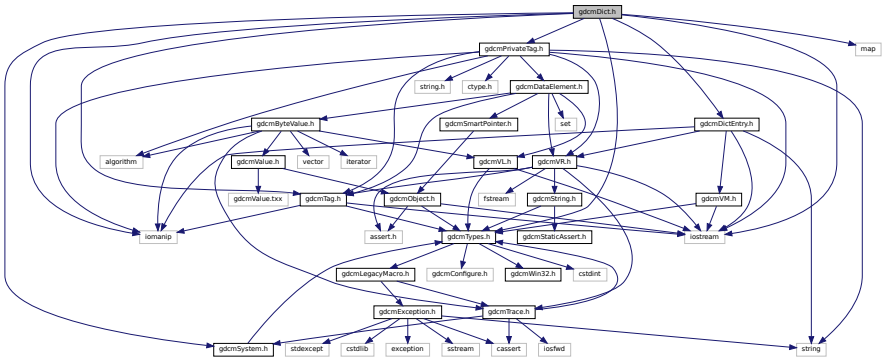
```
1 /*=========================================================================
2
3 Program:  GDCM (Grassroots DICOM). A DICOM library
4
5 Copyright (c) 2006-2011 Mathieu Malaterre
6 All rights reserved.
7 See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.
```

```
8
9 This software is distributed WITHOUT ANY WARRANTY; without even
10 the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
11 PURPOSE.  See the above copyright notice for more information.
12
13 =========================================================================*/
14 #ifndef GDCMCSAHEADERDICTENTRY_H
15 #define GDCMCSAHEADERDICTENTRY_H
16
17 #include "gdcmVR.h"
18 #include "gdcmVM.h"
19
20 #include <string>
21 #include <iostream>
22 #include <iomanip>
23
24 #include <cstring>
25
26 namespace gdcm
27 {
38 class GDCM_EXPORT CSAHeaderDictEntry
39 {
40 public:
41   CSAHeaderDictEntry(const char *name = "", VR const &vr = VR::INVALID, VM const &vm = VM::VM0, const char
       *desc = ""):Name(name),ValueRepresentation(vr),ValueMultiplicity(vm),Description(desc) {
42   }
43
44   friend std::ostream& operator«(std::ostream& _os, const CSAHeaderDictEntry &_val);
45
47   const VR &GetVR()const { return ValueRepresentation; }
48   void SetVR(const VR & vr) { ValueRepresentation = vr; }
49
51   const VM &GetVM()const { return ValueMultiplicity; }
52   void SetVM(VM const & vm) { ValueMultiplicity = vm; }
53
55   const char *GetName()const { return Name.c_str(); }
56   void SetName(const char* name) { Name = name; }
57
59   const char *GetDescription()const { return Description.c_str(); }
60   void SetDescription(const char* desc) { Description = desc; }
61
62   bool operator<(const CSAHeaderDictEntry &entry)const
63 {
64     return strcmp(GetName(),entry.GetName()) < 0;
65     }
66
67 private:
68   std::string Name;
69   VR ValueRepresentation;
70   VM ValueMultiplicity;
71   std::string Description;
72   std::string Type; // TODO
73 };
74
75
76 //-----------------------------------------------------------------------------
77 inline std::ostream& operator«(std::ostream& os, const CSAHeaderDictEntry &val)
78 {
79   if( val.Name.empty() )
80     {
81     os « "[No name]";
82     }
83   else
84     {
85     os « val.Name;
86     }
87   os « "\t" « val.ValueRepresentation « "\t" « val.ValueMultiplicity;
88   if( !val.Description.empty() )
89     {
90     os « "\t" « val.Description;
91     }
92   return os;
93 }
94
95 } // end namespace gdcm
96
97 #endif //GDCMCSAHEADERDICTENTRY_H
```

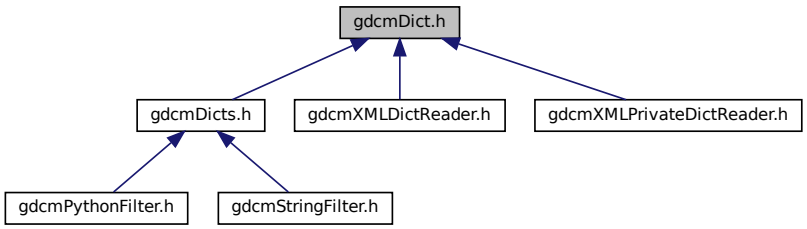
11.93 gdcmDict.h File Reference

```
#include "gdcmTypes.h"
#include "gdcmTag.h"
#include "gdcmPrivateTag.h"
#include "gdcmDictEntry.h"
#include "gdcmSystem.h"
#include <iostream>
#include <iomanip>
#include <map>
```

Include dependency graph for gdcmDict.h:



This graph shows which files directly or indirectly include this file:



Classes

- class gdcm::Dict

  Class to represent a map of DictEntry.
- class gdcm::PrivateDict

  Private Dict.

Namespaces

- namespace gdcm

Functions

- std::ostream & gdcm::operator<< (std::ostream &os, const Dict &val)
- std::ostream & gdcm::operator<< (std::ostream &os, const PrivateDict &val)

11.94 gdcmDict.h

Go to the documentation of this file.

```
1 /*=========================================================================
2
3 Program:  GDCM (Grassroots DICOM). A DICOM library
4
5 Copyright (c) 2006-2011 Mathieu Malaterre
6 All rights reserved.
7 See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.
8
9 This software is distributed WITHOUT ANY WARRANTY; without even
10 the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
11 PURPOSE.  See the above copyright notice for more information.
12
13 =========================================================================*/
14 #ifndef GDCMDICT_H
15 #define GDCMDICT_H
16
17 #include "gdcmTypes.h"
18 #include "gdcmTag.h"
19 #include "gdcmPrivateTag.h"
20 #include "gdcmDictEntry.h"
21 #include "gdcmSystem.h"
22
23 #include <iostream>
24 #include <iomanip>
25 #include <map>
26
27 /*
28 * FIXME / TODO
29 * I need to seriously rewrite this mess.  a class template should work for both a public
30 * and a private dict
31 */
32
33 namespace gdcm
34 {
35 // Data Element Tag
44 class GDCM_EXPORT Dict
45 {
46 public:
47   typedef std::map<Tag, DictEntry> MapDictEntry;
48   typedef MapDictEntry::iterator Iterator;
49   typedef MapDictEntry::const_iterator ConstIterator;
50   //static DictEntry GroupLengthDictEntry; // = DictEntry("Group Length",VR::UL,VM::VM1);
51
52   Dict():DictInternal() {
53     assert( DictInternal.empty() );
54   }
55   Dict &operator=(const Dict &_val) = delete;
56   Dict(const Dict &_val) = delete;
57
58
59   friend std::ostream& operator«(std::ostream& _os, const Dict &_val);
60
61   ConstIterator Begin()const { return DictInternal.begin(); }
62   ConstIterator End()const { return DictInternal.end(); }
63
64   bool IsEmpty()const { return DictInternal.empty(); }
65   void AddDictEntry(const Tag &tag, const DictEntry &de)
```

```
66     {
67 #ifndef NDEBUG
68     MapDictEntry::size_type s = DictInternal.size();
69 #endif
70     DictInternal.insert(
71       MapDictEntry::value_type(tag, de));
72     assert( s < DictInternal.size() );
73     }
74
75   const DictEntry &GetDictEntry(const Tag &tag)const
76 {
77     MapDictEntry::const_iterator it =
78       DictInternal.find(tag);
79     if (it == DictInternal.end())
80       {
81 #ifdef UNKNOWNPUBLICTAG
82       // test.acr
83       if( tag != Tag(0x28,0x15)
84         && tag != Tag(0x28,0x16)
85         && tag != Tag(0x28,0x199)
86         // gdcmData/TheralysGDCM1.dcm
87         && tag != Tag(0x20,0x1)
88         // gdcmData/0019004_Baseline_IMG1.dcm
89         && tag != Tag(0x8348,0x339)
90         && tag != Tag(0xb5e8,0x338)
91         // gdcmData/dicomdir_Acusson_WithPrivate_WithSR
92         && tag != Tag(0x40,0xa125)
93       )
94         {
95         assert( 0 && "Impossible" );
96         }
97 #endif
98       it = DictInternal.find( Tag(0xffff,0xffff) );
99       return it->second;
100       }
101     assert( DictInternal.count(tag) == 1 );
102     return it->second;
103     }
104
106   const char *GetKeywordFromTag(Tag const & tag)const
107 {
108     MapDictEntry::const_iterator it =
109       DictInternal.find(tag);
110     if (it == DictInternal.end())
111       {
112       return nullptr;
113       }
114     assert( DictInternal.count(tag) == 1 );
115     return it->second.GetKeyword();
116     }
117
122   const DictEntry &GetDictEntryByKeyword(const char *keyword, Tag & tag)const
123 {
124     MapDictEntry::const_iterator it =
125       DictInternal.begin();
126     if( keyword )
127       {
128       for(; it != DictInternal.end(); ++it)
129         {
130         if( strcmp( keyword, it->second.GetKeyword() ) == 0 )
131           {
132           // Found a match !
133           tag = it->first;
134           break;
135           }
136         }
137       }
138     else
139       {
140       it = DictInternal.end();
141       }
142     if (it == DictInternal.end())
143       {
144       tag = Tag(0xffff,0xffff);
145       it = DictInternal.find( tag );
146       return it->second;
147       }
148     assert( DictInternal.count(tag) == 1 );
149     return it->second;
150     }
151
```

```
155   const DictEntry &GetDictEntryByName(const char *name, Tag & tag)const
156 {
157     MapDictEntry::const_iterator it =
158       DictInternal.begin();
159     if( name )
160       {
161       for(; it != DictInternal.end(); ++it)
162         {
163         if( strcmp( name, it->second.GetName() ) == 0 )
164           {
165           // Found a match !
166           tag = it->first;
167           break;
168           }
169         }
170       }
171     else
172       {
173       it = DictInternal.end();
174       }
175     if (it == DictInternal.end())
176       {
177       tag = Tag(0xffff,0xffff);
178       it = DictInternal.find( tag );
179       return it->second;
180       }
181     assert( DictInternal.count(tag) == 1 );
182     return it->second;
183     }
184
185 protected:
186   friend class Dicts;
187   void LoadDefault();
188
189 private:
190   MapDictEntry DictInternal;
191 };
192 //-----------------------------------------------------------------------------
193 inline std::ostream& operator<<(std::ostream& os, const Dict &val)
194 {
195   Dict::MapDictEntry::const_iterator it = val.DictInternal.begin();
196   for(;it != val.DictInternal.end(); ++it)
197     {
198     const Tag &t = it->first;
199     const DictEntry &de = it->second;
200     os << t << " " << de << '\n';
201     }
202
203   return os;
204 }
205
206 // TODO
207 // For private dict, element < 0x10 should automatically defined:
208 // Name = "Private Creator"
209 // ValueRepresentation = LO
210 // ValueMultiplicity = 1
211 // Owner = ""
212
216 class GDCM_EXPORT PrivateDict
217 {
218   typedef std::map<PrivateTag, DictEntry> MapDictEntry;
219   friend std::ostream& operator<<(std::ostream& os, const PrivateDict &val);
220 public:
221   PrivateDict() = default;
222   ~PrivateDict() = default;
223   void AddDictEntry(const PrivateTag &tag, const DictEntry &de)
224     {
225 #ifndef NDEBUG
226     MapDictEntry::size_type s = DictInternal.size();
227 #endif
228     DictInternal.insert(
229       MapDictEntry::value_type(tag, de));
230 // The following code should only be used when manually constructing a Private.xml file by hand
231 // it will get rid of VR::UN duplicate (ie.  if a VR != VR::Un can be found)
232 #if defined(NDEBUG) && 0
233     if( s == DictInternal.size() )
234       {
235       MapDictEntry::iterator it =
236         DictInternal.find(tag);
237       assert( it != DictInternal.end() );
238       DictEntry &duplicate = it->second;
```

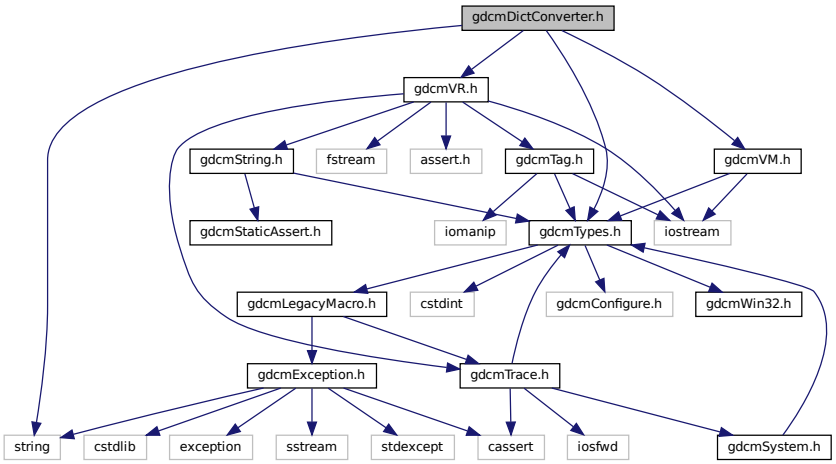
```
239       assert( de.GetVR() == VR::UN || duplicate.GetVR() == VR::UN );
240       assert( de.GetVR() != duplicate.GetVR() );
241       if( duplicate.GetVR() == VR::UN )
242         {
243         assert( de.GetVR() != VR::UN );
244         duplicate.SetVR( de.GetVR() );
245         duplicate.SetVM( de.GetVM() );
246         assert( GetDictEntry(tag).GetVR() != VR::UN );
247         assert( GetDictEntry(tag).GetVR() == de.GetVR() );
248         assert( GetDictEntry(tag).GetVM() == de.GetVM() );
249         }
250       return;
251       }
252 #endif
253     assert( s < DictInternal.size() /*&& std::cout « tag « "," « de « std::endl*/ );
254     }
257   bool RemoveDictEntry(const PrivateTag &tag)
258     {
259     MapDictEntry::size_type s =
260       DictInternal.erase(tag);
261     assert( s == 1 || s == 0 );
262     return s == 1;
263     }
264   bool FindDictEntry(const PrivateTag &tag)const
265 {
266     MapDictEntry::const_iterator it =
267       DictInternal.find(tag);
268     if (it == DictInternal.end())
269       {
270       return false;
271       }
272     return true;
273     }
274   const DictEntry &GetDictEntry(const PrivateTag &tag)const
275 {
276     // if 0x10 -> return Private Creator
277     MapDictEntry::const_iterator it =
278       DictInternal.find(tag);
279     if (it == DictInternal.end())
280       {
281       //assert( 0 && "Impossible" );
282       it = DictInternal.find( PrivateTag(0xffff,0xffff,"GDCM Private Sentinel" ) );
283       assert (it != DictInternal.end());
284       return it->second;
285       }
286     assert( DictInternal.count(tag) == 1 );
287     return it->second;
288     }
289
290
291   void PrintXML()const
292 {
293     MapDictEntry::const_iterator it = DictInternal.begin();
294     std::cout « "<dict edition=\"2008\">\n";
295     for(;it != DictInternal.end(); ++it)
296       {
297       const PrivateTag &t = it->first;
298       const DictEntry &de = it->second;
299       std::cout « "  <entry group=\"" « std::hex « std::setw(4)
300         « std::setfill('0') « t.GetGroup() « "\"" «
301         " element=\"xx" « std::setw(2) « std::setfill('0')« t.GetElement() « "\"" « " vr=\""
302         « de.GetVR() « "\" vm=\"" « de.GetVM() « "\" owner=\""
303         « t.GetOwner();
304       const char *name = de.GetName();
305       if( *name == 0 )
306         {
307         std::cout « "\"/>\n";
308         }
309       else
310         {
311         std::cout « "\" name=\"" « de.GetName() « "\"/>\n";
312         }
313       }
314     std::cout « "</dict>\n";
315     }
316
317   bool IsEmpty()const { return DictInternal.empty(); }
318 protected:
319   friend class Dicts;
320   void LoadDefault();
321
```

```
322 private:
323   PrivateDict &operator=(const PrivateDict &_val) = delete;
324   PrivateDict(const PrivateDict &_val) = delete;
325
326   MapDictEntry DictInternal;
327 };
328 //-----------------------------------------------------------------------------
329 inline std::ostream& operator«(std::ostream& os, const PrivateDict &val)
330 {
331   PrivateDict::MapDictEntry::const_iterator it = val.DictInternal.begin();
332   for(;it != val.DictInternal.end(); ++it)
333     {
334     const PrivateTag &t = it->first;
335     const DictEntry &de = it->second;
336     os « t « " " « de « '\n';
337     }
338
339   return os;
340 }
341
342 } // end namespace gdcm
343
344 #endif //GDCMDICT_H
```

11.95 gdcmDictConverter.h File Reference

```
#include "gdcmTypes.h"
#include "gdcmVR.h"
#include "gdcmVM.h"
#include <string>
```

Include dependency graph for gdcmDictConverter.h:



Classes

- class gdcm::DictConverter

  Class to convert a .dic file into something else:

Namespaces

- namespace gdcm

11.96 gdcmDictConverter.h

Go to the documentation of this file.

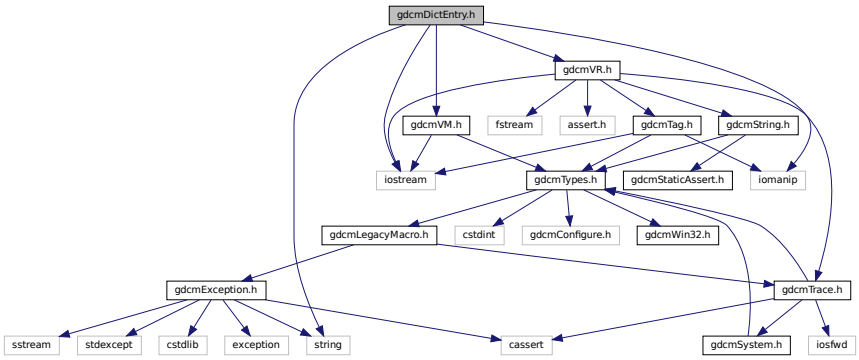
```
1 /*=========================================================================
2
3 Program:  GDCM (Grassroots DICOM). A DICOM library
4
5 Copyright (c) 2006-2011 Mathieu Malaterre
6 All rights reserved.
7 See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.
8
9 This software is distributed WITHOUT ANY WARRANTY; without even
10 the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
11 PURPOSE.  See the above copyright notice for more information.
12
13 =========================================================================*/
14
15 #ifndef GDCMDICTCONVERTER_H
16 #define GDCMDICTCONVERTER_H
17
18 #include "gdcmTypes.h"
19 #include "gdcmVR.h"
20 #include "gdcmVM.h"
21
22 #include <string>
23
24 namespace gdcm
25 {
26
27 class DictConverterInternal;
36 class GDCM_EXPORT DictConverter
37 {
38 public:
39   DictConverter();
40   ~DictConverter();
41   void SetInputFileName(const char* filename);
42   const std::string &GetInputFilename() const;
43   void SetOutputFileName(const char* filename);
44   const std::string &GetOutputFilename() const;
45
46   int GetOutputType()const {
47     return OutputType;
48   }
49   void SetOutputType(int type) {
50     OutputType = type;
51   }
52   const std::string &GetDictName() const;
53   void SetDictName(const char *name);
54
55   void Convert();
56
57   // Leaving them public for now.  Not really user oriented but may be
58   // useful
59   static bool ReadVR(const char *raw, VR::VRType &type);
60   static bool ReadVM(const char *raw, VM::VMType &type);
61   static bool Readuint16(const char *raw, uint16_t &ov);
62
63   enum OutputTypes {
64     DICT_DEFAULT = 0,
65     DICT_DEBUG,
66     DICT_XML
67   };
68
69 protected:
70   void WriteHeader();
71   void WriteFooter();
72   bool ConvertToXML(const char *raw, std::string &cxx);
73   bool ConvertToCXX(const char *raw, std::string &cxx);
74   void AddGroupLength();
75
```

```
76 private:
77   DictConverterInternal *Internal;
78
79   int OutputType;
80 };
81
82 } // end namespace gdcm
83
84 #endif //GDCMDICTCONVERTER_H
```

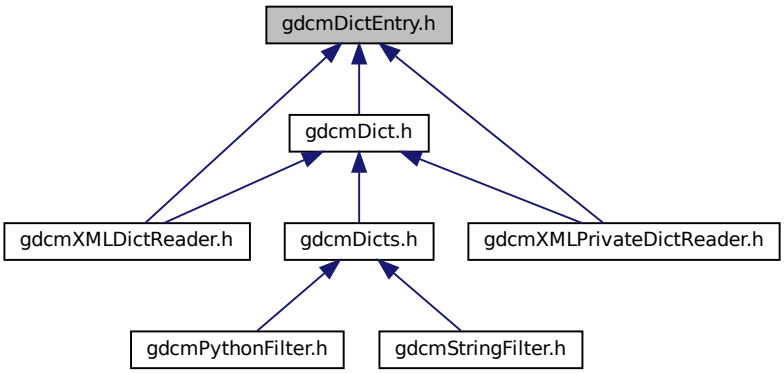
11.97 gdcmDictEntry.h File Reference

```
#include "gdcmVR.h"
#include "gdcmVM.h"
#include <string>
#include <iostream>
#include <iomanip>
```

Include dependency graph for gdcmDictEntry.h:



This graph shows which files directly or indirectly include this file:



Classes

- class gdcm::DictEntry

  Class to represent an Entry in the Dict.

Namespaces

- namespace gdcm

Functions

- std::ostream & gdcm::operator<< (std::ostream &os, const DictEntry &val)

11.98 gdcmDictEntry.h

Go to the documentation of this file.

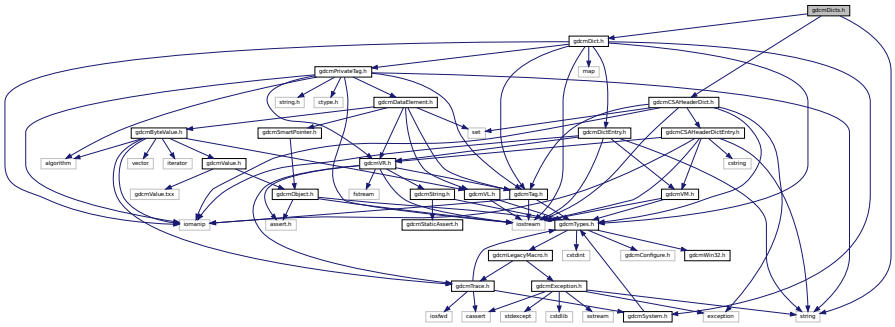
```
1 /*=========================================================================
2
3 Program:  GDCM (Grassroots DICOM). A DICOM library
4
5 Copyright (c) 2006-2011 Mathieu Malaterre
6 All rights reserved.
7 See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.
8
9 This software is distributed WITHOUT ANY WARRANTY; without even
10 the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
11 PURPOSE.  See the above copyright notice for more information.
12
13 =========================================================================*/
14 #ifndef GDCMDICTENTRY_H
15 #define GDCMDICTENTRY_H
16
17 #include "gdcmVR.h"
18 #include "gdcmVM.h"
19
20 #include <string>
21 #include <iostream>
22 #include <iomanip>
23
24 namespace gdcm
25 {
36 class GDCM_EXPORT DictEntry
37 {
38 public:
39   DictEntry(const char *name = "", const char *keyword = "", VR const &vr = VR::INVALID, VM const &vm =
      VM::VM0, bool ret = false):
40     Name(name),
41     Keyword(keyword),
42     ValueRepresentation(vr),
43     ValueMultiplicity(vm),
44     Retired(ret),
45     GroupXX(false),
46     ElementXX(false)
47   {
48   }
49
50   friend std::ostream& operator<<(std::ostream& _os, const DictEntry &_val);
51
53   const VR &GetVR()const { return ValueRepresentation; }
54   void SetVR(const VR & vr) { ValueRepresentation = vr; }
55 //  bool IsValid() const { return ValueRepresentation != VR::VR_END; }
56 //    !Name.empty() /*&& ValueRepresentation && ValueMultiplicity*/; }
57
59   const VM &GetVM()const { return ValueMultiplicity; }
60   void SetVM(VM const & vm) { ValueMultiplicity = vm; }
```

```
61
63   const char *GetName()const { return Name.c_str(); }
64   void SetName(const char* name) { Name = name; }
65
67   const char *GetKeyword()const { return Keyword.c_str(); }
68   void SetKeyword(const char* keyword) { Keyword = keyword; }
69
71   bool GetRetired()const { return Retired; }
72   void SetRetired(bool retired) { Retired = retired; }
73
74   // <entry group="50xx" element="0005" vr="US" vm="1" retired="true" version="3">
76   void SetGroupXX(bool v) { GroupXX = v; }
77
78   // <entry group="0020" element="31xx" vr="CS" vm="1-n" retired="true" version="2">
80   void SetElementXX(bool v) { ElementXX = v; }
81
84   bool IsUnique()const { return ElementXX == false && GroupXX == false; }
85
86 private:
87   //
88   friend class Dict;
89   static bool CheckKeywordAgainstName(const char *name, const char *keyword);
90
91 private:
92   std::string Name;
93   std::string Keyword;
94   VR ValueRepresentation;
95   VM ValueMultiplicity;
96   bool Retired :  1;
97   bool GroupXX : 1;
98   bool ElementXX : 1;
99 };
100
101 #if 0
102 class GDCM_EXPORT PrivateDictEntry :  public DictEntry
103 {
104 public:
105   PrivateDictEntry(const char *name = "", VR::VRType const &vr = VR::INVALID, VM::VMType const &vm = VM::VM0
      , bool ret = false, const char *owner = ""):DictEntry(name,vr,vm,ret),Owner(owner) {}
106   PrivateDictEntry(const char *name, const char *vr, const char *vm):DictEntry(name,vr,vm) {}
107
108   const char *GetOwner()const { return Owner.c_str(); }
109   void SetOwner(const char *owner) { Owner = owner; }
110
111 private:
112   // SIEMENS MED, GEMS_PETD_01 ...
113   std::string Owner;
114 };
115 #endif
116
117 //-----------------------------------------------------------------------------
118 inline std::ostream& operator«(std::ostream& os, const DictEntry &val)
119 {
120   if( val.Name.empty() )
121     {
122     os « "[No name]";
123     }
124   else
125     {
126     os « val.Name;
127     }
128   if( val.Keyword.empty() )
129     {
130     os « "[No keyword]";
131     }
132   else
133     {
134     os « val.Keyword;
135     }
136   os « "\t" « val.ValueRepresentation « "\t" « val.ValueMultiplicity;
137   if( val.Retired )
138     {
139     os « "\t(RET)";
140     }
141   return os;
142 }
143
144 } // end namespace gdcm
145
146 #endif //GDCMDICTENTRY_H
```

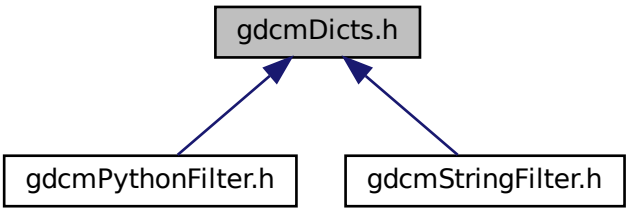
11.99 gdcmDicts.h File Reference

```
#include "gdcmDict.h"
#include "gdcmCSAHeaderDict.h"
#include <string>
```

Include dependency graph for gdcmDicts.h:



This graph shows which files directly or indirectly include this file:



Classes

- class gdcm::Dicts

  Class to manipulate the sum of knowledge (all the dict user load)

Namespaces

- namespace gdcm

Functions

- std::ostream & gdcm::operator<< (std::ostream &os, const Dicts &d)

11.100 gdcmDicts.h

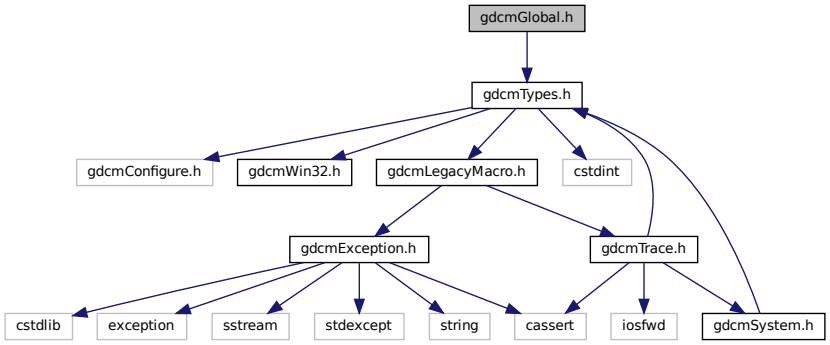
Go to the documentation of this file.

```
1 /*=========================================================================
2
3 Program:  GDCM (Grassroots DICOM). A DICOM library
4
5 Copyright (c) 2006-2011 Mathieu Malaterre
6 All rights reserved.
7 See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.
8
9 This software is distributed WITHOUT ANY WARRANTY; without even
10 the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
11 PURPOSE.  See the above copyright notice for more information.
12
13 =========================================================================*/
14 #ifndef GDCMDICTS_H
15 #define GDCMDICTS_H
16
17 #include "gdcmDict.h"
18 #include "gdcmCSAHeaderDict.h"
19
20 #include <string>
21
22 namespace gdcm
23 {
28 class GDCM_EXPORT Dicts
29 {
30   friend std::ostream& operator<<(std::ostream &_os, const Dicts &d);
31 public:
32   Dicts();
33   ~Dicts();
34   Dicts &operator=(const Dicts &_val) = delete;
35   Dicts(const Dicts &_val) = delete;
36
40   // DataSet::GetPrivateCreator
42   const DictEntry &GetDictEntry(const Tag& tag, const char *owner = nullptr) const;
43
44   const DictEntry &GetDictEntry(const PrivateTag& tag) const;
45
46   //enum PublicTypes {
47   //  DICOMV3_DICT,
48   //  ACRNEMA_DICT,
49   //  NIH_DICT
50   //};
51   const Dict &GetPublicDict() const;
52
53   const PrivateDict &GetPrivateDict() const;
54   PrivateDict &GetPrivateDict();
55
56   const CSAHeaderDict &GetCSAHeaderDict() const;
57
58   bool IsEmpty()const { return GetPublicDict().IsEmpty(); }
59
60 protected:
61   typedef enum {
62     PHILIPS,
63     GEMS,
64     SIEMENS
65   //  ...
66   } ConstructorType;
67   static const char *GetConstructorString(ConstructorType type);
68
69   friend class Global;
70   void LoadDefaults();
71
72 private:
73   // Public dict:
74   Dict PublicDict;
75
76   // Private Dicts:
77   PrivateDict ShadowDict;
78
79   CSAHeaderDict CSADict;
80 };
81 //-----------------------------------------------------------------------------
82 inline std::ostream& operator<<(std::ostream &os, const Dicts &d)
83 {
84   (void)d;
```

```
85   return os;
86 }
87
88
89 } // end namespace gdcm
90
91 #endif //GDCMDICTS_H
```

11.101 gdcmGlobal.h File Reference

`#include "gdcmTypes.h"`
Include dependency graph for gdcmGlobal.h:



Classes

- class gdcm::Global
  Global.

Namespaces

- namespace gdcm

Functions

- std::ostream & gdcm::operator<< (std::ostream &os, const Global &g)

Variables

- static Global gdcm::GlobalInstance

11.102 gdcmGlobal.h

Go to the documentation of this file.

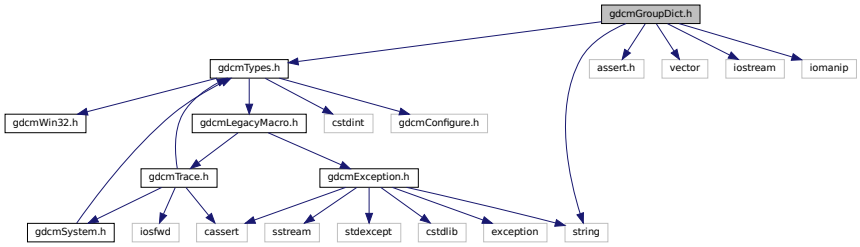
```
1 /*=========================================================================
2
3 Program:  GDCM (Grassroots DICOM). A DICOM library
4
5 Copyright (c) 2006-2011 Mathieu Malaterre
6 All rights reserved.
7 See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.
8
9 This software is distributed WITHOUT ANY WARRANTY; without even
10 the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
11 PURPOSE.  See the above copyright notice for more information.
12
13 =========================================================================*/
14 // Implementation detail was shamelessly borowed from the VTK excellent
15 // implementation of debug leak manager singleton:
16 /*=========================================================================
17
18 Program:      Visualization Toolkit
19 Module:        $RCSfile:  vtkDebugLeaks.cxx,v $
20
21 Copyright (c) Ken Martin, Will Schroeder, Bill Lorensen
22 All rights reserved.
23 See Copyright.txt or http://www.kitware.com/Copyright.htm for details.
24
25 This software is distributed WITHOUT ANY WARRANTY; without even
26 the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
27 PURPOSE.  See the above copyright notice for more information.
28
29 =========================================================================*/
30 #ifndef GDCMGLOBAL_H
31 #define GDCMGLOBAL_H
32
33 #include "gdcmTypes.h"
34
35 namespace gdcm
36 {
37 class GlobalInternal;
38 class Dicts;
39 class Defs;
49 class GDCM_EXPORT Global // why expose the symbol I think I only need to expose the instance...
50 {
51   friend std::ostream& operator<<(std::ostream &_os, const Global &g);
52 public:
53   Global();
54   ~Global();
55   Global &operator=(const Global &_val) = delete;
56   Global(const Global &_val) = delete;
57
60   Dicts const &GetDicts() const;
61   Dicts &GetDicts();
62
65   Defs const &GetDefs() const;
66
68   static Global& GetInstance();
69
73   bool LoadResourcesFiles();
74
77   bool Append(const char *path);
78
81   bool Prepend(const char *path);
82
83 protected:
85   const char *Locate(const char *resfile) const;
86
87 private:
88   // PIMPL:
89   // but we could have also directly exposed a Dicts *Internals;
90   static GlobalInternal *Internals;
91 };
92 //-----------------------------------------------------------------------------
93 inline std::ostream& operator<<(std::ostream &os, const Global &g)
94 {
95   (void)g;
96   return os;
97 }
98
```

```
99 // This instance will show up in any translation unit that uses
100 // Global or that has a singleton.    It will make sure
101 // Global is initialized before it is used and is the last
102 // static object destroyed.
103 static Global GlobalInstance;
104
105 } // end namespace gdcm
106
107 #endif //GDCMGLOBAL_H
```

11.103 gdcmGroupDict.h File Reference

```
#include "gdcmTypes.h"
#include <assert.h>
#include <vector>
#include <string>
#include <iostream>
#include <iomanip>
```

Include dependency graph for gdcmGroupDict.h:



Classes

- class gdcm::GroupDict

  Class to represent the mapping from group number to its abbreviation and name.

Namespaces

- namespace gdcm

Functions

- std::ostream & gdcm::operator<< (std::ostream &_os, const GroupDict &_val)

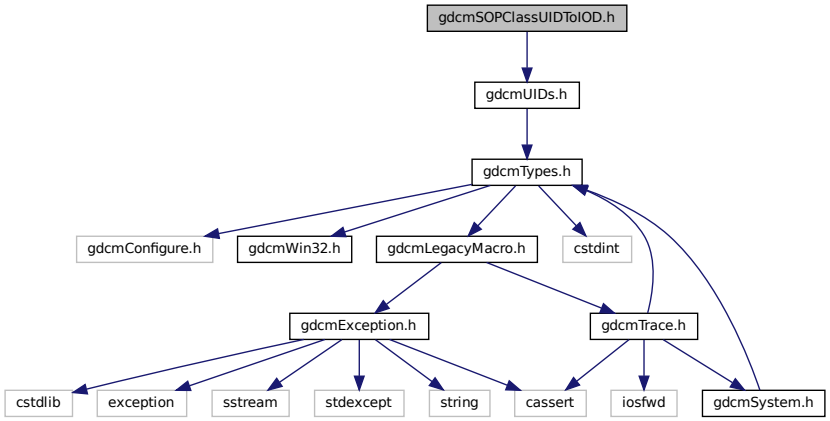
11.104 gdcmGroupDict.h

Go to the documentation of this file.

```
1 /*=========================================================================
2
3 Program:  GDCM (Grassroots DICOM). A DICOM library
4
5 Copyright (c) 2006-2011 Mathieu Malaterre
6 All rights reserved.
7 See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.
8
9 This software is distributed WITHOUT ANY WARRANTY; without even
10 the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
11 PURPOSE.  See the above copyright notice for more information.
12
13 =========================================================================*/
14
15 #ifndef GDCMGROUPDICT_H
16 #define GDCMGROUPDICT_H
17
18 #include "gdcmTypes.h"
19
20 #include <assert.h>
21 #include <vector>
22 #include <string>
23 #include <iostream>
24 #include <iomanip>
25
26 namespace gdcm
27 {
33 class GDCM_EXPORT GroupDict
34 {
35 public:
36   typedef std::vector<std::string> GroupStringVector;
37   GroupDict() { FillDefaultGroupName(); }
38   ~GroupDict() = default;
39
40   friend std::ostream& operator<<(std::ostream& _os, const GroupDict &_val);
41
42   size_t Size()const
43 {
44     assert( Names.size() == Abbreviations.size() );
45     return Names.size(); }
46
47   std::string const &GetAbbreviation(uint16_t num) const;
48
49   std::string const &GetName(uint16_t num) const;
50
51 protected:
52   void Add(std::string const &abbreviation, std::string const &name);
53   void Insert(uint16_t num, std::string const &abbreviation, std::string const &name);
54
55 private:
56   // Generated implementation, see gdcmDefaultGroupNames
57   void FillDefaultGroupName();
58
59   GroupDict &operator=(const GroupDict &_val); // purposely not implemented
60   GroupDict(const GroupDict &_val); // purposely not implemented
61
62   GroupStringVector Abbreviations;
63   GroupStringVector Names;
64 };
65 //-----------------------------------------------------------------------------
66 inline std::ostream& operator<<(std::ostream& _os, const GroupDict &_val)
67 {
68   size_t size = _val.Size();
69   for(size_t i=0; i<size; ++i)
70     {
71     _os << std::hex << std::setw(4) << std::setfill( '0' ) << i << ","
72       << _val.GetAbbreviation((uint16_t)i) << "," << _val.GetName((uint16_t)i) << "\n";
73     }
74   return _os;
75 }
76
77 } // end namespace gdcm
78
79 #endif //GDCMGROUPDICT_H
```

11.105 gdcmSOPClassUIDToIOD.h File Reference

`#include "gdcmUIDs.h"`
Include dependency graph for gdcmSOPClassUIDToIOD.h:



Classes

- class gdcm::SOPClassUIDToIOD

  Class convert a class SOP Class UID into IOD.

Namespaces

- namespace gdcm

11.106 gdcmSOPClassUIDToIOD.h

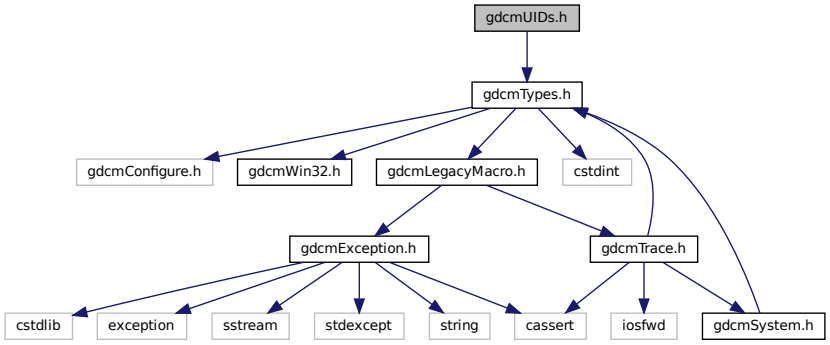
Go to the documentation of this file.

```
1 /*=========================================================================
2
3 Program:  GDCM (Grassroots DICOM). A DICOM library
4
5 Copyright (c) 2006-2011 Mathieu Malaterre
6 All rights reserved.
7 See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.
8
9 This software is distributed WITHOUT ANY WARRANTY; without even
10 the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
11 PURPOSE.  See the above copyright notice for more information.
12
13 =========================================================================*/
14
15 #ifndef GDCMSOPCLASSUIDTOIOD_H
16 #define GDCMSOPCLASSUIDTOIOD_H
17
18 #include "gdcmUIDs.h"
19
```

```
20 namespace gdcm
21 {
22
28 class GDCM_EXPORT SOPClassUIDToIOD
29 {
30 public:
33   static const char *GetIOD(UIDs const & uid);
34
36   static unsigned int GetNumberOfSOPClassToIOD();
37
38   typedef const char* const (SOPClassUIDToIODType)[2];
39   static SOPClassUIDToIODType* GetSOPClassUIDToIODs();
40
41   static SOPClassUIDToIODType& GetSOPClassUIDToIOD(unsigned int i);
42
43   static const char *GetSOPClassUIDFromIOD(const char *iod);
44   static const char *GetIODFromSOPClassUID(const char *sopclassuid);
45 };
46
47 } // end namespace gdcm
48
49 #endif //GDCMSOPCLASSUIDTOIOD_H
```

11.107 gdcmUIDs.h File Reference

`#include "gdcmTypes.h"`
Include dependency graph for gdcmUIDs.h:



This graph shows which files directly or indirectly include this file:

Classes

- class gdcm::UIDs

  all known uids

Namespaces

- namespace gdcm

Functions

- std::ostream & gdcm::operator<< (std::ostream &_os, const UIDs &uid)

11.108 gdcmUIDs.h

Go to the documentation of this file.

```
1
2 // GENERATED FILE DO NOT EDIT
3 // $ xsltproc UIDToC++.xsl Part6.xml > gdcmUIDs.h
4
5 /*=========================================================================
6
7 Program:  GDCM (Grassroots DICOM). A DICOM library
8
9 Copyright (c) 2006-2011 Mathieu Malaterre
10 All rights reserved.
11 See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.
12
13 This software is distributed WITHOUT ANY WARRANTY; without even
14 the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
15 PURPOSE.  See the above copyright notice for more information.
16
17 =========================================================================*/
18
19 #ifndef GDCMUIDS_H
20 #define GDCMUIDS_H
21
22 #include "gdcmTypes.h"
23
24 namespace gdcm
25 {
26
30 class GDCM_EXPORT UIDs
31 {
32 public:
33   typedef enum {
34 uid_1_2_840_10008_1_1 = 1, // Verification SOP Class
35 uid_1_2_840_10008_1_2 = 2, // Implicit VR Little Endian:  Default Transfer Syntax for DICOM
36 uid_1_2_840_10008_1_2_1 = 3, // Explicit VR Little Endian
37 uid_1_2_840_10008_1_2_1_99 = 4, // Deflated Explicit VR Little Endian
38 uid_1_2_840_10008_1_2_2 = 5, // Explicit VR Big Endian
39 uid_1_2_840_10008_1_2_4_50 = 6, // JPEG Baseline (Process 1):  Default Transfer Syntax for Lossy JPEG 8 Bit
      Image Compression
40 uid_1_2_840_10008_1_2_4_51 = 7, // JPEG Extended (Process 2 & 4):  Default Transfer Syntax for Lossy JPEG 12
      Bit Image Compression (Process 4 only)
41 uid_1_2_840_10008_1_2_4_52 = 8, // JPEG Extended (Process 3 & 5)
42 uid_1_2_840_10008_1_2_4_53 = 9, // JPEG Spectral Selection, Non-Hierarchical (Process 6 & 8)
43 uid_1_2_840_10008_1_2_4_54 = 10, // JPEG Spectral Selection, Non-Hierarchical (Process 7 & 9)
44 uid_1_2_840_10008_1_2_4_55 = 11, // JPEG Full Progression, Non-Hierarchical (Process 10 & 12)
45 uid_1_2_840_10008_1_2_4_56 = 12, // JPEG Full Progression, Non-Hierarchical (Process 11 & 13)
46 uid_1_2_840_10008_1_2_4_57 = 13, // JPEG Lossless, Non-Hierarchical (Process 14)
47 uid_1_2_840_10008_1_2_4_58 = 14, // JPEG Lossless, Non-Hierarchical (Process 15)
48 uid_1_2_840_10008_1_2_4_59 = 15, // JPEG Extended, Hierarchical (Process 16 & 18)
49 uid_1_2_840_10008_1_2_4_60 = 16, // JPEG Extended, Hierarchical (Process 17 & 19)
50 uid_1_2_840_10008_1_2_4_61 = 17, // JPEG Spectral Selection, Hierarchical (Process 20 & 22)
51 uid_1_2_840_10008_1_2_4_62 = 18, // JPEG Spectral Selection, Hierarchical (Process 21 & 23)
52 uid_1_2_840_10008_1_2_4_63 = 19, // JPEG Full Progression, Hierarchical (Process 24 & 26)
53 uid_1_2_840_10008_1_2_4_64 = 20, // JPEG Full Progression, Hierarchical (Process 25 & 27)
54 uid_1_2_840_10008_1_2_4_65 = 21, // JPEG Lossless, Hierarchical (Process 28)
55 uid_1_2_840_10008_1_2_4_66 = 22, // JPEG Lossless, Hierarchical (Process 29)
56 uid_1_2_840_10008_1_2_4_70 = 23, // JPEG Lossless, Non-Hierarchical, First-Order Prediction (Process 14
      [Selection Value 1]):  Default Transfer Syntax for Lossless JPEG Image Compression
57 uid_1_2_840_10008_1_2_4_80 = 24, // JPEG-LS Lossless Image Compression
58 uid_1_2_840_10008_1_2_4_81 = 25, // JPEG-LS Lossy (Near-Lossless) Image Compression
59 uid_1_2_840_10008_1_2_4_90 = 26, // JPEG 2000 Image Compression (Lossless Only)
```

```
60 uid_1_2_840_10008_1_2_4_91 = 27, // JPEG 2000 Image Compression
61 uid_1_2_840_10008_1_2_4_92 = 28, // JPEG 2000 Part 2 Multi-component Image Compression (Lossless Only)
62 uid_1_2_840_10008_1_2_4_93 = 29, // JPEG 2000 Part 2 Multi-component Image Compression
63 uid_1_2_840_10008_1_2_4_94 = 30, // JPIP Referenced
64 uid_1_2_840_10008_1_2_4_95 = 31, // JPIP Referenced Deflate
65 uid_1_2_840_10008_1_2_4_100 = 32, // MPEG2 Main Profile @ Main Level
66 uid_1_2_840_10008_1_2_5 = 33, // RLE Lossless
67 uid_1_2_840_10008_1_2_6_1 = 34, // RFC 2557 MIME encapsulation
68 uid_1_2_840_10008_1_2_6_2 = 35, // XML Encoding
69 uid_1_2_840_10008_1_3_10 = 36, // Media Storage Directory Storage
70 uid_1_2_840_10008_1_4_1_1 = 37, // Talairach Brain Atlas Frame of Reference
71 uid_1_2_840_10008_1_4_1_2 = 38, // SPM2 T1 Frame of Reference
72 uid_1_2_840_10008_1_4_1_3 = 39, // SPM2 T2 Frame of Reference
73 uid_1_2_840_10008_1_4_1_4 = 40, // SPM2 PD Frame of Reference
74 uid_1_2_840_10008_1_4_1_5 = 41, // SPM2 EPI Frame of Reference
75 uid_1_2_840_10008_1_4_1_6 = 42, // SPM2 FIL T1 Frame of Reference
76 uid_1_2_840_10008_1_4_1_7 = 43, // SPM2 PET Frame of Reference
77 uid_1_2_840_10008_1_4_1_8 = 44, // SPM2 TRANSM Frame of Reference
78 uid_1_2_840_10008_1_4_1_9 = 45, // SPM2 SPECT Frame of Reference
79 uid_1_2_840_10008_1_4_1_10 = 46, // SPM2 GRAY Frame of Reference
80 uid_1_2_840_10008_1_4_1_11 = 47, // SPM2 WHITE Frame of Reference
81 uid_1_2_840_10008_1_4_1_12 = 48, // SPM2 CSF Frame of Reference
82 uid_1_2_840_10008_1_4_1_13 = 49, // SPM2 BRAINMASK Frame of Reference
83 uid_1_2_840_10008_1_4_1_14 = 50, // SPM2 AVG305T1 Frame of Reference
84 uid_1_2_840_10008_1_4_1_15 = 51, // SPM2 AVG152T1 Frame of Reference
85 uid_1_2_840_10008_1_4_1_16 = 52, // SPM2 AVG152T2 Frame of Reference
86 uid_1_2_840_10008_1_4_1_17 = 53, // SPM2 AVG152PD Frame of Reference
87 uid_1_2_840_10008_1_4_1_18 = 54, // SPM2 SINGLESUBJT1 Frame of Reference
88 uid_1_2_840_10008_1_4_2_1 = 55, // ICBM 452 T1 Frame of Reference
89 uid_1_2_840_10008_1_4_2_2 = 56, // ICBM Single Subject MRI Frame of Reference
90 uid_1_2_840_10008_1_9 = 57, // Basic Study Content Notification SOP Class
91 uid_1_2_840_10008_1_20_1 = 58, // Storage Commitment Push Model SOP Class
92 uid_1_2_840_10008_1_20_1_1 = 59, // Storage Commitment Push Model SOP Instance
93 uid_1_2_840_10008_1_20_2 = 60, // Storage Commitment Pull Model SOP Class
94 uid_1_2_840_10008_1_20_2_1 = 61, // Storage Commitment Pull Model SOP Instance
95 uid_1_2_840_10008_1_40 = 62, // Procedural Event Logging SOP Class
96 uid_1_2_840_10008_1_40_1 = 63, // Procedural Event Logging SOP Instance
97 uid_1_2_840_10008_1_42 = 64, // Substance Administration Logging SOP Class
98 uid_1_2_840_10008_1_42_1 = 65, // Substance Administration Logging SOP Instance
99 uid_1_2_840_10008_2_6_1 = 66, // DICOM UID Registry
100 uid_1_2_840_10008_2_16_4 = 67, // DICOM Controlled Terminology
101 uid_1_2_840_10008_3_1_1_1 = 68, // DICOM Application Context Name
102 uid_1_2_840_10008_3_1_2_1_1 = 69, // Detached Patient Management SOP Class
103 uid_1_2_840_10008_3_1_2_1_4 = 70, // Detached Patient Management Meta SOP Class
104 uid_1_2_840_10008_3_1_2_2_1 = 71, // Detached Visit Management SOP Class
105 uid_1_2_840_10008_3_1_2_3_1 = 72, // Detached Study Management SOP Class
106 uid_1_2_840_10008_3_1_2_3_2 = 73, // Study Component Management SOP Class
107 uid_1_2_840_10008_3_1_2_3_3 = 74, // Modality Performed Procedure Step SOP Class
108 uid_1_2_840_10008_3_1_2_3_4 = 75, // Modality Performed Procedure Step Retrieve SOP Class
109 uid_1_2_840_10008_3_1_2_3_5 = 76, // Modality Performed Procedure Step Notification SOP Class
110 uid_1_2_840_10008_3_1_2_5_1 = 77, // Detached Results Management SOP Class
111 uid_1_2_840_10008_3_1_2_5_4 = 78, // Detached Results Management Meta SOP Class
112 uid_1_2_840_10008_3_1_2_5_5 = 79, // Detached Study Management Meta SOP Class
113 uid_1_2_840_10008_3_1_2_6_1 = 80, // Detached Interpretation Management SOP Class
114 uid_1_2_840_10008_4_2 = 81, // Storage Service Class
115 uid_1_2_840_10008_5_1_1_1 = 82, // Basic Film Session SOP Class
116 uid_1_2_840_10008_5_1_1_2 = 83, // Basic Film Box SOP Class
117 uid_1_2_840_10008_5_1_1_4 = 84, // Basic Grayscale Image Box SOP Class
118 uid_1_2_840_10008_5_1_1_4_1 = 85, // Basic Color Image Box SOP Class
119 uid_1_2_840_10008_5_1_1_4_2 = 86, // Referenced Image Box SOP Class
120 uid_1_2_840_10008_5_1_1_9 = 87, // Basic Grayscale Print Management Meta SOP Class
121 uid_1_2_840_10008_5_1_1_9_1 = 88, // Referenced Grayscale Print Management Meta SOP Class
122 uid_1_2_840_10008_5_1_1_14 = 89, // Print Job SOP Class
123 uid_1_2_840_10008_5_1_1_15 = 90, // Basic Annotation Box SOP Class
124 uid_1_2_840_10008_5_1_1_16 = 91, // Printer SOP Class
125 uid_1_2_840_10008_5_1_1_16_376 = 92, // Printer Configuration Retrieval SOP Class
126 uid_1_2_840_10008_5_1_1_17 = 93, // Printer SOP Instance
127 uid_1_2_840_10008_5_1_1_17_376 = 94, // Printer Configuration Retrieval SOP Instance
128 uid_1_2_840_10008_5_1_1_18 = 95, // Basic Color Print Management Meta SOP Class
129 uid_1_2_840_10008_5_1_1_18_1 = 96, // Referenced Color Print Management Meta SOP Class
130 uid_1_2_840_10008_5_1_1_22 = 97, // VOI LUT Box SOP Class
131 uid_1_2_840_10008_5_1_1_23 = 98, // Presentation LUT SOP Class
132 uid_1_2_840_10008_5_1_1_24 = 99, // Image Overlay Box SOP Class
133 uid_1_2_840_10008_5_1_1_24_1 = 100, // Basic Print Image Overlay Box SOP Class
134 uid_1_2_840_10008_5_1_1_25 = 101, // Print Queue SOP Instance
135 uid_1_2_840_10008_5_1_1_26 = 102, // Print Queue Management SOP Class
136 uid_1_2_840_10008_5_1_1_27 = 103, // Stored Print Storage SOP Class
137 uid_1_2_840_10008_5_1_1_29 = 104, // Hardcopy Grayscale Image Storage SOP Class
138 uid_1_2_840_10008_5_1_1_30 = 105, // Hardcopy Color Image Storage SOP Class
139 uid_1_2_840_10008_5_1_1_31 = 106, // Pull Print Request SOP Class
140 uid_1_2_840_10008_5_1_1_32 = 107, // Pull Stored Print Management Meta SOP Class
```

```
141 uid_1_2_840_10008_5_1_1_33 = 108, // Media Creation Management SOP Class UID
142 uid_1_2_840_10008_5_1_4_1_1_1 = 109, // Computed Radiography Image Storage
143 uid_1_2_840_10008_5_1_4_1_1_1_1 = 110, // Digital X-Ray Image Storage - For Presentation
144 uid_1_2_840_10008_5_1_4_1_1_1_1_1 = 111, // Digital X-Ray Image Storage - For Processing
145 uid_1_2_840_10008_5_1_4_1_1_1_2 = 112, // Digital Mammography X-Ray Image Storage - For Presentation
146 uid_1_2_840_10008_5_1_4_1_1_1_2_1 = 113, // Digital Mammography X-Ray Image Storage - For Processing
147 uid_1_2_840_10008_5_1_4_1_1_1_3 = 114, // Digital Intra-oral X-Ray Image Storage - For Presentation
148 uid_1_2_840_10008_5_1_4_1_1_1_3_1 = 115, // Digital Intra-oral X-Ray Image Storage - For Processing
149 uid_1_2_840_10008_5_1_4_1_1_2 = 116, // CT Image Storage
150 uid_1_2_840_10008_5_1_4_1_1_2_1 = 117, // Enhanced CT Image Storage
151 uid_1_2_840_10008_5_1_4_1_1_3 = 118, // Ultrasound Multi-frame Image Storage
152 uid_1_2_840_10008_5_1_4_1_1_3_1 = 119, // Ultrasound Multi-frame Image Storage
153 uid_1_2_840_10008_5_1_4_1_1_4 = 120, // MR Image Storage
154 uid_1_2_840_10008_5_1_4_1_1_4_1 = 121, // Enhanced MR Image Storage
155 uid_1_2_840_10008_5_1_4_1_1_4_2 = 122, // MR Spectroscopy Storage
156 uid_1_2_840_10008_5_1_4_1_1_5 = 123, // Nuclear Medicine Image Storage
157 uid_1_2_840_10008_5_1_4_1_1_6 = 124, // Ultrasound Image Storage
158 uid_1_2_840_10008_5_1_4_1_1_6_1 = 125, // Ultrasound Image Storage
159 uid_1_2_840_10008_5_1_4_1_1_7 = 126, // Secondary Capture Image Storage
160 uid_1_2_840_10008_5_1_4_1_1_7_1 = 127, // Multi-frame Single Bit Secondary Capture Image Storage
161 uid_1_2_840_10008_5_1_4_1_1_7_2 = 128, // Multi-frame Grayscale Byte Secondary Capture Image Storage
162 uid_1_2_840_10008_5_1_4_1_1_7_3 = 129, // Multi-frame Grayscale Word Secondary Capture Image Storage
163 uid_1_2_840_10008_5_1_4_1_1_7_4 = 130, // Multi-frame True Color Secondary Capture Image Storage
164 uid_1_2_840_10008_5_1_4_1_1_8 = 131, // Standalone Overlay Storage
165 uid_1_2_840_10008_5_1_4_1_1_9 = 132, // Standalone Curve Storage
166 uid_1_2_840_10008_5_1_4_1_1_9_1 = 133, // Waveform Storage - Trial
167 uid_1_2_840_10008_5_1_4_1_1_9_1_1 = 134, // 12-lead ECG Waveform Storage
168 uid_1_2_840_10008_5_1_4_1_1_9_1_2 = 135, // General ECG Waveform Storage
169 uid_1_2_840_10008_5_1_4_1_1_9_1_3 = 136, // Ambulatory ECG Waveform Storage
170 uid_1_2_840_10008_5_1_4_1_1_9_2_1 = 137, // Hemodynamic Waveform Storage
171 uid_1_2_840_10008_5_1_4_1_1_9_3_1 = 138, // Cardiac Electrophysiology Waveform Storage
172 uid_1_2_840_10008_5_1_4_1_1_9_4_1 = 139, // Basic Voice Audio Waveform Storage
173 uid_1_2_840_10008_5_1_4_1_1_10 = 140, // Standalone Modality LUT Storage
174 uid_1_2_840_10008_5_1_4_1_1_11 = 141, // Standalone VOI LUT Storage
175 uid_1_2_840_10008_5_1_4_1_1_11_1 = 142, // Grayscale Softcopy Presentation State Storage SOP Class
176 uid_1_2_840_10008_5_1_4_1_1_11_2 = 143, // Color Softcopy Presentation State Storage SOP Class
177 uid_1_2_840_10008_5_1_4_1_1_11_3 = 144, // Pseudo-Color Softcopy Presentation State Storage SOP Class
178 uid_1_2_840_10008_5_1_4_1_1_11_4 = 145, // Blending Softcopy Presentation State Storage SOP Class
179 uid_1_2_840_10008_5_1_4_1_1_12_1 = 146, // X-Ray Angiographic Image Storage
180 uid_1_2_840_10008_5_1_4_1_1_12_1_1 = 147, // Enhanced XA Image Storage
181 uid_1_2_840_10008_5_1_4_1_1_12_2 = 148, // X-Ray Radiofluoroscopic Image Storage
182 uid_1_2_840_10008_5_1_4_1_1_12_2_1 = 149, // Enhanced XRF Image Storage
183 uid_1_2_840_10008_5_1_4_1_1_13_1_1 = 150, // X-Ray 3D Angiographic Image Storage
184 uid_1_2_840_10008_5_1_4_1_1_13_1_2 = 151, // X-Ray 3D Craniofacial Image Storage
185 uid_1_2_840_10008_5_1_4_1_1_12_3 = 152, // X-Ray Angiographic Bi-Plane Image Storage
186 uid_1_2_840_10008_5_1_4_1_1_20 = 153, // Nuclear Medicine Image Storage
187 uid_1_2_840_10008_5_1_4_1_1_66 = 154, // Raw Data Storage
188 uid_1_2_840_10008_5_1_4_1_1_66_1 = 155, // Spatial Registration Storage
189 uid_1_2_840_10008_5_1_4_1_1_66_2 = 156, // Spatial Fiducials Storage
190 uid_1_2_840_10008_5_1_4_1_1_66_3 = 157, // Deformable Spatial Registration Storage
191 uid_1_2_840_10008_5_1_4_1_1_66_4 = 158, // Segmentation Storage
192 uid_1_2_840_10008_5_1_4_1_1_67 = 159, // Real World Value Mapping Storage
193 uid_1_2_840_10008_5_1_4_1_1_77_1 = 160, // VL Image Storage - Trial
194 uid_1_2_840_10008_5_1_4_1_1_77_2 = 161, // VL Multi-frame Image Storage - Trial
195 uid_1_2_840_10008_5_1_4_1_1_77_1_1 = 162, // VL Endoscopic Image Storage
196 uid_1_2_840_10008_5_1_4_1_1_77_1_1_1 = 163, // Video Endoscopic Image Storage
197 uid_1_2_840_10008_5_1_4_1_1_77_1_2 = 164, // VL Microscopic Image Storage
198 uid_1_2_840_10008_5_1_4_1_1_77_1_2_1 = 165, // Video Microscopic Image Storage
199 uid_1_2_840_10008_5_1_4_1_1_77_1_3 = 166, // VL Slide-Coordinates Microscopic Image Storage
200 uid_1_2_840_10008_5_1_4_1_1_77_1_4 = 167, // VL Photographic Image Storage
201 uid_1_2_840_10008_5_1_4_1_1_77_1_4_1 = 168, // Video Photographic Image Storage
202 uid_1_2_840_10008_5_1_4_1_1_77_1_5_1 = 169, // Ophthalmic Photography 8 Bit Image Storage
203 uid_1_2_840_10008_5_1_4_1_1_77_1_5_2 = 170, // Ophthalmic Photography 16 Bit Image Storage
204 uid_1_2_840_10008_5_1_4_1_1_77_1_5_3 = 171, // Stereometric Relationship Storage
205 uid_1_2_840_10008_5_1_4_1_1_77_1_5_4 = 172, // Ophthalmic Tomography Image Storage
206 uid_1_2_840_10008_5_1_4_1_1_88_1 = 173, // Text SR Storage - Trial
207 uid_1_2_840_10008_5_1_4_1_1_88_2 = 174, // Audio SR Storage - Trial
208 uid_1_2_840_10008_5_1_4_1_1_88_3 = 175, // Detail SR Storage - Trial
209 uid_1_2_840_10008_5_1_4_1_1_88_4 = 176, // Comprehensive SR Storage - Trial
210 uid_1_2_840_10008_5_1_4_1_1_88_11 = 177, // Basic Text SR Storage
211 uid_1_2_840_10008_5_1_4_1_1_88_22 = 178, // Enhanced SR Storage
212 uid_1_2_840_10008_5_1_4_1_1_88_33 = 179, // Comprehensive SR Storage
213 uid_1_2_840_10008_5_1_4_1_1_88_40 = 180, // Procedure Log Storage
214 uid_1_2_840_10008_5_1_4_1_1_88_50 = 181, // Mammography CAD SR Storage
215 uid_1_2_840_10008_5_1_4_1_1_88_59 = 182, // Key Object Selection Document Storage
216 uid_1_2_840_10008_5_1_4_1_1_88_65 = 183, // Chest CAD SR Storage
217 uid_1_2_840_10008_5_1_4_1_1_88_67 = 184, // X-Ray Radiation Dose SR Storage
218 uid_1_2_840_10008_5_1_4_1_1_104_1 = 185, // Encapsulated PDF Storage
219 uid_1_2_840_10008_5_1_4_1_1_104_2 = 186, // Encapsulated CDA Storage
220 uid_1_2_840_10008_5_1_4_1_1_128 = 187, // Positron Emission Tomography Image Storage
221 uid_1_2_840_10008_5_1_4_1_1_129 = 188, // Standalone PET Curve Storage
```

```
222 uid_1_2_840_10008_5_1_4_1_1_481_1 = 189, // RT Image Storage
223 uid_1_2_840_10008_5_1_4_1_1_481_2 = 190, // RT Dose Storage
224 uid_1_2_840_10008_5_1_4_1_1_481_3 = 191, // RT Structure Set Storage
225 uid_1_2_840_10008_5_1_4_1_1_481_4 = 192, // RT Beams Treatment Record Storage
226 uid_1_2_840_10008_5_1_4_1_1_481_5 = 193, // RT Plan Storage
227 uid_1_2_840_10008_5_1_4_1_1_481_6 = 194, // RT Brachy Treatment Record Storage
228 uid_1_2_840_10008_5_1_4_1_1_481_7 = 195, // RT Treatment Summary Record Storage
229 uid_1_2_840_10008_5_1_4_1_1_481_8 = 196, // RT Ion Plan Storage
230 uid_1_2_840_10008_5_1_4_1_1_481_9 = 197, // RT Ion Beams Treatment Record Storage
231 uid_1_2_840_10008_5_1_4_1_2_1_1 = 198, // Patient Root Query/Retrieve Information Model - FIND
232 uid_1_2_840_10008_5_1_4_1_2_1_2 = 199, // Patient Root Query/Retrieve Information Model - MOVE
233 uid_1_2_840_10008_5_1_4_1_2_1_3 = 200, // Patient Root Query/Retrieve Information Model - GET
234 uid_1_2_840_10008_5_1_4_1_2_2_1 = 201, // Study Root Query/Retrieve Information Model - FIND
235 uid_1_2_840_10008_5_1_4_1_2_2_2 = 202, // Study Root Query/Retrieve Information Model - MOVE
236 uid_1_2_840_10008_5_1_4_1_2_2_3 = 203, // Study Root Query/Retrieve Information Model - GET
237 uid_1_2_840_10008_5_1_4_1_2_3_1 = 204, // Patient/Study Only Query/Retrieve Information Model - FIND
238 uid_1_2_840_10008_5_1_4_1_2_3_2 = 205, // Patient/Study Only Query/Retrieve Information Model - MOVE
239 uid_1_2_840_10008_5_1_4_1_2_3_3 = 206, // Patient/Study Only Query/Retrieve Information Model - GET
240 uid_1_2_840_10008_5_1_4_31 = 207, // Modality Worklist Information Model - FIND
241 uid_1_2_840_10008_5_1_4_32_1 = 208, // General Purpose Worklist Information Model - FIND
242 uid_1_2_840_10008_5_1_4_32_2 = 209, // General Purpose Scheduled Procedure Step SOP Class
243 uid_1_2_840_10008_5_1_4_32_3 = 210, // General Purpose Performed Procedure Step SOP Class
244 uid_1_2_840_10008_5_1_4_32 = 211, // General Purpose Worklist Management Meta SOP Class
245 uid_1_2_840_10008_5_1_4_33 = 212, // Instance Availability Notification SOP Class
246 uid_1_2_840_10008_5_1_4_34_1 = 213, // RT Beams Delivery Instruction Storage (Supplement 74 Frozen Draft)
247 uid_1_2_840_10008_5_1_4_34_2 = 214, // RT Conventional Machine Verification (Supplement 74 Frozen Draft)
248 uid_1_2_840_10008_5_1_4_34_3 = 215, // RT Ion Machine Verification (Supplement 74 Frozen Draft)
249 uid_1_2_840_10008_5_1_4_34_4 = 216, // Unified Worklist and Procedure Step Service Class
250 uid_1_2_840_10008_5_1_4_34_4_1 = 217, // Unified Procedure Step - Push SOP Class
251 uid_1_2_840_10008_5_1_4_34_4_2 = 218, // Unified Procedure Step - Watch SOP Class
252 uid_1_2_840_10008_5_1_4_34_4_3 = 219, // Unified Procedure Step - Pull SOP Class
253 uid_1_2_840_10008_5_1_4_34_4_4 = 220, // Unified Procedure Step - Event SOP Class
254 uid_1_2_840_10008_5_1_4_34_5 = 221, // Unified Worklist and Procedure Step SOP Instance
255 uid_1_2_840_10008_5_1_4_37_1 = 222, // General Relevant Patient Information Query
256 uid_1_2_840_10008_5_1_4_37_2 = 223, // Breast Imaging Relevant Patient Information Query
257 uid_1_2_840_10008_5_1_4_37_3 = 224, // Cardiac Relevant Patient Information Query
258 uid_1_2_840_10008_5_1_4_38_1 = 225, // Hanging Protocol Storage
259 uid_1_2_840_10008_5_1_4_38_2 = 226, // Hanging Protocol Information Model - FIND
260 uid_1_2_840_10008_5_1_4_38_3 = 227, // Hanging Protocol Information Model - MOVE
261 uid_1_2_840_10008_5_1_4_41 = 228, // Product Characteristics Query SOP Class
262 uid_1_2_840_10008_5_1_4_42 = 229, // Substance Approval Query SOP Class
263 uid_1_2_840_10008_15_0_3_1 = 230, // dicomDeviceName
264 uid_1_2_840_10008_15_0_3_2 = 231, // dicomDescription
265 uid_1_2_840_10008_15_0_3_3 = 232, // dicomManufacturer
266 uid_1_2_840_10008_15_0_3_4 = 233, // dicomManufacturerModelName
267 uid_1_2_840_10008_15_0_3_5 = 234, // dicomSoftwareVersion
268 uid_1_2_840_10008_15_0_3_6 = 235, // dicomVendorData
269 uid_1_2_840_10008_15_0_3_7 = 236, // dicomAETitle
270 uid_1_2_840_10008_15_0_3_8 = 237, // dicomNetworkConnectionReference
271 uid_1_2_840_10008_15_0_3_9 = 238, // dicomApplicationCluster
272 uid_1_2_840_10008_15_0_3_10 = 239, // dicomAssociationInitiator
273 uid_1_2_840_10008_15_0_3_11 = 240, // dicomAssociationAcceptor
274 uid_1_2_840_10008_15_0_3_12 = 241, // dicomHostname
275 uid_1_2_840_10008_15_0_3_13 = 242, // dicomPort
276 uid_1_2_840_10008_15_0_3_14 = 243, // dicomSOPClass
277 uid_1_2_840_10008_15_0_3_15 = 244, // dicomTransferRole
278 uid_1_2_840_10008_15_0_3_16 = 245, // dicomTransferSyntax
279 uid_1_2_840_10008_15_0_3_17 = 246, // dicomPrimaryDeviceType
280 uid_1_2_840_10008_15_0_3_18 = 247, // dicomRelatedDeviceReference
281 uid_1_2_840_10008_15_0_3_19 = 248, // dicomPreferredCalledAETitle
282 uid_1_2_840_10008_15_0_3_20 = 249, // dicomTLSCyphersuite
283 uid_1_2_840_10008_15_0_3_21 = 250, // dicomAuthorizedNodeCertificateReference
284 uid_1_2_840_10008_15_0_3_22 = 251, // dicomThisNodeCertificateReference
285 uid_1_2_840_10008_15_0_3_23 = 252, // dicomInstalled
286 uid_1_2_840_10008_15_0_3_24 = 253, // dicomStationName
287 uid_1_2_840_10008_15_0_3_25 = 254, // dicomDeviceSerialNumber
288 uid_1_2_840_10008_15_0_3_26 = 255, // dicomInstitutionName
289 uid_1_2_840_10008_15_0_3_27 = 256, // dicomInstitutionAddress
290 uid_1_2_840_10008_15_0_3_28 = 257, // dicomInstitutionDepartmentName
291 uid_1_2_840_10008_15_0_3_29 = 258, // dicomIssuerOfPatientID
292 uid_1_2_840_10008_15_0_3_30 = 259, // dicomPreferredCallingAETitle
293 uid_1_2_840_10008_15_0_3_31 = 260, // dicomSupportedCharacterSet
294 uid_1_2_840_10008_15_0_4_1 = 261, // dicomConfigurationRoot
295 uid_1_2_840_10008_15_0_4_2 = 262, // dicomDevicesRoot
296 uid_1_2_840_10008_15_0_4_3 = 263, // dicomUniqueAETitlesRegistryRoot
297 uid_1_2_840_10008_15_0_4_4 = 264, // dicomDevice
298 uid_1_2_840_10008_15_0_4_5 = 265, // dicomNetworkAE
299 uid_1_2_840_10008_15_0_4_6 = 266, // dicomNetworkConnection
300 uid_1_2_840_10008_15_0_4_7 = 267, // dicomUniqueAETitle
301 uid_1_2_840_10008_15_0_4_8 = 268, // dicomTransferCapability
302 //
```

```
303 uid_1_2_840_10008_5_1_4_1_1_77_1_6 = 269, // VL Whole Slide Microscopy
304 uid_1_2_840_10008_5_1_4_1_1_6_2    = 270, // Enhanced US Volume Storage
305 uid_1_2_840_10008_5_1_4_1_1_66_5   = 271, // Surface Segmentation Storage
306 uid_1_2_840_10008_5_1_4_1_1_13_1_3 = 272, // Breast Tomosynthesis Image Storage
307 uid_1_2_840_10008_5_1_4_1_1_2_2    = 273, // Legacy Converted Enhanced CT
308 uid_1_2_840_10008_5_1_4_1_1_4_4    = 274, // Legacy Converted Enhanced MR
309 uid_1_2_840_10008_5_1_4_1_1_128_1  = 275, // Legacy Converted Enhanced PET
310 uid_1_2_840_10008_1_2_4_101        = 276, // MPEG2 Main Profile High Level
311 uid_1_2_840_10008_1_2_4_102        = 277, // MPEG-4 AVC/H.264 High Profile Lev.  4.1
312 uid_1_2_840_10008_1_2_4_103        = 278, // MPEG-4 AVC/H.264 BD-comp High Profile Lev.  4.1
313
315 //
316 // 2019b
317 //
318 uid_1_2_840_10008_1_5_2              = 279,
319 uid_1_2_840_10008_1_5_3              = 280,
320 uid_1_2_840_10008_1_5_4              = 281,
321 uid_1_2_840_10008_1_5_5              = 282,
322 uid_1_2_840_10008_1_5_6              = 283,
323 uid_1_2_840_10008_1_5_7              = 284,
324 uid_1_2_840_10008_1_5_8              = 285,
325 uid_1_2_840_10008_1_20               = 286,
326 uid_1_2_840_10008_2_16_5             = 287,
327 uid_1_2_840_10008_2_16_6             = 288,
328 uid_1_2_840_10008_2_16_7             = 289,
329 uid_1_2_840_10008_2_16_8             = 290,
330 uid_1_2_840_10008_2_16_9             = 291,
331 uid_1_2_840_10008_2_16_10            = 292,
332 uid_1_2_840_10008_2_16_11            = 293,
333 uid_1_2_840_10008_2_16_12            = 294,
334 uid_1_2_840_10008_2_16_13            = 295,
335 uid_1_2_840_10008_2_16_14            = 296,
336 uid_1_2_840_10008_5_1_1_40           = 297,
337 uid_1_2_840_10008_5_1_1_40_1         = 298,
338 uid_1_2_840_10008_5_1_4_1_1_9_4_2    = 299,
339 uid_1_2_840_10008_5_1_4_1_1_9_5_1    = 300,
340 uid_1_2_840_10008_5_1_4_1_1_9_6_1    = 301,
341 uid_1_2_840_10008_5_1_4_1_1_11_5     = 302,
342 uid_1_2_840_10008_5_1_4_1_1_11_6     = 303,
343 uid_1_2_840_10008_1_2_4_104          = 304,
344 uid_1_2_840_10008_1_2_4_105          = 305,
345 uid_1_2_840_10008_1_2_4_106          = 306,
346 uid_1_2_840_10008_1_2_4_107          = 307,
347 uid_1_2_840_10008_1_2_4_108          = 308,
348 uid_1_2_840_10008_1_5_1              = 309,
349 uid_1_2_840_10008_5_1_4_1_1_11_7     = 310,
350 uid_1_2_840_10008_5_1_4_1_1_11_8     = 311,
351 uid_1_2_840_10008_5_1_4_1_1_11_9     = 312,
352 uid_1_2_840_10008_5_1_4_1_1_11_10    = 313,
353 uid_1_2_840_10008_5_1_4_1_1_11_11    = 314,
354 uid_1_2_840_10008_5_1_4_1_1_12_77    = 315,
355 uid_1_2_840_10008_5_1_4_1_1_13_1_4   = 316,
356 uid_1_2_840_10008_5_1_4_1_1_13_1_5   = 317,
357 uid_1_2_840_10008_5_1_4_1_1_14_1     = 318,
358 uid_1_2_840_10008_5_1_4_1_1_14_2     = 319,
359 uid_1_2_840_10008_5_1_4_1_1_30       = 320,
360 uid_1_2_840_10008_5_1_4_1_1_40       = 321,
361 uid_1_2_840_10008_5_1_4_1_1_66_6     = 322,
362 uid_1_2_840_10008_5_1_4_1_1_68_1     = 323,
363 uid_1_2_840_10008_5_1_4_1_1_68_2     = 324,
364 uid_1_2_840_10008_5_1_4_1_1_77_1_5_5 = 325,
365 uid_1_2_840_10008_5_1_4_1_1_77_1_5_6 = 326,
366 uid_1_2_840_10008_5_1_4_1_1_77_1_5_7 = 327,
367 uid_1_2_840_10008_5_1_4_1_1_77_1_5_8 = 328,
368 uid_1_2_840_10008_5_1_4_1_1_78_1     = 329,
369 uid_1_2_840_10008_5_1_4_1_1_78_2     = 330,
370 uid_1_2_840_10008_5_1_4_1_1_78_3     = 331,
371 uid_1_2_840_10008_5_1_4_1_1_78_4     = 332,
372 uid_1_2_840_10008_5_1_4_1_1_78_5     = 333,
373 uid_1_2_840_10008_5_1_4_1_1_78_6     = 334,
374 uid_1_2_840_10008_5_1_4_1_1_78_7     = 335,
375 uid_1_2_840_10008_5_1_4_1_1_78_8     = 336,
376 uid_1_2_840_10008_5_1_4_1_1_79_1     = 337,
377 uid_1_2_840_10008_5_1_4_1_1_80_1     = 338,
378 uid_1_2_840_10008_5_1_4_1_1_81_1     = 339,
379 uid_1_2_840_10008_5_1_4_1_1_82_1     = 340,
380 uid_1_2_840_10008_5_1_4_1_1_88_34    = 341,
381 uid_1_2_840_10008_5_1_4_1_1_88_35    = 342,
382 uid_1_2_840_10008_5_1_4_1_1_88_68    = 343,
383 uid_1_2_840_10008_5_1_4_1_1_88_69    = 344,
384 uid_1_2_840_10008_5_1_4_1_1_88_70    = 345,
```

```
385 uid_1_2_840_10008_5_1_4_1_1_88_71    = 346,
386 uid_1_2_840_10008_5_1_4_1_1_88_72    = 347,
387 uid_1_2_840_10008_5_1_4_1_1_88_73    = 348,
388 uid_1_2_840_10008_5_1_4_1_1_88_74    = 349,
389 uid_1_2_840_10008_5_1_4_1_1_88_75    = 350,
390 uid_1_2_840_10008_5_1_4_1_1_90_1     = 351,
391 uid_1_2_840_10008_5_1_4_1_1_104_3    = 352,
392 uid_1_2_840_10008_5_1_4_1_1_130      = 353,
393 uid_1_2_840_10008_5_1_4_1_1_131      = 354,
394 uid_1_2_840_10008_5_1_4_1_1_200_1    = 355,
395 uid_1_2_840_10008_5_1_4_1_1_200_2    = 356,
396 uid_1_2_840_10008_5_1_4_1_1_200_3    = 357,
397 uid_1_2_840_10008_5_1_4_1_1_200_4    = 358,
398 uid_1_2_840_10008_5_1_4_1_1_200_5    = 359,
399 uid_1_2_840_10008_5_1_4_1_1_200_6    = 360,
400 uid_1_2_840_10008_5_1_4_1_1_481_10   = 361,
401 uid_1_2_840_10008_5_1_4_1_1_481_11   = 362,
402 uid_1_2_840_10008_5_1_4_1_1_501_1    = 363,
403 uid_1_2_840_10008_5_1_4_1_1_501_2_1  = 364,
404 uid_1_2_840_10008_5_1_4_1_1_501_2_2  = 365,
405 uid_1_2_840_10008_5_1_4_1_1_501_3    = 366,
406 uid_1_2_840_10008_5_1_4_1_1_501_4    = 367,
407 uid_1_2_840_10008_5_1_4_1_1_501_5    = 368,
408 uid_1_2_840_10008_5_1_4_1_1_501_6    = 369,
409 uid_1_2_840_10008_5_1_4_1_1_601_1    = 370,
410 uid_1_2_840_10008_5_1_4_1_1_601_2    = 371,
411 uid_1_2_840_10008_5_1_4_1_2_4_2      = 372,
412 uid_1_2_840_10008_5_1_4_1_2_4_3      = 373,
413 uid_1_2_840_10008_5_1_4_1_2_5_3      = 374,
414 uid_1_2_840_10008_5_1_4_20_1         = 375,
415 uid_1_2_840_10008_5_1_4_20_2         = 376,
416 uid_1_2_840_10008_5_1_4_20_3         = 377,
417 uid_1_2_840_10008_5_1_4_34_5_1       = 378,
418 uid_1_2_840_10008_5_1_4_34_6         = 379,
419 uid_1_2_840_10008_5_1_4_34_6_1       = 380,
420 uid_1_2_840_10008_5_1_4_34_6_2       = 381,
421 uid_1_2_840_10008_5_1_4_34_6_3       = 382,
422 uid_1_2_840_10008_5_1_4_34_6_4       = 383,
423 uid_1_2_840_10008_5_1_4_34_7         = 384,
424 uid_1_2_840_10008_5_1_4_34_8         = 385,
425 uid_1_2_840_10008_5_1_4_34_9         = 386,
426 uid_1_2_840_10008_5_1_4_34_10        = 387,
427 uid_1_2_840_10008_5_1_4_38_4         = 388,
428 uid_1_2_840_10008_5_1_4_39_1         = 389,
429 uid_1_2_840_10008_5_1_4_39_2         = 390,
430 uid_1_2_840_10008_5_1_4_39_3         = 391,
431 uid_1_2_840_10008_5_1_4_39_4         = 392,
432 uid_1_2_840_10008_5_1_4_43_1         = 393,
433 uid_1_2_840_10008_5_1_4_43_2         = 394,
434 uid_1_2_840_10008_5_1_4_43_3         = 395,
435 uid_1_2_840_10008_5_1_4_43_4         = 396,
436 uid_1_2_840_10008_5_1_4_44_1         = 397,
437 uid_1_2_840_10008_5_1_4_44_2         = 398,
438 uid_1_2_840_10008_5_1_4_44_3         = 399,
439 uid_1_2_840_10008_5_1_4_44_4         = 400,
440 uid_1_2_840_10008_5_1_4_45_1         = 401,
441 uid_1_2_840_10008_5_1_4_45_2         = 402,
442 uid_1_2_840_10008_5_1_4_45_3         = 403,
443 uid_1_2_840_10008_5_1_4_45_4         = 404,
444 uid_1_2_840_10008_7_1_1              = 405,
445 uid_1_2_840_10008_7_1_2              = 406,
446 uid_1_2_840_10008_8_1_1              = 407,
447 uid_1_2_840_10008_5_1_4_1_1_4_3      = 408,
448 uid_1_2_840_10008_15_1_1             = 409
449 //
450 //
452
454 //
455 // Optionally private UIDs
456 //
457 #if 0
458 uid_1_2_840_113619_4_2,
459 uid_1_2_840_113619_4_3,
460 uid_1_3_12_2_1107_5_9_1,
461 uid_1_2_840_113619_4_26,
462 uid_1_2_840_113619_4_30,
463 uid_2_16_840_1_113709_1_5_1,
464 uid_2_16_840_1_113709_1_2_2,
465 uid_1_2_840_113543_6_6_1_3_10002,
466 uid_1_2_392_200036_9116_7_8_1_1_1,
467 uid_1_2_392_200036_9125_1_1_2,
```

```
468 uid_1_2_840_113619_4_27,
469 uid_1_3_46_670589_11_0_0_12_1,
470 uid_1_3_46_670589_11_0_0_12_2,
471 uid_1_3_46_670589_11_0_0_12_4,
472 uid_1_3_46_670589_2_3_1_1,
473 uid_1_3_46_670589_2_4_1_1,
474 uid_1_3_46_670589_2_5_1_1,
475 uid_1_3_46_670589_5_0_1,
476 uid_1_3_46_670589_5_0_1_1,
477 uid_1_3_46_670589_5_0_10,
478 uid_1_3_46_670589_5_0_11,
479 uid_1_3_46_670589_5_0_11_1,
480 uid_1_3_46_670589_5_0_12,
481 uid_1_3_46_670589_5_0_13,
482 uid_1_3_46_670589_5_0_14,
483 uid_1_3_46_670589_5_0_2,
484 uid_1_3_46_670589_5_0_2_1,
485 uid_1_3_46_670589_5_0_3,
486 uid_1_3_46_670589_5_0_3_1,
487 uid_1_3_46_670589_5_0_4,
488 uid_1_3_46_670589_5_0_7,
489 uid_1_3_46_670589_5_0_8,
490 uid_1_3_46_670589_5_0_9,
491 uid_1_2_752_24_3_7_6,
492 uid_1_2_752_24_3_7_7,
493 uid_1_2_840_113619_5_2,
494 uid_1_3_46_670589_33_1_4_1
495 #endif
496 //
497 //
499
500 } TSType;
501   typedef enum {
502 VerificationSOPClass = 1, // Verification SOP Class
503 ImplicitVRLittleEndianDefaultTransferSyntaxforDICOM = 2, // Implicit VR Little Endian:  Default Transfer
      Syntax for DICOM
504 ExplicitVRLittleEndian = 3, // Explicit VR Little Endian
505 DeflatedExplicitVRLittleEndian = 4, // Deflated Explicit VR Little Endian
506 ExplicitVRBigEndian = 5, // Explicit VR Big Endian
507 JPEGBaselineProcess1DefaultTransferSyntaxforLossyJPEG8BitImageCompression = 6, // JPEG Baseline (Process 1):
      Default Transfer Syntax for Lossy JPEG 8 Bit Image Compression
508 JPEGExtendedProcess24DefaultTransferSyntaxforLossyJPEG12BitImageCompressionProcess4only = 7, // JPEG
      Extended (Process 2 & 4):  Default Transfer Syntax for Lossy JPEG 12 Bit Image Compression (Process 4
      only)
509 JPEGExtendedProcess35Retired = 8, // JPEG Extended (Process 3 & 5)
510 JPEGSpectralSelectionNonHierarchicalProcess68Retired = 9, // JPEG Spectral Selection, Non-Hierarchical
      (Process 6 & 8)
511 JPEGSpectralSelectionNonHierarchicalProcess79Retired = 10, // JPEG Spectral Selection, Non-Hierarchical
      (Process 7 & 9)
512 JPEGFullProgressionNonHierarchicalProcess1012Retired = 11, // JPEG Full Progression, Non-Hierarchical
      (Process 10 & 12)
513 JPEGFullProgressionNonHierarchicalProcess1113Retired = 12, // JPEG Full Progression, Non-Hierarchical
      (Process 11 & 13)
514 JPEGLosslessNonHierarchicalProcess14 = 13, // JPEG Lossless, Non-Hierarchical (Process 14)
515 JPEGLosslessNonHierarchicalProcess15Retired = 14, // JPEG Lossless, Non-Hierarchical (Process 15)
516 JPEGExtendedHierarchicalProcess1618Retired = 15, // JPEG Extended, Hierarchical (Process 16 & 18)
517 JPEGExtendedHierarchicalProcess1719Retired = 16, // JPEG Extended, Hierarchical (Process 17 & 19)
518 JPEGSpectralSelectionHierarchicalProcess2022Retired = 17, // JPEG Spectral Selection, Hierarchical (Process
      20 & 22)
519 JPEGSpectralSelectionHierarchicalProcess2123Retired = 18, // JPEG Spectral Selection, Hierarchical (Process
      21 & 23)
520 JPEGFullProgressionHierarchicalProcess2426Retired = 19, // JPEG Full Progression, Hierarchical (Process 24 &
      26)
521 JPEGFullProgressionHierarchicalProcess2527Retired = 20, // JPEG Full Progression, Hierarchical (Process 25 &
      27)
522 JPEGLosslessHierarchicalProcess28Retired = 21, // JPEG Lossless, Hierarchical (Process 28)
523 JPEGLosslessHierarchicalProcess29Retired = 22, // JPEG Lossless, Hierarchical (Process 29)
524
      JPEGLosslessNonHierarchicalFirstOrderPredictionProcess14SelectionValue1DefaultTransferSyntaxforLosslessJPEGImageCompression
      = 23, // JPEG Lossless, Non-Hierarchical, First-Order Prediction (Process 14 [Selection Value 1]):
      Default Transfer Syntax for Lossless JPEG Image Compression
525 JPEGLSLosslessImageCompression = 24, // JPEG-LS Lossless Image Compression
526 JPEGLSLossyNearLosslessImageCompression = 25, // JPEG-LS Lossy (Near-Lossless) Image Compression
527 JPEG2000ImageCompressionLosslessOnly = 26, // JPEG 2000 Image Compression (Lossless Only)
528 JPEG2000ImageCompression = 27, // JPEG 2000 Image Compression
529 JPEG2000Part2MulticomponentImageCompressionLosslessOnly = 28, // JPEG 2000 Part 2 Multi-component Image
      Compression (Lossless Only)
530 JPEG2000Part2MulticomponentImageCompression = 29, // JPEG 2000 Part 2 Multi-component Image Compression
531 JPIPReferenced = 30, // JPIP Referenced
532 JPIPReferencedDeflate = 31, // JPIP Referenced Deflate
533 MPEG2MainProfileMainLevel = 32, // MPEG2 Main Profile @ Main Level
```

```
534 RLELossless = 33, // RLE Lossless
535 RFC2557MIMEencapsulation = 34, // RFC 2557 MIME encapsulation
536 XMLEncoding = 35, // XML Encoding
537 MediaStorageDirectoryStorage = 36, // Media Storage Directory Storage
538 TalairachBrainAtlasFrameofReference = 37, // Talairach Brain Atlas Frame of Reference
539 SPM2T1FrameofReference = 38, // SPM2 T1 Frame of Reference
540 SPM2T2FrameofReference = 39, // SPM2 T2 Frame of Reference
541 SPM2PDFrameofReference = 40, // SPM2 PD Frame of Reference
542 SPM2EPIFrameofReference = 41, // SPM2 EPI Frame of Reference
543 SPM2FILT1FrameofReference = 42, // SPM2 FIL T1 Frame of Reference
544 SPM2PETFrameofReference = 43, // SPM2 PET Frame of Reference
545 SPM2TRANSMFrameofReference = 44, // SPM2 TRANSM Frame of Reference
546 SPM2SPECTFrameofReference = 45, // SPM2 SPECT Frame of Reference
547 SPM2GRAYFrameofReference = 46, // SPM2 GRAY Frame of Reference
548 SPM2WHITEFrameofReference = 47, // SPM2 WHITE Frame of Reference
549 SPM2CSFFrameofReference = 48, // SPM2 CSF Frame of Reference
550 SPM2BRAINMASKFrameofReference = 49, // SPM2 BRAINMASK Frame of Reference
551 SPM2AVG305T1FrameofReference = 50, // SPM2 AVG305T1 Frame of Reference
552 SPM2AVG152T1FrameofReference = 51, // SPM2 AVG152T1 Frame of Reference
553 SPM2AVG152T2FrameofReference = 52, // SPM2 AVG152T2 Frame of Reference
554 SPM2AVG152PDFrameofReference = 53, // SPM2 AVG152PD Frame of Reference
555 SPM2SINGLESUBJT1FrameofReference = 54, // SPM2 SINGLESUBJT1 Frame of Reference
556 ICBM452T1FrameofReference = 55, // ICBM 452 T1 Frame of Reference
557 ICBMSingleSubjectMRIFrameofReference = 56, // ICBM Single Subject MRI Frame of Reference
558 BasicStudyContentNotificationSOPClassRetired = 57, // Basic Study Content Notification SOP Class
559 StorageCommitmentPushModelSOPClass = 58, // Storage Commitment Push Model SOP Class
560 StorageCommitmentPushModelSOPInstance = 59, // Storage Commitment Push Model SOP Instance
561 StorageCommitmentPullModelSOPClassRetired = 60, // Storage Commitment Pull Model SOP Class
562 StorageCommitmentPullModelSOPInstanceRetired = 61, // Storage Commitment Pull Model SOP Instance
563 ProceduralEventLoggingSOPClass = 62, // Procedural Event Logging SOP Class
564 ProceduralEventLoggingSOPInstance = 63, // Procedural Event Logging SOP Instance
565 SubstanceAdministrationLoggingSOPClass = 64, // Substance Administration Logging SOP Class
566 SubstanceAdministrationLoggingSOPInstance = 65, // Substance Administration Logging SOP Instance
567 DICOMUIDRegistry = 66, // DICOM UID Registry
568 DICOMControlledTerminology = 67, // DICOM Controlled Terminology
569 DICOMApplicationContextName = 68, // DICOM Application Context Name
570 DetachedPatientManagementSOPClassRetired = 69, // Detached Patient Management SOP Class
571 DetachedPatientManagementMetaSOPClassRetired = 70, // Detached Patient Management Meta SOP Class
572 DetachedVisitManagementSOPClassRetired = 71, // Detached Visit Management SOP Class
573 DetachedStudyManagementSOPClassRetired = 72, // Detached Study Management SOP Class
574 StudyComponentManagementSOPClassRetired = 73, // Study Component Management SOP Class
575 ModalityPerformedProcedureStepSOPClass = 74, // Modality Performed Procedure Step SOP Class
576 ModalityPerformedProcedureStepRetrieveSOPClass = 75, // Modality Performed Procedure Step Retrieve SOP Class
577 ModalityPerformedProcedureStepNotificationSOPClass = 76, // Modality Performed Procedure Step Notification
      SOP Class
578 DetachedResultsManagementSOPClassRetired = 77, // Detached Results Management SOP Class
579 DetachedResultsManagementMetaSOPClassRetired = 78, // Detached Results Management Meta SOP Class
580 DetachedStudyManagementMetaSOPClassRetired = 79, // Detached Study Management Meta SOP Class
581 DetachedInterpretationManagementSOPClassRetired = 80, // Detached Interpretation Management SOP Class
582 StorageServiceClass = 81, // Storage Service Class
583 BasicFilmSessionSOPClass = 82, // Basic Film Session SOP Class
584 BasicFilmBoxSOPClass = 83, // Basic Film Box SOP Class
585 BasicGrayscaleImageBoxSOPClass = 84, // Basic Grayscale Image Box SOP Class
586 BasicColorImageBoxSOPClass = 85, // Basic Color Image Box SOP Class
587 ReferencedImageBoxSOPClassRetired = 86, // Referenced Image Box SOP Class
588 BasicGrayscalePrintManagementMetaSOPClass = 87, // Basic Grayscale Print Management Meta SOP Class
589 ReferencedGrayscalePrintManagementMetaSOPClassRetired = 88, // Referenced Grayscale Print Management Meta
      SOP Class
590 PrintJobSOPClass = 89, // Print Job SOP Class
591 BasicAnnotationBoxSOPClass = 90, // Basic Annotation Box SOP Class
592 PrinterSOPClass = 91, // Printer SOP Class
593 PrinterConfigurationRetrievalSOPClass = 92, // Printer Configuration Retrieval SOP Class
594 PrinterSOPInstance = 93, // Printer SOP Instance
595 PrinterConfigurationRetrievalSOPInstance = 94, // Printer Configuration Retrieval SOP Instance
596 BasicColorPrintManagementMetaSOPClass = 95, // Basic Color Print Management Meta SOP Class
597 ReferencedColorPrintManagementMetaSOPClassRetired = 96, // Referenced Color Print Management Meta SOP Class
598 VOILUTBoxSOPClass = 97, // VOI LUT Box SOP Class
599 PresentationLUTSOPClass = 98, // Presentation LUT SOP Class
600 ImageOverlayBoxSOPClassRetired = 99, // Image Overlay Box SOP Class
601 BasicPrintImageOverlayBoxSOPClassRetired = 100, // Basic Print Image Overlay Box SOP Class
602 PrintQueueSOPInstanceRetired = 101, // Print Queue SOP Instance
603 PrintQueueManagementSOPClassRetired = 102, // Print Queue Management SOP Class
604 StoredPrintStorageSOPClassRetired = 103, // Stored Print Storage SOP Class
605 HardcopyGrayscaleImageStorageSOPClassRetired = 104, // Hardcopy Grayscale Image Storage SOP Class
606 HardcopyColorImageStorageSOPClassRetired = 105, // Hardcopy Color Image Storage SOP Class
607 PullPrintRequestSOPClassRetired = 106, // Pull Print Request SOP Class
608 PullStoredPrintManagementMetaSOPClassRetired = 107, // Pull Stored Print Management Meta SOP Class
609 MediaCreationManagementSOPClassUID = 108, // Media Creation Management SOP Class UID
610 ComputedRadiographyImageStorage = 109, // Computed Radiography Image Storage
611 DigitalXRayImageStorageForPresentation = 110, // Digital X-Ray Image Storage - For Presentation
612 DigitalXRayImageStorageForProcessing = 111, // Digital X-Ray Image Storage - For Processing
```

```
613 DigitalMammographyXRayImageStorageForPresentation = 112, // Digital Mammography X-Ray Image Storage - For
      Presentation
614 DigitalMammographyXRayImageStorageForProcessing = 113, // Digital Mammography X-Ray Image Storage - For
      Processing
615 DigitalIntraoralXRayImageStorageForPresentation = 114, // Digital Intra-oral X-Ray Image Storage - For
      Presentation
616 DigitalIntraoralXRayImageStorageForProcessing = 115, // Digital Intra-oral X-Ray Image Storage - For
      Processing
617 CTImageStorage = 116, // CT Image Storage
618 EnhancedCTImageStorage = 117, // Enhanced CT Image Storage
619 UltrasoundMultiframeImageStorageRetired = 118, // Ultrasound Multi-frame Image Storage
620 UltrasoundMultiframeImageStorage = 119, // Ultrasound Multi-frame Image Storage
621 MRImageStorage = 120, // MR Image Storage
622 EnhancedMRImageStorage = 121, // Enhanced MR Image Storage
623 MRSpectroscopyStorage = 122, // MR Spectroscopy Storage
624 NuclearMedicineImageStorageRetired = 123, // Nuclear Medicine Image Storage
625 UltrasoundImageStorageRetired = 124, // Ultrasound Image Storage
626 UltrasoundImageStorage = 125, // Ultrasound Image Storage
627 SecondaryCaptureImageStorage = 126, // Secondary Capture Image Storage
628 MultiframeSingleBitSecondaryCaptureImageStorage = 127, // Multi-frame Single Bit Secondary Capture Image
      Storage
629 MultiframeGrayscaleByteSecondaryCaptureImageStorage = 128, // Multi-frame Grayscale Byte Secondary Capture
      Image Storage
630 MultiframeGrayscaleWordSecondaryCaptureImageStorage = 129, // Multi-frame Grayscale Word Secondary Capture
      Image Storage
631 MultiframeTrueColorSecondaryCaptureImageStorage = 130, // Multi-frame True Color Secondary Capture Image
      Storage
632 StandaloneOverlayStorageRetired = 131, // Standalone Overlay Storage
633 StandaloneCurveStorageRetired = 132, // Standalone Curve Storage
634 WaveformStorageTrialRetired = 133, // Waveform Storage - Trial
635 ECG12leadWaveformStorage = 134, // 12-lead ECG Waveform Storage
636 GeneralECGWaveformStorage = 135, // General ECG Waveform Storage
637 AmbulatoryECGWaveformStorage = 136, // Ambulatory ECG Waveform Storage
638 HemodynamicWaveformStorage = 137, // Hemodynamic Waveform Storage
639 CardiacElectrophysiologyWaveformStorage = 138, // Cardiac Electrophysiology Waveform Storage
640 BasicVoiceAudioWaveformStorage = 139, // Basic Voice Audio Waveform Storage
641 StandaloneModalityLUTStorageRetired = 140, // Standalone Modality LUT Storage
642 StandaloneVOILUTStorageRetired = 141, // Standalone VOI LUT Storage
643 GrayscaleSoftcopyPresentationStateStorageSOPClass = 142, // Grayscale Softcopy Presentation State Storage
      SOP Class
644 ColorSoftcopyPresentationStateStorageSOPClass = 143, // Color Softcopy Presentation State Storage SOP Class
645 PseudoColorSoftcopyPresentationStateStorageSOPClass = 144, // Pseudo-Color Softcopy Presentation State
      Storage SOP Class
646 BlendingSoftcopyPresentationStateStorageSOPClass = 145, // Blending Softcopy Presentation State Storage SOP
      Class
647 XRayAngiographicImageStorage = 146, // X-Ray Angiographic Image Storage
648 EnhancedXAImageStorage = 147, // Enhanced XA Image Storage
649 XRayRadiofluoroscopicImageStorage = 148, // X-Ray Radiofluoroscopic Image Storage
650 EnhancedXRFImageStorage = 149, // Enhanced XRF Image Storage
651 XRay3DAngiographicImageStorage = 150, // X-Ray 3D Angiographic Image Storage
652 XRay3DCraniofacialImageStorage = 151, // X-Ray 3D Craniofacial Image Storage
653 XRayAngiographicBiPlaneImageStorageRetired = 152, // X-Ray Angiographic Bi-Plane Image Storage
654 NuclearMedicineImageStorage = 153, // Nuclear Medicine Image Storage
655 RawDataStorage = 154, // Raw Data Storage
656 SpatialRegistrationStorage = 155, // Spatial Registration Storage
657 SpatialFiducialsStorage = 156, // Spatial Fiducials Storage
658 DeformableSpatialRegistrationStorage = 157, // Deformable Spatial Registration Storage
659 SegmentationStorage = 158, // Segmentation Storage
660 RealWorldValueMappingStorage = 159, // Real World Value Mapping Storage
661 VLImageStorageTrialRetired = 160, // VL Image Storage - Trial
662 VLMultiframeImageStorageTrialRetired = 161, // VL Multi-frame Image Storage - Trial
663 VLEndoscopicImageStorage = 162, // VL Endoscopic Image Storage
664 VideoEndoscopicImageStorage = 163, // Video Endoscopic Image Storage
665 VLMicroscopicImageStorage = 164, // VL Microscopic Image Storage
666 VideoMicroscopicImageStorage = 165, // Video Microscopic Image Storage
667 VLSlideCoordinatesMicroscopicImageStorage = 166, // VL Slide-Coordinates Microscopic Image Storage
668 VLPhotographicImageStorage = 167, // VL Photographic Image Storage
669 VideoPhotographicImageStorage = 168, // Video Photographic Image Storage
670 OphthalmicPhotography8BitImageStorage = 169, // Ophthalmic Photography 8 Bit Image Storage
671 OphthalmicPhotography16BitImageStorage = 170, // Ophthalmic Photography 16 Bit Image Storage
672 StereometricRelationshipStorage = 171, // Stereometric Relationship Storage
673 OphthalmicTomographyImageStorage = 172, // Ophthalmic Tomography Image Storage
674 TextSRStorageTrialRetired = 173, // Text SR Storage - Trial
675 AudioSRStorageTrialRetired = 174, // Audio SR Storage - Trial
676 DetailSRStorageTrialRetired = 175, // Detail SR Storage - Trial
677 ComprehensiveSRStorageTrialRetired = 176, // Comprehensive SR Storage - Trial
678 BasicTextSRStorage = 177, // Basic Text SR Storage
679 EnhancedSRStorage = 178, // Enhanced SR Storage
680 ComprehensiveSRStorage = 179, // Comprehensive SR Storage
681 ProcedureLogStorage = 180, // Procedure Log Storage
682 MammographyCADSRStorage = 181, // Mammography CAD SR Storage
```

```
683 KeyObjectSelectionDocumentStorage = 182, // Key Object Selection Document Storage
684 ChestCADSRStorage = 183, // Chest CAD SR Storage
685 XRayRadiationDoseSRStorage = 184, // X-Ray Radiation Dose SR Storage
686 EncapsulatedPDFStorage = 185, // Encapsulated PDF Storage
687 EncapsulatedCDAStorage = 186, // Encapsulated CDA Storage
688 PositronEmissionTomographyImageStorage = 187, // Positron Emission Tomography Image Storage
689 StandalonePETCurveStorageRetired = 188, // Standalone PET Curve Storage
690 RTImageStorage = 189, // RT Image Storage
691 RTDoseStorage = 190, // RT Dose Storage
692 RTStructureSetStorage = 191, // RT Structure Set Storage
693 RTBeamsTreatmentRecordStorage = 192, // RT Beams Treatment Record Storage
694 RTPlanStorage = 193, // RT Plan Storage
695 RTBrachyTreatmentRecordStorage = 194, // RT Brachy Treatment Record Storage
696 RTTreatmentSummaryRecordStorage = 195, // RT Treatment Summary Record Storage
697 RTIonPlanStorage = 196, // RT Ion Plan Storage
698 RTIonBeamsTreatmentRecordStorage = 197, // RT Ion Beams Treatment Record Storage
699 PatientRootQueryRetrieveInformationModelFIND = 198, // Patient Root Query/Retrieve Information Model - FIND
700 PatientRootQueryRetrieveInformationModelMOVE = 199, // Patient Root Query/Retrieve Information Model - MOVE
701 PatientRootQueryRetrieveInformationModelGET = 200, // Patient Root Query/Retrieve Information Model - GET
702 StudyRootQueryRetrieveInformationModelFIND = 201, // Study Root Query/Retrieve Information Model - FIND
703 StudyRootQueryRetrieveInformationModelMOVE = 202, // Study Root Query/Retrieve Information Model - MOVE
704 StudyRootQueryRetrieveInformationModelGET = 203, // Study Root Query/Retrieve Information Model - GET
705 PatientStudyOnlyQueryRetrieveInformationModelFINDRetired = 204, // Patient/Study Only Query/Retrieve
      Information Model - FIND
706 PatientStudyOnlyQueryRetrieveInformationModelMOVERetired = 205, // Patient/Study Only Query/Retrieve
      Information Model - MOVE
707 PatientStudyOnlyQueryRetrieveInformationModelGETRetired = 206, // Patient/Study Only Query/Retrieve
      Information Model - GET
708 ModalityWorklistInformationModelFIND = 207, // Modality Worklist Information Model - FIND
709 GeneralPurposeWorklistInformationModelFIND = 208, // General Purpose Worklist Information Model - FIND
710 GeneralPurposeScheduledProcedureStepSOPClass = 209, // General Purpose Scheduled Procedure Step SOP Class
711 GeneralPurposePerformedProcedureStepSOPClass = 210, // General Purpose Performed Procedure Step SOP Class
712 GeneralPurposeWorklistManagementMetaSOPClass = 211, // General Purpose Worklist Management Meta SOP Class
713 InstanceAvailabilityNotificationSOPClass = 212, // Instance Availability Notification SOP Class
714 RTBeamsDeliveryInstructionStorageSupplement74FrozenDraft = 213, // RT Beams Delivery Instruction Storage
      (Supplement 74 Frozen Draft)
715 RTConventionalMachineVerificationSupplement74FrozenDraft = 214, // RT Conventional Machine Verification
      (Supplement 74 Frozen Draft)
716 RTIonMachineVerificationSupplement74FrozenDraft = 215, // RT Ion Machine Verification (Supplement 74 Frozen
      Draft)
717 UnifiedWorklistandProcedureStepServiceClass = 216, // Unified Worklist and Procedure Step Service Class
718 UnifiedProcedureStepPushSOPClass = 217, // Unified Procedure Step - Push SOP Class
719 UnifiedProcedureStepWatchSOPClass = 218, // Unified Procedure Step - Watch SOP Class
720 UnifiedProcedureStepPullSOPClass = 219, // Unified Procedure Step - Pull SOP Class
721 UnifiedProcedureStepEventSOPClass = 220, // Unified Procedure Step - Event SOP Class
722 UnifiedWorklistandProcedureStepSOPInstance = 221, // Unified Worklist and Procedure Step SOP Instance
723 GeneralRelevantPatientInformationQuery = 222, // General Relevant Patient Information Query
724 BreastImagingRelevantPatientInformationQuery = 223, // Breast Imaging Relevant Patient Information Query
725 CardiacRelevantPatientInformationQuery = 224, // Cardiac Relevant Patient Information Query
726 HangingProtocolStorage = 225, // Hanging Protocol Storage
727 HangingProtocolInformationModelFIND = 226, // Hanging Protocol Information Model - FIND
728 HangingProtocolInformationModelMOVE = 227, // Hanging Protocol Information Model - MOVE
729 ProductCharacteristicsQuerySOPClass = 228, // Product Characteristics Query SOP Class
730 SubstanceApprovalQuerySOPClass = 229, // Substance Approval Query SOP Class
731 dicomDeviceName = 230, // dicomDeviceName
732 dicomDescription = 231, // dicomDescription
733 dicomManufacturer = 232, // dicomManufacturer
734 dicomManufacturerModelName = 233, // dicomManufacturerModelName
735 dicomSoftwareVersion = 234, // dicomSoftwareVersion
736 dicomVendorData = 235, // dicomVendorData
737 dicomAETitle = 236, // dicomAETitle
738 dicomNetworkConnectionReference = 237, // dicomNetworkConnectionReference
739 dicomApplicationCluster = 238, // dicomApplicationCluster
740 dicomAssociationInitiator = 239, // dicomAssociationInitiator
741 dicomAssociationAcceptor = 240, // dicomAssociationAcceptor
742 dicomHostname = 241, // dicomHostname
743 dicomPort = 242, // dicomPort
744 dicomSOPClass = 243, // dicomSOPClass
745 dicomTransferRole = 244, // dicomTransferRole
746 dicomTransferSyntax = 245, // dicomTransferSyntax
747 dicomPrimaryDeviceType = 246, // dicomPrimaryDeviceType
748 dicomRelatedDeviceReference = 247, // dicomRelatedDeviceReference
749 dicomPreferredCalledAETitle = 248, // dicomPreferredCalledAETitle
750 dicomTLSCyphersuite = 249, // dicomTLSCyphersuite
751 dicomAuthorizedNodeCertificateReference = 250, // dicomAuthorizedNodeCertificateReference
752 dicomThisNodeCertificateReference = 251, // dicomThisNodeCertificateReference
753 dicomInstalled = 252, // dicomInstalled
754 dicomStationName = 253, // dicomStationName
755 dicomDeviceSerialNumber = 254, // dicomDeviceSerialNumber
756 dicomInstitutionName = 255, // dicomInstitutionName
757 dicomInstitutionAddress = 256, // dicomInstitutionAddress
```

```
758 dicomInstitutionDepartmentName = 257, // dicomInstitutionDepartmentName
759 dicomIssuerOfPatientID = 258, // dicomIssuerOfPatientID
760 dicomPreferredCallingAETitle = 259, // dicomPreferredCallingAETitle
761 dicomSupportedCharacterSet = 260, // dicomSupportedCharacterSet
762 dicomConfigurationRoot = 261, // dicomConfigurationRoot
763 dicomDevicesRoot = 262, // dicomDevicesRoot
764 dicomUniqueAETitlesRegistryRoot = 263, // dicomUniqueAETitlesRegistryRoot
765 dicomDevice = 264, // dicomDevice
766 dicomNetworkAE = 265, // dicomNetworkAE
767 dicomNetworkConnection = 266, // dicomNetworkConnection
768 dicomUniqueAETitle = 267, // dicomUniqueAETitle
769 dicomTransferCapability = 268, // dicomTransferCapability
770 //
771 VLWholeSlideMicroscopyImageStorage           = 269,
772 EnhancedUSVolumeStorage                      = 270,
773 SurfaceSegmentationStorage                   = 271,
774 BreastTomosynthesisImageStorage              = 272,
775 LegacyConvertedEnhancedCTImageStorage        = 273,
776 LegacyConvertedEnhancedMRImageStorage        = 274,
777 LegacyConvertedEnhancedPETImageStorage       = 275,
778 MPEG2MainProfileHighLevel                    = 276,
779 MPEG4AVCH_264HighProfileLevel4_1             = 277,
780 MPEG4AVCH_264BDcompatibleHighProfileLevel4_1 = 278,
781
783 //
784 // 2019b
785 //
786 PETColorPaletteSOPInstance                                          = 279,
787 HotMetalBlueColorPaletteSOPInstance                                 = 280,
788 PET20StepColorPaletteSOPInstance                                    = 281,
789 SpringColorPaletteSOPInstance                                       = 282,
790 SummerColorPaletteSOPInstance                                       = 283,
791 FallColorPaletteSOPInstance                                         = 284,
792 WinterColorPaletteSOPInstance                                       = 285,
793 Papyrus3ImplicitVRLittleEndian                                      = 286,
794 AdultMouseAnatomyOntology                                           = 287,
795 UberonOntology                                                     = 288,
796 IntegratedTaxonomicInformationSystemITISTaxonomicSerialNumberTSN   = 289,
797 MouseGenomeInitiativeMGI                                           = 290,
798 PubChemCompoundCID                                                 = 291,
799 ICD11                                                              = 292,
800 NewYorkUniversityMelanomaClinicalCooperativeGroup                  = 293,
801 MayoClinicNonradiologicalImagesSBSAnatomicalSurfaceRegionGuide     = 294,
802 ImageBiomarkerStandardisationInitiative                            = 295,
803 RadiomicsOntology                                                  = 296,
804 DisplaySystemSOPClass                                              = 297,
805 DisplaySystemSOPInstance                                           = 298,
806 GeneralAudioWaveformStorage                                        = 299,
807 ArterialPulseWaveformStorage                                       = 300,
808 RespiratoryWaveformStorage                                         = 301,
809 XAXRFGrayscaleSoftcopyPresentationStateStorage                     = 302,
810 GrayscalePlanarMPRVolumetricPresentationStateStorage               = 303,
811 MPEG4AVCH_264HighProfileLevel4_2For2DVideo                         = 304,
812 MPEG4AVCH_264HighProfileLevel4_2For3DVideo                         = 305,
813 MPEG4AVCH_264StereoHighProfileLevel4_2                             = 306,
814 HEVCH_265MainProfileLevel5_1                                       = 307,
815 HEVCH_265Main10ProfileLevel5_1                                     = 308,
816 HotIronColorPaletteSOPInstance                                     = 309,
817 CompositingPlanarMPRVolumetricPresentationStateStorage             = 310,
818 AdvancedBlendingPresentationStateStorage                           = 311,
819 VolumeRenderingVolumetricPresentationStateStorage                  = 312,
820 SegmentedVolumeRenderingVolumetricPresentationStateStorage         = 313,
821 MultipleVolumeRenderingVolumetricPresentationStateStorage          = 314,
822 Null0                                                              = 315,
823 BreastProjectionXRayImageStorageForPresentation                    = 316,
824 BreastProjectionXRayImageStorageForProcessing                      = 317,
825 IntravascularOpticalCoherenceTomographyImageStorageForPresentation = 318,
826 IntravascularOpticalCoherenceTomographyImageStorageForProcessing   = 319,
827 ParametricMapStorage                                               = 320,
828 Null1                                                              = 321,
829 TractographyResultsStorage                                         = 322,
830 SurfaceScanMeshStorage                                             = 323,
831 SurfaceScanPointCloudStorage                                       = 324,
832 WideFieldOphthalmicPhotographyStereographicProjectionImageStorage  = 325,
833 WideFieldOphthalmicPhotography3DCoordinatesImageStorage            = 326,
834 OphthalmicOpticalCoherenceTomographyEnFaceImageStorage             = 327,
835 OphthalmicOpticalCoherenceTomographyBscanVolumeAnalysisStorage     = 328,
836 LensometryMeasurementsStorage                                      = 329,
837 AutorefractionMeasurementsStorage                                  = 330,
838 KeratometryMeasurementsStorage                                     = 331,
839 SubjectiveRefractionMeasurementsStorage                            = 332,
```

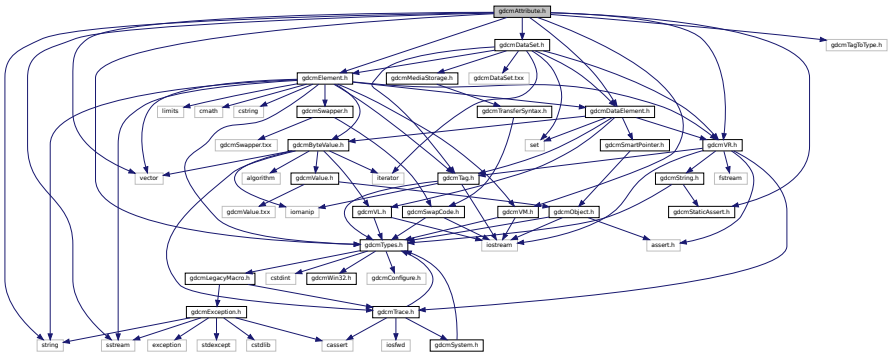
```
840 VisualAcuityMeasurementsStorage                                 = 333,
841 SpectaclePrescriptionReportStorage                              = 334,
842 OphthalmicAxialMeasurementsStorage                              = 335,
843 IntraocularLensCalculationsStorage                              = 336,
844 MacularGridThicknessandVolumeReportStorage                       = 337,
845 OphthalmicVisualFieldStaticPerimetryMeasurementsStorage         = 338,
846 OphthalmicThicknessMapStorage                                   = 339,
847 CornealTopographyMapStorage                                     = 340,
848 Comprehensive3DSRStorage                                        = 341,
849 ExtensibleSRStorage                                             = 342,
850 RadiopharmaceuticalRadiationDoseSRStorage                       = 343,
851 ColonCADSRStorage                                               = 344,
852 ImplantationPlanSRStorage                                       = 345,
853 AcquisitionContextSRStorage                                     = 346,
854 SimplifiedAdultEchoSRStorage                                    = 347,
855 PatientRadiationDoseSRStorage                                   = 348,
856 PlannedImagingAgentAdministrationSRStorage                      = 349,
857 PerformedImagingAgentAdministrationSRStorage                    = 350,
858 ContentAssessmentResultsStorage                                 = 351,
859 EncapsulatedSTLStorage                                          = 352,
860 EnhancedPETImageStorage                                         = 353,
861 BasicStructuredDisplayStorage                                   = 354,
862 CTDefinedProcedureProtocolStorage                               = 355,
863 CTPerformedProcedureProtocolStorage                             = 356,
864 ProtocolApprovalStorage                                         = 357,
865 ProtocolApprovalInformationModelFIND                            = 358,
866 ProtocolApprovalInformationModelMOVE                            = 359,
867 ProtocolApprovalInformationModelGET                             = 360,
868 RTPhysicianIntentStorage                                        = 361,
869 RTSegmentAnnotationStorage                                      = 362,
870 DICOSCTImageStorage                                             = 363,
871 DICOSDigitalXRayImageStorageForPresentation                     = 364,
872 DICOSDigitalXRayImageStorageForProcessing                       = 365,
873 DICOSThreatDetectionReportStorage                               = 366,
874 DICOS2DAITStorage                                               = 367,
875 DICOS3DAITStorage                                               = 368,
876 DICOSQuadrupoleResonanceQRStorage                               = 369,
877 EddyCurrentImageStorage                                         = 370,
878 EddyCurrentMultiframeImageStorage                               = 371,
879 CompositeInstanceRootRetrieveMOVE                               = 372,
880 CompositeInstanceRootRetrieveGET                                = 373,
881 CompositeInstanceRetrieveWithoutBulkDataGET                     = 374,
882 DefinedProcedureProtocolInformationModelFIND                    = 375,
883 DefinedProcedureProtocolInformationModelMOVE                    = 376,
884 DefinedProcedureProtocolInformationModelGET                     = 377,
885 UPSFilteredGlobalSubscriptionSOPInstance                        = 378,
886 UnifiedWorklistandProcedureStepServiceClass1                    = 379,
887 UnifiedProcedureStepPushSOPClass1                               = 380,
888 UnifiedProcedureStepWatchSOPClass1                              = 381,
889 UnifiedProcedureStepPullSOPClass1                               = 382,
890 UnifiedProcedureStepEventSOPClass1                              = 383,
891 RTBeamsDeliveryInstructionStorage                               = 384,
892 RTConventionalMachineVerification                               = 385,
893 RTIonMachineVerification                                        = 386,
894 RTBrachyApplicationSetupDeliveryInstructionStorage              = 387,
895 HangingProtocolInformationModelGET                              = 388,
896 ColorPaletteStorage                                             = 389,
897 ColorPaletteQueryRetrieveInformationModelFIND                   = 390,
898 ColorPaletteQueryRetrieveInformationModelMOVE                   = 391,
899 ColorPaletteQueryRetrieveInformationModelGET                    = 392,
900 GenericImplantTemplateStorage                                   = 393,
901 GenericImplantTemplateInformationModelFIND                      = 394,
902 GenericImplantTemplateInformationModelMOVE                      = 395,
903 GenericImplantTemplateInformationModelGET                       = 396,
904 ImplantAssemblyTemplateStorage                                  = 397,
905 ImplantAssemblyTemplateInformationModelFIND                     = 398,
906 ImplantAssemblyTemplateInformationModelMOVE                     = 399,
907 ImplantAssemblyTemplateInformationModelGET                      = 400,
908 ImplantTemplateGroupStorage                                     = 401,
909 ImplantTemplateGroupInformationModelFIND                        = 402,
910 ImplantTemplateGroupInformationModelMOVE                        = 403,
911 ImplantTemplateGroupInformationModelGET                         = 404,
912 NativeDICOMModel                                                = 405,
913 AbstractMultiDimensionalImageModel                              = 406,
914 DICOMContentMappingResource                                     = 407,
915 EnhancedMRColorImageStorage                                     = 408,
916 UniversalCoordinatedTime                                        = 409
917 //
918 //
920
922 //
```

```
923 // Optionally private UIDs
924 //
925 #if 0
926 Private_1_2_840_113619_4_2,
927 Private_1_2_840_113619_4_3,
928 Private_1_3_12_2_1107_5_9_1,
929 Private_1_2_840_113619_4_26,
930 Private_1_2_840_113619_4_30,
931 Private_2_16_840_1_113709_1_5_1,
932 Private_2_16_840_1_113709_1_2_2,
933 Private_1_2_840_113543_6_6_1_3_10002,
934 Private_1_2_392_200036_9116_7_8_1_1_1,
935 Private_1_2_392_200036_9125_1_1_2,
936 Private_1_2_840_113619_4_27,
937 Private_1_3_46_670589_11_0_0_12_1,
938 Private_1_3_46_670589_11_0_0_12_2,
939 Private_1_3_46_670589_11_0_0_12_4,
940 Private_1_3_46_670589_2_3_1_1,
941 Private_1_3_46_670589_2_4_1_1,
942 Private_1_3_46_670589_2_5_1_1,
943 Private_1_3_46_670589_5_0_1,
944 Private_1_3_46_670589_5_0_1_1,
945 Private_1_3_46_670589_5_0_10,
946 Private_1_3_46_670589_5_0_11,
947 Private_1_3_46_670589_5_0_11_1,
948 Private_1_3_46_670589_5_0_12,
949 Private_1_3_46_670589_5_0_13,
950 Private_1_3_46_670589_5_0_14,
951 Private_1_3_46_670589_5_0_2,
952 Private_1_3_46_670589_5_0_2_1,
953 Private_1_3_46_670589_5_0_3,
954 Private_1_3_46_670589_5_0_3_1,
955 Private_1_3_46_670589_5_0_4,
956 Private_1_3_46_670589_5_0_7,
957 Private_1_3_46_670589_5_0_8,
958 Private_1_3_46_670589_5_0_9,
959 Private_1_2_752_24_3_7_6,
960 Private_1_2_752_24_3_7_7,
961 Private_1_2_840_113619_5_2,
962 Private_1_3_46_670589_33_1_4_1
963 #endif
964 //
965 //
967
968 } TSName;
969
970
971   typedef const char* const (*TransferSyntaxStringsType)[2];
972   static TransferSyntaxStringsType GetTransferSyntaxStrings();
973   static const char * const *GetTransferSyntaxString(unsigned int ts);
974   static unsigned int GetNumberOfTransferSyntaxStrings();
975
976
977   // TODO: Because I would like a dual signature for TSType and TSName, C++ won't let me do it...
978   static const char* GetUIDString(/*TSType*/ unsigned int ts);
979   static const char* GetUIDName(/*TSType*/ unsigned int ts);
980
983   bool SetFromUID(const char *str);
984
987   const char *GetName() const;
988
991   const char *GetString() const;
992
993   operator TSType ()const { return TSField; }
994
995 private:
996   TSType TSField;
997 };
998 //-----------------------------------------------------------------------------
999 inline std::ostream &operator«(std::ostream &_os, const UIDs &uid)
1000 {
1001   _os « uid.GetString() « " -> " « uid.GetName();
1002   return _os;
1003
1004 }
1005
1006 } // end namespace gdcm
1007
1008 #endif //GDCMUIDS_H
```

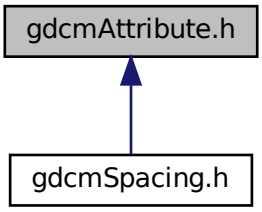
11.109 gdcmAttribute.h File Reference

```
#include "gdcmTypes.h"
#include "gdcmVR.h"
#include "gdcmTagToType.h"
#include "gdcmVM.h"
#include "gdcmElement.h"
#include "gdcmDataElement.h"
#include "gdcmDataSet.h"
#include "gdcmStaticAssert.h"
#include <string>
#include <vector>
#include <sstream>
```

Include dependency graph for gdcmAttribute.h:



This graph shows which files directly or indirectly include this file:



Classes

- class gdcm::Attribute< Group, Element, TVR, TVM >

  Attribute *class This class use template metaprograming tricks to let the user know when the template instantiation does not match the public dictionary.*

- class gdcm::Attribute< Group, Element, TVR, VM::VM1 >
- class gdcm::Attribute< Group, Element, TVR, VM::VM1_3 >
- class gdcm::Attribute< Group, Element, TVR, VM::VM1_8 >
- class gdcm::Attribute< Group, Element, TVR, VM::VM1_n >
- class gdcm::Attribute< Group, Element, TVR, VM::VM2_2n >
- class gdcm::Attribute< Group, Element, TVR, VM::VM2_n >
- class gdcm::Attribute< Group, Element, TVR, VM::VM3_3n >
- class gdcm::Attribute< Group, Element, TVR, VM::VM3_n >
- class gdcm::VRVLSize< 0 >
- class gdcm::VRVLSize< 1 >

Namespaces

- namespace gdcm

11.110 gdcmAttribute.h

Go to the documentation of this file.

```
1 /*=========================================================================
2
3 Program:  GDCM (Grassroots DICOM). A DICOM library
4
5 Copyright (c) 2006-2011 Mathieu Malaterre
6 All rights reserved.
7 See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.
8
9 This software is distributed WITHOUT ANY WARRANTY; without even
10 the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
11 PURPOSE.  See the above copyright notice for more information.
12
13 =========================================================================*/
14 #ifndef GDCMATTRIBUTE_H
15 #define GDCMATTRIBUTE_H
16
17 #include "gdcmTypes.h"
18 #include "gdcmVR.h"
19 #include "gdcmTagToType.h"
20 #include "gdcmVM.h"
21 #include "gdcmElement.h"
22 #include "gdcmDataElement.h"
23 #include "gdcmDataSet.h"
24 #include "gdcmStaticAssert.h"
25
26 #include <string>
27 #include <vector>
28 #include <sstream>
29
30 namespace gdcm_ns
31 {
32
33 struct void_;
34
35 // Declaration, also serve as forward declaration
36 template<int T> class VRVLSize;
37
38 // Implementation when VL is coded on 16 bits:
39 template<> class VRVLSize<0> {
40 public:
41   static inline uint16_t Read(std::istream &_is) {
42     uint16_t l;
43     _is.read((char*)&l, 2);
44     return l;
45     }
46
47   static inline void Write(std::ostream &os)  { (void)os;
```

```
48     }
49 };
50 // Implementation when VL is coded on 32 bits:
51 template<> class VRVLSize<1> {
52 public:
53   static inline uint32_t Read(std::istream &_is) {
54     char dummy[2];
55     _is.read(dummy, 2);
56
57     uint32_t l;
58     _is.read((char*)&l, 4);
59     return l;
60     }
61
62   static inline void Write(std::ostream &os)  { (void)os;
63     }
64 };
65
66 template<uint16_t Group, uint16_t Element,
67    long long TVR = TagToType<Group, Element>::VRType, // can the user override this value ?
68    int TVM = TagToType<Group, Element>::VMType // can the user override this value ?
69    /*typename SQAttribute = void_*/ > // if only I had variadic template...
70 class Attribute
71 {
72 public:
73   typedef typename VRToType<TVR>::Type ArrayType;
74   enum { VMType = VMToLength<TVM>::Length };
75   ArrayType Internal[VMToLength<TVM>::Length];
76
77   // Make sure that user specified VR/VM are compatible with the public dictionary:
78   GDCM_STATIC_ASSERT( ((VR::VRType)TVR & (VR::VRType)(TagToType<Group, Element>::VRType)) );
79   GDCM_STATIC_ASSERT( ((VM::VMType)TVM & (VM::VMType)(TagToType<Group, Element>::VMType)) );
80   GDCM_STATIC_ASSERT( ((((VR::VRType)TVR & VR::VR_VM1) && ((VM::VMType)TVM == VM::VM1) )
81                       || !((VR::VRType)TVR & VR::VR_VM1) ) );
82
83   static Tag GetTag() { return Tag(Group,Element); }
84   static VR  GetVR()  { return (VR::VRType)TVR; }
85    static VM  GetVM()  { return (VM::VMType)TVM; }
86
87   // The following two methods do make sense only in case of public element,
88   // when the template is intanciated with private element the VR/VM are simply
89   // defaulted to allow everything (see gdcmTagToType.h default template for TagToType)
90   static VR  GetDictVR() { return (VR::VRType)(TagToType<Group, Element>::VRType); }
91   static VM  GetDictVM() { return (VM::VMType)(TagToType<Group, Element>::VMType); }
92
93   // Some extra dummy checks:
94   // Data Elements with a VR of SQ, OF, OW, OB or UN shall always have a Value Multiplicity of one.
95
96   unsigned int GetNumberOfValues()const {
97     return VMToLength<TVM>::Length;
98   }
99   // Implementation of Print is common to all Mode (ASCII/Binary)
100  // TODO: Can we print a \ when in ASCII...well I don't think so
101  // it would mean we used a bad VM then, right ?
102  void Print(std::ostream &os)const {
103    os « GetTag() « " ";
104    os « TagToType<Group,Element>::GetVRString()  « " ";
105    os « TagToType<Group,Element>::GetVMString()  « " ";
106    os « Internal[0]; // VM is at least guarantee to be one
107    for(unsigned int i=1; i<GetNumberOfValues(); ++i)
108      os « "," « Internal[i];
109    }
110
111  // copy:
112  //ArrayType GetValue(unsigned int idx = 0) {
113  //  assert( idx < GetNumberOfValues() );
114  //  return Internal[idx];
115  //}
116  //ArrayType operator[] (unsigned int idx) {
117  //  return GetValue(idx);
118  //}
119  // FIXME: is this always a good idea ?
120  // I do not think so, I prefer operator
121  //operator ArrayType () const { return Internal[0]; }
122
123  bool operator==(const Attribute &att)const
124{
125    return std::equal(Internal, Internal+GetNumberOfValues(),
126      att.GetValues());
127    }
128  bool operator!=(const Attribute &att)const
```

```
144 {
145     return !std::equal(Internal, Internal+GetNumberOfValues(),
146       att.GetValues());
147     }
148   bool operator<(const Attribute &att)const
149 {
150     return std::lexicographical_compare(Internal, Internal+GetNumberOfValues(),
151       att.GetValues(), att.GetValues() + att.GetNumberOfValues() );
152     }
153
154   ArrayType &GetValue(unsigned int idx = 0) {
155     assert( idx < GetNumberOfValues() );
156     return Internal[idx];
157   }
158   ArrayType & operator[] (unsigned int idx) {
159     return GetValue(idx);
160   }
161   // const reference
162   ArrayType const &GetValue(unsigned int idx = 0)const {
163     assert( idx < GetNumberOfValues() );
164     return Internal[idx];
165   }
166   ArrayType const & operator[] (unsigned int idx)const {
167     return GetValue(idx);
168   }
169   void SetValue(ArrayType v, unsigned int idx = 0) {
170     assert( idx < GetNumberOfValues() );
171     Internal[idx] = v;
172   }
173   void SetValues(const ArrayType* array, unsigned int numel = VMType ) {
174     assert( array && numel && numel == GetNumberOfValues() );
175     // std::copy is smarted than a memcpy, and will call memcpy when POD type
176     std::copy(array, array+numel, Internal);
177   }
178   const ArrayType* GetValues()const {
179     return Internal;
180   }
181
182   // API to talk to the run-time layer:  gdcm::DataElement
183   DataElement GetAsDataElement()const {
184     DataElement ret( GetTag() );
185     std::ostringstream os;
186     // os.imbue(std::locale::classic()); // This is not required AFAIK
187     EncodingImplementation<VRToEncoding<TVR>::Mode>::Write(Internal,
188       GetNumberOfValues(),os);
189     ret.SetVR( GetVR() );
190     assert( ret.GetVR() != VR::SQ );
191     if( (VR::VRType)VRToEncoding<TVR>::Mode == VR::VRASCII )
192       {
193       if( GetVR() != VR::UI )
194         {
195         if( os.str().size() % 2 )
196           {
197           os « " ";
198           }
199         }
200       }
201     VL::Type osStrSize = (VL::Type)os.str().size();
202     ret.SetByteValue( os.str().c_str(), osStrSize );
203     return ret;
204   }
205
206   void SetFromDataElement(DataElement const &de) {
207     // This is kind of hackish but since I do not generate other element than the first one:  0x6000 I
    should be ok:
208     assert( Tag(Group,Element) == de.GetTag() || Group == 0x6000 || Group == 0x5000 );
209     assert( GetVR() != VR::INVALID );
210     assert( GetVR().Compatible( de.GetVR() ) || de.GetVR() == VR::INVALID ); // In case of VR::INVALID
    cannot use the & operator
211     if( de.IsEmpty() ) return;
212     const ByteValue *bv = de.GetByteValue();
213 #ifdef GDCM_WORDS_BIGENDIAN
214     if( de.GetVR() == VR::UN /*|| de.GetVR() == VR::INVALID*/ )
215 #else
216     if( de.GetVR() == VR::UN || de.GetVR() == VR::INVALID )
217 #endif
218       {
219       SetByteValue(bv);
220       }
221     else
222       {
```

```
223       SetByteValueNoSwap(bv);
224       }
225   }
226   void Set(DataSet const &ds) {
227     SetFromDataElement( ds.GetDataElement( Tag(Group,Element) ) );
228   }
229   void SetFromDataSet(DataSet const &ds) {
230     if( ds.FindDataElement( Tag(Group,Element) ) &&
231       !ds.GetDataElement( Tag(Group,Element) ).IsEmpty() )
232       {
233       SetFromDataElement( ds.GetDataElement( Tag(Group,Element) ) );
234       }
235   }
236 protected:
237   void SetByteValueNoSwap(const ByteValue *bv) {
238     if( !bv ) return; // That would be bad...
239     assert( bv->GetPointer() && bv->GetLength() ); // [123]C element can be empty
240     //if( VRToEncoding<TVR>::Mode == VR::VRBINARY )
241     //  {
242     //  // always do a copy !
243     //  SetValues(bv->GetPointer(), bv->GetLength());
244     //  }
245     //else
246       {
247       std::stringstream ss;
248       std::string s = std::string( bv->GetPointer(), bv->GetLength() );
249       ss.str( s );
250       EncodingImplementation<VRToEncoding<TVR>::Mode>::ReadNoSwap(Internal,
251         GetNumberOfValues(),ss);
252       }
253   }
254   void SetByteValue(const ByteValue *bv) {
255     if( !bv ) return; // That would be bad...
256     assert( bv->GetPointer() && bv->GetLength() ); // [123]C element can be empty
257     //if( VRToEncoding<TVR>::Mode == VR::VRBINARY )
258     //  {
259     //  // always do a copy !
260     //  SetValues(bv->GetPointer(), bv->GetLength());
261     //  }
262     //else
263       {
264       std::stringstream ss;
265       std::string s = std::string( bv->GetPointer(), bv->GetLength() );
266       ss.str( s );
267       EncodingImplementation<VRToEncoding<TVR>::Mode>::Read(Internal,
268         GetNumberOfValues(),ss);
269       }
270   }
271 #if 0 // TODO  FIXME the implicit way:
272   // explicit:
273   void Read(std::istream &_is) {
274     const uint16_t cref[] = { Group, Element };
275     uint16_t c[2];
276     _is.read((char*)&c, sizeof(c));
277     assert( c[0] == cref[0] && c[1] == cref[1] );
278     char vr[2];
279     _is.read(vr, 2); // Check consistency ?
280     const uint32_t lref = GetLength() * sizeof( typename VRToType<TVR>::Type );
281     uint32_t l = VRVLSize< (TVR & VR::VL32) >::Read(_is);
282     l /= sizeof( typename VRToType<TVR>::Type );
283     return EncodingImplementation<VRToEncoding<TVR>::Mode>::Read(Internal,
284       l,_is);
285   }
286   void Write(std::ostream &_os)const {
287     uint16_t c[] = { Group, Element };
288     _os.write((char*)&c, 4);
289     uint32_t l = GetLength() * sizeof( typename VRToType<TVR>::Type );
290     _os.write((char*)&l, 4);
291     return EncodingImplementation<VRToEncoding<TVR>::Mode>::Write(Internal,
292       GetLength(),_os);
293     }
294   void Read(std::istream &_is) {
295     uint16_t cref[] = { Group, Element };
296     uint16_t c[2];
297     _is.read((char*)&c, 4);
298     const uint32_t lref = GetLength() * sizeof( typename VRToType<TVR>::Type );
299     uint32_t l;
300     _is.read((char*)&l, 4);
301     l /= sizeof( typename VRToType<TVR>::Type );
302      return EncodingImplementation<VRToEncoding<TVR>::Mode>::Read(Internal,
303       l,_is);
```

```
304     }
305   void Write(std::ostream &_os)const {
306     uint16_t c[] = { Group, Element };
307     _os.write((char*)&c, 4);
308     uint32_t l = GetLength() * sizeof( typename VRToType<TVR>::Type );
309     _os.write((char*)&l, 4);
310     return EncodingImplementation<VRToEncoding<TVR>::Mode>::Write(Internal,
311       GetLength(),_os);
312     }
313 #endif
314
315 };
316
317 template<uint16_t Group, uint16_t Element, long long TVR >
318 class Attribute<Group,Element,TVR,VM::VM1>
319 {
320 public:
321   typedef typename VRToType<TVR>::Type ArrayType;
322   enum { VMType = VMToLength<VM::VM1>::Length };
323   //ArrayType Internal[VMToLength<TVM>::Length];
324   ArrayType Internal;
325   GDCM_STATIC_ASSERT( VMToLength<VM::VM1>::Length == 1 );
326
327   // Make sure that user specified VR/VM are compatible with the public dictionary:
328   GDCM_STATIC_ASSERT( ((VR::VRType)TVR & (VR::VRType)(TagToType<Group, Element>::VRType)) );
329   GDCM_STATIC_ASSERT( ((VM::VMType)VM::VM1 & (VM::VMType)(TagToType<Group, Element>::VMType)) );
330   GDCM_STATIC_ASSERT( ((((VR::VRType)TVR & VR::VR_VM1) && ((VM::VMType)VM::VM1 == VM::VM1) )
331                       || !((VR::VRType)TVR & VR::VR_VM1) ) );
332
333   static Tag GetTag() { return Tag(Group,Element); }
334   static VR  GetVR()  { return (VR::VRType)TVR; }
335   static VM  GetVM()  { return (VM::VMType)VM::VM1; }
336
337   // The following two methods do make sense only in case of public element,
338   // when the template is intanciated with private element the VR/VM are simply
339   // defaulted to allow everything (see gdcmTagToType.h default template for TagToType)
340   static VR  GetDictVR() { return (VR::VRType)(TagToType<Group, Element>::VRType); }
341   static VM  GetDictVM() { return (VM::VMType)(TagToType<Group, Element>::VMType); }
342
343   // Some extra dummy checks:
344   // Data Elements with a VR of SQ, OF, OW, OB or UN shall always have a Value Multiplicity of one.
345
346   unsigned int GetNumberOfValues()const {
347     return VMToLength<VM::VM1>::Length;
348   }
349   // Implementation of Print is common to all Mode (ASCII/Binary)
350   // TODO: Can we print a \ when in ASCII...well I don't think so
351   // it would mean we used a bad VM then, right ?
352   void Print(std::ostream &os)const {
353     os « GetTag() « " ";
354     os « TagToType<Group,Element>::GetVRString()  « " ";
355     os « TagToType<Group,Element>::GetVMString()  « " ";
356     os « Internal; // VM is at least guarantee to be one
357   }
358   // copy:
359   //ArrayType GetValue(unsigned int idx = 0) {
360   //  assert( idx < GetNumberOfValues() );
361   //  return Internal[idx];
362   //}
363   //ArrayType operator[] (unsigned int idx) {
364   //  return GetValue(idx);
365   //}
366   // FIXME: is this always a good idea ?
367   // I do not think so, I prefer operator
368   //operator ArrayType () const { return Internal[0]; }
369
370   bool operator==(const Attribute &att)const
371 {
372     return std::equal(&Internal, &Internal+GetNumberOfValues(),
373       att.GetValues());
374     }
375   bool operator!=(const Attribute &att)const
376 {
377     return !std::equal(&Internal, &Internal+GetNumberOfValues(),
378       att.GetValues());
379     }
380   bool operator<(const Attribute &att)const
381 {
382     return std::lexicographical_compare(&Internal, &Internal+GetNumberOfValues(),
383       att.GetValues(), att.GetValues() + att.GetNumberOfValues() );
384     }
```

```
385
386   ArrayType &GetValue() {
387 //    assert( idx < GetNumberOfValues() );
388     return Internal;
389   }
390 //  ArrayType & operator[] (unsigned int idx) {
391 //    return GetValue(idx);
392 //  }
393   // const reference
394   ArrayType const &GetValue()const {
395     //assert( idx < GetNumberOfValues() );
396     return Internal;
397   }
398   //ArrayType const & operator[] () const {
399   //  return GetValue();
400   //}
401   void SetValue(ArrayType v) {
402 //    assert( idx < GetNumberOfValues() );
403     Internal = v;
404   }
405 /*  void SetValues(const ArrayType* array, unsigned int numel = VMType ) {
406 assert( array && numel && numel == GetNumberOfValues() );
407 // std::copy is smarted than a memcpy, and will call memcpy when POD type
408 std::copy(array, array+numel, Internal);
409 }
410 */
411
412   // FIXME Should we remove this function ?
413   const ArrayType* GetValues()const {
414     return &Internal;
415   }
416
417   // API to talk to the run-time layer:  gdcm::DataElement
418   DataElement GetAsDataElement()const {
419     DataElement ret( Tag(Group,Element) );
420     std::ostringstream os;
421     // os.imbue(std::locale::classic()); // This is not required AFAIK
422     EncodingImplementation<VRToEncoding<TVR>::Mode>::Write(&Internal,
423       GetNumberOfValues(),os);
424     ret.SetVR( GetVR() );
425     assert( ret.GetVR() != VR::SQ );
426     if( (VR::VRType)VRToEncoding<TVR>::Mode == VR::VRASCII )
427       {
428       if( GetVR() != VR::UI )
429         {
430         if( os.str().size() % 2 )
431           {
432           os « " ";
433           }
434         }
435       }
436     VL::Type osStrSize = (VL::Type)os.str().size();
437     ret.SetByteValue( os.str().c_str(), osStrSize );
438     return ret;
439   }
440
441   void SetFromDataElement(DataElement const &de) {
442     // This is kind of hackish but since I do not generate other element than the first one:  0x6000 I
    should be ok:
443     assert( Tag(Group,Element) == de.GetTag() || Group == 0x6000 || Group == 0x5000 );
444     assert( GetVR() != VR::INVALID );
445     assert( GetVR().Compatible( de.GetVR() ) || de.GetVR() == VR::INVALID ); // In case of VR::INVALID
    cannot use the & operator
446     if( de.IsEmpty() ) return;
447     const ByteValue *bv = de.GetByteValue();
448 #ifdef GDCM_WORDS_BIGENDIAN
449     if( de.GetVR() == VR::UN /*|| de.GetVR() == VR::INVALID*/ )
450 #else
451     if( de.GetVR() == VR::UN || de.GetVR() == VR::INVALID )
452 #endif
453       {
454       SetByteValue(bv);
455       }
456     else
457       {
458       SetByteValueNoSwap(bv);
459       }
460   }
461   void Set(DataSet const &ds) {
462     SetFromDataElement( ds.GetDataElement( Tag(Group,Element) ) );
463   }
```

```
464   void SetFromDataSet(DataSet const &ds) {
465     if( ds.FindDataElement( Tag(Group,Element) ) &&
466       !ds.GetDataElement( Tag(Group,Element) ).IsEmpty() )
467       {
468       SetFromDataElement( ds.GetDataElement( Tag(Group,Element) ) );
469       }
470   }
471 protected:
472   void SetByteValueNoSwap(const ByteValue *bv) {
473     if( !bv ) return; // That would be bad...
474     assert( bv->GetPointer() && bv->GetLength() ); // [123]C element can be empty
475     //if( VRToEncoding<TVR>::Mode == VR::VRBINARY )
476     //  {
477     //  // always do a copy !
478     //  SetValues(bv->GetPointer(), bv->GetLength());
479     //  }
480     //else
481       {
482       std::stringstream ss;
483       std::string s = std::string( bv->GetPointer(), bv->GetLength() );
484       ss.str( s );
485       EncodingImplementation<VRToEncoding<TVR>::Mode>::ReadNoSwap(&Internal,
486         GetNumberOfValues(),ss);
487       }
488   }
489   void SetByteValue(const ByteValue *bv) {
490     if( !bv ) return; // That would be bad...
491     assert( bv->GetPointer() && bv->GetLength() ); // [123]C element can be empty
492     //if( VRToEncoding<TVR>::Mode == VR::VRBINARY )
493     //  {
494     //  // always do a copy !
495     //  SetValues(bv->GetPointer(), bv->GetLength());
496     //  }
497     //else
498       {
499       std::stringstream ss;
500       std::string s = std::string( bv->GetPointer(), bv->GetLength() );
501       ss.str( s );
502       EncodingImplementation<VRToEncoding<TVR>::Mode>::Read(&Internal,
503         GetNumberOfValues(),ss);
504       }
505   }
506 #if 0 // TODO  FIXME the implicit way:
507   // explicit:
508   void Read(std::istream &_is) {
509     const uint16_t cref[] = { Group, Element };
510     uint16_t c[2];
511     _is.read((char*)&c, sizeof(c));
512     assert( c[0] == cref[0] && c[1] == cref[1] );
513     char vr[2];
514     _is.read(vr, 2); // Check consistency ?
515     const uint32_t lref = GetLength() * sizeof( typename VRToType<TVR>::Type );
516     uint32_t l = VRVLSize< (TVR & VR::VL32) >::Read(_is);
517     l /= sizeof( typename VRToType<TVR>::Type );
518     return EncodingImplementation<VRToEncoding<TVR>::Mode>::Read(Internal,
519       l,_is);
520   }
521   void Write(std::ostream &_os)const {
522     uint16_t c[] = { Group, Element };
523     _os.write((char*)&c, 4);
524     uint32_t l = GetLength() * sizeof( typename VRToType<TVR>::Type );
525     _os.write((char*)&l, 4);
526     return EncodingImplementation<VRToEncoding<TVR>::Mode>::Write(Internal,
527       GetLength(),_os);
528     }
529   void Read(std::istream &_is) {
530     uint16_t cref[] = { Group, Element };
531     uint16_t c[2];
532     _is.read((char*)&c, 4);
533     const uint32_t lref = GetLength() * sizeof( typename VRToType<TVR>::Type );
534     uint32_t l;
535     _is.read((char*)&l, 4);
536     l /= sizeof( typename VRToType<TVR>::Type );
537      return EncodingImplementation<VRToEncoding<TVR>::Mode>::Read(Internal,
538       l,_is);
539     }
540   void Write(std::ostream &_os)const {
541     uint16_t c[] = { Group, Element };
542     _os.write((char*)&c, 4);
543     uint32_t l = GetLength() * sizeof( typename VRToType<TVR>::Type );
544     _os.write((char*)&l, 4);
```

```
545     return EncodingImplementation<VRToEncoding<TVR>::Mode>::Write(Internal,
546       GetLength(),_os);
547     }
548 #endif
549
550 };
551
552 // No need to repeat default template arg, since primary template
553 // will be used to generate the default arguments
554 template<uint16_t Group, uint16_t Element, long long TVR >
555 class Attribute<Group,Element,TVR,VM::VM1_n>
556 {
557 public:
558   typedef typename VRToType<TVR>::Type ArrayType;
559
560   // Make sure that user specified VR/VM are compatible with the public dictionary:
561   GDCM_STATIC_ASSERT( ((VR::VRType)TVR & (VR::VRType)(TagToType<Group, Element>::VRType)) );
562   GDCM_STATIC_ASSERT( (VM::VM1_n & (VM::VMType)(TagToType<Group, Element>::VMType)) );
563   GDCM_STATIC_ASSERT( (((((VR::VRType)TVR & VR::VR_VM1) && ((VM::VMType)TagToType<Group,Element>::VMType ==
      VM::VM1) )
564                     || !((VR::VRType)TVR & VR::VR_VM1) ) );
565
566   static Tag GetTag() { return Tag(Group,Element); }
567   static VR  GetVR()  { return (VR::VRType)TVR; }
568   static VM  GetVM()  { return VM::VM1_n; }
569
570   static VR  GetDictVR() { return (VR::VRType)(TagToType<Group, Element>::VRType); }
571   static VM  GetDictVM() { return GetVM(); }
572
573   // This the way to prevent default initialization
574   explicit Attribute() { Internal=nullptr; Length=0; Own = true; }
575   ~Attribute() {
576     if( Own ) {
577       delete[] Internal;
578     }
579     Internal = nullptr; // paranoid
580   }
581
582   unsigned int GetNumberOfValues()const { return Length; }
583
584   void SetNumberOfValues(unsigned int numel)
585     {
586     SetValues(nullptr, numel, true);
587     }
588
589   const ArrayType* GetValues()const {
590     return Internal;
591   }
592   void Print(std::ostream &os)const {
593     os « GetTag() « " ";
594     os « GetVR()  « " ";
595     os « GetVM()  « " ";
596     os « Internal[0]; // VM is at least guarantee to be one
597     for(unsigned int i=1; i<GetNumberOfValues(); ++i)
598       os « "," « Internal[i];
599     }
600   ArrayType &GetValue(unsigned int idx = 0) {
601     assert( idx < GetNumberOfValues() );
602     return Internal[idx];
603   }
604   ArrayType &operator[] (unsigned int idx) {
605     return GetValue(idx);
606   }
607   // const reference
608   ArrayType const &GetValue(unsigned int idx = 0)const {
609     assert( idx < GetNumberOfValues() );
610     return Internal[idx];
611   }
612   ArrayType const & operator[] (unsigned int idx)const {
613     return GetValue(idx);
614   }
615   void SetValue(unsigned int idx, ArrayType v) {
616     assert( idx < GetNumberOfValues() );
617     Internal[idx] = v;
618   }
619   void SetValue(ArrayType v) { SetValue(0, v); }
620
621   void SetValues(const ArrayType *array, unsigned int numel, bool own = false)
622     {
623     if( Internal ) // were we used before ?
624       {
```

```
625       // yes !
626       if( Own ) delete[] Internal;
627       Internal = nullptr;
628       }
629     Own = own;
630     Length = numel;
631     assert( Internal == nullptr );
632     if( own ) // make a copy:
633       {
634       Internal = new ArrayType[numel];
635       if( array && numel )
636         std::copy(array, array+numel, Internal);
637       }
638     else // pass pointer
639       {
640       Internal = const_cast<ArrayType*>(array);
641       }
642     // postcondition
643     assert( numel == GetNumberOfValues() );
644     }
645
646   DataElement GetAsDataElement()const {
647     DataElement ret( GetTag() );
648     std::ostringstream os;
649     if( Internal )
650       {
651       EncodingImplementation<VRToEncoding<TVR>::Mode>::Write(Internal,
652         GetNumberOfValues(),os);
653       if( (VR::VRType)VRToEncoding<TVR>::Mode == VR::VRASCII )
654         {
655         if( GetVR() != VR::UI )
656           {
657           if( os.str().size() % 2 )
658             {
659             os « " ";
660             }
661           }
662         }
663       }
664     ret.SetVR( GetVR() );
665     assert( ret.GetVR() != VR::SQ );
666     VL::Type osStrSize = (VL::Type) os.str().size();
667     ret.SetByteValue( os.str().c_str(), osStrSize);
668     return ret;
669   }
670   void SetFromDataElement(DataElement const &de) {
671     // This is kind of hackish but since I do not generate other element than the first one:  0x6000 I
    should be ok:
672     assert( GetTag() == de.GetTag() || GetTag().GetGroup() == 0x6000
673       || GetTag().GetGroup() == 0x5000 );
674     assert( GetVR().Compatible( de.GetVR() ) ); // In case of VR::INVALID cannot use the & operator
675     assert( !de.IsEmpty() );
676     const ByteValue *bv = de.GetByteValue();
677     SetByteValue(bv);
678   }
679   void Set(DataSet const &ds) {
680     SetFromDataElement( ds.GetDataElement( GetTag() ) );
681   }
682   void SetFromDataSet(DataSet const &ds) {
683     if( ds.FindDataElement( GetTag() ) &&
684       !ds.GetDataElement( GetTag() ).IsEmpty() )
685       {
686       SetFromDataElement( ds.GetDataElement( GetTag() ) );
687       }
688   }
689 protected:
690   void SetByteValue(const ByteValue *bv) {
691     assert( bv ); // FIXME
692     std::stringstream ss;
693     std::string s = std::string( bv->GetPointer(), bv->GetLength() );
694     Length = bv->GetLength(); // HACK FIXME
695     ss.str( s );
696     ArrayType *internal;
697     ArrayType buffer[256];
698     if( bv->GetLength() < 256 )
699       {
700       internal = buffer;
701       }
702     else
703       {
704       internal = new ArrayType[(VL::Type)bv->GetLength()]; // over allocation
```

```
705       }
706     EncodingImplementation<VRToEncoding<TVR>::Mode>::ReadComputeLength(internal, Length, ss);
707     SetValues( internal, Length, true );
708     if( !(bv->GetLength() < 256) )
709       {
710       delete[] internal;
711       }
712     //EncodingImplementation<VRToEncoding<TVR>::Mode>::Read(Internal,
713     //  GetNumberOfValues(),ss);
714   }
715
716 private:
717   ArrayType *Internal;
718   unsigned int Length;
719   bool Own :  1;
720 };
721
722 template<uint16_t Group, uint16_t Element, long long TVR>
723 class Attribute<Group,Element,TVR,VM::VM1_3> :  public Attribute<Group,Element,TVR,VM::VM1_n>
724 {
725 public:
726   VM  GetVM()const { return VM::VM1_3; }
727 };
728
729 template<uint16_t Group, uint16_t Element, long long TVR>
730 class Attribute<Group,Element,TVR,VM::VM1_8> :  public Attribute<Group,Element,TVR,VM::VM1_n>
731 {
732 public:
733   VM  GetVM()const { return VM::VM1_8; }
734 };
735
736 template<uint16_t Group, uint16_t Element, long long TVR>
737 class Attribute<Group,Element,TVR,VM::VM2_n> :  public Attribute<Group,Element,TVR,VM::VM1_n>
738 {
739 public:
740   VM  GetVM()const { return VM::VM2_n; }
741 };
742
743 template<uint16_t Group, uint16_t Element, long long TVR>
744 class Attribute<Group,Element,TVR,VM::VM2_2n> :  public Attribute<Group,Element,TVR,VM::VM2_n>
745 {
746 public:
747   static VM  GetVM() { return VM::VM2_2n; }
748 };
749
750 template<uint16_t Group, uint16_t Element, long long TVR>
751 class Attribute<Group,Element,TVR,VM::VM3_n> :  public Attribute<Group,Element,TVR,VM::VM1_n>
752 {
753 public:
754   static VM  GetVM() { return VM::VM3_n; }
755 };
756
757 template<uint16_t Group, uint16_t Element, long long TVR>
758 class Attribute<Group,Element,TVR,VM::VM3_3n> :  public Attribute<Group,Element,TVR,VM::VM3_n>
759 {
760 public:
761   static VM  GetVM() { return VM::VM3_3n; }
762 };
763
764
765 // For particular case for ASCII string
766 // WARNING: This template explicitly instantiates a particular
767 // EncodingImplementation THEREFORE it is required to be declared after the
768 // EncodingImplementation is needs (doh!)
769 #if 0
770 template<int TVM>
771 class Attribute<TVM>
772 {
773 public:
774   Attribute(const char array[])
775     {
776     unsigned int i = 0;
777     const char sep = '\\';
778     std::string sarray = array;
779     std::string::size_type pos1 = 0;
780     std::string::size_type pos2 = sarray.find(sep, pos1+1);
781     while(pos2 != std::string::npos)
782       {
783       Internal[i++] = sarray.substr(pos1, pos2-pos1);
784       pos1 = pos2+1;
785       pos2 = sarray.find(sep, pos1+1);
```

```
786       }
787     Internal[i] = sarray.substr(pos1, pos2-pos1);
788     // Shouldn't we do the contrary, since we know how many separators
789     // (and default behavior is to discard anything after the VM declared
790     assert( GetLength()-1 == i );
791     }
792
793   unsigned long GetLength()const {
794     return VMToLength<TVM>::Length;
795   }
796   // Implementation of Print is common to all Mode (ASCII/Binary)
797   void Print(std::ostream &_os)const {
798     _os « Internal[0]; // VM is at least guarantee to be one
799     for(int i=1; i<VMToLength<TVM>::Length; ++i)
800       _os « "," « Internal[i];
801     }
802
803   void Read(std::istream &_is) {
804     EncodingImplementation<VR::VRASCII>::Read(Internal, GetLength(),_is);
805     }
806   void Write(std::ostream &_os)const {
807     EncodingImplementation<VR::VRASCII>::Write(Internal, GetLength(),_os);
808     }
809 private:
810   typename String Internal[VMToLength<TVM>::Length];
811 };
812
813 template< int TVM>
814 class Attribute<VR::PN, TVM> :  public StringAttribute<TVM>
815 {
816 };
817 #endif
818
819 #if 0
820
821 // Implementation for the undefined length (dynamically allocated array)
822 template<int TVR>
823 class Attribute<TVR, VM::VM1_n>
824 {
825 public:
826   // This the way to prevent default initialization
827   explicit Attribute() { Internal=0; Length=0; }
828   ~Attribute() {
829     delete[] Internal;
830     Internal = 0;
831   }
832
833   // Length manipulation
834   // SetLength should really be protected anyway...all operation
835   // should go through SetArray
836   unsigned long GetLength()const { return Length; }
837   typedef typename VRToType<TVR>::Type ArrayType;
838   void SetLength(unsigned long len) {
839     const unsigned int size = sizeof(ArrayType);
840     if( len ) {
841       if( len > Length ) {
842         // perform realloc
843         assert( (len / size) * size == len );
844         ArrayType *internal = new ArrayType[len / size];
845         memcpy(internal, Internal, Length * size);
846         delete[] Internal;
847         Internal = internal;
848         }
849       }
850     Length = len / size;
851   }
852
853   // If save is set to zero user should not delete the pointer
854   //void SetArray(const typename VRToType<TVR>::Type *array, int len, bool save = false)
855   void SetArray(const ArrayType *array, unsigned long len,
856     bool save = false) {
857     if( save ) {
858       SetLength(len); // realloc
859       memcpy(Internal, array, len/*/sizeof(ArrayType)*/);
860       }
861     else {
862       // TODO rewrite this stupid code:
863       Length = len;
864       //Internal = array;
865       assert(0);
866       }
```

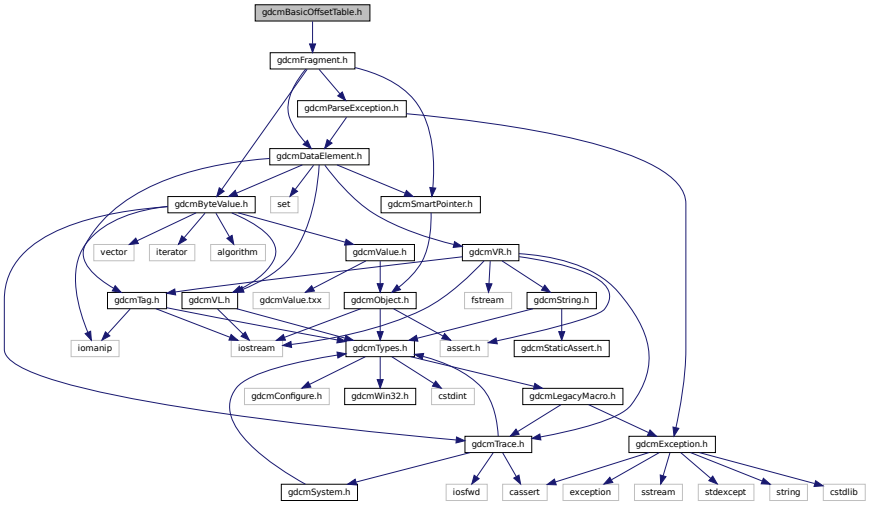
```
867   }
868   // Implementation of Print is common to all Mode (ASCII/Binary)
869   void Print(std::ostream &_os)const {
870     assert( Length );
871     assert( Internal );
872     _os « Internal[0]; // VM is at least guarantee to be one
873     const unsigned long length = GetLength() < 25 ?  GetLength() :  25;
874     for(unsigned long i=1; i<length; ++i)
875       _os « "," « Internal[i];
876     }
877   void Read(std::istream &_is) {
878     EncodingImplementation<VRToEncoding<TVR>::Mode>::Read(Internal,
879       GetLength(),_is);
880     }
881   void Write(std::ostream &_os)const {
882     EncodingImplementation<VRToEncoding<TVR>::Mode>::Write(Internal,
883       GetLength(),_os);
884     }
885
886   Attribute(const Attribute&_val) {
887     if( this != &_val) {
888       *this = _val;
889       }
890     }
891
892   Attribute &operator=(const Attribute &_val) {
893     Length = 0; // SYITF
894     Internal = 0;
895     SetArray(_val.Internal, _val.Length, true);
896     return *this;
897     }
898
899 private:
900   typename VRToType<TVR>::Type *Internal;
901   unsigned long Length; // unsigned int ??
902 };
903
904 //template <int TVM = VM::VM1_n>
905 //class Attribute<VR::OB, TVM > :  public Attribute<VR::OB, VM::VM1_n> {};
906
907 // Partial specialization for derivatives of 1-n :  2-n, 3-n ...
908 template<int TVR>
909 class Attribute<TVR, VM::VM2_n> :  public Attribute<TVR, VM::VM1_n>
910 {
911 public:
912   typedef Attribute<TVR, VM::VM1_n> Parent;
913   void SetLength(int len) {
914     if( len <= 1 ) return;
915     Parent::SetLength(len);
916   }
917 };
918 template<int TVR>
919 class Attribute<TVR, VM::VM2_2n> :  public Attribute<TVR, VM::VM2_n>
920 {
921 public:
922   typedef Attribute<TVR, VM::VM2_n> Parent;
923   void SetLength(int len) {
924     if( len % 2 ) return;
925     Parent::SetLength(len);
926   }
927 };
928 template<int TVR>
929 class Attribute<TVR, VM::VM3_n> :  public Attribute<TVR, VM::VM1_n>
930 {
931 public:
932   typedef Attribute<TVR, VM::VM1_n> Parent;
933   void SetLength(int len) {
934     if( len <= 2 ) return;
935     Parent::SetLength(len);
936   }
937 };
938 template<int TVR>
939 class Attribute<TVR, VM::VM3_3n> :  public Attribute<TVR, VM::VM3_n>
940 {
941 public:
942   typedef Attribute<TVR, VM::VM3_n> Parent;
943   void SetLength(int len) {
944     if( len % 3 ) return;
945     Parent::SetLength(len);
946   }
947 };
```

```
948
949
950 //template<int T> struct VRToLength;
951 //template <> struct VRToLength<VR::AS>
952 //{ enum { Length  = VM::VM1 }; }
953 //template<>
954 //class Attribute<VR::AS> :  public Attribute<VR::AS, VRToLength<VR::AS>::Length >
955
956 // only 0010 1010 AS 1 Patient's Age
957 template<>
958 class Attribute<VR::AS, VM::VM5>
959 {
960 public:
961   char Internal[VMToLength<VM::VM5>::Length];
962   void Print(std::ostream &_os)const {
963     _os « Internal;
964     }
965 };
966
967 template <>
968 class Attribute<VR::OB, VM::VM1> :  public Attribute<VR::OB, VM::VM1_n> {};
969 // Make it impossible to compile any other cases:
970 template <int TVM> class Attribute<VR::OB, TVM>;
971
972 // Same for OW:
973 template <>
974 class Attribute<VR::OW, VM::VM1> :  public Attribute<VR::OW, VM::VM1_n> {};
975 // Make it impossible to compile any other cases:
976 template <int TVM> class Attribute<VR::OW, TVM>;
977 #endif
978
979 #if 0
980 template<>
981 class Attribute<0x7fe0,0x0010, VR::OW, VM::VM1>
982 {
983 public:
984   char *Internal;
985   unsigned long Length; // unsigned int ??
986
987   void Print(std::ostream &_os)const {
988     _os « Internal[0];
989     }
990   void SetBytes(char *bytes, unsigned long length) {
991     Internal = bytes;
992     Length = length;
993   }
994   void Read(std::istream &_is) {
995      uint16_t c[2];
996     _is.read((char*)&c, 4);
997     uint32_t l;
998     _is.read((char*)&l, 4);
999     Length = l;
1000     _is.read( Internal, Length );
1001     }
1002   void Write(std::ostream &_os)const {
1003      uint16_t c[] = {0x7fe0, 0x0010};
1004     _os.write((char*)&c, 4);
1005     _os.write((char*)&Length, 4);
1006     _os.write( Internal, Length );
1007     }
1008 };
1009 #endif
1010
1011 /*
1012 // Removing Attribute for SQ for now...
1013 template<uint16_t Group, uint16_t Element, typename SQA>
1014 class Attribute<Group,Element, VR::SQ, VM::VM1, SQA>
1015 {
1016 public:
1017 SQA sqa;
1018 void Print(std::ostream &_os) const {
1019 _os « Tag(Group,Element);
1020 sqa.Print(_os « std::endl « '\t');
1021 }
1022 void Write(std::ostream &_os) const {
1023 uint16_t c[] = {Group, Element};
1024 _os.write((char*)&c, 4);
1025 uint32_t undef = 0xffffffff;
1026 _os.write((char*)&undef, 4);
1027 uint16_t item_beg[] = {0xfffe,0xe000};
1028 _os.write((char*)&item_beg, 4);
```

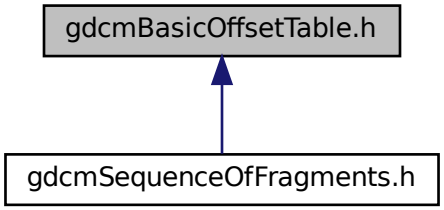
```
1029 _os.write((char*)&undef, 4);
1030 sqa.Write(_os);
1031 uint16_t item_end[] = {0xfffe,0xe00d};
1032 _os.write((char*)&item_end, 4);
1033 uint32_t zero = 0x0;
1034 _os.write((char*)&zero, 4);
1035 uint16_t seq_end[] = {0xfffe, 0xe0dd};
1036 _os.write((char*)&seq_end, 4);
1037 _os.write((char*)&zero, 4);
1038 }
1039 };
1040 */
1041
1047 } // namespace gdcm_ns
1048
1049 #endif //GDCMATTRIBUTE_H
```

11.111 gdcmBasicOffsetTable.h File Reference

`#include "gdcmFragment.h"`
Include dependency graph for gdcmBasicOffsetTable.h:



This graph shows which files directly or indirectly include this file:



Classes

- class gdcm::BasicOffsetTable

  Class to represent a BasicOffsetTable.

Namespaces

- namespace gdcm

Functions

- std::ostream & gdcm::operator<< (std::ostream &os, const BasicOffsetTable &val)

11.112 gdcmBasicOffsetTable.h

Go to the documentation of this file.

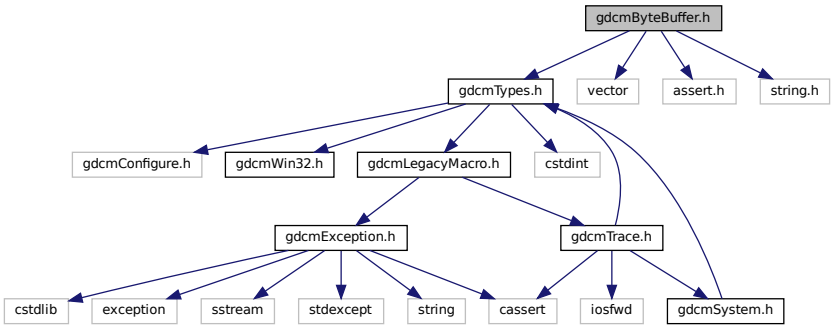
```
1 /*=========================================================================
2
3 Program:  GDCM (Grassroots DICOM). A DICOM library
4
5 Copyright (c) 2006-2011 Mathieu Malaterre
6 All rights reserved.
7 See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.
8
9 This software is distributed WITHOUT ANY WARRANTY; without even
10 the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
11 PURPOSE.  See the above copyright notice for more information.
12
13 =========================================================================*/
14
15 #ifndef GDCMBASICOFFSETTABLE_H
16 #define GDCMBASICOFFSETTABLE_H
17
18 #include "gdcmFragment.h"
19
20 namespace gdcm_ns
21 {
26 class GDCM_EXPORT BasicOffsetTable :  public Fragment
27 {
28 //protected:
29 //  void SetTag(const Tag &t);
30 public:
31   BasicOffsetTable() :  Fragment() {}
32   friend std::ostream &operator«(std::ostream &os, const BasicOffsetTable &val);
33
34 /*
35 VL GetLength() const {
36 assert( !ValueLengthField.IsUndefined() );
37 assert( !ValueField || ValueField->GetLength() == ValueLengthField );
38 return TagField.GetLength() + ValueLengthField.GetLength()
39 + ValueLengthField;
40 }
41 */
42
43   template <typename TSwap>
44   std::istream &Read(std::istream &is) {
45     // Superclass
46     const Tag itemStart(0xfffe, 0xe000);
47     const Tag seqDelItem(0xfffe,0xe0dd);
48     if( !TagField.Read<TSwap>(is) )
49       {
50       assert(0 && "Should not happen");
51       return is;
52       }
53     //assert( TagField == itemStart );
```

```
54      if( TagField != itemStart )
55        {
56        // Bug_Siemens_PrivateIconNoItem.dcm
57        //gdcmDebugMacro( "Could be Bug_Siemens_PrivateIconNoItem.dcm" );
58        ParseException pe;
59        pe.SetLastElement(*this);
60        //throw "SIEMENS Icon thingy";
61        throw pe;
62        }
63      if( !ValueLengthField.Read<TSwap>(is) )
64        {
65        assert(0 && "Should not happen");
66        return is;
67        }
68      // Self
69      SmartPointer<ByteValue> bv = new ByteValue;
70      bv->SetLength(ValueLengthField);
71      if( !bv->Read<TSwap>(is) )
72        {
73        gdcmAssertAlwaysMacro(0 && "Should not happen");
74        return is;
75        }
76      ValueField = bv;
77      return is;
78      }
79
80  /*
81  template <typename TSwap>
82  std::ostream &Write(std::ostream &os) const {
83  const Tag itemStart(0xfffe, 0xe000);
84  const Tag seqDelItem(0xfffe,0xe0dd);
85  if( !TagField.Write<TSwap>(os) )
86  {
87  assert(0 && "Should not happen");
88  return os;
89  }
90  assert( TagField == itemStart );
91  if( !ValueLengthField.Write<TSwap>(os) )
92  {
93  assert(0 && "Should not happen");
94  return os;
95  }
96  if( ValueLengthField )
97  {
98  // Self
99  const ByteValue *bv = GetByteValue();
100 assert( bv );
101 assert( bv->GetLength() == ValueLengthField );
102 if( !bv->Write<TSwap>(os) )
103 {
104 assert(0 && "Should not happen");
105 return os;
106 }
107 }
108 return os;
109 }
110 */
111 };
112 //-----------------------------------------------------------------------------
113 inline std::ostream &operator«(std::ostream &os, const BasicOffsetTable &val)
114 {
115   os « " BasicOffsetTable Length=" « val.ValueLengthField « std::endl;
116   if( val.ValueField )
117     {
118     const ByteValue *bv = val.GetByteValue();
119     assert( bv );
120     os « *bv;
121     }
122
123   return os;
124 }
125
126
127 } // end namespace gdcm_ns
128
129 #endif //GDCMBASICOFFSETTABLE_H
```

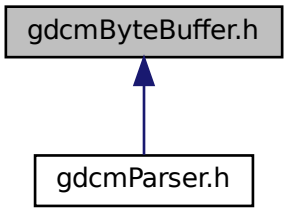
11.113 gdcmByteBuffer.h File Reference

```
#include "gdcmTypes.h"
#include <vector>
#include <assert.h>
#include <string.h>
```

Include dependency graph for gdcmByteBuffer.h:



This graph shows which files directly or indirectly include this file:



Classes

- class gdcm::ByteBuffer

  ByteBuffer.

Namespaces

- namespace gdcm

11.114 gdcmByteBuffer.h

Go to the documentation of this file.

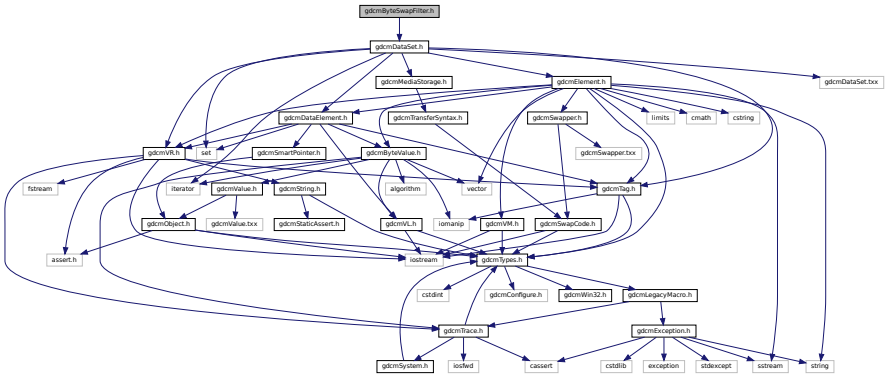
```
1 /*=========================================================================
2
3 Program:  GDCM (Grassroots DICOM). A DICOM library
4
5 Copyright (c) 2006-2011 Mathieu Malaterre
6 All rights reserved.
7 See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.
8
9 This software is distributed WITHOUT ANY WARRANTY; without even
10 the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
11 PURPOSE.  See the above copyright notice for more information.
12
13 =========================================================================*/
14 #ifndef GDCMBYTEBUFFER_H
15 #define GDCMBYTEBUFFER_H
16
17 #include "gdcmTypes.h"
18 #include <vector>
19 #include <assert.h>
20 #include <string.h> // memmove
21
22 #error should not be used
23
24 namespace gdcm
25 {
34 class ByteBuffer
35 {
36   static const int InitBufferSize = 1024;
37 public:
38   ByteBuffer() :  Start(0), End(0),Limit(0) {}
39   char *Get(int len)
40     {
41     char *buffer = &Internal[0];
42     if (len > Limit - End)
43       {
44       // FIXME avoid integer overflow
45       int neededSize = len + (End - Start);
46       if (neededSize  <= Limit - buffer)
47         {
48         memmove(buffer, Start, End - Start);
49         End = buffer + (End - Start);
50         Start = buffer;
51         }
52       else
53         {
54         char *newBuf;
55         int bufferSize = Limit - Start;
56         if ( bufferSize == 0 )
57           {
58           bufferSize = InitBufferSize;
59           }
60         do
61           {
62           bufferSize *= 2;
63           } while (bufferSize < neededSize);
64         //newBuf = malloc(bufferSize);
65         try
66           {
67           Internal.reserve(bufferSize);
68           newBuf = &Internal[0];
69           }
70         catch(...)
71           {
72           //errorCode = NoMemoryError;
73           return 0;
74           }
75         Limit = newBuf + bufferSize;
76
77         if (Start)
78           {
79           memcpy(newBuf, Start, End - Start);
80           }
81         End = newBuf + (End - Start);
82         Start = /*buffer =*/ newBuf;
83         }
84       }
```

```
85     assert( (int)Internal.capacity() >= len );
86     return End;
87     }
88
89   void UpdatePosition() {}
90   void ShiftEnd(int len) {
91     End += len;
92   }
93   const char *GetStart()const {
94     return Start;
95   }
96
97 private:
98   typedef std::vector<char> CharVector;
99   const char *Start;
100         char *End;
101    const char *Limit;
102    CharVector Internal;
103 };
104
105 } // end namespace gdcm
106
107 #endif //GDCMBYTEBUFFER_H
```

11.115 gdcmByteSwapFilter.h File Reference

`#include "gdcmDataSet.h"`

Include dependency graph for gdcmByteSwapFilter.h:



Classes

- class gdcm::ByteSwapFilter

  ByteSwapFilter.

Namespaces

- namespace gdcm

11.116 gdcmByteSwapFilter.h

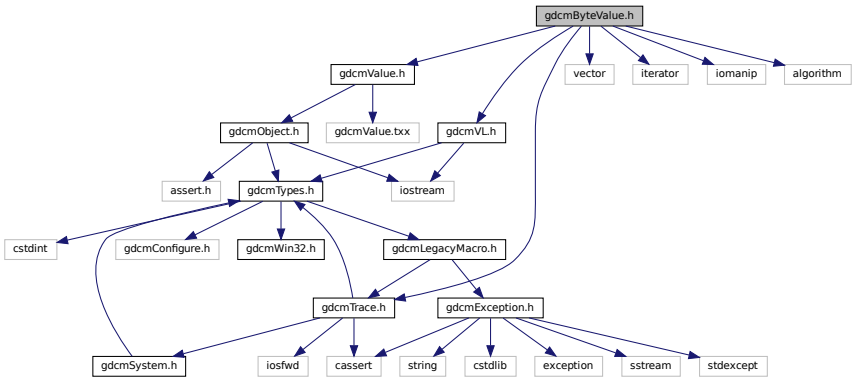
Go to the documentation of this file.

```
1 /*=========================================================================
2
3 Program:  GDCM (Grassroots DICOM). A DICOM library
4
5 Copyright (c) 2006-2011 Mathieu Malaterre
6 All rights reserved.
7 See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.
8
9 This software is distributed WITHOUT ANY WARRANTY; without even
10 the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
11 PURPOSE.  See the above copyright notice for more information.
12
13 =========================================================================*/
14 #ifndef GDCMBYTESWAPFILTER_H
15 #define GDCMBYTESWAPFILTER_H
16
17 #include "gdcmDataSet.h"
18
19 namespace gdcm
20 {
21
27 class GDCM_EXPORT ByteSwapFilter
28 {
29 public:
30   ByteSwapFilter(DataSet& ds):DS(ds),ByteSwapTag(false) {}
31   ~ByteSwapFilter();
32   ByteSwapFilter(const ByteSwapFilter &) = delete;
33   ByteSwapFilter& operator=(const ByteSwapFilter &) = delete;
34
35   bool ByteSwap();
36   void SetByteSwapTag(bool b) { ByteSwapTag = b; }
37
38 private:
39   DataSet &DS;
40   bool ByteSwapTag;
41
42 };
43
44 } // end namespace gdcm
45
46 #endif //GDCMBYTESWAPFILTER_H
```

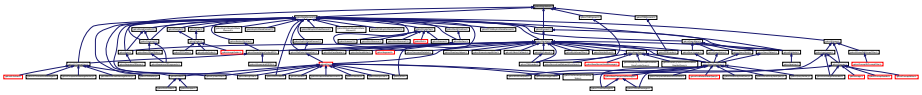
11.117 gdcmByteValue.h File Reference

```
#include "gdcmValue.h"
#include "gdcmTrace.h"
#include "gdcmVL.h"
#include <vector>
#include <iterator>
#include <iomanip>
#include <algorithm>
```

Include dependency graph for gdcmByteValue.h:



This graph shows which files directly or indirectly include this file:



Classes

- class gdcm::ByteValue

  Class to represent binary value (array of bytes)

Namespaces

- namespace gdcm

11.118 gdcmByteValue.h

Go to the documentation of this file.

```
1 /*=========================================================================
2
3 Program:  GDCM (Grassroots DICOM). A DICOM library
4
5 Copyright (c) 2006-2011 Mathieu Malaterre
6 All rights reserved.
7 See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.
8
9 This software is distributed WITHOUT ANY WARRANTY; without even
10 the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
11 PURPOSE.  See the above copyright notice for more information.
12
13 =========================================================================*/
14 #ifndef GDCMBYTEVALUE_H
15 #define GDCMBYTEVALUE_H
```

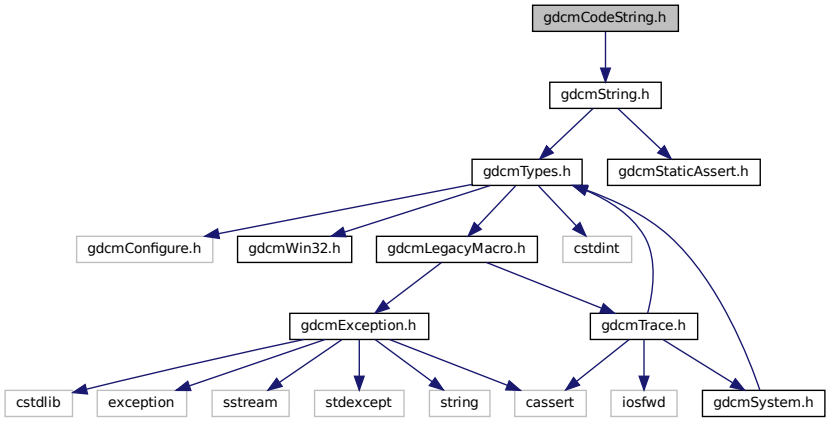
```
16
17 #include "gdcmValue.h"
18 #include "gdcmTrace.h"
19 #include "gdcmVL.h"
20
21 #include <vector>
22 #include <iterator>
23 #include <iomanip>
24 #include <algorithm>
25
26 namespace gdcm_ns
27 {
28 #if !defined(SWIGPYTHON) && !defined(SWIGCSHARP) && !defined(SWIGJAVA) && !defined(SWIGPHP)
29 using namespace gdcm;
30 #endif
35 class GDCM_EXPORT ByteValue :  public Value
36 {
37 public:
38   ByteValue(const char* array = nullptr, VL const &vl = 0):
39     Internal(array, array+vl),Length(vl) {
40       if( vl.IsOdd() )
41         {
42         gdcmDebugMacro( "Odd length" );
43         Internal.resize(vl+1);
44         ++Length;
45         }
46   }
47
49   ByteValue(std::vector<char> &v):Internal(v),Length((uint32_t)v.size()) {}
50   //ByteValue(std::ostringstream const &os) {
51   //  (void)os;
52   //  assert(0); // TODO
53   //}
54   ~ByteValue()override {
55     Internal.clear();
56   }
57
58   // When 'dumping' dicom file we still have some information from
59   // Either the VR: eg LO (private tag)
60   void PrintASCII(std::ostream &os, VL maxlength ) const;
61
62   void PrintHex(std::ostream &os, VL maxlength) const;
63
64   // Either from Element Number (== 0x0000)
65   void PrintGroupLength(std::ostream &os) {
66     assert( Length == 2 );
67     (void)os;
68   }
69
70   bool IsEmpty()const {
71 #if 0
72     if( Internal.empty() ) assert( Length == 0 );
73     return Internal.empty();
74 #else
75   return Length == 0;
76 #endif
77   }
78   VL GetLength()const override { return Length; }
79
80   VL ComputeLength()const { return Length + Length % 2; }
81   // Does a reallocation
82   void SetLength(VL vl) override;
83
84   operator const std::vector<char>& () const { return Internal; }
85
86   ByteValue &operator=(const ByteValue &val) {
87     Internal = val.Internal;
88     Length = val.Length;
89     return *this;
90     }
91
92   bool operator==(const ByteValue &val)const {
93     if( Length != val.Length )
94       return false;
95     if( Internal == val.Internal )
96       return true;
97     return false;
98     }
99   bool operator==(const Value &val)const override
100 {
101     const ByteValue &bv = dynamic_cast<const ByteValue&>(val);
```

```
102     return Length == bv.Length && Internal == bv.Internal;
103     }
104
105   void Append(ByteValue const & bv);
106
107   void Clear()override {
108     Internal.clear();
109   }
110   // Use that only if you understand what you are doing
111   const char *GetPointer()const {
112     if(!Internal.empty()) return &Internal[0];
113     return nullptr;
114   }
115   // Use that only if you really understand what you are doing
116   const void *GetVoidPointer()const {
117     if(!Internal.empty()) return &Internal[0];
118     return nullptr;
119   }
120   void *GetVoidPointer() {
121     if(!Internal.empty()) return &Internal[0];
122     return nullptr;
123   }
124   void Fill(char c) {
125     //if( Internal.empty() ) return;
126     std::vector<char>::iterator it = Internal.begin();
127     for(; it != Internal.end(); ++it) *it = c;
128   }
129   bool GetBuffer(char *buffer, unsigned long length) const;
130   bool WriteBuffer(std::ostream &os)const {
131     if( Length ) {
132       //assert( Internal.size() <= Length );
133       assert( !(Internal.size() % 2) );
134       os.write(&Internal[0], Internal.size() );
135       }
136     return true;
137   }
138
139   template <typename TSwap, typename TType>
140   std::istream &Read(std::istream &is, bool readvalues = true) {
141     // If Length is odd we have detected that in SetLength
142     // and calling std::vector::resize make sure to allocate *AND*
143     // initialize values to 0 so we are sure to have a \0 at the end
144     // even in this case
145     if(Length)
146       {
147       if( readvalues )
148         {
149         is.read(&Internal[0], Length);
150         assert( Internal.size() == Length || Internal.size() == Length + 1 );
151         TSwap::SwapArray((TType*)GetVoidPointer(), Internal.size() / sizeof(TType) );
152         }
153       else
154         {
155         is.seekg(Length, std::ios::cur);
156         }
157       }
158     return is;
159   }
160
161   template <typename TSwap>
162   std::istream &Read(std::istream &is) {
163     return Read<TSwap,uint8_t>(is);
164   }
165
166
167   template <typename TSwap, typename TType>
168   std::ostream const &Write(std::ostream &os)const {
169     assert( !(Internal.size() % 2) );
170     if( !Internal.empty() ) {
171       //os.write(&Internal[0], Internal.size());
172       std::vector<char> copy = Internal;
173       TSwap::SwapArray((TType*)(void*)&copy[0], Internal.size() / sizeof(TType) );
174       os.write(&copy[0], copy.size());
175       }
176     return os;
177   }
178
179   template <typename TSwap>
180   std::ostream const &Write(std::ostream &os)const {
181     return Write<TSwap,uint8_t>(os);
182   }
```

```
183
190   bool IsPrintable(VL length)const {
191     assert( length <= Length );
192     for(unsigned int i=0; i<length; i++)
193       {
194       if ( i == (length-1) && Internal[i] == '\0') continue;
195       if ( !( isprint((unsigned char)Internal[i]) || isspace((unsigned char)Internal[i]) ) )
196         {
197         //gdcmWarningMacro( "Cannot print :" « i );
198         return false;
199         }
200       }
201     return true;
202     }
203
205   void PrintPNXML(std::ostream &os) const;
206   void PrintASCIIXML(std::ostream &os) const;
207   void PrintHexXML(std::ostream &os) const;
208 protected:
209   void Print(std::ostream &os)const override {
210   // This is perfectly valid to have a Length = 0 , so we cannot check
211   // the length for printing
212   if( !Internal.empty() )
213     {
214     if( IsPrintable(Length) )
215       {
216       // WARNING: Internal.end() != Internal.begin()+Length
217       std::vector<char>::size_type length = Length;
218       if( Internal.back() == 0 ) --length;
219       std::copy(Internal.begin(), Internal.begin()+length,
220         std::ostream_iterator<char>(os));
221       }
222     else
223       os « "Loaded:" « Internal.size();
224     }
225   else
226     {
227     //os « "Not Loaded";
228     os « "(no value available)";
229     }
230   }
231 /*
232 //Introduce check for invalid XML characters
233 friend std::ostream& operator«(std::ostream &os,const char c);
234 */
235
236   void SetLengthOnly(VL vl)override {
237     Length = vl;
238   }
239
240 private:
241   std::vector<char> Internal;
242
243   // WARNING Length IS NOT Internal.size() some *featured* DICOM
244   // implementation define odd length, we always load them as even number
245   // of byte, so we need to keep the right Length
246   VL Length;
247 };
248
249 } // end namespace gdcm_ns
250
251 #endif //GDCMBYTEVALUE_H
```

11.119 gdcmCodeString.h File Reference

`#include "gdcmString.h"`
Include dependency graph for gdcmCodeString.h:



Classes

- class gdcm::CodeString
  CodeString.

Namespaces

- namespace gdcm

Functions

- bool gdcm::operator!= (const CodeString &ref, const CodeString &cs)
- std::ostream & gdcm::operator<< (std::ostream &os, const CodeString &str)
- bool gdcm::operator== (const CodeString &ref, const CodeString &cs)

11.120 gdcmCodeString.h

Go to the documentation of this file.

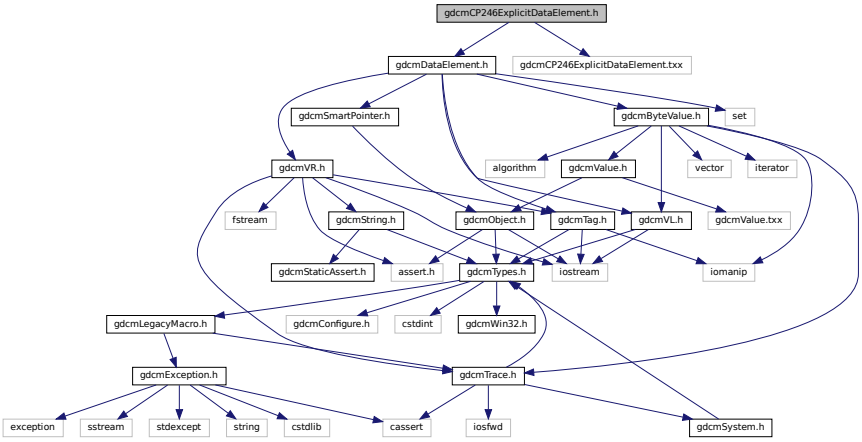
```
1 /*=========================================================================
2
3 Program:  GDCM (Grassroots DICOM). A DICOM library
4
5 Copyright (c) 2006-2011 Mathieu Malaterre
6 All rights reserved.
7 See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.
```

```
8
9 This software is distributed WITHOUT ANY WARRANTY; without even
10 the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
11 PURPOSE.  See the above copyright notice for more information.
12
13 =========================================================================*/
14 #ifndef GDCMCODESTRING_H
15 #define GDCMCODESTRING_H
16
17 #include "gdcmString.h"
18
19 namespace gdcm
20 {
21
39 // Note to myself:  because note all wrapped language support exception
40 // we could not support throwing an exception during object construction.
41 class GDCM_EXPORT CodeString
42 {
43   friend std::ostream& operator« (std::ostream& os, const CodeString& str);
44   friend bool operator==(const CodeString &ref, const CodeString& cs);
45   friend bool operator!=(const CodeString &ref, const CodeString& cs);
46   typedef String<'\\',16> InternalClass;
47 public:
48   typedef InternalClass::value_type             value_type;
49   typedef InternalClass::pointer                pointer;
50   typedef InternalClass::reference              reference;
51   typedef InternalClass::const_reference        const_reference;
52   typedef InternalClass::size_type              size_type;
53   typedef InternalClass::difference_type        difference_type;
54   typedef InternalClass::iterator               iterator;
55   typedef InternalClass::const_iterator         const_iterator;
56   typedef InternalClass::reverse_iterator       reverse_iterator;
57   typedef InternalClass::const_reverse_iterator const_reverse_iterator;
58
60   CodeString():  Internal() {}
61   CodeString(const value_type* s):  Internal(s) { Internal = Internal.Trim(); }
62   CodeString(const value_type* s, size_type n):  Internal(s, n) {
63     Internal = Internal.Trim(); }
64   CodeString(const InternalClass& s, size_type pos=0, size_type n=InternalClass::npos):
65     Internal(s, pos, n) { Internal = Internal.Trim(); }
66
68   bool IsValid() const;
69
71   std::string GetAsString()const {
72     return Internal;
73   }
74
76   size_type Size()const { return Internal.size(); }
77
78 protected:
79   std::string TrimInternal()const {
80     return Internal.Trim();
81   }
82
83 private:
84   String<'\\',16> Internal;
85 };
86
87 inline std::ostream& operator« (std::ostream& os, const CodeString& str)
88 {
89   os « str.Internal;
90   return os;
91 }
92
93 inline bool operator==(const CodeString &ref, const CodeString& cs)
94 {
95   return ref.Internal == cs.Internal;
96 }
97 inline bool operator!=(const CodeString &ref, const CodeString& cs)
98 {
99   return ref.Internal != cs.Internal;
100 }
101
102
103 } // end namespace gdcm
104
105 #endif //GDCMCODESTRING_H
```

11.121 gdcmCP246ExplicitDataElement.h File Reference

```
#include "gdcmDataElement.h"
#include "gdcmCP246ExplicitDataElement.txx"
```

Include dependency graph for gdcmCP246ExplicitDataElement.h:



Classes

- class gdcm::CP246ExplicitDataElement

  Class to read/write a DataElement as CP246Explicit Data Element.

Namespaces

- namespace gdcm

11.122 gdcmCP246ExplicitDataElement.h

Go to the documentation of this file.

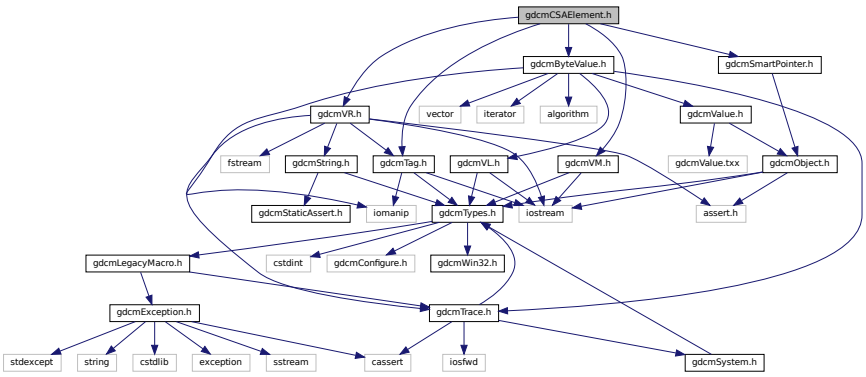
```
1 /*=========================================================================
2
3 Program:  GDCM (Grassroots DICOM). A DICOM library
4
5 Copyright (c) 2006-2011 Mathieu Malaterre
6 All rights reserved.
7 See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.
8
9 This software is distributed WITHOUT ANY WARRANTY; without even
10 the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
11 PURPOSE.  See the above copyright notice for more information.
12
13 =========================================================================*/
14 #ifndef GDCMCP246EXPLICITDATAELEMENT_H
15 #define GDCMCP246EXPLICITDATAELEMENT_H
16
17 #include "gdcmDataElement.h"
18
```

```
19 namespace gdcm
20 {
21 // Data Element (CP246Explicit)
28 class GDCM_EXPORT CP246ExplicitDataElement :  public DataElement
29 {
30 public:
31   VL GetLength() const;
32
33   template <typename TSwap>
34   std::istream &Read(std::istream &is);
35
36   template <typename TSwap>
37   std::istream &ReadPreValue(std::istream &is);
38
39   template <typename TSwap>
40   std::istream &ReadValue(std::istream &is, bool readvalues = true);
41
42   template <typename TSwap>
43   std::istream &ReadWithLength(std::istream &is, VL & length);
44
45   // PURPOSELY do not provide an implementation for writing !
46   //template <typename TSwap>
47   //const std::ostream &Write(std::ostream &os) const;
48 };
49
50 } // end namespace gdcm
51
52 #include "gdcmCP246ExplicitDataElement.txx"
53
54 #endif //GDCMCP246EXPLICITDATAELEMENT_H
```

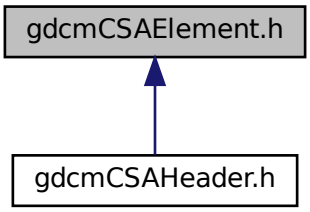
11.123 gdcmCSAElement.h File Reference

```
#include "gdcmTag.h"
#include "gdcmVM.h"
#include "gdcmVR.h"
#include "gdcmByteValue.h"
#include "gdcmSmartPointer.h"
```

Include dependency graph for gdcmCSAElement.h:



This graph shows which files directly or indirectly include this file:



Classes

- class gdcm::CSAElement

  Class to represent a CSA Element.

Namespaces

- namespace gdcm

Functions

- std::ostream & gdcm::operator<< (std::ostream &os, const CSAElement &val)

11.124 gdcmCSAElement.h

Go to the documentation of this file.

```
1 /*=========================================================================
2
3 Program:  GDCM (Grassroots DICOM). A DICOM library
4
5 Copyright (c) 2006-2011 Mathieu Malaterre
6 All rights reserved.
7 See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.
8
9 This software is distributed WITHOUT ANY WARRANTY; without even
10 the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
11 PURPOSE.  See the above copyright notice for more information.
12
13 =========================================================================*/
14 #ifndef GDCMCSAELEMENT_H
15 #define GDCMCSAELEMENT_H
16
17 #include "gdcmTag.h"
18 #include "gdcmVM.h"
19 #include "gdcmVR.h"
20 #include "gdcmByteValue.h"
21 #include "gdcmSmartPointer.h"
```

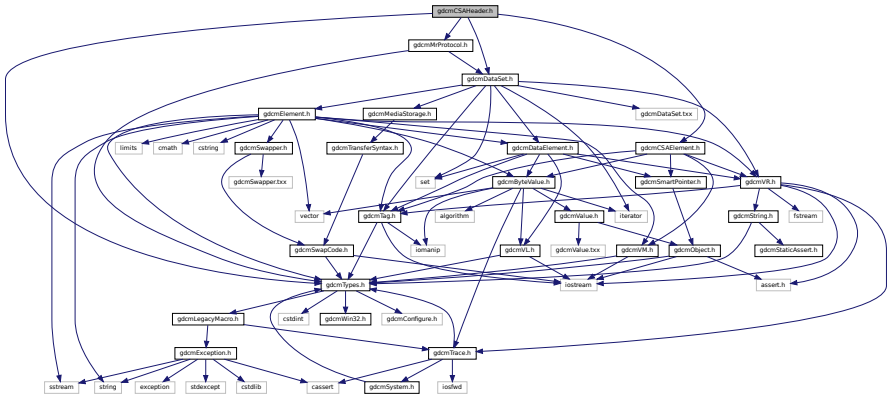
```
22
23 namespace gdcm
24 {
29 class GDCM_EXPORT CSAElement
30 {
31 public:
32   CSAElement(unsigned int kf = 0):KeyField(kf) {}
33
34   friend std::ostream& operator<<(std::ostream &os, const CSAElement &val);
35
37   unsigned int GetKey()const { return KeyField; }
38   void SetKey(unsigned int key) { KeyField = key; }
39
41   const char *GetName()const { return NameField.c_str(); }
42   void SetName(const char *name) { NameField = name; }
43
45   const VM& GetVM()const { return ValueMultiplicityField; }
46   void SetVM(const VM &vm) { ValueMultiplicityField = vm; }
47
49   VR const &GetVR()const { return VRField; }
50   void SetVR(VR const &vr) { VRField = vr; }
51
53   unsigned int GetSyngoDT()const { return SyngoDTField; }
54   void SetSyngoDT(unsigned int syngodt) { SyngoDTField = syngodt; }
55
57   unsigned int GetNoOfItems()const { return NoOfItemsField; }
58   void SetNoOfItems(unsigned int items) { NoOfItemsField = items; }
59
61   Value const &GetValue()const { return *DataField; }
62   Value &GetValue() { return *DataField; }
63   void SetValue(Value const & vl) {
64     //assert( DataField == 0 );
65     DataField = vl;
66   }
68   bool IsEmpty()const { return DataField == nullptr; }
69
71   void SetByteValue(const char *array, VL length)
72     {
73     ByteValue *bv = new ByteValue(array,length);
74     SetValue( *bv );
75     }
78   const ByteValue* GetByteValue()const {
79     // Get the raw pointer from the gdcm::SmartPointer
80     const ByteValue *bv = dynamic_cast<const ByteValue*>(DataField.GetPointer());
81     return bv; // Will return NULL if not ByteValue
82   }
83
84   CSAElement(const CSAElement &_val)
85     {
86     if( this != &_val)
87       {
88       *this = _val;
89       }
90     }
91
92   bool operator<(const CSAElement &de)const
93 {
94     return GetKey() < de.GetKey();
95     }
96   CSAElement &operator=(const CSAElement &de)
97     = default;
98
99   bool operator==(const CSAElement &de)const
100 {
101     return KeyField == de.KeyField
102       && NameField == de.NameField
103       && ValueMultiplicityField == de.ValueMultiplicityField
104       && VRField == de.VRField
105       && SyngoDTField == de.SyngoDTField
106       //&& ValueField == de.ValueField;
107       ;
108     }
109
110 protected:
111   unsigned int KeyField;
112   std::string NameField;
113   VM ValueMultiplicityField;
114   VR VRField;
115   unsigned int SyngoDTField;
116   unsigned int NoOfItemsField;
117   typedef SmartPointer<Value> DataPtr;
```

```
118   DataPtr DataField;
119 };
120 //-----------------------------------------------------------------------------
121 inline std::ostream& operator«(std::ostream &os, const CSAElement &val)
122 {
123   os « val.KeyField;
124   os « " - '" « val.NameField;
125   os « "' VM " « val.ValueMultiplicityField;
126   os « ", VR " « val.VRField;
127   os « ", SyngoDT " « val.SyngoDTField;
128   os « ", NoOfItems " « val.NoOfItemsField;
129   os « ", Data ";
130   if( val.DataField )
131     {
132     //val.DataField->Print( os « "'" );
133     const ByteValue * bv = dynamic_cast<ByteValue*>(&*val.DataField);
134     assert( bv );
135     const char * p = bv->GetPointer();
136     std::string str(p, p + bv->GetLength() );
137     if( val.ValueMultiplicityField == VM::VM1 )
138       {
139       os « "'" « str.c_str() « "'";
140       }
141     else
142       {
143       std::istringstream is( str );
144       std::string s;
145       bool sep = false;
146       while( std::getline(is, s, '\\' ) )
147         {
148         if( sep )
149           {
150           os « '\\';
151           }
152         sep = true;
153         os « "'" « s.c_str() « "'";
154         }
155       //bv->Print( os « "'" );
156       //os « "'";
157       }
158     }
159   return os;
160 }
161
162 } // end namespace gdcm
163
164 #endif //GDCMCSAELEMENT_H
```

11.125 gdcmCSAHeader.h File Reference

```
#include "gdcmTypes.h"
#include "gdcmDataSet.h"
#include "gdcmCSAElement.h"
#include "gdcmMrProtocol.h"
```

Include dependency graph for gdcmCSAHeader.h:



Classes

- class gdcm::CSAHeader

  Class for CSAHeader.

Namespaces

- namespace gdcm

Functions

- std::ostream & gdcm::operator<< (std::ostream &os, const CSAHeader &d)

11.126 gdcmCSAHeader.h

Go to the documentation of this file.

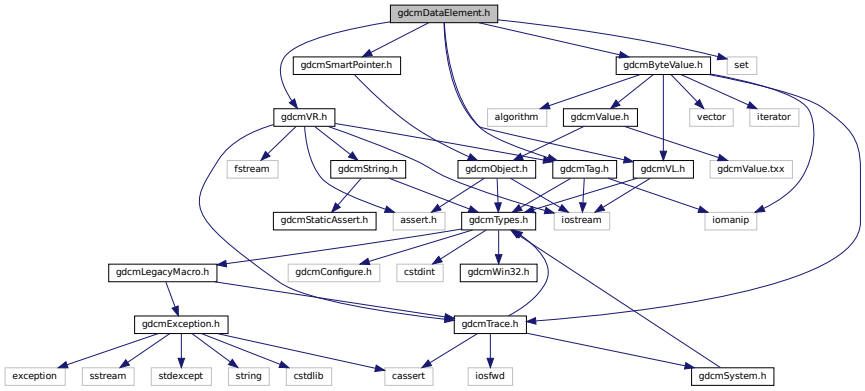
```
1 /*=========================================================================
2
3 Program:  GDCM (Grassroots DICOM). A DICOM library
4
5 Copyright (c) 2006-2011 Mathieu Malaterre
6 All rights reserved.
7 See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.
8
9 This software is distributed WITHOUT ANY WARRANTY; without even
10 the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
11 PURPOSE.  See the above copyright notice for more information.
12
13 =========================================================================*/
14 #ifndef GDCMCSAHEADER_H
15 #define GDCMCSAHEADER_H
16
17 #include "gdcmTypes.h"
18 #include "gdcmDataSet.h"
19 #include "gdcmCSAElement.h"
20 #include "gdcmMrProtocol.h"
```

```
21
22 namespace gdcm
23 {
24 /*
25 * Everything done in this code is for the sole purpose of writing interoperable
26 * software under Sect.  1201 (f) Reverse Engineering exception of the DMCA.
27 * If you believe anything in this code violates any law or any of your rights,
28 * please contact us (gdcm-developers@lists.sourceforge.net) so that we can
29 * find a solution.
30 */
31 //-----------------------------------------------------------------------------
32
33 class DataElement;
34 class PrivateTag;
63 class GDCM_EXPORT CSAHeader
64 {
65   friend std::ostream& operator<<(std::ostream &_os, const CSAHeader &d);
66 public :
67   CSAHeader():InternalDataSet(),InternalType(UNKNOWN),InterfileData(nullptr) {};
68   ~CSAHeader() = default;
69
71   typedef enum {
72     UNKNOWN = 0,
73     SV10,
74     NOMAGIC,
75     DATASET_FORMAT,
76     INTERFILE,
77     ZEROED_OUT
78   } CSAHeaderType;
79
81   bool LoadFromDataElement(DataElement const &de);
82
84   void Print(std::ostream &os) const;
85
87   const DataSet& GetDataSet()const { return InternalDataSet; }
88
90   const char * GetInterfile()const { return InterfileData; }
91
94   CSAHeaderType GetFormat() const;
95
98   static const PrivateTag & GetCSAImageHeaderInfoTag();
99
102   static const PrivateTag & GetCSASeriesHeaderInfoTag();
103
106   static const PrivateTag & GetCSADataInfo();
107
110   const CSAElement &GetCSAElementByName(const char *name);
111
114   bool FindCSAElementByName(const char *name);
115
117   bool GetMrProtocol( const DataSet & ds, MrProtocol & mrProtocol );
118
119 protected:
120   const CSAElement& GetCSAEEnd() const;
121
122 private:
123   std::set<CSAElement> InternalCSADataSet;
124   DataSet InternalDataSet;
125   CSAHeaderType InternalType;
126   Tag DataElementTag;
127   static CSAElement CSAEEnd;
128   const char *InterfileData;
129 };
130 //-----------------------------------------------------------------------------
131 inline std::ostream& operator<<(std::ostream &os, const CSAHeader &d)
132 {
133   d.Print( os );
134   return os;
135 }
136
137 } // end namespace gdcm
138 //-----------------------------------------------------------------------------
139 #endif //GDCMCSAHEADER_H
```

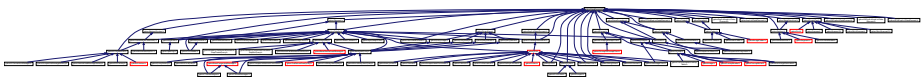
11.127 gdcmDataElement.h File Reference

```
#include "gdcmTag.h"
#include "gdcmVL.h"
#include "gdcmVR.h"
#include "gdcmByteValue.h"
#include "gdcmSmartPointer.h"
#include <set>
```

Include dependency graph for gdcmDataElement.h:



This graph shows which files directly or indirectly include this file:



Classes

- class gdcm::DataElement

  Class to represent a Data Element either Implicit or Explicit.

Namespaces

- namespace gdcm

Functions

- bool gdcm::operator!= (const DataElement &lhs, const DataElement &rhs)
- std::ostream & gdcm::operator<< (std::ostream &os, const DataElement &val)

11.128 gdcmDataElement.h

Go to the documentation of this file.

```
1 /*=========================================================================
2
3 Program:  GDCM (Grassroots DICOM). A DICOM library
4
5 Copyright (c) 2006-2011 Mathieu Malaterre
6 All rights reserved.
7 See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.
8
9 This software is distributed WITHOUT ANY WARRANTY; without even
10 the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
11 PURPOSE.  See the above copyright notice for more information.
12
13 =========================================================================*/
14 #ifndef GDCMDATAELEMENT_H
15 #define GDCMDATAELEMENT_H
16
17 #include "gdcmTag.h"
18 #include "gdcmVL.h"
19 #include "gdcmVR.h"
20 #include "gdcmByteValue.h"
21 #include "gdcmSmartPointer.h"
22
23 #include <set>
24
25 namespace gdcm_ns
26 {
27 // Data Element
28 // Contains multiple fields:
29 // -> Tag
30 // -> Optional VR (Explicit Transfer Syntax)
31 // -> ValueLength
32 // -> Value
33 // TODO: This class SHOULD be pure virtual.  I don't want a user
34 // to shoot himself in the foot.
35
36 class SequenceOfItems;
37 class SequenceOfFragments;
58 class GDCM_EXPORT DataElement
59 {
60 public:
61   DataElement(const Tag& t = Tag(0), const VL& vl = 0, const VR &vr =
       VR::INVALID):TagField(t),ValueLengthField(vl),VRField(vr),ValueField(nullptr) {}
62   //DataElement( Attribute const &att );
63
64   friend std::ostream& operator<<(std::ostream &_os, const DataElement &_val);
65
67   const Tag& GetTag()const { return TagField; }
68   Tag& GetTag() { return TagField; }
71   void SetTag(const Tag &t) { TagField = t; }
72
74   const VL& GetVL()const { return ValueLengthField; }
75   VL& GetVL() { return ValueLengthField; }
79   void SetVL(const VL &vl) { ValueLengthField = vl; }
80   void SetVLToUndefined();
81
84   VR const &GetVR()const { return VRField; }
88   void SetVR(VR const &vr) {
89     if( vr.IsVRFile() )
90       VRField = vr;
91   }
92
94   Value const &GetValue()const { gdcmAssertAlwaysMacro(ValueField); return *ValueField; }
95   Value &GetValue() {
96     gdcmAssertAlwaysMacro(ValueField);
97     return *ValueField;
98   }
100   void SetValue(Value const & vl) {
101     //assert( ValueField == 0 );
102     ValueField = vl;
103     ValueLengthField = vl.GetLength();
104   }
106   bool IsEmpty()const { return ValueField == nullptr || (GetByteValue() && GetByteValue()->IsEmpty()); }
107
109   void Empty() { ValueField = nullptr; ValueLengthField = 0; }
110
112   void Clear()
```

```
113     {
114     TagField = 0;
115     VRField = VR::INVALID;
116     ValueField = nullptr;
117     ValueLengthField = 0;
118     }
119
120   // Helper:
126   void SetByteValue(const char *array, VL length)
127     {
128     ByteValue *bv = new ByteValue(array,length);
129     SetValue( *bv );
130     }
133   const ByteValue* GetByteValue()const {
134     // Get the raw pointer from the gdcm::SmartPointer
135     const ByteValue *bv = dynamic_cast<const ByteValue*>(ValueField.GetPointer());
136     return bv; // Will return NULL if not ByteValue
137   }
138
145   SmartPointer<SequenceOfItems> GetValueAsSQ() const;
146
149   const SequenceOfFragments* GetSequenceOfFragments() const;
150   SequenceOfFragments* GetSequenceOfFragments();
151
153   bool IsUndefinedLength()const {
154     return ValueLengthField.IsUndefined();
155   }
156
157   DataElement(const DataElement &_val)
158     {
159     if( this != &_val)
160       {
161       *this = _val;
162       }
163     }
164
165   bool operator<(const DataElement &de)const
166 {
167     return GetTag() < de.GetTag();
168     }
169   DataElement &operator=(const DataElement &)
170     = default;
171
172   bool operator==(const DataElement &de)const
173 {
174     bool b = TagField == de.TagField
175       && ValueLengthField == de.ValueLengthField
176       && VRField == de.VRField;
177     if( !ValueField && !de.ValueField )
178       {
179       return b;
180       }
181     if( ValueField && de.ValueField )
182       {
183       return b && (*ValueField == *de.ValueField);
184       }
185     // ValueField != de.ValueField
186     return false;
187     }
188
189   // The following functionalities are dependent on:
190   // # The Transfer Syntax:  Explicit or Implicit
191   // # The Byte encoding:  Little Endian / Big Endian
192
193   /*
194 * The following was inspired by a C++ idiom:  Curiously Recurring Template Pattern
195 * Ref:  http://en.wikipedia.org/wiki/Curiously_Recurring_Template_Pattern
196 * The typename TDE is typically a derived class *without* any data
197 * while TSwap is a simple template parameter to achieve byteswapping (and allow factorization of
198 * highly identical code)
199 */
200   template <typename TDE>
201   VL GetLength()const {
202     return static_cast<const TDE*>(this)->GetLength();
203   }
204
205   template <typename TDE, typename TSwap>
206   std::istream &Read(std::istream &is) {
207     return static_cast<TDE*>(this)->template Read<TSwap>(is);
208   }
209
```

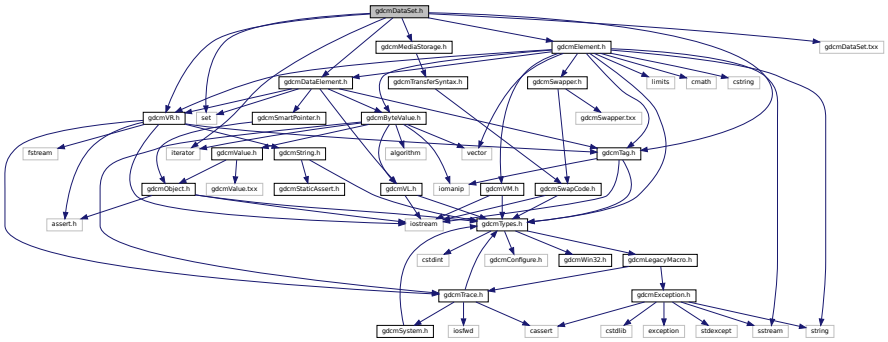
```
210   template <typename TDE, typename TSwap>
211   std::istream &ReadOrSkip(std::istream &is, std::set<Tag> const &skiptags) {
212     (void)skiptags;
213     return static_cast<TDE*>(this)->template Read<TSwap>(is);
214   }
215
216   template <typename TDE, typename TSwap>
217   std::istream &ReadPreValue(std::istream &is, std::set<Tag> const &skiptags) {
218     (void)skiptags;
219     return static_cast<TDE*>(this)->template ReadPreValue<TSwap>(is);
220   }
221   template <typename TDE, typename TSwap>
222   std::istream &ReadValue(std::istream &is, std::set<Tag> const &skiptags) {
223     (void)skiptags;
224     return static_cast<TDE*>(this)->template ReadValue<TSwap>(is);
225   }
226   template <typename TDE, typename TSwap>
227   std::istream &ReadValueWithLength(std::istream &is, VL & length, std::set<Tag> const &skiptags) {
228     (void)skiptags;
229     return static_cast<TDE*>(this)->template ReadValueWithLength<TSwap>(is, length);
230   }
231
232   template <typename TDE, typename TSwap>
233   std::istream &ReadWithLength(std::istream &is, VL &length) {
234     return static_cast<TDE*>(this)->template ReadWithLength<TSwap>(is,length);
235   }
236
237   template <typename TDE, typename TSwap>
238   const std::ostream &Write(std::ostream &os)const {
239     return static_cast<const TDE*>(this)->template Write<TSwap>(os);
240   }
241
242 protected:
243   Tag TagField;
244   // This is the value read from the file, might be different from the length of Value Field
245   VL ValueLengthField; // Can be 0xFFFFFFFF
246
247   // Value Representation
248   VR VRField;
249   typedef SmartPointer<Value> ValuePtr;
250   ValuePtr ValueField;
251
252   void SetValueFieldLength( VL vl, bool readvalues );
253 };
254 //-----------------------------------------------------------------------------
255 inline std::ostream& operator«(std::ostream &os, const DataElement &val)
256 {
257   os « val.TagField;
258   os « "\t" « val.VRField;
259   os « "\t" « val.ValueLengthField;
260   if( val.ValueField )
261     {
262     val.ValueField->Print( os « "\t" );
263     }
264   return os;
265 }
266
267 inline bool operator!=(const DataElement& lhs, const DataElement& rhs)
268 {
269   return !  ( lhs == rhs );
270 }
271
272 } // end namespace gdcm_ns
273
274 #endif //GDCMDATAELEMENT_H
```

11.129 gdcmDataSet.h File Reference

```
#include "gdcmDataElement.h"
#include "gdcmTag.h"
#include "gdcmVR.h"
#include "gdcmElement.h"
#include "gdcmMediaStorage.h"
```

```
#include <set>
#include <iterator>
#include "gdcmDataSet.txx"
```

Include dependency graph for gdcmDataSet.h:



This graph shows which files directly or indirectly include this file:

Classes

- class gdcm::DataElementException
- class gdcm::DataSet

  Class to represent a Data Set (which contains Data Elements)

Namespaces

- namespace gdcm

Functions

- std::ostream & gdcm::operator<< (std::ostream &os, const DataSet &val)

11.130 gdcmDataSet.h

Go to the documentation of this file.

```
1 /*=========================================================================
2
3 Program:  GDCM (Grassroots DICOM). A DICOM library
4
5 Copyright (c) 2006-2011 Mathieu Malaterre
6 All rights reserved.
7 See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.
8
9 This software is distributed WITHOUT ANY WARRANTY; without even
10 the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
11 PURPOSE.  See the above copyright notice for more information.
12
13 =========================================================================*/
14 #ifndef GDCMDATASET_H
15 #define GDCMDATASET_H
16
17 #include "gdcmDataElement.h"
18 #include "gdcmTag.h"
19 #include "gdcmVR.h"
20 #include "gdcmElement.h"
21 #include "gdcmMediaStorage.h"
22
23 #include <set>
24 #include <iterator>
25
26 namespace gdcm_ns
27 {
28 class GDCM_EXPORT DataElementException :  public std::exception {};
29
30 class PrivateTag;
55 class GDCM_EXPORT DataSet
56 {
57   friend class CSAHeader;
58 public:
59   typedef std::set<DataElement> DataElementSet;
60   typedef DataElementSet::const_iterator ConstIterator;
61   typedef DataElementSet::iterator Iterator;
62   typedef DataElementSet::size_type SizeType;
63   //typedef typename DataElementSet::iterator iterator;
64   ConstIterator Begin()const { return DES.begin(); }
65   Iterator Begin() { return DES.begin(); }
66   ConstIterator End()const { return DES.end(); }
67   Iterator End() { return DES.end(); }
68   const DataElementSet &GetDES()const { return DES; }
69   DataElementSet &GetDES() { return DES; }
70   void Clear() {
71     DES.clear();
72     assert( DES.empty() );
73   }
74
75   SizeType Size()const {
76     return DES.size();
77   }
78
79   void Print(std::ostream &os, std::string const &indent = "")const {
80     // CT_Phillips_JPEG2K_Decompr_Problem.dcm has a SQ of length == 0
81     //int s = DES.size();
82     //assert( s );
83     //std::copy(DES.begin(), DES.end(),
84     //  std::ostream_iterator<DataElement>(os, "\n"));
85     ConstIterator it = DES.begin();
86     for( ; it != DES.end(); ++it)
87       {
88       os « indent « *it « "\n";
89       }
90   }
91
92   template <typename TDE>
93   unsigned int ComputeGroupLength(Tag const &tag)const
94 {
95     assert( tag.GetElement() == 0x0 );
96     const DataElement r(tag);
97     ConstIterator it = DES.find(r);
98     unsigned int res = 0;
99     for( ++it; it != DES.end()
100       && it->GetTag().GetGroup() == tag.GetGroup(); ++it)
```

```
101       {
102       assert( it->GetTag().GetElement() != 0x0 );
103       assert( it->GetTag().GetGroup() == tag.GetGroup() );
104       res += it->GetLength<TDE>();
105       }
106     return res;
107     }
108
109   template <typename TDE>
110   VL GetLength()const {
111     if( DES.empty() ) return 0;
112     assert( !DES.empty() );
113     VL ll = 0;
114     assert( ll == 0 );
115     ConstIterator it = DES.begin();
116     for( ; it != DES.end(); ++it)
117       {
118       assert( !(it->GetLength<TDE>().IsUndefined()) );
119       if ( it->GetTag() != Tag(0xfffe,0xe00d) )
120         {
121         ll += it->GetLength<TDE>();
122         }
123       }
124     return ll;
125   }
128   void Insert(const DataElement& de) {
129     // FIXME: there is a special case where a dataset can have value < 0x8, see:
130     // $ gdcmdump --csa gdcmData/SIEMENS-JPEG-CorruptFrag.dcm
131     if( de.GetTag().GetGroup() >= 0x0008 || de.GetTag().GetGroup() == 0x4 )
132       {
133       // prevent user error:
134       if( de.GetTag() == Tag(0xfffe,0xe00d)
135       || de.GetTag() == Tag(0xfffe,0xe0dd)
136       || de.GetTag() == Tag(0xfffe,0xe000) )
137         {
138         }
139       else
140         {
141         InsertDataElement( de );
142         }
143       }
144     else
145       {
146       gdcmErrorMacro( "Cannot add element with group < 0x0008 and != 0x4 in the dataset:  " « de.GetTag() );
147       }
148   }
150   void Replace(const DataElement& de) {
151     ConstIterator it = DES.find(de);
152   if( it != DES.end() )
153   {
154     // detect loop:
155     gdcmAssertAlwaysMacro( &*it != &de );
156     DES.erase(it);
157   }
158     DES.insert(de);
159   }
161   void ReplaceEmpty(const DataElement& de) {
162     ConstIterator it = DES.find(de);
163     if( it != DES.end() && it->IsEmpty() )
164     {
165       // detect loop:
166     gdcmAssertAlwaysMacro( &*it != &de );
167     DES.erase(it);
168     }
169   DES.insert(de);
170   }
172   SizeType Remove(const Tag& tag) {
173     DataElementSet::size_type count = DES.erase(tag);
174     assert( count == 0 || count == 1 );
175     return count;
176   }
177
181   //DataElement& GetDataElement(const Tag &t) {
182   //  DataElement r(t);
183   //  Iterator it = DES.find(r);
184   //  if( it != DES.end() )
185   //    return *it;
186   //  return GetDEEnd();
187   //  }
188   const DataElement& GetDataElement(const Tag &t)const {
189     const DataElement r(t);
```

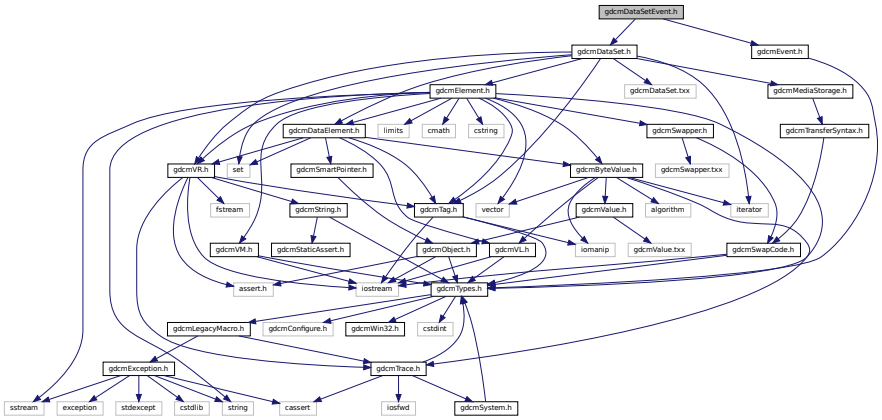
```
190     ConstIterator it = DES.find(r);
191     if( it != DES.end() )
192       return *it;
193     return GetDEEnd();
194     }
195   const DataElement& operator[] (const Tag &t)const { return GetDataElement(t); }
196   const DataElement& operator() (uint16_t group, uint16_t element)const { return GetDataElement(
      Tag(group,element) ); }
197
200   std::string GetPrivateCreator(const Tag &t) const;
201
203   PrivateTag GetPrivateTag(const Tag &t) const;
204
206   bool FindDataElement(const PrivateTag &t) const;
208   const DataElement& GetDataElement(const PrivateTag &t) const;
209
210   // DUMB: this only search within the level of the current DataSet
211   bool FindDataElement(const Tag &t)const {
212     const DataElement r(t);
213     //ConstIterator it = DES.find(r);
214     if( DES.find(r) != DES.end() )
215       {
216       return true;
217       }
218     return false;
219     }
220
221   // WARNING:
222   // This only search at the same level as the DataSet is !
223   const DataElement& FindNextDataElement(const Tag &t)const {
224     const DataElement r(t);
225     ConstIterator it = DES.lower_bound(r);
226     if( it != DES.end() )
227       return *it;
228     return GetDEEnd();
229     }
230
232   bool IsEmpty()const { return DES.empty(); };
233
234   DataSet& operator=(DataSet const &)
235   = default;
236
237   template <typename TDE, typename TSwap>
238   std::istream &ReadNested(std::istream &is);
239
240   template <typename TDE, typename TSwap>
241   std::istream &Read(std::istream &is);
242
243   template <typename TDE, typename TSwap>
244   std::istream &ReadUpToTag(std::istream &is, const Tag &t, std::set<Tag> const & skiptags);
245
246   template <typename TDE, typename TSwap>
247   std::istream &ReadUpToTagWithLength(std::istream &is, const Tag &t, std::set<Tag> const & skiptags, VL &
      length);
248
249   template <typename TDE, typename TSwap>
250   std::istream &ReadSelectedTags(std::istream &is, const std::set<Tag> & tags, bool readvalues = true);
251   template <typename TDE, typename TSwap>
252   std::istream &ReadSelectedTagsWithLength(std::istream &is, const std::set<Tag> & tags, VL & length, bool
      readvalues = true);
253
254   template <typename TDE, typename TSwap>
255   std::istream &ReadSelectedPrivateTags(std::istream &is, const std::set<PrivateTag> & tags, bool readvalues
      = true);
256   template <typename TDE, typename TSwap>
257   std::istream &ReadSelectedPrivateTagsWithLength(std::istream &is, const std::set<PrivateTag> & tags, VL &
      length, bool readvalues = true);
258
259   template <typename TDE, typename TSwap>
260   std::ostream const &Write(std::ostream &os) const;
261
262   template <typename TDE, typename TSwap>
263   std::istream &ReadWithLength(std::istream &is, VL &length);
264
265   MediaStorage GetMediaStorage() const;
266
267 protected:
268   /* GetDEEnd is a Win32 only issue, one cannot use a dllexported
269  * static member data in an inline function, otherwise symbol
270  * will get reported as missing in any dll using the inlined function
271  */
```

```
272   const DataElement& GetDEEnd() const;
273
274   // This function is not safe, it does not check for the value of the tag
275   // so depending whether we are getting called from a dataset or file meta header
276   // the condition is different
277   void InsertDataElement(const DataElement& de) {
278     //if( de.GetTag() == Tag(0xfffe,0xe00d) ) return;
279     //if( de.GetTag() == Tag(0xfffe,0xe0dd) ) return;
280 #ifndef NDEBUG
281     std::pair<Iterator,bool> pr = DES.insert(de);
282     if( pr.second == false )
283       {
284       gdcmWarningMacro( "DataElement:  " << de << " was already found, skipping duplicate entry.\n"
285         "Original entry kept is:  " << *pr.first );
286       }
287 #else
288     DES.insert(de);
289 #endif
290     assert( de.IsEmpty() || de.GetVL() == de.GetValue().GetLength() );
291     }
292
293 protected:
294   // Internal function, that will compute the actual Tag (if found) of
295   // a requested Private Tag (XXXX,YY,"PRIVATE")
296   Tag ComputeDataElement(const PrivateTag & t) const;
297
298 private:
299   DataElementSet DES;
300   static DataElement DEEnd;
301   friend std::ostream& operator<<(std::ostream &_os, const DataSet &);
302 };
303 //-----------------------------------------------------------------------------
304 inline std::ostream& operator<<(std::ostream &os, const DataSet &val)
305 {
306   val.Print(os);
307   return os;
308 }
309
310 #if defined(SWIGPYTHON) || defined(SWIGCSHARP) || defined(SWIGJAVA) || defined(SWIGPHP)
311 /*
312 * HACK: I need this temp class to be able to manipulate a std::set from python,
313 * swig does not support wrapping of simple class like std::set...
314 */
315 class SWIGDataSet
316 {
317 public:
318   SWIGDataSet(DataSet &des):Internal(des),it(des.Begin()) {}
319   const DataElement& GetCurrent()const { return *it; }
320   void Start() { it = Internal.Begin(); }
321   bool IsAtEnd()const { return it == Internal.End(); }
322   void Next() { ++it; }
323 private:
324   DataSet & Internal;
325   DataSet::ConstIterator it;
326 };
327 #endif /* SWIG */
328
334 } // end namespace gdcm_ns
335
336 #include "gdcmDataSet.txx"
337
338 #endif //GDCMDATASET_H
```

11.131 gdcmDataSetEvent.h File Reference

```
#include "gdcmEvent.h"
#include "gdcmDataSet.h"
```

Include dependency graph for gdcmDataSetEvent.h:



Classes

- class gdcm::DataSetEvent

  DataSetEvent.

Namespaces

- namespace gdcm

11.132 gdcmDataSetEvent.h

Go to the documentation of this file.

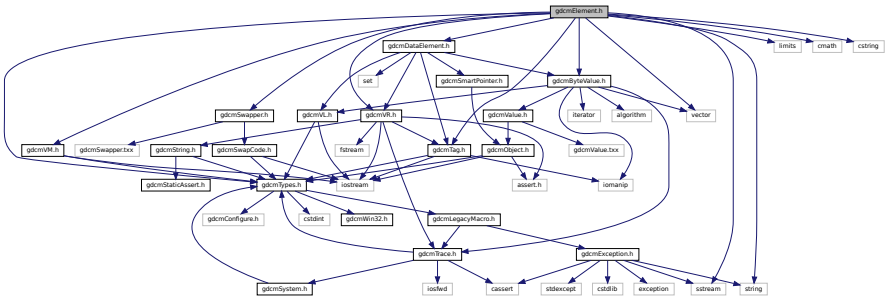
```
1 /*=========================================================================
2
3 Program:  GDCM (Grassroots DICOM). A DICOM library
4
5 Copyright (c) 2006-2011 Mathieu Malaterre
6 All rights reserved.
7 See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.
8
9 This software is distributed WITHOUT ANY WARRANTY; without even
10 the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
11 PURPOSE.  See the above copyright notice for more information.
12
13 =========================================================================*/
14 #ifndef GDCMDATASETEVENT_H
15 #define GDCMDATASETEVENT_H
16
17 #include "gdcmEvent.h"
18 #include "gdcmDataSet.h"
19
20 namespace gdcm
21 {
22
29 class DataSetEvent :  public AnyEvent
30 {
31 public:
32   typedef DataSetEvent Self;
33   typedef AnyEvent Superclass;
```

```
34   DataSetEvent(DataSet const *ds = nullptr):m_DataSet(ds) {}
35   ~DataSetEvent() override = default;
36   void operator=(const Self&) = delete;
37   const DataSet *m_DataSet;
38   const char * GetEventName()const override { return "DataSetEvent"; }
39   bool CheckEvent(const ::gdcm::Event* e)const override
40 { return (dynamic_cast<const Self*>(e) == nullptr ?  false :  true) ; }
41   ::gdcm::Event* MakeObject()const override
42 { return new Self; }
43   DataSetEvent(const Self&s) :  AnyEvent(s){};
44
45   DataSet const & GetDataSet()const { return *m_DataSet; }
46 };
47
48
49 } // end namespace gdcm
50
51 #endif //GDCMANONYMIZEEVENT_H
```

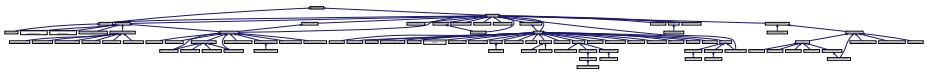
11.133 gdcmElement.h File Reference

```
#include "gdcmTypes.h"
#include "gdcmVR.h"
#include "gdcmTag.h"
#include "gdcmVM.h"
#include "gdcmByteValue.h"
#include "gdcmDataElement.h"
#include "gdcmSwapper.h"
#include <string>
#include <vector>
#include <sstream>
#include <limits>
#include <cmath>
#include <cstring>
```

Include dependency graph for gdcmElement.h:



This graph shows which files directly or indirectly include this file:



Classes

- class gdcm::Element< TVR, TVM >

  Element class.
- class gdcm::Element< TVR, VM::VM1_2 >
- class gdcm::Element< TVR, VM::VM1_n >
- class gdcm::Element< TVR, VM::VM2_2n >
- class gdcm::Element< TVR, VM::VM2_n >
- class gdcm::Element< TVR, VM::VM3_3n >
- class gdcm::Element< TVR, VM::VM3_4 >
- class gdcm::Element< TVR, VM::VM3_n >
- class gdcm::Element< VR::AS, VM::VM5 >
- class gdcm::Element< VR::OB, VM::VM1 >
- class gdcm::Element< VR::OW, VM::VM1 >
- class gdcm::ElementDisableCombinations< TVR, TVM >

  A class which is used to produce compile errors for an invalid combination of template parameters.
- class gdcm::ElementDisableCombinations< VR::OB, VM::VM1_n >
- class gdcm::ElementDisableCombinations< VR::OW, VM::VM1_n >
- class gdcm::EncodingImplementation< VR::VRASCII >
- class gdcm::EncodingImplementation< VR::VRBINARY >
- struct gdcm::ignore_char

Namespaces

- namespace gdcm

Functions

- static int gdcm::add1 (char *buf, int n)
- ignore_char const gdcm::backslash ('\\')
- static void gdcm::clean (char *mant)
- static int gdcm::doround (char *buf, unsigned int n)
- std::istream & gdcm::operator>> (std::istream &in, ignore_char const &ic)
- static int gdcm::roundat (char *buf, unsigned int i, int iexp)
- template<typename Float >
  static void gdcm::x16printf (char *buf, int size, Float f)

11.134 gdcmElement.h

Go to the documentation of this file.

```
1 /*=========================================================================
2
3 Program:  GDCM (Grassroots DICOM). A DICOM library
4
5 Copyright (c) 2006-2011 Mathieu Malaterre
6 All rights reserved.
7 See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.
8
9 This software is distributed WITHOUT ANY WARRANTY; without even
10 the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
```

```
11 PURPOSE.  See the above copyright notice for more information.
12
13 =========================================================================*/
14 #ifndef GDCMELEMENT_H
15 #define GDCMELEMENT_H
16
17 #include "gdcmTypes.h"
18 #include "gdcmVR.h"
19 #include "gdcmTag.h"
20 #include "gdcmVM.h"
21 #include "gdcmByteValue.h"
22 #include "gdcmDataElement.h"
23 #include "gdcmSwapper.h"
24
25 #include <string>
26 #include <vector>
27 #include <sstream>
28 #include <limits>
29 #include <cmath>
30 #include <cstring>
31
32 namespace gdcm_ns
33 {
34
35 // Forward declaration
41 template<long long T> class EncodingImplementation;
42
43
51 template <long long TVR, int TVM>
52 class ElementDisableCombinations {};
53 template <>
54 class  ElementDisableCombinations<VR::OB, VM::VM1_n> {};
55 template <>
56 class  ElementDisableCombinations<VR::OW, VM::VM1_n> {};
57 // Make it impossible to compile these other cases
58 template <int TVM>
59 class  ElementDisableCombinations<VR::OB, TVM>;
60 template <int TVM>
61 class ElementDisableCombinations<VR::OW, TVM>;
62
68 template<long long TVR, int TVM>
69 class Element
70 {
71   enum { ElementDisableCombinationsCheck = sizeof ( ElementDisableCombinations<TVR, TVM> ) };
72 public:
73   typename VRToType<TVR>::Type Internal[VMToLength<TVM>::Length];
74   typedef typename VRToType<TVR>::Type Type;
75
76   static VR  GetVR()  { return (VR::VRType)TVR; }
77   static VM  GetVM()  { return (VM::VMType)TVM; }
78
79   unsigned long GetLength()const {
80     return VMToLength<TVM>::Length;
81   }
82   // Implementation of Print is common to all Mode (ASCII/Binary)
83   // TODO: Can we print a \ when in ASCII...well I don't think so
84   // it would mean we used a bad VM then, right?
85   void Print(std::ostream &_os)const {
86     _os « Internal[0]; // VM is at least guarantee to be one
87     for(int i=1; i<VMToLength<TVM>::Length; ++i)
88       _os « "," « Internal[i];
89     }
90
91   const typename VRToType<TVR>::Type *GetValues()const {
92     return Internal;
93   }
94   const typename VRToType<TVR>::Type &GetValue(unsigned int idx = 0)const {
95     assert( idx < VMToLength<TVM>::Length );
96     return Internal[idx];
97   }
98   typename VRToType<TVR>::Type &GetValue(unsigned int idx = 0) {
99     assert( idx < VMToLength<TVM>::Length );
100     return Internal[idx];
101   }
102   typename VRToType<TVR>::Type operator[] (unsigned int idx)const {
103     return GetValue(idx);
104   }
105   void SetValue(typename VRToType<TVR>::Type v, unsigned int idx = 0) {
106     assert( idx < VMToLength<TVM>::Length );
107     Internal[idx] = v;
108   }
```

```
109
110   void SetFromDataElement(DataElement const &de) {
111     const ByteValue *bv = de.GetByteValue();
112     if( !bv ) return;
113 #ifdef GDCM_WORDS_BIGENDIAN
114     if( de.GetVR() == VR::UN /*|| de.GetVR() == VR::INVALID*/ )
115 #else
116     if( de.GetVR() == VR::UN || de.GetVR() == VR::INVALID )
117 #endif
118       {
119       Set(de.GetValue());
120       }
121     else
122       {
123       SetNoSwap(de.GetValue());
124       }
125   }
126
127   DataElement GetAsDataElement()const {
128     DataElement ret;
129     std::ostringstream os;
130     EncodingImplementation<VRToEncoding<TVR>::Mode>::Write(Internal,
131       GetLength(),os);
132     ret.SetVR( (VR::VRType)TVR );
133     assert( ret.GetVR() != VR::SQ );
134     if( (VR::VRType)VRToEncoding<TVR>::Mode == VR::VRASCII )
135       {
136       if( GetVR() != VR::UI )
137         {
138         if( os.str().size() % 2 )
139           {
140           os << " ";
141           }
142         }
143       }
144     VL::Type osStrSize = (VL::Type)os.str().size();
145     ret.SetByteValue( os.str().c_str(), osStrSize );
146
147     return ret;
148   }
149
150   void Read(std::istream &_is) {
151     return EncodingImplementation<VRToEncoding<TVR>::Mode>::Read(Internal,
152       GetLength(),_is);
153     }
154   void Write(std::ostream &_os)const {
155     return EncodingImplementation<VRToEncoding<TVR>::Mode>::Write(Internal,
156       GetLength(),_os);
157     }
158
159   // FIXME: remove this function
160   // this is only used in gdcm::SplitMosaicFilter / to pass value of a CSAElement
161   void Set(Value const &v) {
162     const ByteValue *bv = dynamic_cast<const ByteValue*>(&v);
163     if( bv ) {
164       //memcpy(Internal, bv->GetPointer(), bv->GetLength());
165       std::stringstream ss;
166       std::string s = std::string( bv->GetPointer(), bv->GetLength() );
167       ss.str( s );
168       EncodingImplementation<VRToEncoding<TVR>::Mode>::Read(Internal,
169         GetLength(),ss);
170     }
171   }
172 protected:
173   void SetNoSwap(Value const &v) {
174     const ByteValue *bv = dynamic_cast<const ByteValue*>(&v);
175     assert( bv ); // That would be bad...
176     //memcpy(Internal, bv->GetPointer(), bv->GetLength());
177     std::stringstream ss;
178     std::string s = std::string( bv->GetPointer(), bv->GetLength() );
179     ss.str( s );
180     EncodingImplementation<VRToEncoding<TVR>::Mode>::ReadNoSwap(Internal,
181       GetLength(),ss);
182   }
183 };
184
185 struct ignore_char {
186   ignore_char(char c):  m_char(c) {}
187   char m_char;
188 };
189 ignore_char const backslash('\\');
```

```
190
191   inline std::istream& operator» (std::istream& in, ignore_char const& ic) {
192     if (!in.eof())
193       in.clear(in.rdstate() & ~std::ios_base::failbit);
194     if (in.get() != ic.m_char)
195       in.setstate(std::ios_base::failbit);
196     return in;
197   }
198
199
200 // Implementation to perform formatted read and write
201 template<> class EncodingImplementation<VR::VRASCII> {
202 public:
203   template<typename T> // FIXME this should be VRToType<TVR>::Type
204   static inline void ReadComputeLength(T* data, unsigned int &length,
205                           std::istream &_is) {
206     assert( data );
207     //assert( length ); // != 0
208     length = 0;
209     assert( _is );
210 #if 0
211     char sep;
212     while( _is » data[length++] )
213       {
214       // Get the separator in between the values
215       assert( _is );
216       _is.get(sep);
217       assert( sep == '\\' || sep == ' ' ); // FIXME: Bad use of assert
218       if( sep == ' ' ) length--; // FIXME
219       }
220 #else
221   while( _is » std::ws » data[length++] » std::ws » backslash )
222     {
223     }
224 #endif
225     }
226
227   template<typename T> // FIXME this should be VRToType<TVR>::Type
228   static inline void Read(T* data, unsigned long length,
229                           std::istream &_is) {
230     assert( data );
231     assert( length ); // != 0
232     assert( _is );
233     // FIXME BUG: what if » operation fails ?
234     // gdcmData/MR00010001.dcm / SpacingBetweenSlices
235     _is » std::ws » data[0];
236     char sep;
237     //std::cout « "GetLength:  " « af->GetLength() « std::endl;
238     for(unsigned long i=1; i<length;++i) {
239       //assert( _is );
240       // Get the separator in between the values
241       _is » std::ws » sep; //_is.get(sep);
242       //assert( sep == '\\' ); // FIXME: Bad use of assert
243       _is » std::ws » data[i];
244       }
245     }
246
247   template<typename T>
248   static inline void ReadNoSwap(T* data, unsigned long length,
249                           std::istream &_is) {
250     Read(data,length,_is);
251 }
252   template<typename T>
253   static inline void Write(const T* data, unsigned long length,
254                            std::ostream &_os)  {
255     assert( data );
256     assert( length );
257     assert( _os );
258     _os « data[0];
259     for(unsigned long i=1; i<length; ++i) {
260       assert( _os );
261       _os « "\\" « data[i];
262       }
263     }
264 };
265
266 //#define VRDS16ILLEGAL
267
268 #ifdef VRDS16ILLEGAL
269 template < typename Float >
270 std::string to_string ( Float data ) {
```

```
271   std::stringstream in;
272   // in.imbue(std::locale::classic()); // This is not required AFAIK
273   int const digits =
274     static_cast< int >(
275     - std::log( std::numeric_limits<Float>::epsilon() )
276     / static_cast< Float >( std::log( 10.0 ) ) );
277   if ( in « std::dec « std::setprecision(/*2+*/digits) « data ) {
278     return ( in.str() );
279   } else {
280     throw "Impossible Conversion"; // should not happen ...
281   }
282 }
283 #else
284 //
      http://stackoverflow.com/questions/32631178/writing-ieee-754-1985-double-as-ascii-on-a-limited-16-bytes-string
285
286 static inline void clean(char *mant) {
287   char *ix = mant + strlen(mant) - 1;
288   while(('0' == *ix) && (ix > mant)) {
289     *ix-- = '\0';
290   }
291   if ('.'  == *ix) {
292     *ix = '\0';
293   }
294 }
295
296 static int add1(char *buf, int n) {
297   if (n < 0) return 1;
298   if (buf[n] == '9') {
299     buf[n] = '0';
300     return add1(buf, n-1);
301   }
302   else {
303     buf[n] = (char)(buf[n] + 1);
304   }
305   return 0;
306 }
307
308 static int doround(char *buf, unsigned int n) {
309   char c;
310   if (n >= strlen(buf)) return 0;
311   c = buf[n];
312   buf[n] = 0;
313   if ((c >= '5') && (c <= '9')) return add1(buf, n-1);
314   return 0;
315 }
316
317 static int roundat(char *buf, unsigned int i, int iexp) {
318   if (doround(buf, i) != 0) {
319     iexp += 1;
320     switch(iexp) {
321     case -2:
322       strcpy(buf, ".01");
323       break;
324     case -1:
325       strcpy(buf, ".1");
326       break;
327     case 0:
328       strcpy(buf, "1.");
329       break;
330     case 1:
331       strcpy(buf, "10");
332       break;
333     case 2:
334       strcpy(buf, "100");
335       break;
336     default:
337       sprintf(buf, "1e%d", iexp);
338     }
339     return 1;
340   }
341   return 0;
342 }
343
344 template < typename Float >
345 static void x16printf(char *buf, int size, Float f) {
346   char line[40];
347   char *mant = line + 1;
348   int iexp, lexp, i;
349   char exp[6];
350
```

```
351   if (f < 0) {
352     f = -f;
353     size -= 1;
354     *buf++ = '-';
355   }
356   sprintf(line, "%1.16e", f);
357   if (line[0] == '-') {
358     f = -f;
359     size -= 1;
360     *buf++ = '-';
361     sprintf(line, "%1.16e", f);
362   }
363   *mant = line[0];
364   i = (int)strcspn(mant, "eE");
365   mant[i] = '\0';
366   iexp = (int)strtol(mant + i + 1, nullptr, 10);
367   lexp = sprintf(exp, "e%d", iexp);
368   if ((iexp >= size) || (iexp < -3)) {
369     i = roundat(mant, size - 1 -lexp, iexp);
370     if(i == 1) {
371       strcpy(buf, mant);
372       return;
373     }
374     buf[0] = mant[0];
375     buf[1] = '.';
376     strncpy(buf + i + 2, mant + 1, size - 2 - lexp);
377     buf[size-lexp] = 0;
378     clean(buf);
379     strcat(buf, exp);
380   }
381   else if (iexp >= size - 2) {
382     roundat(mant, iexp + 1, iexp);
383     strcpy(buf, mant);
384   }
385   else if (iexp >= 0) {
386     i = roundat(mant, size - 1, iexp);
387     if (i == 1) {
388       strcpy(buf, mant);
389       return;
390     }
391     strncpy(buf, mant, iexp + 1);
392     buf[iexp + 1] = '.';
393     strncpy(buf + iexp + 2, mant + iexp + 1, size - iexp - 1);
394     buf[size] = 0;
395     clean(buf);
396   }
397   else {
398     int j;
399     i = roundat(mant, size + 1 + iexp, iexp);
400     if (i == 1) {
401       strcpy(buf, mant);
402       return;
403     }
404     buf[0] = '.';
405     for(j=0; j< -1 - iexp; j++) {
406       buf[j+1] = '0';
407     }
408     strncpy(buf - iexp, mant, size + 1 + iexp);
409     buf[size] = 0;
410     clean(buf);
411   }
412 }
413 #endif
414
415 template<> inline void EncodingImplementation<VR::VRASCII>::Write(const double* data, unsigned long length,
      std::ostream &_os)  {
416     assert( data );
417     assert( length );
418     assert( _os );
419 #ifdef VRDS16ILLEGAL
420     _os << to_string(data[0]);
421 #else
422     char buf[16+1];
423     x16printf(buf, 16, data[0]);
424     _os << buf;
425 #endif
426     for(unsigned long i=1; i<length; ++i) {
427       assert( _os );
428 #ifdef VRDS16ILLEGAL
429       _os << "\\" << to_string(data[i]);
430 #else
```

```
431       x16printf(buf, 16, data[i]);
432       _os « "\\" « buf;
433 #endif
434       }
435     }
436
437
438 // Implementation to perform binary read and write
439 // TODO rewrite operation so that either:
440 // #1.  dummy implementation use a pointer to Internal and do ++p (faster)
441 // #2.  Actually do some meta programming to unroll the loop
442 // (no notion of order in VM ...)
443 template<> class EncodingImplementation<VR::VRBINARY> {
444 public:
445   template<typename T> // FIXME this should be VRToType<TVR>::Type
446     static inline void ReadComputeLength(T* data, unsigned int &length,
447       std::istream &_is) {
448     const unsigned int type_size = sizeof(T);
449     assert( data ); // Can we read from pointer ?
450     //assert( length );
451     length /= type_size;
452     assert( _is ); // Is stream valid ?
453     _is.read( reinterpret_cast<char*>(data+0), type_size);
454     for(unsigned long i=1; i<length; ++i) {
455       assert( _is );
456       _is.read( reinterpret_cast<char*>(data+i), type_size );
457     }
458     }
459   template<typename T>
460   static inline void ReadNoSwap(T* data, unsigned long length,
461     std::istream &_is) {
462     const unsigned int type_size = sizeof(T);
463     assert( data ); // Can we read from pointer ?
464     assert( length );
465     assert( _is ); // Is stream valid ?
466     _is.read( reinterpret_cast<char*>(data+0), type_size);
467     for(unsigned long i=1; i<length; ++i) {
468       assert( _is );
469       _is.read( reinterpret_cast<char*>(data+i), type_size );
470     }
471     //ByteSwap<T>::SwapRangeFromSwapCodeIntoSystem(data,
472     //  _is.GetSwapCode(), length);
473     //SwapperNoOp::SwapArray(data,length);
474   }
475   template<typename T>
476   static inline void Read(T* data, unsigned long length,
477     std::istream &_is) {
478     const unsigned int type_size = sizeof(T);
479     assert( data ); // Can we read from pointer ?
480     assert( length );
481     assert( _is ); // Is stream valid ?
482     _is.read( reinterpret_cast<char*>(data+0), type_size);
483     for(unsigned long i=1; i<length; ++i) {
484       assert( _is );
485       _is.read( reinterpret_cast<char*>(data+i), type_size );
486     }
487     //ByteSwap<T>::SwapRangeFromSwapCodeIntoSystem(data,
488     //  _is.GetSwapCode(), length);
489     SwapperNoOp::SwapArray(data,length);
490   }
491   template<typename T>
492   static inline void Write(const T* data, unsigned long length,
493     std::ostream &_os) {
494     const unsigned int type_size = sizeof(T);
495     assert( data ); // Can we write into pointer ?
496     assert( length );
497     assert( _os ); // Is stream valid ?
498     //ByteSwap<T>::SwapRangeFromSwapCodeIntoSystem((T*)data,
499     //  _os.GetSwapCode(), length);
500     T swappedData = SwapperNoOp::Swap(data[0]);
501     _os.write( reinterpret_cast<const char*>(&swappedData), type_size);
502     for(unsigned long i=1; i<length;++i) {
503       assert( _os );
504       swappedData = SwapperNoOp::Swap(data[i]);
505       _os.write( reinterpret_cast<const char*>(&swappedData), type_size );
506     }
507     //ByteSwap<T>::SwapRangeFromSwapCodeIntoSystem((T*)data,
508     //  _os.GetSwapCode(), length);
509   }
510 };
511
```

```
512 // For particular case for ASCII string
513 // WARNING: This template explicitly instantiates a particular
514 // EncodingImplementation THEREFORE it is required to be declared after the
515 // EncodingImplementation is needs (doh!)
516 #if 0
517 template<int TVM>
518 class Element<TVM>
519 {
520 public:
521   Element(const char array[])
522     {
523     unsigned int i = 0;
524     const char sep = '\\';
525     std::string sarray = array;
526     std::string::size_type pos1 = 0;
527     std::string::size_type pos2 = sarray.find(sep, pos1+1);
528     while(pos2 != std::string::npos)
529       {
530       Internal[i++] = sarray.substr(pos1, pos2-pos1);
531       pos1 = pos2+1;
532       pos2 = sarray.find(sep, pos1+1);
533       }
534     Internal[i] = sarray.substr(pos1, pos2-pos1);
535     // Shouldn't we do the contrary, since we know how many separators
536     // (and default behavior is to discard anything after the VM declared
537     assert( GetLength()-1 == i );
538     }
539
540   unsigned long GetLength()const {
541     return VMToLength<TVM>::Length;
542   }
543   // Implementation of Print is common to all Mode (ASCII/Binary)
544   void Print(std::ostream &_os)const {
545     _os << Internal[0]; // VM is at least guarantee to be one
546     for(int i=1; i<VMToLength<TVM>::Length; ++i)
547       _os << "," << Internal[i];
548     }
549
550   void Read(std::istream &_is) {
551     EncodingImplementation<VR::VRASCII>::Read(Internal, GetLength(),_is);
552     }
553   void Write(std::ostream &_os)const {
554     EncodingImplementation<VR::VRASCII>::Write(Internal, GetLength(),_os);
555     }
556 private:
557   typename String Internal[VMToLength<TVM>::Length];
558 };
559
560 template< int TVM>
561 class Element<VR::PN, TVM> :  public StringElement<TVM>
562 {
563   enum { ElementDisableCombinationsCheck = sizeof ( ElementDisableCombinations<VR::PN, TVM> ) };
564 };
565 #endif
566
567 // Implementation for the undefined length (dynamically allocated array)
568 template<long long TVR>
569 class Element<TVR, VM::VM1_n>
570 {
571   enum { ElementDisableCombinationsCheck = sizeof ( ElementDisableCombinations<TVR, VM::VM1_n> ) };
572 public:
573   // This the way to prevent default initialization
574   explicit Element() { Internal=nullptr; Length=0; Save = false; }
575   ~Element() {
576     if( Save ) {
577       delete[] Internal;
578     }
579     Internal = nullptr;
580   }
581
582   static VR  GetVR()  { return (VR::VRType)TVR; }
583   static VM  GetVM()  { return VM::VM1_n; }
584
585   // Length manipulation
586   // SetLength should really be protected anyway...all operation
587   // should go through SetArray
588   unsigned long GetLength()const { return Length; }
589   typedef typename VRToType<TVR>::Type Type;
590
591   void SetLength(unsigned long len) {
592     const unsigned int size = sizeof(Type);
```

```
593     if( len ) {
594       if( len > Length ) {
595         // perform realloc
596         assert( (len / size) * size == len );
597         Type *internal = new Type[len / size];
598         assert( Save == false );
599         Save = true; // ????
600         if( Internal )
601           {
602           memcpy(internal, Internal, len);
603           delete[] Internal;
604           }
605         Internal = internal;
606         }
607       }
608     Length = len / size;
609   }
610
611   // If save is set to zero user should not delete the pointer
612   //void SetArray(const typename VRToType<TVR>::Type *array, int len, bool save = false)
613   void SetArray(const Type *array, unsigned long len,
614     bool save = false) {
615     if( save ) {
616       SetLength(len); // realloc
617       memcpy(Internal, array, len/*/sizeof(Type)*/);
618       assert( Save == false );
619       }
620     else {
621       // TODO rewrite this stupid code:
622       assert( Length == 0 );
623       assert( Internal == nullptr );
624       assert( Save == false );
625       Length = len / sizeof(Type);
626       //assert( (len / sizeof(Type)) * sizeof(Type) == len );
627       // MR00010001.dcm is a tough kid:  0019,105a is supposed to be VR::FL, VM::VM3 but
628       // length is 14 bytes instead of 12 bytes.  Simply consider value is total garbage.
629       if( (len / sizeof(Type)) * sizeof(Type) != len ) { Internal = nullptr; Length = 0; }
630       else Internal = const_cast<Type*>(array);
631       }
632       Save = save;
633   }
634   void SetValue(typename VRToType<TVR>::Type v, unsigned int idx = 0) {
635     assert( idx < Length );
636     Internal[idx] = v;
637   }
638   const typename VRToType<TVR>::Type &GetValue(unsigned int idx = 0)const {
639     assert( idx < Length );
640     return Internal[idx];
641   }
642   typename VRToType<TVR>::Type &GetValue(unsigned int idx = 0) {
643     //assert( idx < Length );
644     return Internal[idx];
645   }
646   typename VRToType<TVR>::Type operator[] (unsigned int idx)const {
647     return GetValue(idx);
648   }
649   void Set(Value const &v) {
650     const ByteValue *bv = dynamic_cast<const ByteValue*>(&v);
651     assert( bv ); // That would be bad...
652     if( (VR::VRType)(VRToEncoding<TVR>::Mode) == VR::VRBINARY )
653       {
654       const Type* array = (const Type*)bv->GetVoidPointer();
655       if( array ) {
656         assert( array ); // That would be bad...
657         assert( Internal == nullptr );
658         SetArray(array, bv->GetLength() ); }
659       }
660     else
661       {
662       std::stringstream ss;
663       std::string s = std::string( bv->GetPointer(), bv->GetLength() );
664       ss.str( s );
665       EncodingImplementation<VRToEncoding<TVR>::Mode>::Read(Internal,
666         GetLength(),ss);
667       }
668   }
669   void SetFromDataElement(DataElement const &de) {
670     const ByteValue *bv = de.GetByteValue();
671     if( !bv ) return;
672 #ifdef GDCM_WORDS_BIGENDIAN
673     if( de.GetVR() == VR::UN /*|| de.GetVR() == VR::INVALID*/ )
```

```
674 #else
675     if( de.GetVR() == VR::UN || de.GetVR() == VR::INVALID )
676 #endif
677       {
678       Set(de.GetValue());
679       }
680     else
681       {
682       SetNoSwap(de.GetValue());
683       }
684   }
685
686
687   // Need to be placed after definition of EncodingImplementation<VR::VRASCII>
688   void WriteASCII(std::ostream &os)const {
689     return EncodingImplementation<VR::VRASCII>::Write(Internal, GetLength(), os);
690     }
691
692   // Implementation of Print is common to all Mode (ASCII/Binary)
693   void Print(std::ostream &_os)const {
694     assert( Length );
695     assert( Internal );
696     _os « Internal[0]; // VM is at least guarantee to be one
697     const unsigned long length = GetLength() < 25 ?  GetLength() :  25;
698     for(unsigned long i=1; i<length; ++i)
699       _os « "," « Internal[i];
700     }
701   void Read(std::istream &_is) {
702     if( !Internal ) return;
703     EncodingImplementation<VRToEncoding<TVR>::Mode>::Read(Internal,
704       GetLength(),_is);
705     }
706   //void ReadComputeLength(std::istream &_is) {
707   //  if( !Internal ) return;
708   //  EncodingImplementation<VRToEncoding<TVR>::Mode>::ReadComputeLength(Internal,
709   //    Length,_is);
710   //  }
711   void Write(std::ostream &_os)const {
712     EncodingImplementation<VRToEncoding<TVR>::Mode>::Write(Internal,
713       GetLength(),_os);
714     }
715
716   DataElement GetAsDataElement()const {
717     DataElement ret;
718     ret.SetVR( (VR::VRType)TVR );
719     assert( ret.GetVR() != VR::SQ );
720     if( Internal )
721       {
722       std::ostringstream os;
723       EncodingImplementation<VRToEncoding<TVR>::Mode>::Write(Internal,
724         GetLength(),os);
725       if( (VR::VRType)VRToEncoding<TVR>::Mode == VR::VRASCII )
726         {
727         if( GetVR() != VR::UI )
728           {
729           if( os.str().size() % 2 )
730             {
731             os « " ";
732             }
733           }
734         }
735       VL::Type osStrSize = (VL::Type)os.str().size();
736       ret.SetByteValue( os.str().c_str(), osStrSize );
737       }
738     return ret;
739   }
740
741   Element(const Element&_val) {
742     if( this != &_val) {
743       *this = _val;
744       }
745     }
746
747   Element &operator=(const Element &_val) {
748     Length = 0; // SYITF
749     Internal = 0;
750     SetArray(_val.Internal, _val.Length, true);
751     return *this;
752     }
753 protected:
754   void SetNoSwap(Value const &v) {
```

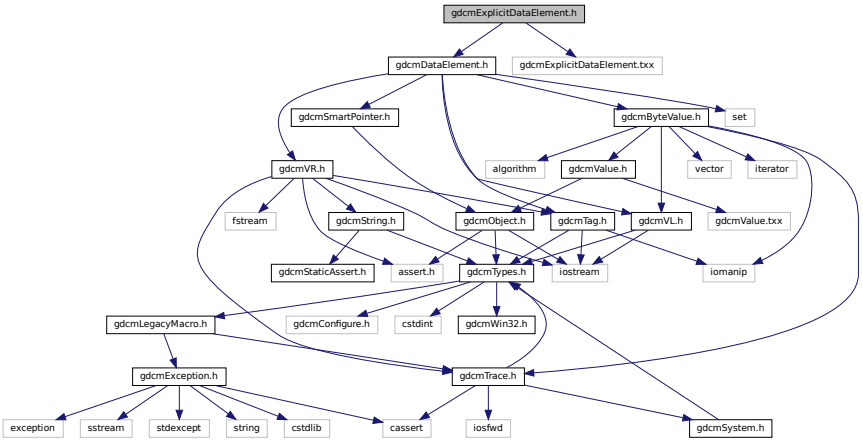
```
755     const ByteValue *bv = dynamic_cast<const ByteValue*>(&v);
756     assert( bv ); // That would be bad...
757     if( (VR::VRType)(VRToEncoding<TVR>::Mode) == VR::VRBINARY )
758       {
759       const Type* array = (const Type*)bv->GetPointer();
760       if( array ) {
761         assert( array ); // That would be bad...
762         assert( Internal == nullptr );
763         SetArray(array, bv->GetLength() ); }
764       }
765     else
766       {
767       std::stringstream ss;
768       std::string s = std::string( bv->GetPointer(), bv->GetLength() );
769       ss.str( s );
770       EncodingImplementation<VRToEncoding<TVR>::Mode>::ReadNoSwap(Internal,
771         GetLength(),ss);
772       }
773   }
774
775 private:
776   typename VRToType<TVR>::Type *Internal;
777   unsigned long Length; // unsigned int ??
778   bool Save;
779 };
780
781 //template <int TVM = VM::VM1_n>
782 //class Element<VR::OB, TVM > :  public Element<VR::OB, VM::VM1_n> {};
783
784 // Partial specialization for derivatives of 1-n :  2-n, 3-n ...
785 template<long long TVR>
786 class Element<TVR, VM::VM1_2> :  public Element<TVR, VM::VM1_n>
787 {
788 public:
789   typedef Element<TVR, VM::VM1_n> Parent;
790   void SetLength(int len) {
791     if( len != 1 && len != 2 ) return;
792     Parent::SetLength(len);
793   }
794 };
795 template<long long TVR>
796 class Element<TVR, VM::VM2_n> :  public Element<TVR, VM::VM1_n>
797 {
798   enum { ElementDisableCombinationsCheck = sizeof ( ElementDisableCombinations<TVR, VM::VM2_n> ) };
799 public:
800   typedef Element<TVR, VM::VM1_n> Parent;
801   void SetLength(int len) {
802     if( len <= 1 ) return;
803     Parent::SetLength(len);
804   }
805 };
806 template<long long TVR>
807 class Element<TVR, VM::VM2_2n> :  public Element<TVR, VM::VM2_n>
808 {
809   enum { ElementDisableCombinationsCheck = sizeof ( ElementDisableCombinations<TVR, VM::VM2_2n> ) };
810 public:
811   typedef Element<TVR, VM::VM2_n> Parent;
812   void SetLength(int len) {
813     if( len % 2 ) return;
814     Parent::SetLength(len);
815   }
816 };
817 template<long long TVR>
818 class Element<TVR, VM::VM3_n> :  public Element<TVR, VM::VM1_n>
819 {
820   enum { ElementDisableCombinationsCheck = sizeof ( ElementDisableCombinations<TVR, VM::VM3_n> ) };
821 public:
822   typedef Element<TVR, VM::VM1_n> Parent;
823   void SetLength(int len) {
824     if( len <= 2 ) return;
825     Parent::SetLength(len);
826   }
827 };
828 template<long long TVR>
829 class Element<TVR, VM::VM3_3n> :  public Element<TVR, VM::VM3_n>
830 {
831   enum { ElementDisableCombinationsCheck = sizeof ( ElementDisableCombinations<TVR, VM::VM3_3n> ) };
832 public:
833   typedef Element<TVR, VM::VM3_n> Parent;
834   void SetLength(int len) {
835     if( len % 3 ) return;
```

```
836     Parent::SetLength(len);
837   }
838 };
839 template<long long TVR>
840 class Element<TVR, VM::VM3_4> :  public Element<TVR, VM::VM1_n>
841 {
842 public:
843   typedef Element<TVR, VM::VM1_n> Parent;
844   void SetLength(int len) {
845     if( len != 3 && len != 4 ) return;
846     Parent::SetLength(len);
847   }
848 };
849
850
851 //template<int T> struct VRToLength;
852 //template <> struct VRToLength<VR::AS>
853 //{ enum { Length  = VM::VM1 }; }
854 //template<>
855 //class Element<VR::AS> :  public Element<VR::AS, VRToLength<VR::AS>::Length >
856
857 // only 0010 1010 AS 1 Patient's Age
858 template<>
859 class Element<VR::AS, VM::VM5>
860 {
861   enum { ElementDisableCombinationsCheck = sizeof ( ElementDisableCombinations<VR::AS, VM::VM5> ) };
862 public:
863   char Internal[VMToLength<VM::VM5>::Length * sizeof( VRToType<VR::AS>::Type )];
864   void Print(std::ostream &_os)const {
865     _os « Internal;
866     }
867   unsigned long GetLength()const {
868     return VMToLength<VM::VM5>::Length;
869   }
870 };
871
872
873 template <>
874 class Element<VR::OB, VM::VM1> :  public Element<VR::OB, VM::VM1_n> {};
875
876 // Same for OW:
877 template <>
878 class Element<VR::OW, VM::VM1> :  public Element<VR::OW, VM::VM1_n> {};
879
880
881 } // namespace gdcm_ns
882
883 #endif //GDCMELEMENT_H
```

11.135 gdcmExplicitDataElement.h File Reference

```
#include "gdcmDataElement.h"
#include "gdcmExplicitDataElement.txx"
```

Include dependency graph for gdcmExplicitDataElement.h:



This graph shows which files directly or indirectly include this file:

Classes

- class gdcm::ExplicitDataElement

  Class to read/write a DataElement as Explicit Data Element.

Namespaces

- namespace gdcm

11.136 gdcmExplicitDataElement.h

Go to the documentation of this file.

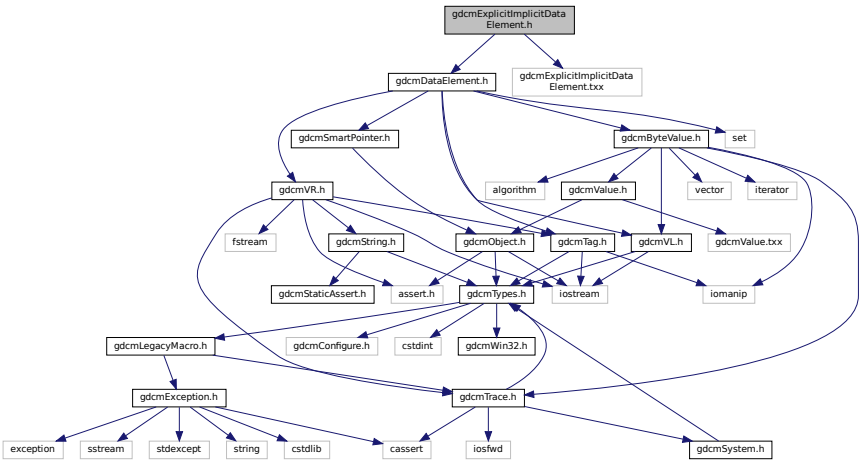
```
1 /*=========================================================================
2
3 Program:  GDCM (Grassroots DICOM). A DICOM library
4
5 Copyright (c) 2006-2011 Mathieu Malaterre
6 All rights reserved.
7 See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.
8
9 This software is distributed WITHOUT ANY WARRANTY; without even
10 the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
11 PURPOSE.  See the above copyright notice for more information.
```

```
12
13 =========================================================================*/
14 #ifndef GDCMEXPLICITDATAELEMENT_H
15 #define GDCMEXPLICITDATAELEMENT_H
16
17 #include "gdcmDataElement.h"
18
19 namespace gdcm_ns
20 {
25 class GDCM_EXPORT ExplicitDataElement :  public DataElement
26 {
27 public:
28   VL GetLength() const;
29
30   template <typename TSwap>
31   std::istream &Read(std::istream &is);
32
33   template <typename TSwap>
34   std::istream &ReadPreValue(std::istream &is);
35
36   template <typename TSwap>
37   std::istream &ReadValue(std::istream &is, bool readvalues = true);
38
39   template <typename TSwap>
40   std::istream &ReadWithLength(std::istream &is, VL & length);
41
42   template <typename TSwap>
43   const std::ostream &Write(std::ostream &os) const;
44 };
45
46 } // end namespace gdcm_ns
47
48 #include "gdcmExplicitDataElement.txx"
49
50 #endif //GDCMEXPLICITDATAELEMENT_H
```

11.137 gdcmExplicitImplicitDataElement.h File Reference

```
#include "gdcmDataElement.h"
#include "gdcmExplicitImplicitDataElement.txx"
```

Include dependency graph for gdcmExplicitImplicitDataElement.h:



Classes

- class gdcm::ExplicitImplicitDataElement

Class to read/write a DataElement as ExplicitImplicit Data Element.

Namespaces

- namespace gdcm

11.138 gdcmExplicitImplicitDataElement.h

Go to the documentation of this file.

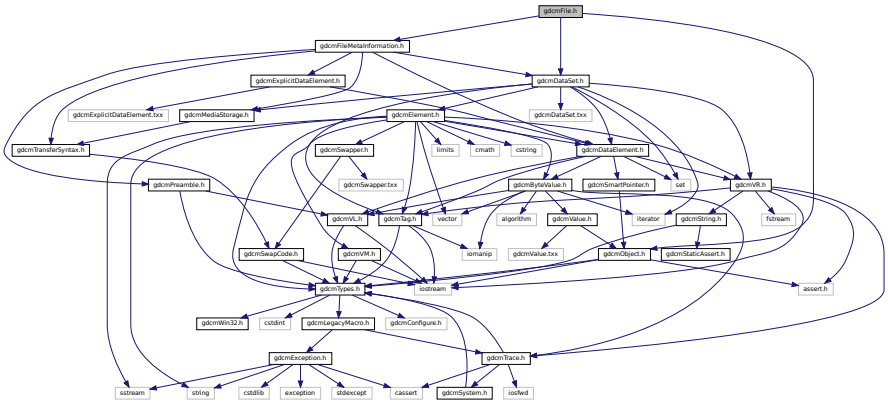
```
1 /*=========================================================================
2
3 Program:  GDCM (Grassroots DICOM). A DICOM library
4
5 Copyright (c) 2006-2011 Mathieu Malaterre
6 All rights reserved.
7 See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.
8
9 This software is distributed WITHOUT ANY WARRANTY; without even
10 the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
11 PURPOSE.  See the above copyright notice for more information.
12
13 =========================================================================*/
14 #ifndef GDCMEXPLICITIMPLICITDATAELEMENT_H
15 #define GDCMEXPLICITIMPLICITDATAELEMENT_H
16
17 #include "gdcmDataElement.h"
18
19 namespace gdcm
20 {
21 // Data Element (ExplicitImplicit)
29 class GDCM_EXPORT ExplicitImplicitDataElement :  public DataElement
30 {
31 public:
32   VL GetLength() const;
33
34   template <typename TSwap>
35   std::istream &Read(std::istream &is);
36
37   template <typename TSwap>
38   std::istream &ReadPreValue(std::istream &is);
39
40   template <typename TSwap>
41   std::istream &ReadValue(std::istream &is, bool readvalues = true);
42
43   template <typename TSwap>
44   std::istream &ReadWithLength(std::istream &is, VL & length)
45     {
46     (void)length;
47     return Read<TSwap>(is);
48     }
49
50   // PURPOSELY do not provide an implementation for writing !
51   //template <typename TSwap>
52   //const std::ostream &Write(std::ostream &os) const;
53 };
54
55 } // end namespace gdcm
56
57 #include "gdcmExplicitImplicitDataElement.txx"
58
59 #endif //GDCMEXPLICITIMPLICITDATAELEMENT_H
```

11.139 gdcmFile.h File Reference

```
#include "gdcmObject.h"
#include "gdcmDataSet.h"
```

```
#include "gdcmFileMetaInformation.h"
```

Include dependency graph for gdcmFile.h:



This graph shows which files directly or indirectly include this file:

Classes

- class gdcm::File

  a DICOM File

Namespaces

- namespace gdcm

Functions

- std::ostream & gdcm::operator<< (std::ostream &os, const File &val)

11.140 gdcmFile.h

Go to the documentation of this file.

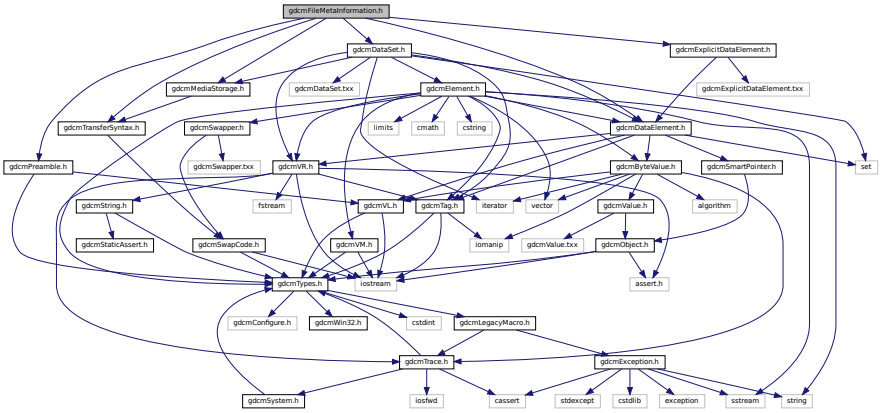
```
1 /*=========================================================================
2
3 Program:  GDCM (Grassroots DICOM). A DICOM library
4
5 Copyright (c) 2006-2011 Mathieu Malaterre
6 All rights reserved.
```

```
7 See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.
8
9 This software is distributed WITHOUT ANY WARRANTY; without even
10 the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
11 PURPOSE.  See the above copyright notice for more information.
12
13 =========================================================================*/
14 #ifndef GDCMFILE_H
15 #define GDCMFILE_H
16
17 #include "gdcmObject.h"
18 #include "gdcmDataSet.h"
19 #include "gdcmFileMetaInformation.h"
20
21 namespace gdcm_ns
22 {
23
33 class GDCM_EXPORT File :  public Object
34 {
35 public:
36   File();
37   ~File() override;
38
39   friend std::ostream &operator«(std::ostream &os, const File &val);
40
42   std::istream &Read(std::istream &is);
43
45   std::ostream const &Write(std::ostream &os) const;
46
48   const FileMetaInformation &GetHeader()const { return Header; }
49
51   FileMetaInformation &GetHeader() { return Header; }
52
54   void SetHeader( const FileMetaInformation &fmi ) { Header = fmi; }
55
57   const DataSet &GetDataSet()const { return DS; }
58
60   DataSet &GetDataSet() { return DS; }
61
63   void SetDataSet( const DataSet &ds) { DS = ds; }
64
65 private:
66   FileMetaInformation Header;
67   DataSet DS;
68 };
69 //-----------------------------------------------------------------------------
70 inline std::ostream& operator«(std::ostream &os, const File &val)
71 {
72   os « val.GetHeader() « std::endl;
73   //os « val.GetDataSet() « std::endl; // FIXME
74   assert(0);
75   return os;
76 }
77
78 } // end namespace gdcm_ns
79
80 #endif //GDCMFILE_H
```

11.141 gdcmFileMetaInformation.h File Reference

```
#include "gdcmPreamble.h"
#include "gdcmDataSet.h"
#include "gdcmDataElement.h"
#include "gdcmMediaStorage.h"
#include "gdcmTransferSyntax.h"
#include "gdcmExplicitDataElement.h"
```

Include dependency graph for gdcmFileMetaInformation.h:



This graph shows which files directly or indirectly include this file:

Classes

- class gdcm::FileMetaInformation

  Class to represent a File Meta Information.

Namespaces

- namespace gdcm

Functions

- std::ostream & gdcm::operator<< (std::ostream &os, const FileMetaInformation &val)

11.142 gdcmFileMetaInformation.h

Go to the documentation of this file.

```
1 /*=========================================================================
2
3 Program:  GDCM (Grassroots DICOM). A DICOM library
4
5 Copyright (c) 2006-2011 Mathieu Malaterre
6 All rights reserved.
```

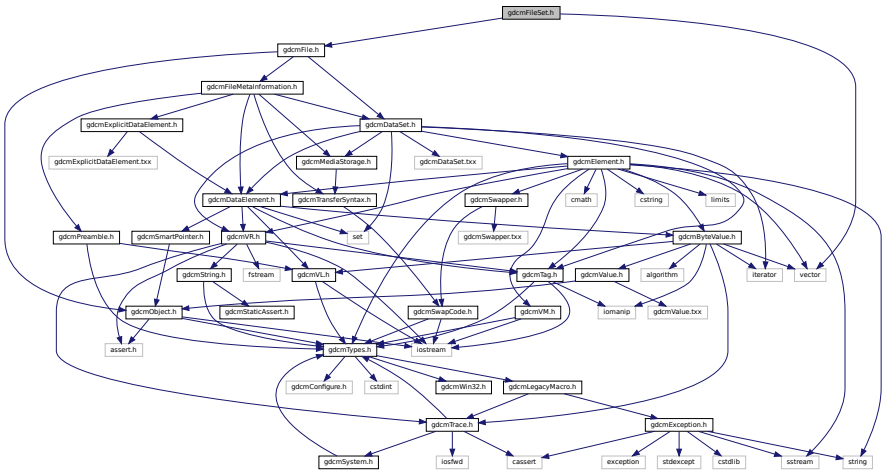
```
7 See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.
8
9 This software is distributed WITHOUT ANY WARRANTY; without even
10 the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
11 PURPOSE.  See the above copyright notice for more information.
12
13 =========================================================================*/
14 #ifndef GDCMFILEMETAINFORMATION_H
15 #define GDCMFILEMETAINFORMATION_H
16
17 #include "gdcmPreamble.h"
18 #include "gdcmDataSet.h"
19 #include "gdcmDataElement.h"
20 #include "gdcmMediaStorage.h"
21 #include "gdcmTransferSyntax.h"
22 #include "gdcmExplicitDataElement.h"
23
24 namespace gdcm_ns
25 {
40 class GDCM_EXPORT FileMetaInformation :  public DataSet
41 {
42 public:
43   // FIXME: TransferSyntax::TS_END -> TransferSyntax::ImplicitDataElement
44   FileMetaInformation();
45   ~FileMetaInformation();
46
47   friend std::ostream &operator<<(std::ostream &_os, const FileMetaInformation &_val);
48
49   bool IsValid()const { return true; }
50
51   TransferSyntax::NegociatedType GetMetaInformationTS()const { return MetaInformationTS; }
52   void SetDataSetTransferSyntax(const TransferSyntax &ts);
53   const TransferSyntax &GetDataSetTransferSyntax()const { return DataSetTS; }
54   MediaStorage GetMediaStorage() const;
55   std::string GetMediaStorageAsString() const;
56
57   // FIXME: no virtual function means:  duplicate code...
58   void Insert(const DataElement& de) {
59     if( de.GetTag().GetGroup() == 0x0002 )
60       {
61       InsertDataElement( de );
62       }
63     else
64       {
65       gdcmErrorMacro( "Cannot add element with group != 0x0002 in the file meta header:  " << de );
66       }
67   }
68   void Replace(const DataElement& de) {
69     Remove(de.GetTag());
70     Insert(de);
71   }
72
74   std::istream &Read(std::istream &is);
75   std::istream &ReadCompat(std::istream &is);
76
78   std::ostream &Write(std::ostream &os) const;
79
81   void FillFromDataSet(DataSet const &ds);
82
84   const Preamble &GetPreamble()const { return P; }
85   Preamble &GetPreamble() { return P; }
86   void SetPreamble(const Preamble &p) { P = p; }
87
89   static void SetImplementationClassUID(const char * imp);
90   static void AppendImplementationClassUID(const char * imp);
91   static const char *GetImplementationClassUID();
92   static void SetImplementationVersionName(const char * version);
93   static const char *GetImplementationVersionName();
94   static void SetSourceApplicationEntityTitle(const char * title);
95   static const char *GetSourceApplicationEntityTitle();
96
97   FileMetaInformation(FileMetaInformation const &fmi):DataSet(fmi)
98     {
99     DataSetTS = fmi.DataSetTS;
100     MetaInformationTS = fmi.MetaInformationTS;
101     DataSetMS = fmi.DataSetMS;
102     }
103   FileMetaInformation& operator=(const FileMetaInformation& fmi)
104     {
105     DataSetTS = fmi.DataSetTS;
106     MetaInformationTS = fmi.MetaInformationTS;
```

```
107     DataSetMS = fmi.DataSetMS;
108     return *this;
109     }
110
111     VL GetFullLength()const {
112       return P.GetLength() + DataSet::GetLength<ExplicitDataElement>();
113     }
114
115 protected:
116   void ComputeDataSetTransferSyntax(); // FIXME
117
118   template <typename TSwap>
119   std::istream &ReadCompatInternal(std::istream &is);
120
121   void Default();
122   void ComputeDataSetMediaStorageSOPClass();
123
124   TransferSyntax DataSetTS;
125   TransferSyntax::NegociatedType MetaInformationTS;
126   MediaStorage::MSType DataSetMS;
127
128 protected:
129   static const char * GetFileMetaInformationVersion();
130   static const char * GetGDCMImplementationClassUID();
131   static const char * GetGDCMImplementationVersionName();
132   static const char * GetGDCMSourceApplicationEntityTitle();
133
134 private:
135   Preamble P;
136
137 //static stuff:
138   static const char GDCM_FILE_META_INFORMATION_VERSION[];
139   static const char GDCM_IMPLEMENTATION_CLASS_UID[];
140   static const char GDCM_IMPLEMENTATION_VERSION_NAME[];
141   static const char GDCM_SOURCE_APPLICATION_ENTITY_TITLE[];
142   static std::string ImplementationClassUID;
143   static std::string ImplementationVersionName;
144   static std::string SourceApplicationEntityTitle;
145 };
146 //-----------------------------------------------------------------------------
147 inline std::ostream& operator«(std::ostream &os, const FileMetaInformation &val)
148 {
149   os « val.GetPreamble() « std::endl;
150   val.Print( os );
151   return os;
152 }
153
154 } // end namespace gdcm_ns
155
156 #endif //GDCMFILEMETAINFORMATION_H
```

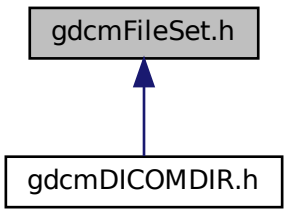
11.143 gdcmFileSet.h File Reference

```
#include "gdcmFile.h"
#include <vector>
```

Include dependency graph for gdcmFileSet.h:



This graph shows which files directly or indirectly include this file:



Classes

- class gdcm::FileSet

Namespaces

- namespace gdcm

Functions

- std::ostream & gdcm::operator<< (std::ostream &os, const FileSet &f)

11.144 gdcmFileSet.h

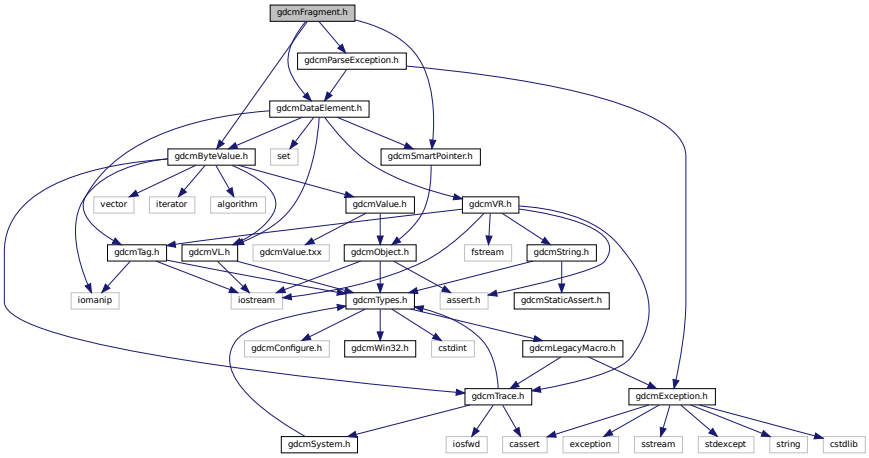
Go to the documentation of this file.

```
1 /*=========================================================================
2
3 Program:  GDCM (Grassroots DICOM). A DICOM library
4
5 Copyright (c) 2006-2011 Mathieu Malaterre
6 All rights reserved.
7 See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.
8
9 This software is distributed WITHOUT ANY WARRANTY; without even
10 the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
11 PURPOSE.  See the above copyright notice for more information.
12
13 =========================================================================*/
14 #ifndef GDCMFILESET_H
15 #define GDCMFILESET_H
16
17 #include "gdcmFile.h"
18 #include <vector>
19
20 namespace gdcm
21 {
27 class GDCM_EXPORT FileSet
28 {
29   friend std::ostream& operator<<(std::ostream &_os, const FileSet &d);
30 public:
31   FileSet():Files() {}
32   typedef std::string FileType;
33   typedef std::vector<FileType> FilesType;
34
36   void AddFile(File const & ) {}
37
40   bool AddFile(const char *filename);
41
42   void SetFiles(FilesType const &files);
43   FilesType const &GetFiles()const {
44     return Files;
45   }
46
47 private:
48   FilesType Files;
49 };
50 //-----------------------------------------------------------------------------
51 inline std::ostream& operator<<(std::ostream &os, const FileSet &f)
52 {
53   (void)f; // FIXME
54   return os;
55 }
56
57 } // end namespace gdcm
58
59 #endif //GDCMFILESET_H
```

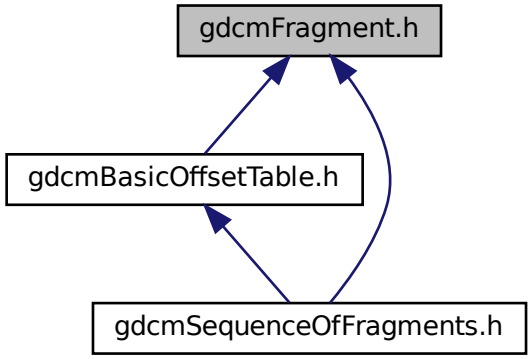
11.145 gdcmFragment.h File Reference

```
#include "gdcmDataElement.h"
#include "gdcmByteValue.h"
#include "gdcmSmartPointer.h"
#include "gdcmParseException.h"
```

Include dependency graph for gdcmFragment.h:



This graph shows which files directly or indirectly include this file:



Classes

- class gdcm::Fragment

  Class to represent a Fragment.

Namespaces

- namespace gdcm

Functions

- std::ostream & gdcm::operator<< (std::ostream &os, const Fragment &val)

11.146 gdcmFragment.h

Go to the documentation of this file.

```
1 /*=========================================================================
2
3 Program:  GDCM (Grassroots DICOM). A DICOM library
4
5 Copyright (c) 2006-2011 Mathieu Malaterre
6 All rights reserved.
7 See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.
8
9 This software is distributed WITHOUT ANY WARRANTY; without even
10 the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
11 PURPOSE.  See the above copyright notice for more information.
12
13 =========================================================================*/
14 #ifndef GDCMFRAGMENT_H
15 #define GDCMFRAGMENT_H
16
17 #include "gdcmDataElement.h"
18 #include "gdcmByteValue.h"
19 #include "gdcmSmartPointer.h"
20 #include "gdcmParseException.h"
21
22 namespace gdcm_ns
23 {
24
25 // Implementation detail:
26 // I think Fragment should be a protected subclass of DataElement:
27 // looking somewhat like this:
28 /*
29 class GDCM_EXPORT Fragment :  protected DataElement
30 {
31 public:
32 using DataElement::GetTag;
33 using DataElement::GetVL;
34 using DataElement::SetByteValue;
35 using DataElement::GetByteValue;
36 using DataElement::GetValue;
37 */
38 // Instead I am only hiding the SetTag member...
39
43 class GDCM_EXPORT Fragment :  public DataElement
44 {
45 //protected:
46 //  void SetTag(const Tag &t);
47 public:
48   Fragment() :  DataElement(Tag(0xfffe, 0xe000), 0) {}
49   friend std::ostream &operator«(std::ostream &os, const Fragment &val);
50
51   VL GetLength() const;
52
53   VL ComputeLength() const;
54
55   template <typename TSwap>
56   std::istream &Read(std::istream &is)
57     {
58     ReadPreValue<TSwap>(is);
59     return ReadValue<TSwap>(is);
60     }
61
62   template <typename TSwap>
63   std::istream &ReadPreValue(std::istream &is)
64     {
65     const Tag itemStart(0xfffe, 0xe000);
66     const Tag seqDelItem(0xfffe,0xe0dd);
67
68     TagField.Read<TSwap>(is);
69     if( !is )
70       {
```

```
71        //  BogusItemStartItemEnd.dcm
72        throw Exception( "Problem #1" );
73        }
74      if( !ValueLengthField.Read<TSwap>(is) )
75        {
76        // GENESIS_SIGNA-JPEG-CorruptFrag.dcm
77        // JPEG fragment is declared to have 61902, but in fact really is only 61901
78        // so we end up reading 0xddff,0x00e0, and VL = 0x0 (1 byte)
79        throw Exception( "Problem #2" );
80        }
81 #ifdef GDCM_SUPPORT_BROKEN_IMPLEMENTATION
82      if( TagField != itemStart && TagField != seqDelItem )
83        {
84        throw Exception( "Problem #3" );
85        }
86 #endif
87      return is;
88      }
89
90    template <typename TSwap>
91    std::istream &ReadValue(std::istream &is)
92      {
93      // Superclass
94      const Tag itemStart(0xfffe, 0xe000);
95      const Tag seqDelItem(0xfffe,0xe0dd);
96      // Self
97      SmartPointer<ByteValue> bv = new ByteValue;
98      bv->SetLength(ValueLengthField);
99      if( !bv->Read<TSwap>(is) )
100       {
101       // Fragment is incomplete, but is a itemStart, let's try to push it anyway...
102       gdcmWarningMacro( "Fragment could not be read" );
103       //bv->SetLength(is.gcount());
104       ValueField = bv;
105       ParseException pe;
106       pe.SetLastElement( *this );
107       throw pe;
108       }
109     ValueField = bv;
110     return is;
111     }
112
113   template <typename TSwap>
114   std::istream &ReadBacktrack(std::istream &is)
115     {
116     const Tag itemStart(0xfffe, 0xe000);
117     const Tag seqDelItem(0xfffe,0xe0dd);
118
119     bool cont = true;
120     const std::streampos start = is.tellg();
121     const int max = 10;
122     int offset = 0;
123     while( cont )
124       {
125       TagField.Read<TSwap>(is);
126       assert( is );
127       if( TagField != itemStart && TagField != seqDelItem )
128         {
129         ++offset;
130         is.seekg( (std::streampos)((size_t)start - offset) );
131         gdcmWarningMacro( "Fuzzy Search, backtrack:  " << (start - is.tellg()) << " Offset:  " << is.tellg() );
132         if( offset > max )
133           {
134           gdcmErrorMacro( "Giving up" );
135           throw "Impossible to backtrack";
136           }
137         }
138       else
139         {
140         cont = false;
141         }
142       }
143     assert( TagField == itemStart || TagField == seqDelItem );
144     if( !ValueLengthField.Read<TSwap>(is) )
145       {
146       return is;
147       }
148
149     // Self
150     SmartPointer<ByteValue> bv = new ByteValue;
151     bv->SetLength(ValueLengthField);
```

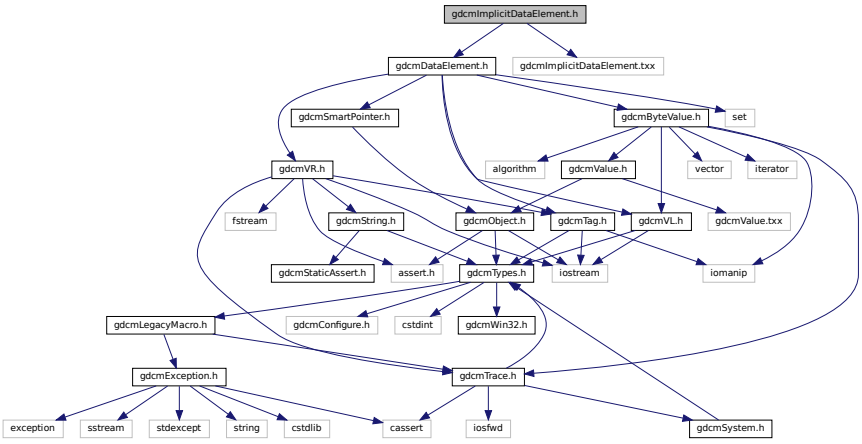
```
152     if( !bv->Read<TSwap>(is) )
153       {
154       // Fragment is incomplete, but is a itemStart, let's try to push it anyway...
155       gdcmWarningMacro( "Fragment could not be read" );
156       //bv->SetLength(is.gcount());
157       ValueField = bv;
158       ParseException pe;
159       pe.SetLastElement( *this );
160       throw pe;
161       }
162     ValueField = bv;
163     return is;
164     }
165
166
167   template <typename TSwap>
168   std::ostream &Write(std::ostream &os)const {
169     const Tag itemStart(0xfffe, 0xe000);
170     const Tag seqDelItem(0xfffe,0xe0dd);
171     if( !TagField.Write<TSwap>(os) )
172       {
173       assert(0 && "Should not happen");
174       return os;
175       }
176     assert( TagField == itemStart
177         || TagField == seqDelItem );
178     const ByteValue *bv = GetByteValue();
179     // VL
180     // The following piece of code is hard to read in order to support such broken file as:
181     // CompressedLossy.dcm
182     if( IsEmpty() )
183       {
184       //assert( bv );
185       VL zero = 0;
186       if( !zero.Write<TSwap>(os) )
187         {
188         assert(0 && "Should not happen");
189         return os;
190         }
191       }
192     else
193       {
194       assert( ValueLengthField );
195       assert( !ValueLengthField.IsUndefined() );
196       const VL actuallen = bv->ComputeLength();
197       assert( actuallen == ValueLengthField || actuallen == ValueLengthField + 1 );
198       if( !actuallen.Write<TSwap>(os) )
199         {
200         assert(0 && "Should not happen");
201         return os;
202         }
203       }
204     // Value
205     if( ValueLengthField && bv )
206       {
207       // Self
208       assert( bv );
209       assert( bv->GetLength() == ValueLengthField );
210       if( !bv->Write<TSwap>(os) )
211         {
212         assert(0 && "Should not happen");
213         return os;
214         }
215       }
216     return os;
217     }
218 };
219 //-----------------------------------------------------------------------------
220 inline std::ostream &operator«(std::ostream &os, const Fragment &val)
221 {
222   os « "Tag:   " « val.TagField;
223   os « "\tVL: " « val.ValueLengthField;
224   if( val.ValueField )
225     {
226     os « "\t" « *(val.ValueField);
227     }
228
229   return os;
230 }
231
232 } // end namespace gdcm_ns
```

```
233
234 #endif //GDCMFRAGMENT_H
```

11.147 gdcmImplicitDataElement.h File Reference

```
#include "gdcmDataElement.h"
#include "gdcmImplicitDataElement.txx"
```

Include dependency graph for gdcmImplicitDataElement.h:



Classes

- class gdcm::ImplicitDataElement

  Class to represent an Implicit VR Data Element.

Namespaces

- namespace gdcm

11.148 gdcmImplicitDataElement.h

Go to the documentation of this file.

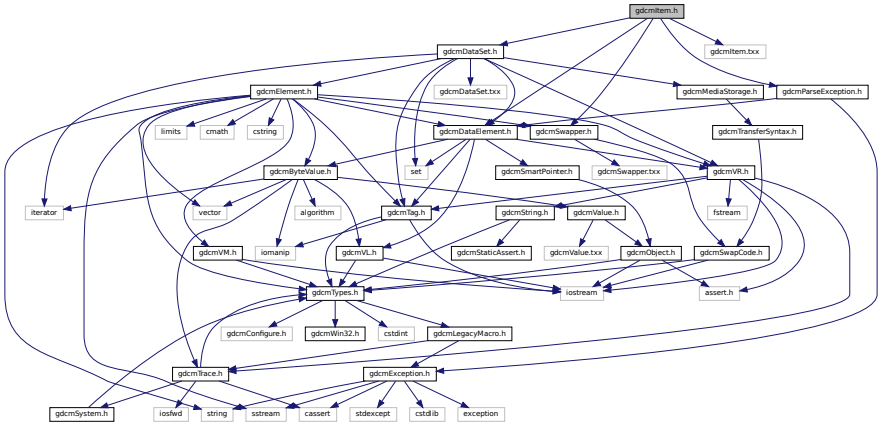
```
1 /*=========================================================================
2
3 Program:  GDCM (Grassroots DICOM). A DICOM library
4
5 Copyright (c) 2006-2011 Mathieu Malaterre
6 All rights reserved.
7 See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.
8
9 This software is distributed WITHOUT ANY WARRANTY; without even
10 the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
11 PURPOSE.  See the above copyright notice for more information.
```

```
12
13 =========================================================================*/
14 #ifndef GDCMIMPLICITDATAELEMENT_H
15 #define GDCMIMPLICITDATAELEMENT_H
16
17 #include "gdcmDataElement.h"
18
19 namespace gdcm_ns
20 {
21
26 class GDCM_EXPORT ImplicitDataElement :  public DataElement
27 {
28 public:
29   VL GetLength() const;
30
31   template <typename TSwap>
32   std::istream &Read(std::istream& is);
33
34   template <typename TSwap>
35   std::istream &ReadPreValue(std::istream& is);
36
37   template <typename TSwap>
38   std::istream &ReadValue(std::istream& is, bool readvalues = true);
39
40   template <typename TSwap>
41   std::istream &ReadWithLength(std::istream& is, VL & length, bool readvalues = true);
42
43   template <typename TSwap>
44   std::istream &ReadValueWithLength(std::istream& is, VL & length, bool readvalues = true);
45
46   template <typename TSwap>
47   const std::ostream &Write(std::ostream& os) const;
48 };
49
50 } // end namespace gdcm_ns
51
52 #include "gdcmImplicitDataElement.txx"
53
54 #endif //GDCMIMPLICITDATAELEMENT_H
```

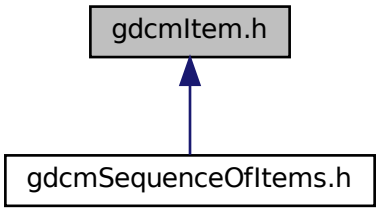
11.149 gdcmItem.h File Reference

```
#include "gdcmDataElement.h"
#include "gdcmDataSet.h"
#include "gdcmParseException.h"
#include "gdcmSwapper.h"
#include "gdcmItem.txx"
```

Include dependency graph for gdcmItem.h:



This graph shows which files directly or indirectly include this file:



Classes

- class gdcm::Item

  Class to represent an Item.

Namespaces

- namespace gdcm

Functions

- std::ostream & gdcm::operator<< (std::ostream &os, const Item &val)

11.150 gdcmItem.h

Go to the documentation of this file.

```
1 /*=========================================================================
2
3 Program:  GDCM (Grassroots DICOM). A DICOM library
4
5 Copyright (c) 2006-2011 Mathieu Malaterre
6 All rights reserved.
7 See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.
8
9 This software is distributed WITHOUT ANY WARRANTY; without even
10 the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
11 PURPOSE.  See the above copyright notice for more information.
12
13 =========================================================================*/
14
15 #ifndef GDCMITEM_H
16 #define GDCMITEM_H
17
18 #include "gdcmDataElement.h"
19 #include "gdcmDataSet.h"
20 #include "gdcmParseException.h"
21 #include "gdcmSwapper.h"
```

```
22
23 #ifdef GDCM_SUPPORT_BROKEN_IMPLEMENTATION
24 #include "gdcmByteSwapFilter.h"
25 #endif
26
27 namespace gdcm_ns
28 {
29
30 class DataSet;
45 class GDCM_EXPORT Item :  public DataElement
46 {
47 public:
48   Item() :  DataElement(Tag(0xfffe, 0xe000), 0xFFFFFFFF), NestedDataSet() {}
49   friend std::ostream& operator« (std::ostream &os, const Item &val);
50
51   void Clear() {
52     this->DataElement::Clear();
53     NestedDataSet.Clear();
54     }
55
56   template <typename TDE>
57   VL GetLength() const;
58
59   void InsertDataElement(const DataElement & de) {
60     NestedDataSet.Insert(de);
61     // Update the length
62     if( !IsUndefinedLength() )
63       {
64       assert( 0 && "InsertDataElement" );
65       //ValueLengthField += de.GetLength();
66       }
67     }
68   const DataElement& GetDataElement(const Tag& t)const
69 {
70     return NestedDataSet.GetDataElement(t);
71     }
72
73   // Completely defines it with the nested dataset
74   // destroy anything present
75   void SetNestedDataSet(const DataSet& nested)
76     {
77     NestedDataSet = nested;
78     }
79   // Return a const ref to the Nested Data Set
80   const DataSet &GetNestedDataSet()const
81 {
82     return NestedDataSet;
83     }
84   DataSet &GetNestedDataSet()
85     {
86     return NestedDataSet;
87     }
88
89   //Value const & GetValue() const { return *NestedDataSet; }
90
91   Item(Item const &val):DataElement(val)
92     {
93     NestedDataSet = val.NestedDataSet;
94     }
95
96   template <typename TDE, typename TSwap>
97   std::istream &Read(std::istream &is) {
98     // Superclass
99     {
100         DataSet &nested = NestedDataSet;
101         nested.Clear();
102         assert( nested.IsEmpty() );
103      }
104     if( !TagField.Read<TSwap>(is) )
105       {
106       throw Exception("Should not happen (item)");
107       return is;
108       }
109 #ifdef GDCM_SUPPORT_BROKEN_IMPLEMENTATION
110     // MR_Philips_Intera_SwitchIndianess_noLgtSQItem_in_trueLgtSeq.dcm
111     if( TagField == Tag(0xfeff, 0x00e0)
112       || TagField == Tag(0xfeff, 0xdde0) )
113       {
114       gdcmWarningMacro( "ByteSwaping Private SQ: " « TagField );
115       // Invert previously read TagField since wrong endianness:
116       TagField = Tag( SwapperDoOp::Swap( TagField.GetGroup() ), SwapperDoOp::Swap( TagField.GetElement() )
```

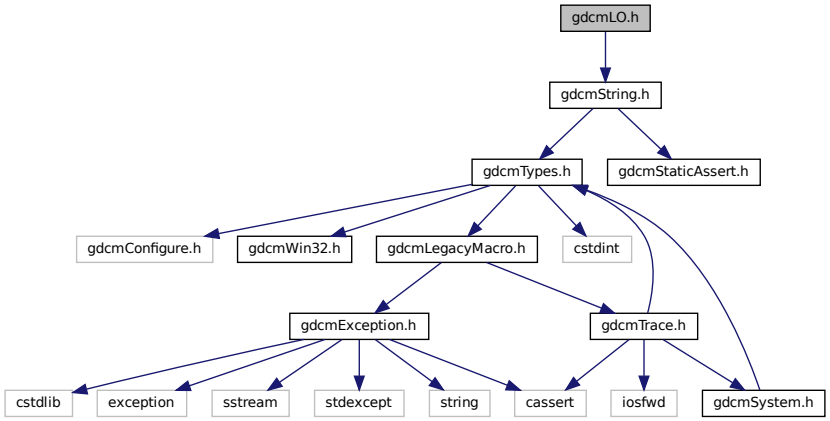
```
  );
117       assert ( TagField == Tag(0xfffe, 0xe000)
118         || TagField == Tag(0xfffe, 0xe0dd) );
119
120       if( !ValueLengthField.Read<SwapperDoOp>(is) )
121         {
122         assert(0 && "Should not happen");
123         return is;
124         }
125       // Self
126       // Some file written by GDCM 1.0 we write 0xFFFFFFFF instead of 0x0
127       if( TagField == Tag(0xfffe,0xe0dd) )
128         {
129         if( ValueLengthField )
130           {
131           gdcmErrorMacro( "ValueLengthField is not 0" );
132           }
133         }
134       //else if( ValueLengthField == 0 )
135       //  {
136       //  //assert( TagField == Tag( 0xfffe, 0xe0dd) );
137       //  if( TagField != Tag( 0xfffe, 0xe0dd) )
138       //    {
139       //    gdcmErrorMacro( "SQ: " « TagField « " has a length of 0" );
140       //    }
141       //  }
142       else if( ValueLengthField.IsUndefined() )
143         {
144         DataSet &nested = NestedDataSet;
145         nested.Clear();
146         assert( nested.IsEmpty() );
147         std::streampos start = is.tellg();
148         try
149           {
150           nested.template ReadNested<TDE,SwapperDoOp>(is);
151           ByteSwapFilter bsf(nested);
152           bsf.ByteSwap();
153           }
154         catch(ParseException &pe)
155           {
156           (void)pe;
157           // MR_Philips_Intera_PrivateSequenceExplicitVR_in_SQ_2001_e05f_item_wrong_lgt_use_NOSHADOWSEQ.dcm
158           // You have to byteswap the length but not the tag...sigh
159           gdcmWarningMacro( "Attempt to read nested Item without byteswapping the Value Length." );
160           start -= is.tellg();
161           assert( start < 0 );
162           is.seekg( start, std::ios::cur );
163           nested.Clear();
164           nested.template ReadNested<TDE,SwapperNoOp>(is);
165           ByteSwapFilter bsf(nested);
166           // Tag are read in big endian, need to byteswap them back...
167           bsf.SetByteSwapTag(true);
168           bsf.ByteSwap();
169           }
170         catch(Exception &e)
171           {
172           // MR_Philips_Intera_No_PrivateSequenceImplicitVR.dcm
173           throw e;
174           }
175         catch(...)
176           {
177           assert(0);
178           }
179         }
180       else /* if( ValueLengthField.IsUndefined() ) */
181         {
182         DataSet &nested = NestedDataSet;
183         nested.Clear();
184         assert( nested.IsEmpty() );
185         nested.template ReadWithLength<TDE,SwapperDoOp>(is, ValueLengthField);
186         ByteSwapFilter bsf(nested);
187         bsf.ByteSwap();
188         }
189       return is;
190       }
191     // http://groups.google.com/group/comp.protocols.dicom/msg/c07efcf5e759fc83
192     // Bug_Philips_ItemTag_3F3F.dcm
193     if( TagField == Tag(0x3f3f, 0x3f00) )
194       {
195       //TagField = Tag(0xfffe, 0xe000);
196       }
```

```
197 #endif
198     if( TagField != Tag(0xfffe, 0xe000) && TagField != Tag(0xfffe, 0xe0dd) )
199       {
200       gdcmDebugMacro( "Invalid Item, found tag:  " « TagField);
201       throw Exception( "Not a valid Item" );
202       }
203     assert( TagField == Tag(0xfffe, 0xe000) || TagField == Tag(0xfffe, 0xe0dd) );
204
205     if( !ValueLengthField.Read<TSwap>(is) )
206       {
207       assert(0 && "Should not happen");
208       return is;
209       }
210     // Self
211     if( TagField == Tag(0xfffe,0xe0dd) )
212       {
213       // Some file written by GDCM 1.0 were written with 0xFFFFFFFF instead of 0x0
214       if( ValueLengthField )
215         {
216         gdcmDebugMacro( "ValueLengthField is not 0 but " « ValueLengthField );
217         }
218       }
219     else if( ValueLengthField.IsUndefined() )
220       {
221       DataSet &nested = NestedDataSet;
222       nested.Clear();
223       assert( nested.IsEmpty() );
224       nested.template ReadNested<TDE,TSwap>(is);
225       }
226     else /* if( ValueLengthField.IsUndefined() ) */
227       {
228       assert( !ValueLengthField.IsUndefined() );
229       DataSet &nested = NestedDataSet;
230       nested.Clear();
231       assert( nested.IsEmpty() );
232       nested.template ReadWithLength<TDE,TSwap>(is, ValueLengthField);
233       }
234
235     return is;
236   }
237
238   template <typename TDE, typename TSwap>
239   const std::ostream &Write(std::ostream &os)const {
240 #ifdef GDCM_SUPPORT_BROKEN_IMPLEMENTATION
241     if( TagField == Tag(0x3f3f,0x3f00) && false )
242       {
243       Tag t(0xfffe, 0xe000);
244       t.Write<TSwap>(os);
245       }
246     else
247 #endif
248       {
249       assert ( TagField == Tag(0xfffe, 0xe000)
250         || TagField == Tag(0xfffe, 0xe0dd) );
251       // Not sure how this happen
252       if( TagField == Tag(0xfffe, 0xe0dd) )
253         {
254         gdcmWarningMacro( "SeqDelItem found in defined length Sequence" );
255         assert( ValueLengthField == 0 );
256         assert( NestedDataSet.Size() == 0 );
257         }
258       if( !TagField.Write<TSwap>(os) )
259         {
260         assert(0 && "Should not happen");
261         return os;
262         }
263       }
264     if( ValueLengthField.IsUndefined() )
265       {
266       if( !ValueLengthField.Write<TSwap>(os) )
267         {
268         assert(0 && "Should not happen");
269         return os;
270         }
271       }
272     else
273       {
274       const VL dummy = NestedDataSet.GetLength<TDE>();
275       assert( dummy % 2 == 0 );
276       //assert( ValueLengthField == dummy );
277       if( !dummy.Write<TSwap>(os) )
```

```
278         {
279         assert(0 && "Should not happen");
280         return os;
281         }
282       }
283     // Self
284     NestedDataSet.Write<TDE,TSwap>(os);
285     if( ValueLengthField.IsUndefined() )
286       {
287       const Tag itemDelItem(0xfffe,0xe00d);
288       itemDelItem.Write<TSwap>(os);
289       VL zero = 0;
290       zero.Write<TSwap>(os);
291       }
292
293     return os;
294   }
295
296 /*
297 There are three special SQ related Data Elements that are not ruled by the VR encoding rules conveyed
298 by the Transfer Syntax.  They shall be encoded as Implicit VR. These special Data Elements are Item
299 (FFFE,E000), Item Delimitation Item (FFFE,E00D), and Sequence Delimitation Item (FFFE,E0DD).
300 However, the Data Set within the Value Field of the Data Element Item (FFFE,E000) shall be encoded
301 according to the rules conveyed by the Transfer Syntax.
302 */
303   bool FindDataElement(const Tag &t)const {
304     return NestedDataSet.FindDataElement( t );
305   }
306
307 private:
308   /* NESTED DATA SET  a Data Set contained within a Data Element of an other Data Set.
309 * May be nested recursively.
310 * Only Data Elements with VR = SQ  may, themselves, contain Data Sets
311 */
312   DataSet NestedDataSet;
313 };
314 //-----------------------------------------------------------------------------
315 inline std::ostream& operator«(std::ostream& os, const Item &val)
316 {
317   os « val.TagField;
318   os « "\t" « val.ValueLengthField « "\n";
319   val.NestedDataSet.Print( os, "\t" );
320
321   return os;
322 }
323
324
325 } // end namespace gdcm_ns
326
327 #include "gdcmItem.txx"
328
329 #endif //GDCMITEM_H
```

11.151 gdcmLO.h File Reference

`#include "gdcmString.h"`
Include dependency graph for gdcmLO.h:



Classes

- class gdcm::LO
  LO.

Namespaces

- namespace gdcm

11.152 gdcmLO.h

Go to the documentation of this file.

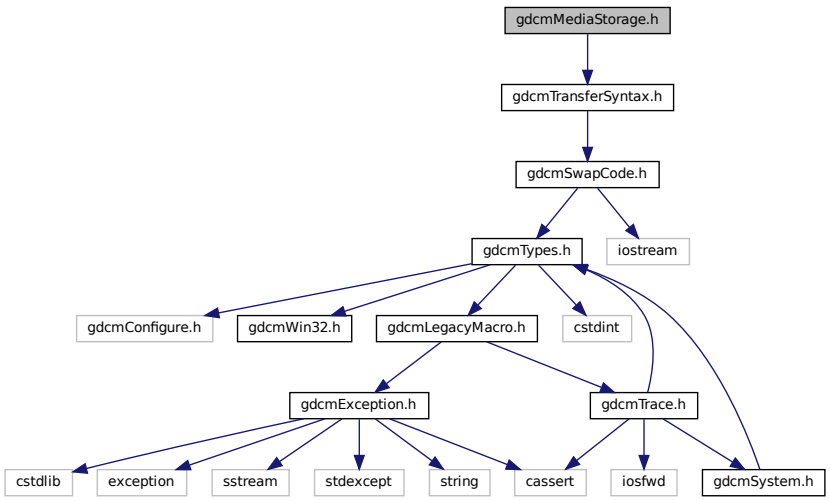
```
1 /*=========================================================================
2
3 Program:  GDCM (Grassroots DICOM). A DICOM library
4
5 Copyright (c) 2006-2011 Mathieu Malaterre
6 All rights reserved.
7 See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.
8
9 This software is distributed WITHOUT ANY WARRANTY; without even
10 the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
11 PURPOSE.  See the above copyright notice for more information.
12
13 =========================================================================*/
14 #ifndef GDCMLO_H
15 #define GDCMLO_H
16
17 #include "gdcmString.h"
18
19 namespace gdcm
```

```
20 {
21
27 class /*GDCM_EXPORT*/ LO :  public String<'\\',64> /* PLEASE do not export me */
28 {
29 public:
30   // typedef are not inherited:
31   typedef String<'\\',64> Superclass;
32   typedef Superclass::value_type             value_type;
33   typedef Superclass::pointer                pointer;
34   typedef Superclass::reference              reference;
35   typedef Superclass::const_reference        const_reference;
36   typedef Superclass::size_type              size_type;
37   typedef Superclass::difference_type        difference_type;
38   typedef Superclass::iterator               iterator;
39   typedef Superclass::const_iterator         const_iterator;
40   typedef Superclass::reverse_iterator       reverse_iterator;
41   typedef Superclass::const_reverse_iterator const_reverse_iterator;
42
43   // LO constructors.
44   LO():  Superclass() {}
45   LO(const value_type* s):  Superclass(s) {}
46   LO(const value_type* s, size_type n):  Superclass(s, n) {}
47   LO(const Superclass& s, size_type pos=0, size_type n=npos):
48     Superclass(s, pos, n) {}
49
50   bool IsValid()const {
51     if( !Superclass::IsValid() ) return false;
52     // Implementation specific:
53     return true;
54   }
55 };
56
57 } // end namespace gdcm
58
59 #endif //GDCMLO_H
```

11.153 gdcmMediaStorage.h File Reference

`#include "gdcmTransferSyntax.h"`

Include dependency graph for gdcmMediaStorage.h:



This graph shows which files directly or indirectly include this file:

Classes

- class gdcm::MediaStorage

  MediaStorage.

Namespaces

- namespace gdcm

Functions

- std::ostream & gdcm::operator<< (std::ostream &_os, const MediaStorage &ms)

11.154 gdcmMediaStorage.h

Go to the documentation of this file.

```
1 /*=========================================================================
2
3 Program:  GDCM (Grassroots DICOM). A DICOM library
4
5 Copyright (c) 2006-2011 Mathieu Malaterre
6 All rights reserved.
7 See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.
8
9 This software is distributed WITHOUT ANY WARRANTY; without even
10 the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
11 PURPOSE.  See the above copyright notice for more information.
12
13 =========================================================================*/
14 #ifndef GDCMMEDIASTORAGE_H
15 #define GDCMMEDIASTORAGE_H
16
17 #include "gdcmTransferSyntax.h"
18
19 namespace gdcm { class Tag; }
20 namespace gdcm_ns
21 {
22 #if !defined(SWIGPYTHON) && !defined(SWIGCSHARP) && !defined(SWIGJAVA) && !defined(SWIGPHP)
23 using namespace gdcm;
24 #endif
25 class DataSet;
26 class FileMetaInformation;
27 class File;
28
29 // WARNING: This class will be deprecated in the future.  There is no reason to extend this class.
30 // Please check the gdcm::UIDs class if adding new well known UID.
31
43 class GDCM_EXPORT MediaStorage
44 {
45 public:
46   typedef enum {
47     MediaStorageDirectoryStorage = 0,
48     ComputedRadiographyImageStorage,
```

```
49      DigitalXRayImageStorageForPresentation,
50      DigitalXRayImageStorageForProcessing,
51      DigitalMammographyImageStorageForPresentation,
52      DigitalMammographyImageStorageForProcessing,
53      DigitalIntraoralXrayImageStorageForPresentation,
54      DigitalIntraoralXRayImageStorageForProcessing,
55      CTImageStorage,
56      EnhancedCTImageStorage,
57      UltrasoundImageStorageRetired,
58      UltrasoundImageStorage,
59      UltrasoundMultiFrameImageStorageRetired,
60      UltrasoundMultiFrameImageStorage,
61      MRImageStorage,
62      EnhancedMRImageStorage,
63      MRSpectroscopyStorage,
64      NuclearMedicineImageStorageRetired,
65      SecondaryCaptureImageStorage,
66      MultiframeSingleBitSecondaryCaptureImageStorage,
67      MultiframeGrayscaleByteSecondaryCaptureImageStorage,
68      MultiframeGrayscaleWordSecondaryCaptureImageStorage,
69      MultiframeTrueColorSecondaryCaptureImageStorage,
70      StandaloneOverlayStorage,
71      StandaloneCurveStorage,
72      LeadECGWaveformStorage, // 12-
73      GeneralECGWaveformStorage,
74      AmbulatoryECGWaveformStorage,
75      HemodynamicWaveformStorage,
76      CardiacElectrophysiologyWaveformStorage,
77      BasicVoiceAudioWaveformStorage,
78      StandaloneModalityLUTStorage,
79      StandaloneVOILUTStorage,
80      GrayscaleSoftcopyPresentationStateStorageSOPClass,
81      XRayAngiographicImageStorage,
82      XRayRadiofluoroscopingImageStorage,
83      XRayAngiographicBiPlaneImageStorageRetired,
84      NuclearMedicineImageStorage,
85      RawDataStorage,
86      SpacialRegistrationStorage, // Spatial
87      SpacialFiducialsStorage, // Spatial..
88      PETImageStorage,
89      RTImageStorage,
90      RTDoseStorage,
91      RTStructureSetStorage,
92      RTPlanStorage,
93      CSANonImageStorage,
94      Philips3D,
95      EnhancedSR,
96      BasicTextSR,
97      HardcopyGrayscaleImageStorage,
98      ComprehensiveSR,
99      DetachedStudyManagementSOPClass,
100      EncapsulatedPDFStorage,
101      EncapsulatedCDAStorage,
102      StudyComponentManagementSOPClass,
103      DetachedVisitManagementSOPClass,
104      DetachedPatientManagementSOPClass,
105      VideoEndoscopicImageStorage,
106      GeneralElectricMagneticResonanceImageStorage,
107      GEPrivate3DModelStorage,
108      ToshibaPrivateDataStorage,
109      MammographyCADSR,
110      KeyObjectSelectionDocument,
111      HangingProtocolStorage,
112      ModalityPerformedProcedureStepSOPClass,
113      PhilipsPrivateMRSyntheticImageStorage,
114      VLPhotographicImageStorage,
115      SegmentationStorage, // "1.2.840.10008.5.1.4.1.1.66.4"
116      RTIonPlanStorage, // 1.2.840.10008.5.1.4.1.1.481.8
117      XRay3DAngiographicImageStorage, // 1.2.840.10008.5.1.4.1.1.13.1.1
118      EnhancedXAImageStorage,
119      RTIonBeamsTreatmentRecordStorage, // 1.2.840.10008.5.1.4.1.1.481.9
120      SurfaceSegmentationStorage, // "1.2.840.10008.5.1.4.1.1.66.5"
121      VLWholeSlideMicroscopyImageStorage, // 1.2.840.10008.5.1.4.1.1.77.1.6
122      RTTreatmentSummaryRecordStorage, // 1.2.840.10008.5.1.4.1.1.481.7
123      EnhancedUSVolumeStorage, // 1.2.840.10008.5.1.4.1.1.6.2
124      XRayRadiationDoseSR, // 1.2.840.10008.5.1.4.1.1.88.67
125      VLEndoscopicImageStorage, // 1.2.840.10008.5.1.4.1.1.77.1.1
126      BreastTomosynthesisImageStorage, // 1.2.840.10008.5.1.4.1.1.13.1.3
127      FujiPrivateCRImageStorage, // 1.2.392.200036.9125.1.1.2
128      OphthalmicPhotography8BitImageStorage, // 1.2.840.10008.5.1.4.1.1.77.1.5.1
129      OphthalmicTomographyImageStorage, // 1.2.840.10008.5.1.4.1.1.77.1.5.4
```

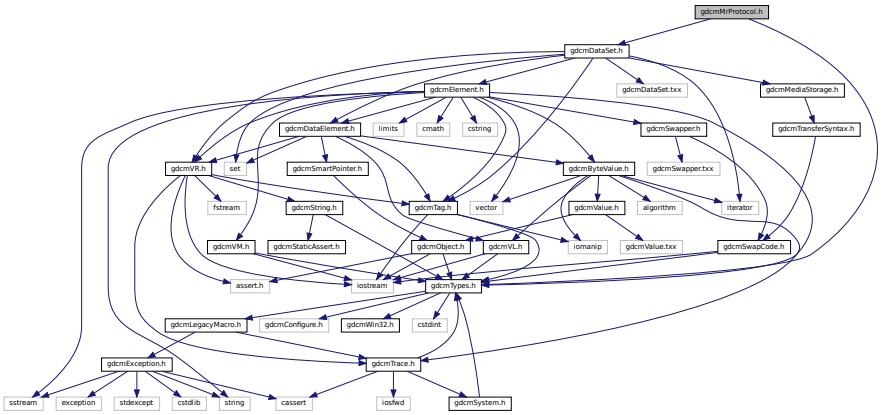
```
130     VLMicroscopicImageStorage,
131     EnhancedPETImageStorage,
132     VideoPhotographicImageStorage,
133     XRay3DCraniofacialImageStorage,
134     IVOCTForPresentation,
135     IVOCTForProcessing,
136     LegacyConvertedEnhancedCTImageStorage,
137     LegacyConvertedEnhancedMRImageStorage,
138     LegacyConvertedEnhancedPETImageStorage,
139     BreastProjectionXRayImageStorageForPresentation,
140     BreastProjectionXRayImageStorageForProcessing,
141     HardcopyColorImageStorage,
142     EnhancedMRColorImageStorage,
143     FujiPrivateMammoCRImageStorage,
144     OphthalmicPhotography16BitImageStorage,
145     VideoMicroscopicImageStorage,
146     MS_END
147   } MSType; // Media Storage Type
148
149 typedef enum {
150     NoObject = 0, // DICOMDIR
151     Video, // Most common, include image, video and volume
152     Waveform, // Isn't it simply a 1D video ?
153     Audio, // ???
154     PDF,
155     URI, // URL...
156     Segmentation, // TODO
157     ObjectEnd
158   } ObjectType;
159
161   static const char* GetMSString(MSType ts);
162
164   const char* GetString() const;
165   static MSType GetMSType(const char *str);
166
167   MediaStorage(MSType type = MS_END):MSField(type) {}
168
171   static bool IsImage(MSType ts);
172
173   operator MSType ()const { return MSField; }
174
175   const char *GetModality() const;
176   unsigned int GetModalityDimension() const;
177
178   static unsigned int GetNumberOfMSType();
179   static unsigned int GetNumberOfMSString();
180   static unsigned int GetNumberOfModality();
181
182
187   bool SetFromFile(File const &file);
188
191   bool SetFromDataSet(DataSet const &ds); // Will get the SOP Class UID
192   bool SetFromHeader(FileMetaInformation const &fmi); // Will get the Media Storage SOP Class UID
193   bool SetFromModality(DataSet const &ds);
194   void GuessFromModality(const char *modality, unsigned int dimension = 2);
195
196   friend std::ostream &operator<<(std::ostream &os, const MediaStorage &ms);
197
198   bool IsUndefined()const { return MSField == MS_END; }
199
200 protected:
201   void SetFromSourceImageSequence(DataSet const &ds);
202
203 private:
204   bool SetFromDataSetOrHeader(DataSet const &ds, const Tag & tag);
205
206   std::string GetFromDataSetOrHeader(DataSet const &ds, const Tag & tag);
207   std::string GetFromHeader(FileMetaInformation const &fmi);
208   std::string GetFromDataSet(DataSet const &ds);
209
210 private:
211   MSType MSField;
212 };
213 //-----------------------------------------------------------------------------
214 inline std::ostream &operator<<(std::ostream &_os, const MediaStorage &ms)
215 {
216   const char *msstring = MediaStorage::GetMSString(ms);
217   _os << (msstring ?  msstring :  "INVALID MEDIA STORAGE");
218   return _os;
219
220 }
```

```
221
222 } // end namespace gdcm_ns
223
224 #endif // GDCMMEDIASTORAGE_H
```

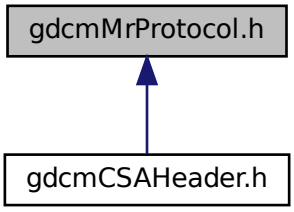
11.155 gdcmMrProtocol.h File Reference

```
#include "gdcmTypes.h"
#include "gdcmDataSet.h"
```

Include dependency graph for gdcmMrProtocol.h:



This graph shows which files directly or indirectly include this file:



Classes

- class gdcm::MrProtocol

  Class for MrProtocol.
- struct gdcm::MrProtocol::Slice
- struct gdcm::MrProtocol::SliceArray
- struct gdcm::MrProtocol::Vector3

Namespaces

- namespace gdcm

Functions

- std::ostream & gdcm::operator<< (std::ostream &os, const MrProtocol &d)

11.156 gdcmMrProtocol.h

Go to the documentation of this file.

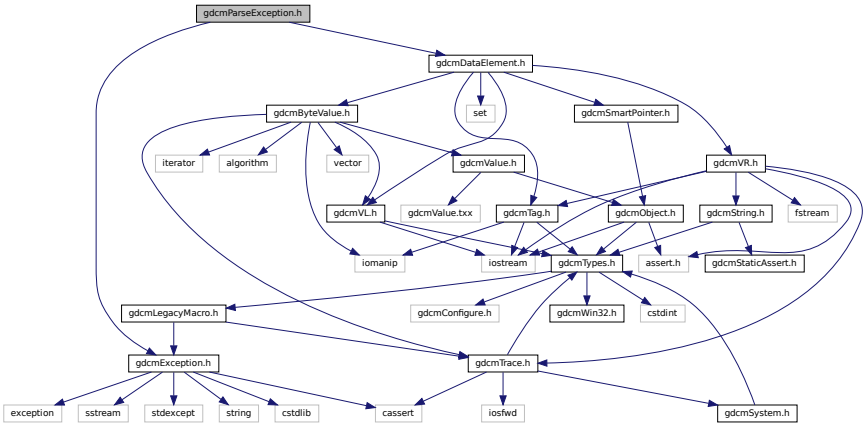
```
1 /*=========================================================================
2
3 Program:  GDCM (Grassroots DICOM). A DICOM library
4
5 Copyright (c) 2006-2011 Mathieu Malaterre
6 All rights reserved.
7 See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.
8
9 This software is distributed WITHOUT ANY WARRANTY; without even
10 the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
11 PURPOSE.  See the above copyright notice for more information.
12
13 =========================================================================*/
14 #ifndef GDCMMRPROTOCOL_H
15 #define GDCMMRPROTOCOL_H
16
17 #include "gdcmTypes.h"
18 #include "gdcmDataSet.h"
19
20 namespace gdcm
21 {
22 class ByteValue;
23 /*
24 * Everything done in this code is for the sole purpose of writing interoperable
25 * software under Sect.  1201 (f) Reverse Engineering exception of the DMCA.
26 * If you believe anything in this code violates any law or any of your rights,
27 * please contact us (gdcm-developers@lists.sourceforge.net) so that we can
28 * find a solution.
29 */
30 //-----------------------------------------------------------------------------
31
32 class DataElement;
36 class GDCM_EXPORT MrProtocol
37 {
38   friend std::ostream& operator<<(std::ostream &_os, const MrProtocol &d);
39 public :
40   MrProtocol();
41   ~MrProtocol();
42
43   bool Load( const ByteValue * bv, const char * str, int version );
44   void Print(std::ostream &os) const;
45
46   int GetVersion() const;
47
48   const char * GetMrProtocolByName(const char *name) const;
49
50   bool FindMrProtocolByName(const char *name) const;
51
52   struct Vector3
53   {
54     double dSag;
55     double dCor;
56     double dTra;
57   };
58   struct Slice
59   {
60     Vector3 Normal;
61     Vector3 Position;
62   };
```

```
63   struct SliceArray
64   {
65     std::vector< Slice > Slices;
66   };
67   bool GetSliceArray( MrProtocol::SliceArray & sa ) const;
68
69 private:
70   struct Element;
71   struct Internals;
72   Internals *Pimpl;
73 };
74 //-----------------------------------------------------------------------------
75 inline std::ostream& operator«(std::ostream &os, const MrProtocol &d)
76 {
77   d.Print( os );
78   return os;
79 }
80
81 } // end namespace gdcm
82 //-----------------------------------------------------------------------------
83 #endif //GDCMMRPROTOCOL_H
```

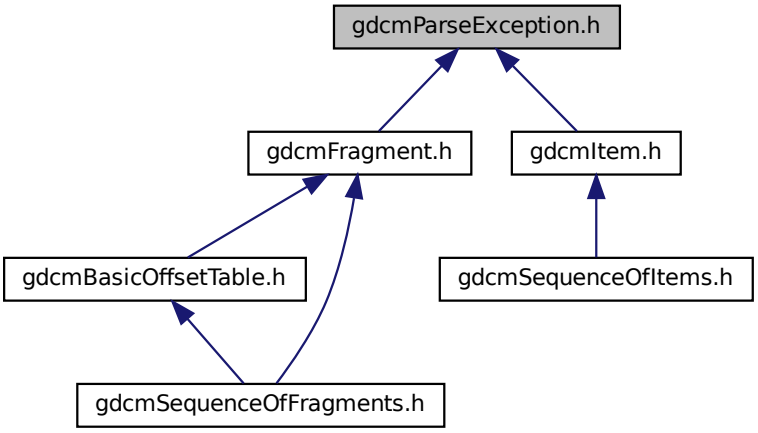
11.157 gdcmParseException.h File Reference

```
#include "gdcmException.h"
#include "gdcmDataElement.h"
```

Include dependency graph for gdcmParseException.h:



This graph shows which files directly or indirectly include this file:



Classes

- class gdcm::ParseException

  ParseException Standard exception handling object.

Namespaces

- namespace gdcm

11.158 gdcmParseException.h

Go to the documentation of this file.

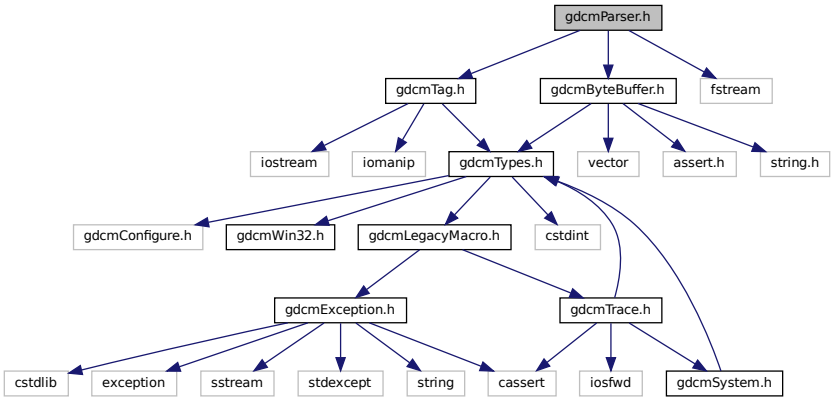
```
1 /*=========================================================================
2
3 Program:  GDCM (Grassroots DICOM). A DICOM library
4
5 Copyright (c) 2006-2011 Mathieu Malaterre
6 All rights reserved.
7 See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.
8
9 This software is distributed WITHOUT ANY WARRANTY; without even
10 the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
11 PURPOSE.  See the above copyright notice for more information.
12
13 =========================================================================*/
14 #ifndef GDCMPARSEEXCEPTION_H
15 #define GDCMPARSEEXCEPTION_H
16
17 #include "gdcmException.h"
18 #include "gdcmDataElement.h"
19
20 // Disable clang warning "dynamic exception specifications are deprecated".
21 // We need to be C++03 and C++11 compatible, and if we remove the 'throw()'
```

```
22 // specifier we'll get an error in C++03 by not matching the superclass.
23 #if defined(__clang__) && defined(__has_warning)
24 # if __has_warning("-Wdeprecated")
25 #  pragma clang diagnostic push
26 #  pragma clang diagnostic ignored "-Wdeprecated"
27 # endif
28 #endif
29
30 namespace gdcm_ns
31 {
36 class ParseException :  public Exception
37 {
38 public:
39   ParseException() = default;
40   ~ParseException() throw()override {};
41
43   ParseException &operator= ( const ParseException &orig )
44     {
45     LastElement = orig.LastElement;
46     return *this;
47     }
48   ParseException(const ParseException& orig):Exception(orig)
49     {
50     LastElement = orig.LastElement;
51     }
52
54 /*  virtual bool operator==( const ParseException &orig )
55 {
56 return true;
57 }*/
58
59 /*
60 // Multiple calls to what ??
61 const char* what() const throw()
62 {
63 static std::string strwhat;
64 std::ostringstream oswhat;
65 oswhat « File « ":" « Line « ":\n";
66 oswhat « Description;
67 strwhat = oswhat.str();
68 return strwhat.c_str();
69 }
70 */
71   void SetLastElement(DataElement& de)
72     {
73     LastElement = de;
74     }
75   const DataElement& GetLastElement()const { return LastElement; }
76
77 private:
78   // Store last parsed element before error:
79   DataElement LastElement;
80 };
81
82 } // end namespace gdcm_ns
83
84 // Undo warning suppression.
85 #if defined(__clang__) && defined(__has_warning)
86 # if __has_warning("-Wdeprecated")
87 #  pragma clang diagnostic pop
88 # endif
89 #endif
90
91 #endif
```

11.159 gdcmParser.h File Reference

```
#include "gdcmTag.h"
#include "gdcmByteBuffer.h"
#include <fstream>
```

Include dependency graph for gdcmParser.h:



Classes

- class gdcm::Parser

  Parser ala XML_Parser from expat (SAX)

Namespaces

- namespace gdcm

11.160 gdcmParser.h

Go to the documentation of this file.

```
1 /*=========================================================================
2
3 Program:  GDCM (Grassroots DICOM). A DICOM library
4
5 Copyright (c) 2006-2011 Mathieu Malaterre
6 All rights reserved.
7 See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.
8
9 This software is distributed WITHOUT ANY WARRANTY; without even
10 the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
11 PURPOSE.  See the above copyright notice for more information.
12
13 =========================================================================*/
14
15 #ifndef GDCMPARSER_H
16 #define GDCMPARSER_H
17
18 #include "gdcmTag.h"
19 #error do not use
20 #include "gdcmByteBuffer.h"
21
22 #include <fstream> // std::ifstream
23
24 namespace gdcm
25 {
32 class GDCM_EXPORT Parser /*:  private IStream*/
```

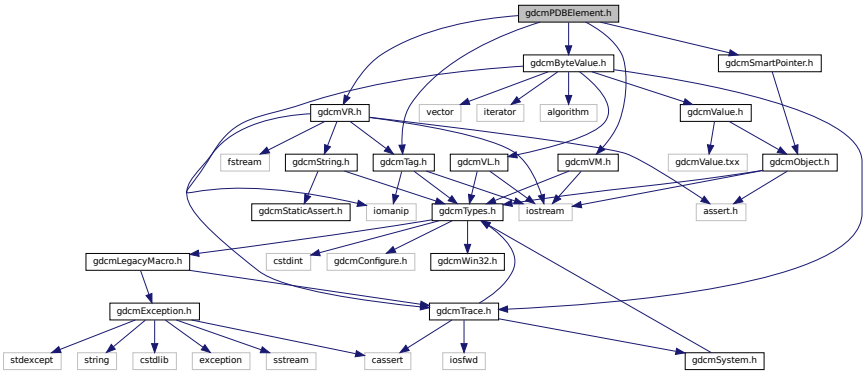
```
33 {
34 public:
35   typedef enum {
36     NoError,
37     NoMemoryError,
38     SyntaxError,
39     NoElementsError,
40     TagMismatchError,
41     DuplicateAttributeError,
42     JunkAfterDocElementError,
43     UndefinedEntityError,
44     UnexpectedStateError
45   } ErrorType;
46
47   Parser() :  UserData(0),Buffer(),ErrorCode(NoError) {}
48   ~Parser() {}
49
50   // Parse some more of the document.  The string s is a buffer containing
51   // part (or perhaps all) of the document.  The number of bytes of s that
52   // are part of the document is indicated by len.  This means that s
53   // doesn't have to be null terminated.  It also means that if len is
54   // larger than the number of bytes in the block of memory that s points
55   // at, then a memory fault is likely.  The isFinal parameter informs the
56   // parser that this is the last piece of the document.  Frequently, the
57   // last piece is empty (i.e.  len is zero.)  If a parse error occurred,
58   // it returns 0.  Otherwise it returns a non-zero value.
59   bool Parse(const char* s, int len, bool isFinal);
60
61   // Set handlers for start and end tags.  Attributes are passed to the
62   // start handler as a pointer to a vector of char pointers.  Each
63   // attribute seen in a start (or empty) tag occupies 2 consecutive places
64   // in this vector:  the attribute name followed by the attribute value.
65   // These pairs are terminated by a null pointer.
66   typedef void (*StartElementHandler) (void *userData,
67                                        const Tag &tag,
68                                        const char *atts[]);
69   typedef void (*EndElementHandler) (void *userData, const Tag &name);
70   void SetElementHandler(StartElementHandler start, EndElementHandler end);
71
72   // Return what type of error has occurred.
73   ErrorType GetErrorCode() const;
74
75   // Return a string describing the error corresponding to code.
76   // The code should be one of the enums that can be returned from
77   // GetErrorCode.
78   static const char *GetErrorString(ErrorType const &err);
79
80   // Return the byte offset of the position.
81   unsigned long GetCurrentByteIndex() const;
82
83   // Miscellaneous functions
84
85   // The functions in this section either obtain state information from
86   // the parser or can be used to dynamically set parser options.
87
88   // This sets the user data pointer that gets passed to handlers.
89   void SetUserData(void *userData);
90
91   // This returns the user data pointer that gets passed to handlers.
92   void * GetUserData() const;
93
94 protected:
95
96   // This is just like Parse, except in this case expat provides the buffer.
97   // By obtaining the buffer from expat with the GetBuffer function,
98   // the application can avoid double copying of the input.
99   bool ParseBuffer(int len, bool isFinal);
100
101    // Obtain a buffer of size len to read a piece of the document into.
102    // A NULL value is returned if expat can't allocate enough memory for
103    // this buffer.  This has to be called prior to every call to ParseBuffer.
104    char *GetBuffer(int len);
105
106    ErrorType Process();
107
108 private:
109   std::ifstream Stream;
110   void* UserData;
111   ByteBuffer Buffer;
112   ErrorType ErrorCode;
113
```

```
114   StartElementHandler StartElement;
115   EndElementHandler EndElement;
116 };
117
118 } // end namespace gdcm
119
120 #endif //GDCMPARSER_H
```

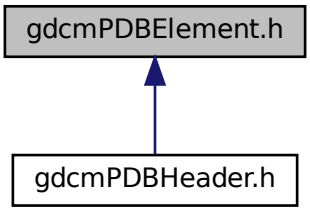
11.161 gdcmPDBElement.h File Reference

```
#include "gdcmTag.h"
#include "gdcmVM.h"
#include "gdcmVR.h"
#include "gdcmByteValue.h"
#include "gdcmSmartPointer.h"
```

Include dependency graph for gdcmPDBElement.h:



This graph shows which files directly or indirectly include this file:



Classes

- class gdcm::PDBElement

  Class to represent a PDB Element.

Namespaces

- namespace gdcm

Functions

- std::ostream & gdcm::operator<< (std::ostream &os, const PDBElement &val)

11.162 gdcmPDBElement.h

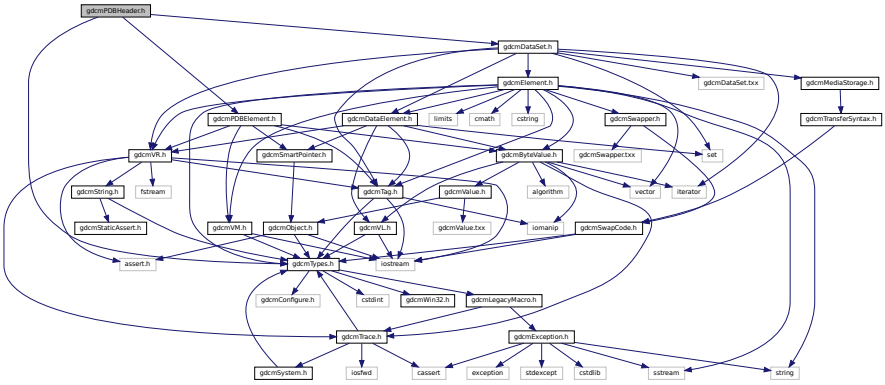
Go to the documentation of this file.

```
1 /*=========================================================================
2
3 Program:  GDCM (Grassroots DICOM). A DICOM library
4
5 Copyright (c) 2006-2011 Mathieu Malaterre
6 All rights reserved.
7 See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.
8
9 This software is distributed WITHOUT ANY WARRANTY; without even
10 the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
11 PURPOSE.  See the above copyright notice for more information.
12
13 =========================================================================*/
14 #ifndef GDCMPDBELEMENT_H
15 #define GDCMPDBELEMENT_H
16
17 #include "gdcmTag.h"
18 #include "gdcmVM.h"
19 #include "gdcmVR.h"
20 #include "gdcmByteValue.h"
21 #include "gdcmSmartPointer.h"
22
23 namespace gdcm
24 {
29 class GDCM_EXPORT PDBElement
30 {
31 public:
32   PDBElement() = default;
33
34   friend std::ostream& operator<<(std::ostream &os, const PDBElement &val);
35
37   const char *GetName()const { return NameField.c_str(); }
38   void SetName(const char *name) { NameField = name; }
39
41   const char *GetValue()const { return ValueField.c_str(); }
42   void SetValue(const char *value) { ValueField = value; }
43
44   bool operator==(const PDBElement &de)const
45 {
46     return ValueField == de.ValueField
47       && NameField == de.NameField;
48     }
49
50 protected:
51   std::string NameField;
52   std::string ValueField;
53 };
54 //-----------------------------------------------------------------------------
55 inline std::ostream& operator<<(std::ostream &os, const PDBElement &val)
56 {
57   os << val.NameField;
58   os << " \"";
59   os << val.ValueField;
60   os << "\"";
61
62   return os;
63 }
64
65 } // end namespace gdcm
66
67 #endif //GDCMPDBELEMENT_H
```

11.163 gdcmPDBHeader.h File Reference

```
#include "gdcmTypes.h"
#include "gdcmDataSet.h"
#include "gdcmPDBElement.h"
```

Include dependency graph for gdcmPDBHeader.h:



Classes

- class gdcm::PDBHeader

  Class for PDBHeader.

Namespaces

- namespace gdcm

Functions

- std::ostream & gdcm::operator<< (std::ostream &os, const PDBHeader &d)

11.164 gdcmPDBHeader.h

Go to the documentation of this file.

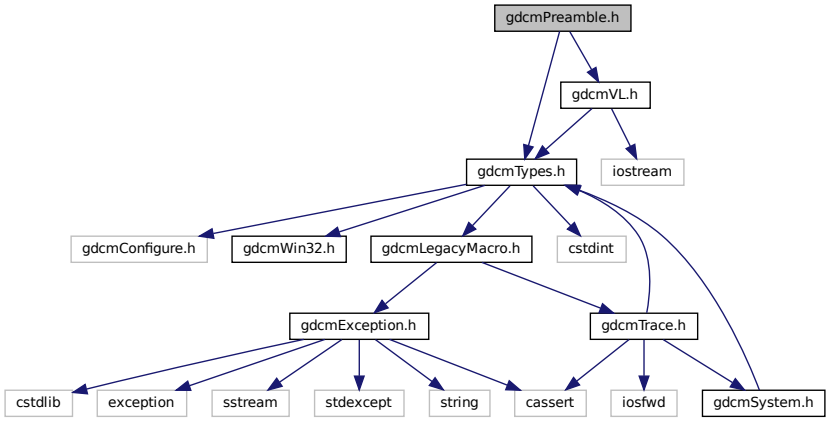
```
1 /*=========================================================================
2
3 Program:  GDCM (Grassroots DICOM). A DICOM library
4
5 Copyright (c) 2006-2011 Mathieu Malaterre
6 All rights reserved.
7 See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.
8
9 This software is distributed WITHOUT ANY WARRANTY; without even
10 the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
11 PURPOSE.  See the above copyright notice for more information.
12
```

```
13 =========================================================================*/
14 #ifndef GDCMPDBHEADER_H
15 #define GDCMPDBHEADER_H
16
17 #include "gdcmTypes.h"
18 #include "gdcmDataSet.h"
19 #include "gdcmPDBElement.h"
20
21 namespace gdcm
22 {
23
24 /*
25 * Everything done in this code is for the sole purpose of writing interoperable
26 * software under Sect.  1201 (f) Reverse Engineering exception of the DMCA.
27 * If you believe anything in this code violates any law or any of your rights,
28 * please contact us (gdcm-developers@lists.sourceforge.net) so that we can
29 * find a solution.
30 */
31 //-----------------------------------------------------------------------------
32
33 class DataElement;
34 class PrivateTag;
53 class GDCM_EXPORT PDBHeader
54 {
55   friend std::ostream& operator<<(std::ostream &_os, const PDBHeader &d);
56 public :
57   PDBHeader() = default;
58   ~PDBHeader() = default;
59
61   bool LoadFromDataElement(DataElement const &de);
62
64   void Print(std::ostream &os) const;
65
67   static const PrivateTag & GetPDBInfoTag();
68
71   const PDBElement &GetPDBElementByName(const char *name);
72
74   bool FindPDBElementByName(const char *name);
75
76 protected:
77   const PDBElement& GetPDBEEnd() const;
78
79 private:
80   int readprotocoldatablock(const char *input, size_t inputlen, bool verbose);
81   std::vector<PDBElement> InternalPDBDataSet;
82   static PDBElement PDBEEnd;
83   bool IsXML;
84   std::string xmltxt;
85 };
86 //-----------------------------------------------------------------------------
87 inline std::ostream& operator<<(std::ostream &os, const PDBHeader &d)
88 {
89   d.Print( os );
90   return os;
91 }
92
93 } // end namespace gdcm
94 //-----------------------------------------------------------------------------
95 #endif //GDCMPDBHEADER_H
```

11.165 gdcmPreamble.h File Reference

```
#include "gdcmTypes.h"
#include "gdcmVL.h"
```

Include dependency graph for gdcmPreamble.h:



This graph shows which files directly or indirectly include this file:

Classes

- class gdcm::Preamble

  DICOM Preamble (Part 10)

Namespaces

- namespace gdcm

Functions

- std::ostream & gdcm::operator<< (std::ostream &os, const Preamble &val)

11.166 gdcmPreamble.h

Go to the documentation of this file.

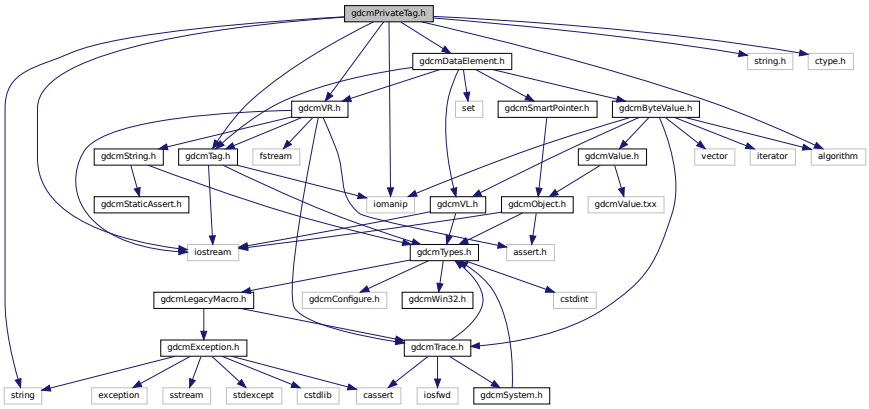
```
1 /*=========================================================================
2
3 Program:  GDCM (Grassroots DICOM). A DICOM library
4
5 Copyright (c) 2006-2011 Mathieu Malaterre
6 All rights reserved.
7 See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.
8
9 This software is distributed WITHOUT ANY WARRANTY; without even
10 the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
11 PURPOSE.  See the above copyright notice for more information.
12
13 =========================================================================*/
14 #ifndef GDCMPREAMBLE_H
15 #define GDCMPREAMBLE_H
16
17 #include "gdcmTypes.h"
18 #include "gdcmVL.h"
19
20 namespace gdcm
21 {
22
26 class GDCM_EXPORT Preamble
27 {
28 public:
29   Preamble();
30   ~Preamble();
31
32   friend std::ostream &operator<<(std::ostream &_os, const Preamble &_val);
33
35   void Clear();
36
38   void Valid();
39   void Create();
40   void Remove();
41
43   std::istream &Read(std::istream &is);
44
46   std::ostream const &Write(std::ostream &os) const;
47
49   void Print(std::ostream &os) const;
50
52   const char *GetInternal()const { return Internal; }
53
55   bool IsEmpty()const { return !Internal; }
56
58   VL GetLength()const { return 128 + 4; }
59
60   Preamble(Preamble const &)
61     {
62     Create();
63     }
64   Preamble& operator=(Preamble const &)
65     {
66     Create();
67     return *this;
68     }
69 protected:
70   //
71   bool IsValid()const {
72     // is (IsValid == true) => Internal was read
73     return true;
74   }
75
76 private:
77   char *Internal;
78 };
79 //-----------------------------------------------------------------------------
80 inline std::ostream& operator<<(std::ostream &os, const Preamble &val)
81 {
82   os << val.Internal;
83   return os;
84 }
85
86 } // end namespace gdcm
87
```

```
88 #endif //GDCMPREAMBLE_H
```

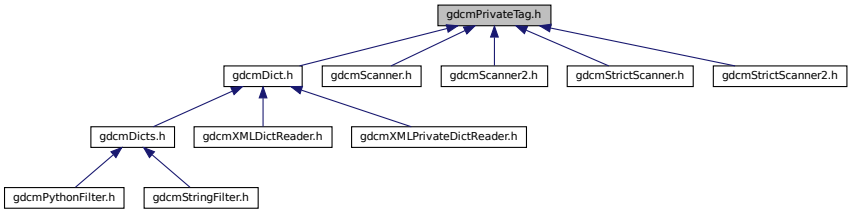
11.167 gdcmPrivateTag.h File Reference

```
#include "gdcmTag.h"
#include "gdcmVR.h"
#include "gdcmDataElement.h"
#include <iostream>
#include <iomanip>
#include <string>
#include <algorithm>
#include <string.h>
#include <ctype.h>
```

Include dependency graph for gdcmPrivateTag.h:



This graph shows which files directly or indirectly include this file:



Classes

- class gdcm::PrivateTag

  Class to represent a Private DICOM Data Element (Attribute) Tag (Group, Element, Owner)

Namespaces

- namespace gdcm

Functions

- std::ostream & gdcm::operator<< (std::ostream &os, const PrivateTag &val)

11.168 gdcmPrivateTag.h

Go to the documentation of this file.

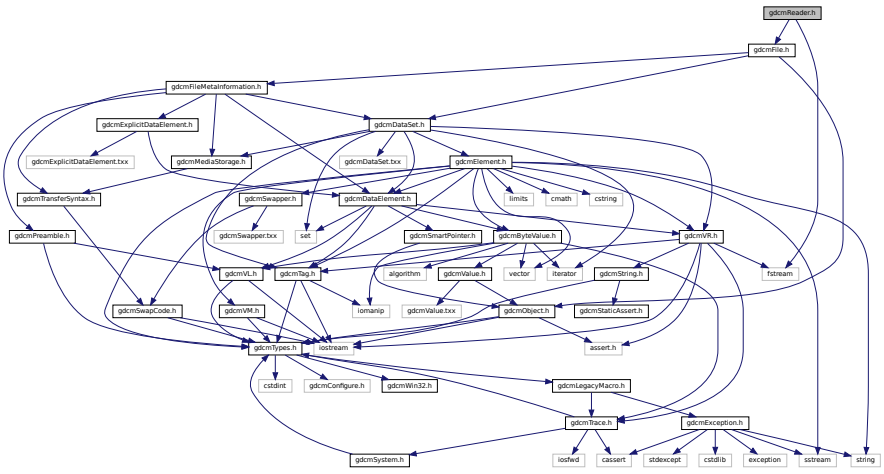
```
1 /*=========================================================================
2
3 Program:  GDCM (Grassroots DICOM). A DICOM library
4
5 Copyright (c) 2006-2011 Mathieu Malaterre
6 All rights reserved.
7 See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.
8
9 This software is distributed WITHOUT ANY WARRANTY; without even
10 the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
11 PURPOSE.  See the above copyright notice for more information.
12
13 =========================================================================*/
14 #ifndef GDCMPRIVATETAG_H
15 #define GDCMPRIVATETAG_H
16
17 #include "gdcmTag.h"
18 #include "gdcmVR.h"
19 #include "gdcmDataElement.h"
20
21 #include <iostream>
22 #include <iomanip>
23 #include <string>
24 #include <algorithm>
25
26 #include <string.h> // strlen
27 #include <ctype.h> // tolower
28
29 namespace gdcm_ns
30 {
31
37 // TODO: We could save some space since we only store 8bits for element
38 class GDCM_EXPORT PrivateTag :  public Tag
39 {
40   friend std::ostream& operator<<(std::ostream &_os, const PrivateTag &_val);
41 public:
42   PrivateTag(uint16_t group = 0, uint16_t element = 0, const char *owner = ""):Tag(group,element),Owner(owner
       ?  LOComp::Trim(owner) :  "") {
43     // truncate the high bits
44     SetElement( (uint8_t)element );
45   }
46   PrivateTag( Tag const & t, const char *owner = ""):Tag(t),Owner(owner ?  LOComp::Trim(owner) :  "") {
47     // truncate the high bits
48     SetElement( (uint8_t)t.GetElement());
49   }
50
51   const char *GetOwner()const { return Owner.c_str(); }
52   void SetOwner(const char *owner) { if(owner) Owner = LOComp::Trim(owner); }
53
54   PrivateTag &operator=(const PrivateTag &_val)
55     {
56     SetElementTag( _val.GetElementTag() );
57     Owner = _val.Owner;
58     return *this;
59     }
60
61   bool operator==(const Tag &_val)const
62 {
63     return GetElementTag() == _val.GetElementTag();
```

```
64     }
65   bool operator==(const PrivateTag &_val)const
66 {
67     return GetElementTag() == _val.GetElementTag() && Owner == _val.Owner;
68     }
69   bool operator!=(const Tag &_val)const
70 {
71     return GetElementTag() != _val.GetElementTag();
72     }
73   bool operator!=(const PrivateTag &_val)const
74 {
75     return GetElementTag() != _val.GetElementTag() || Owner != _val.Owner;
76     }
77
78   bool operator<(const PrivateTag &_val) const;
79
82   bool ReadFromCommaSeparatedString(const char *str);
83
84   DataElement GetAsDataElement() const;
85
86 private:
87   // SIEMENS MED, GEMS_PETD_01 ...
88   std::string Owner;
89 };
90
91 inline std::ostream& operator«(std::ostream &os, const PrivateTag &val)
92 {
93   //assert( !val.Owner.empty() );
94   os.setf( std::ios::right );
95   os « std::hex « '(' « std::setw( 4 ) « std::setfill( '0' )
96     « val[0] « ',' « std::setw( 2 ) « std::setfill( '0' )
97     « val[1] « ',';
98   os « val.Owner;
99   os « ')' « std::setfill( ' ' ) « std::dec;
100   return os;
101 }
102
103 } // end namespace gdcm_ns
104
105 #endif //GDCMPRIVATETAG_H
```

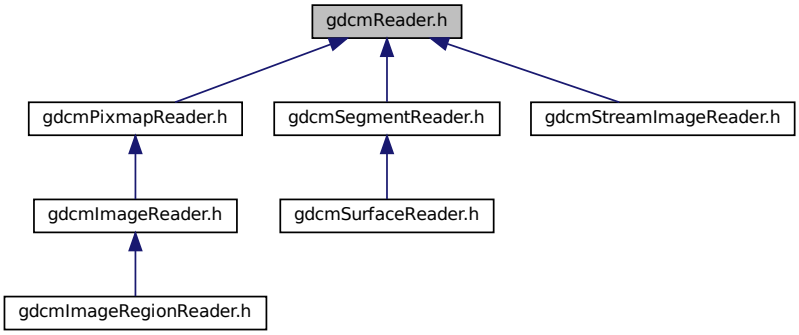
11.169 gdcmReader.h File Reference

```
#include "gdcmFile.h"
#include <fstream>
```

Include dependency graph for gdcmReader.h:



This graph shows which files directly or indirectly include this file:



Classes

- class gdcm::Reader

  Reader ala DOM (Document Object Model)

Namespaces

- namespace gdcm

11.170 gdcmReader.h

Go to the documentation of this file.

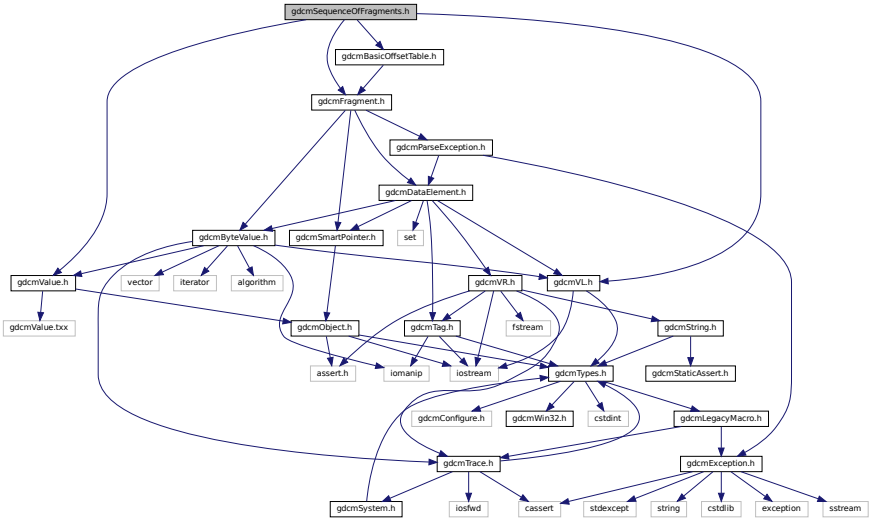
```
1 /*=========================================================================
2
3 Program:  GDCM (Grassroots DICOM). A DICOM library
4
5 Copyright (c) 2006-2011 Mathieu Malaterre
6 All rights reserved.
7 See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.
8
9 This software is distributed WITHOUT ANY WARRANTY; without even
10 the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
11 PURPOSE.  See the above copyright notice for more information.
12
13 =========================================================================*/
14 #ifndef GDCMREADER_H
15 #define GDCMREADER_H
16
17 #include "gdcmFile.h"
18
19 #include <fstream>
20
21 namespace gdcm_ns
22 {
23   class StreamImageReader;
53 class GDCM_EXPORT Reader
54 {
55 public:
56   Reader();
```

```
57   virtual ~Reader();
58
60   virtual bool Read(); // Execute()
61
64   void SetFileName(const char *filename_native);
65
67   void SetStream(std::istream &input_stream) {
68     Stream = &input_stream;
69   }
70
72   const File &GetFile()const { return *F; }
73
75   File &GetFile() { return *F; }
76
78   void SetFile(File& file) { F = &file; }
79
82   bool ReadUpToTag(const Tag & tag, std::set<Tag> const & skiptags = std::set<Tag>() );
83
85   bool ReadSelectedTags(std::set<Tag> const & tags, bool readvalues = true);
86
88   bool ReadSelectedPrivateTags(std::set<PrivateTag> const & ptags, bool readvalues = true);
89
92   bool CanRead() const;
93
96   size_t GetStreamCurrentPosition() const;
97
98 protected:
99   bool ReadPreamble();
100   bool ReadMetaInformation();
101   bool ReadDataSet();
102
103   SmartPointer<File> F;
104
105   friend class StreamImageReader; //need to be friended to be able to grab the GetStreamPtr
106
107   //this function is added for the StreamImageReader, which needs to read
108   //up to the pixel data and then stops right before reading the pixel data.
109   //it's used to get that position, so that reading can continue
110   //apace once the read function is called.
111   //so, this function gets the stream directly, and then allows for position information
112   //from the tellg function, and allows for stream/pointer manip in order
113   //to read the pixel data.    Note, of course, that reading pixel elements
114   //will still have to be subject to endianness swaps, if necessary.
115   std::istream* GetStreamPtr()const { return Stream; }
116
117 private:
118   template <typename T_Caller>
119   bool InternalReadCommon(const T_Caller &caller);
120   TransferSyntax GuessTransferSyntax();
121   std::istream *Stream;
122   std::ifstream *Ifstream;
123
124   // prevent copy/move to avoid 2 ifstream leak
125   Reader(const Reader &) = delete;
126   Reader &operator=(const Reader &) = delete;
127   Reader(const Reader &&) = delete;
128   Reader &operator=(const Reader &&) = delete;
129 };
130
137 } // end namespace gdcm_ns
138
139
140 #endif //GDCMREADER_H
```

11.171 gdcmSequenceOfFragments.h File Reference

```
#include "gdcmValue.h"
#include "gdcmVL.h"
#include "gdcmFragment.h"
#include "gdcmBasicOffsetTable.h"
```

Include dependency graph for gdcmSequenceOfFragments.h:



Classes

- class gdcm::SequenceOfFragments

  Class to represent a Sequence Of Fragments.

Namespaces

- namespace gdcm

11.172 gdcmSequenceOfFragments.h

Go to the documentation of this file.

```
1 /*=========================================================================
2
3 Program:  GDCM (Grassroots DICOM). A DICOM library
4
5 Copyright (c) 2006-2011 Mathieu Malaterre
6 All rights reserved.
7 See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.
8
9 This software is distributed WITHOUT ANY WARRANTY; without even
10 the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
11 PURPOSE.  See the above copyright notice for more information.
12
13 =========================================================================*/
14 #ifndef GDCMSEQUENCEOFFRAGMENTS_H
15 #define GDCMSEQUENCEOFFRAGMENTS_H
16
17 #include "gdcmValue.h"
18 #include "gdcmVL.h"
19 #include "gdcmFragment.h"
20 #include "gdcmBasicOffsetTable.h"
21
22 namespace gdcm_ns
```

```
23 {
24
25   // FIXME gdcmSequenceOfItems and gdcmSequenceOfFragments
26   // should be rethink (duplicate code)
31 class GDCM_EXPORT SequenceOfFragments :  public Value
32 {
33 public:
34   // Typdefs:
35   typedef std::vector<Fragment> FragmentVector;
36   typedef FragmentVector::size_type SizeType;
37   typedef FragmentVector::iterator Iterator;
38   typedef FragmentVector::const_iterator ConstIterator;
39   Iterator Begin() { return Fragments.begin(); }
40   Iterator End() { return Fragments.end(); }
41   ConstIterator Begin()const { return Fragments.begin(); }
42   ConstIterator End()const { return Fragments.end(); }
43
45   SequenceOfFragments():Table(),SequenceLengthField(0xFFFFFFFF) { }
46
48   VL GetLength()const override {
49     return SequenceLengthField;
50   }
51
53   void SetLength(VL length)override {
54     SequenceLengthField = length;
55   }
56
58   void Clear() override;
59
61   void AddFragment(Fragment const &item);
62
63   // Compute the length of all fragments (and framents only!).
64   // Basically the size of the PixelData as stored (in bytes).
65   unsigned long ComputeByteLength() const;
66
67   // Compute the length of fragments (in bytes)+ length of tag...
68   // to be used for computation of Group Length
69   VL ComputeLength() const;
70
71   // Get the buffer
72   bool GetBuffer(char *buffer, unsigned long length) const;
73   bool GetFragBuffer(unsigned int fragNb, char *buffer, unsigned long &length) const;
74
75   SizeType GetNumberOfFragments() const;
76   const Fragment& GetFragment(SizeType num) const;
77
78   // Write the buffer of each fragment (call WriteBuffer on all Fragments, which are
79   // ByteValue).  No Table information is written.
80   bool WriteBuffer(std::ostream &os) const;
81
82   const BasicOffsetTable &GetTable()const { return Table; }
83   BasicOffsetTable &GetTable() { return Table; }
84
85 template <typename TSwap>
86 std::istream& Read(std::istream &is, bool readvalues = true)
87 {
88   assert( SequenceLengthField.IsUndefined() );
89   ReadPreValue<TSwap>(is);
90   return ReadValue<TSwap>(is, readvalues);
91 }
92
93 template <typename TSwap>
94 std::istream& ReadPreValue(std::istream &is)
95 {
96   // First item is the basic offset table:
97 #if 0
98   try
99     {
100     Table.Read<TSwap>(is);
101     gdcmDebugMacro( "Table:  " « Table );
102     }
103   catch(...)
104     {
105     // throw "SIEMENS Icon thingy";
106     // Bug_Siemens_PrivateIconNoItem.dcm
107     // First thing first let's rewind
108     is.seekg(-4, std::ios::cur);
109     // FF D8 <=> Start of Image (SOI) marker
110     // FF E0 <=> APP0 Reserved for Application Use
111     if ( Table.GetTag() == Tag(0xd8ff,0xe0ff) )
112       {
```

```
113       Table = BasicOffsetTable(); // clear up stuff
114       //Table.SetByteValue( "", 0 );
115       Fragment frag;
116       if( FillFragmentWithJPEG( frag, is ) )
117         {
118         Fragments.push_back( frag );
119         }
120       return is;
121       }
122     else
123       {
124       throw "Catch me if you can";
125       //assert(0);
126       }
127     }
128 #else
129   Table.Read<TSwap>(is);
130   gdcmDebugMacro( "Table:  " « Table );
131 #endif
132   return is;
133 }
134
135 template <typename TSwap>
136 std::istream& ReadValue(std::istream &is, bool /*readvalues*/)
137 {
138   const Tag seqDelItem(0xfffe,0xe0dd);
139   // not used for now...
140   Fragment frag;
141   try
142     {
143     while( frag.Read<TSwap>(is) && frag.GetTag() != seqDelItem )
144       {
145       //gdcmDebugMacro( "Frag:  " « frag );
146       Fragments.push_back( frag );
147       }
148     assert( frag.GetTag() == seqDelItem && frag.GetVL() == 0 );
149     }
150   catch(Exception &ex)
151     {
152     (void)ex;
153 #ifdef GDCM_SUPPORT_BROKEN_IMPLEMENTATION
154     // that's ok !  In all cases the whole file was read, because
155     // Fragment::Read only fail on eof() reached 1.
156     // SIEMENS-JPEG-CorruptFrag.dcm is more difficult to deal with, we have a
157     // partial fragment, read we decide to add it anyway to the stack of
158     // fragments (eof was reached so we need to clear error bit)
159     if( frag.GetTag() == Tag(0xfffe,0xe000)  )
160       {
161       gdcmWarningMacro( "Pixel Data Fragment could be corrupted.  Use file at own risk" );
162       Fragments.push_back( frag );
163       is.clear(); // clear the error bit
164       }
165     // 2.  GENESIS_SIGNA-JPEG-CorruptFrag.dcm
166     else if ( frag.GetTag() == Tag(0xddff,0x00e0) )
167       {
168       assert( Fragments.size() == 1 );
169       const ByteValue *bv = Fragments[0].GetByteValue();
170       assert( (unsigned char)bv->GetPointer()[ bv->GetLength() - 1 ] == 0xfe );
171       // Yes this is an extra copy, this is a bug anyway, go fix YOUR code
172       Fragments[0].SetByteValue( bv->GetPointer(), bv->GetLength() - 1 );
173       gdcmWarningMacro( "JPEG Fragment length was declared with an extra byte"
174         " at the end:  stripped !" );
175       is.clear(); // clear the error bit
176       }
177     // 3.  LEICA/WSI
178     else if ( (frag.GetTag().GetGroup() == 0x00ff)
179       && ((frag.GetTag().GetElement() & 0x00ff) == 0xe0) )
180       {
181       // Looks like there is a mess with offset and odd byte array
182       // We are going first to backtrack one byte back, and then use a
183       // ReadBacktrack function which in turn may backtrack up to 10 bytes
184       // backward.  This appears to be working on a set of DICOM/WSI files from
185       // LEICA
186       gdcmWarningMacro( "Trying to fix the even-but-odd value length bug #1" );
187       assert( Fragments.size() );
188       const size_t lastf = Fragments.size() - 1;
189       const ByteValue *bv = Fragments[ lastf ].GetByteValue();
190       const char *a = bv->GetPointer();
191       gdcmAssertAlwaysMacro( (unsigned char)a[ bv->GetLength() - 1 ] == 0xfe );
192       Fragments[ lastf ].SetByteValue( bv->GetPointer(), bv->GetLength() - 1 );
193       is.seekg( -9, std::ios::cur );
```

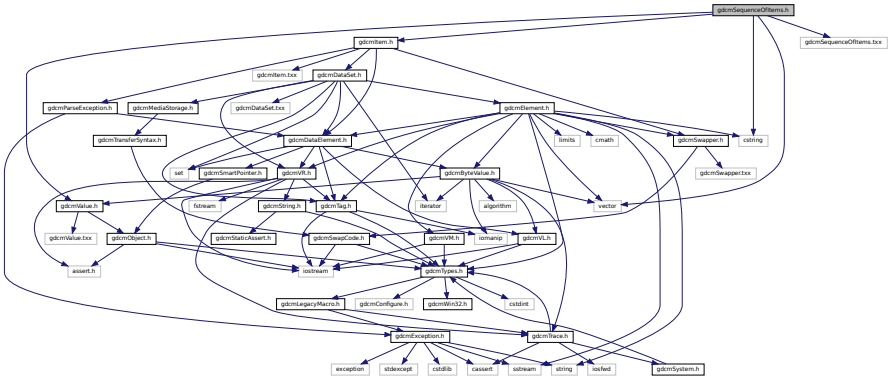
```
194       assert( is.good() );
195       while( frag.ReadBacktrack<TSwap>(is) && frag.GetTag() != seqDelItem )
196         {
197         gdcmDebugMacro( "Frag:  " « frag );
198         Fragments.push_back( frag );
199         }
200       assert( frag.GetTag() == seqDelItem && frag.GetVL() == 0 );
201       }
202     // 4.  LEICA/WSI (bis)
203     else if ( frag.GetTag().GetGroup() == 0xe000 )
204       {
205       // Looks like there is a mess with offset and odd byte array
206       // We are going first to backtrack one byte back, and then use a
207       // ReadBacktrack function which in turn may backtrack up to 10 bytes
208       // backward.  This appears to be working on a set of DICOM/WSI files from
209       // LEICA
210       gdcmWarningMacro( "Trying to fix the even-but-odd value length bug #2" );
211       assert( Fragments.size() );
212       const size_t lastf = Fragments.size() - 1;
213       const ByteValue *bv = Fragments[ lastf ].GetByteValue();
214       const char *a = bv->GetPointer();
215       gdcmAssertAlwaysMacro( (unsigned char)a[ bv->GetLength() - 2 ] == 0xfe );
216       Fragments[ lastf ].SetByteValue( bv->GetPointer(), bv->GetLength() - 2 );
217       is.seekg( -10, std::ios::cur );
218       assert( is.good() );
219       while( frag.ReadBacktrack<TSwap>(is) && frag.GetTag() != seqDelItem )
220         {
221         gdcmDebugMacro( "Frag:  " « frag );
222         Fragments.push_back( frag );
223         }
224       assert( frag.GetTag() == seqDelItem && frag.GetVL() == 0 );
225       }
226     // 5.  LEICA/WSI (ter)
227     else if ( (frag.GetTag().GetGroup() & 0x00ff) == 0x00e0
228     && (frag.GetTag().GetElement() & 0xff00) == 0x0000 )
229       {
230       // Looks like there is a mess with offset and odd byte array
231       // We are going first to backtrack one byte back, and then use a
232       // ReadBacktrack function which in turn may backtrack up to 10 bytes
233       // backward.  This appears to be working on a set of DICOM/WSI files from
234       // LEICA
235       gdcmWarningMacro( "Trying to fix the even-but-odd value length bug #3" );
236       assert( Fragments.size() );
237       const size_t lastf = Fragments.size() - 1;
238       const ByteValue *bv = Fragments[ lastf ].GetByteValue();
239       const char *a = bv->GetPointer();
240       gdcmAssertAlwaysMacro( (unsigned char)a[ bv->GetLength() - 3 ] == 0xfe );
241       Fragments[ lastf ].SetByteValue( bv->GetPointer(), bv->GetLength() - 3 );
242       is.seekg( -11, std::ios::cur );
243       assert( is.good() );
244       while( frag.ReadBacktrack<TSwap>(is) && frag.GetTag() != seqDelItem )
245         {
246         gdcmDebugMacro( "Frag:  " « frag );
247         Fragments.push_back( frag );
248         }
249       assert( frag.GetTag() == seqDelItem && frag.GetVL() == 0 );
250       }
251     else
252       {
253       // 3.  gdcm-JPEG-LossLess3a.dcm:  easy case, an extra tag was found
254       // instead of terminator (eof is the next char)
255       gdcmWarningMacro( "Reading failed at Tag:" « frag.GetTag() « " Index #"
256         « Fragments.size() « " Offset " « is.tellg() « ".  Use file at own risk."
257         « ex.what() );
258       }
259 #endif /* GDCM_SUPPORT_BROKEN_IMPLEMENTATION */
260     }
261
262   return is;
263 }
264
265 template <typename TSwap>
266 std::ostream const &Write(std::ostream &os)const
267 {
268   if( !Table.Write<TSwap>(os) )
269     {
270     assert(0 && "Should not happen");
271     return os;
272     }
273   for(ConstIterator it = Begin();it != End(); ++it)
274     {
```

```
275     it->Write<TSwap>(os);
276     }
277   // seq del item is not stored, write it !
278   const Tag seqDelItem(0xfffe,0xe0dd);
279   seqDelItem.Write<TSwap>(os);
280   VL zero = 0;
281   zero.Write<TSwap>(os);
282
283   return os;
284 }
285
286 //#if defined(SWIGPYTHON) || defined(SWIGCSHARP) || defined(SWIGJAVA)
287   // For now leave it there, this does not make sense in the C++ layer
288   // Create a new object
289   static SmartPointer<SequenceOfFragments> New()
290   {
291      return new SequenceOfFragments();
292   }
293 //#endif
294
295 protected:
296 public:
297   void Print(std::ostream &os)const override {
298     os « "SQ L= " « SequenceLengthField « "\n";
299     os « "Table:" « Table « "\n";
300     for(ConstIterator it = Begin();it != End(); ++it)
301       {
302       os « "  " « *it « "\n";
303       }
304     assert( SequenceLengthField.IsUndefined() );
305       {
306       const Tag seqDelItem(0xfffe,0xe0dd);
307       VL zero = 0;
308       os « seqDelItem;
309       os « "\t" « zero;
310       }
311   }
312   bool operator==(const Value &val)const override
313 {
314     const SequenceOfFragments &sqf = dynamic_cast<const SequenceOfFragments&>(val);
315     return Table == sqf.Table &&
316       SequenceLengthField == sqf.SequenceLengthField &&
317       Fragments == sqf.Fragments;
318     }
319
320 private:
321   BasicOffsetTable Table;
322   VL SequenceLengthField;
323   FragmentVector Fragments;
324
325 private:
326   bool FillFragmentWithJPEG( Fragment & frag, std::istream & is );
327 };
328
335 } // end namespace gdcm_ns
336
337 #endif //GDCMSEQUENCEOFFRAGMENTS_H
```

11.173 gdcmSequenceOfItems.h File Reference

```
#include "gdcmValue.h"
#include "gdcmItem.h"
#include <vector>
#include <cstring>
#include "gdcmSequenceOfItems.txx"
```

Include dependency graph for gdcmSequenceOfItems.h:



Classes

- class gdcm::SequenceOfItems

  Class to represent a Sequence Of Items.

Namespaces

- namespace gdcm

11.174 gdcmSequenceOfItems.h

Go to the documentation of this file.

```
1 /*=========================================================================
2
3 Program:  GDCM (Grassroots DICOM). A DICOM library
4
5 Copyright (c) 2006-2011 Mathieu Malaterre
6 All rights reserved.
7 See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.
8
9 This software is distributed WITHOUT ANY WARRANTY; without even
10 the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
11 PURPOSE.  See the above copyright notice for more information.
12
13 =========================================================================*/
14
15 #ifndef GDCMSEQUENCEOFITEMS_H
16 #define GDCMSEQUENCEOFITEMS_H
17
18 #include "gdcmValue.h"
19 #include "gdcmItem.h"
20
21 #include <vector>
22 #include <cstring> // strcmp
23
24 namespace gdcm_ns
25 {
26
39 class GDCM_EXPORT SequenceOfItems :  public Value
40 {
41 public:
```

```
42   // Typdefs:
43   typedef std::vector< Item > ItemVector;
44   typedef ItemVector::size_type SizeType;
45   typedef ItemVector::iterator Iterator;
46   typedef ItemVector::const_iterator ConstIterator;
47   Iterator Begin() { return Items.begin(); }
48   Iterator End() { return Items.end(); }
49   ConstIterator Begin()const { return Items.begin(); }
50   ConstIterator End()const { return Items.end(); }
51
53   SequenceOfItems():SequenceLengthField(0xFFFFFFFF) { }
54   //SequenceOfItems(VL const &vl = 0xFFFFFFFF):SequenceLengthField(vl),NType(type) { }
55
57   VL GetLength()const override { return SequenceLengthField; }
59   void SetLength(VL length)override {
60     SequenceLengthField = length;
61   }
63   void SetLengthToUndefined();
65   bool IsUndefinedLength()const {
66     return SequenceLengthField.IsUndefined();
67   }
68
69   template <typename TDE>
70   VL ComputeLength() const;
71
73   void Clear() override;
74
76   void AddItem(Item const &item);
77
79   Item & AddNewUndefinedLengthItem();
80
83   bool RemoveItemByIndex( const SizeType index );
84
85   bool IsEmpty()const { return Items.empty(); };
86   SizeType GetNumberOfItems()const {  return Items.size(); }
87   void SetNumberOfItems(SizeType n) {  Items.resize(n); }
88
89   /* WARNING: first item is #1 (see DICOM standard)
90  *  Each Item shall be implicitly assigned an ordinal position starting with the value 1 for the
91  * first Item in the Sequence, and incremented by 1 with each subsequent Item.  The last Item in the
92  * Sequence shall have an ordinal position equal to the number of Items in the Sequence.
93  */
94   const Item &GetItem(SizeType position) const;
95   Item &GetItem(SizeType position);
96
97   SequenceOfItems &operator=(const SequenceOfItems &val) {
98     SequenceLengthField = val.SequenceLengthField;
99     Items = val.Items;
100     return *this;
101     }
102
103    template <typename TDE, typename TSwap>
104    std::istream &Read(std::istream &is, bool readvalues = true)
105      {
106      (void)readvalues;
107      const Tag seqDelItem(0xfffe,0xe0dd);
108      if( SequenceLengthField.IsUndefined() )
109        {
110        Item item;
111        while( item.Read<TDE,TSwap>(is) && item.GetTag() != seqDelItem )
112          {
113          //gdcmDebugMacro( "Item:  " « item );
114          assert( item.GetTag() != seqDelItem );
115          Items.push_back( item );
116          item.Clear();
117          }
118        //assert( item.GetTag() == seqDelItem && item.GetVL() == 0 );
119        }
120      else
121        {
122        Item item;
123        VL l = 0;
124        //is.seekg( SequenceLengthField, std::ios::cur ); return is;
125        while( l != SequenceLengthField )
126          {
127          try
128            {
129            item.Read<TDE,TSwap>(is);
130            }
131          catch( Exception &ex )
132            {
```

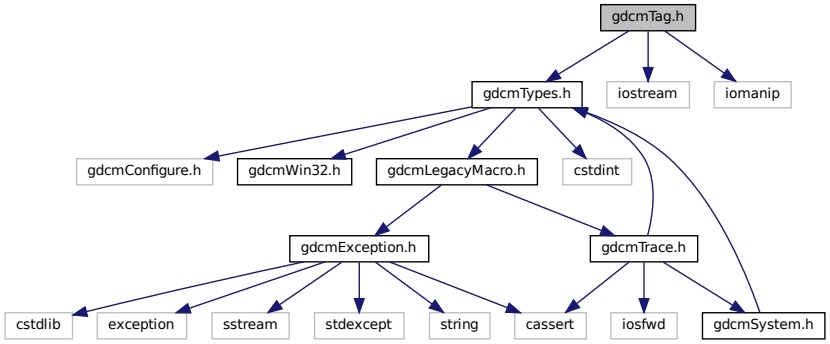
```
133           if( strcmp( ex.GetDescription(), "Changed Length" ) == 0 )
134             {
135             VL newlength = l + item.template GetLength<TDE>();
136             if( newlength > SequenceLengthField )
137               {
138               // BogugsItemAndSequenceLength.dcm
139               gdcmWarningMacro( "SQ length is wrong" );
140               SequenceLengthField = newlength;
141               }
142             }
143           else
144             {
145             throw ex;
146             }
147           }
148 #ifdef GDCM_SUPPORT_BROKEN_IMPLEMENTATION
149         if( item.GetTag() == seqDelItem )
150           {
151           gdcmWarningMacro( "SeqDelItem found in defined length Sequence.  Skipping" );
152           assert( item.GetVL() == 0 );
153           assert( item.GetNestedDataSet().Size() == 0 );
154           // we need to pay attention that the length of the Sequence of Items will be wrong
155           // this way.  Indeed by not adding this item we are changing the size of this sqi
156           }
157         else // Not a seq del item marker
158 #endif
159           {
160           // By design we never load them.  If we were to load those attribute
161           // as normal item it would become very complex to convert a sequence
162           // from defined length to undefined length with the risk to write two
163           // seq del marker
164           Items.push_back( item );
165           }
166         l += item.template GetLength<TDE>();
167         if( l > SequenceLengthField )
168           {
169           gdcmDebugMacro( "Found:  Length of Item larger than expected" );
170           throw "Length of Item larger than expected";
171           }
172         assert( l <= SequenceLengthField );
173         //std::cerr « "sqi debug len:  " « is.tellg() « " " «  l « " " «  SequenceLengthField « std::endl;
174 #ifdef GDCM_SUPPORT_BROKEN_IMPLEMENTATION
175         // MR_Philips_Intera_No_PrivateSequenceImplicitVR.dcm
176         // (0x2005, 0x1080):  for some reason computation of length fails...
177         if( SequenceLengthField == 778 && l == 774 )
178           {
179           gdcmWarningMacro( "PMS: Super bad hack" );
180           SequenceLengthField = l;
181           throw Exception( "Wrong Length" );
182           //l = SequenceLengthField;
183           }
184         // Bug_Philips_ItemTag_3F3F
185         // (0x2005, 0x1080):  Because we do not handle fully the bug at the item
186         // level we need to check here too
187         else if ( SequenceLengthField == 444 && l == 3*71 )
188           {
189           // This one is a double bug.  Item length is wrong and impact SQ length
190           gdcmWarningMacro( "PMS: Super bad hack" );
191           l = SequenceLengthField;
192           }
193 #endif
194         }
195       assert( l == SequenceLengthField );
196       }
197     return is;
198     }
199
200   template <typename TDE,typename TSwap>
201   std::ostream const &Write(std::ostream &os)const
202 {
203     typename ItemVector::const_iterator it = Items.begin();
204     for(;it != Items.end(); ++it)
205       {
206       it->Write<TDE,TSwap>(os);
207       }
208     if( SequenceLengthField.IsUndefined() )
209       {
210       // seq del item is not stored, write it !
211       const Tag seqDelItem(0xfffe,0xe0dd);
212       seqDelItem.Write<TSwap>(os);
213       VL zero = 0;
```

```
214       zero.Write<TSwap>(os);
215       }
216
217     return os;
218     }
219
220 //protected:
221   void Print(std::ostream &os)const override {
222     os « "\t(" « SequenceLengthField « ")\n";
223     ItemVector::const_iterator it =
224       Items.begin();
225     for(;it != Items.end(); ++it)
226       {
227       os « "  " « *it;
228       }
229     if( SequenceLengthField.IsUndefined() )
230       {
231       const Tag seqDelItem(0xfffe,0xe0dd);
232       VL zero = 0;
233       os « seqDelItem;
234       os « "\t" « zero;
235       }
236   }
237
238   static SmartPointer<SequenceOfItems> New()
239   {
240      return new SequenceOfItems;
241   }
242   bool FindDataElement(const Tag &t) const;
243
244   bool operator==(const Value &val)const override
245 {
246     const SequenceOfItems &sqi = dynamic_cast<const SequenceOfItems&>(val);
247     return SequenceLengthField == sqi.SequenceLengthField &&
248       Items == sqi.Items;
249     }
250
251 private:
252 public:
254   VL SequenceLengthField;
256   ItemVector Items;
257 };
258
259 } // end namespace gdcm_ns
260
261 #include "gdcmSequenceOfItems.txx"
262
263 #endif //GDCMSEQUENCEOFITEMS_H
```

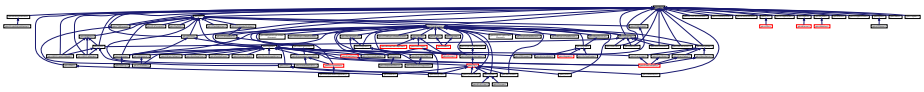
11.175 gdcmTag.h File Reference

```
#include "gdcmTypes.h"
#include <iostream>
#include <iomanip>
```

Include dependency graph for gdcmTag.h:



This graph shows which files directly or indirectly include this file:



Classes

- class gdcm::Tag

  Class to represent a DICOM Data Element (Attribute) Tag (Group, Element).

Namespaces

- namespace gdcm

Functions

- std::ostream & gdcm::operator<< (std::ostream &_os, const Tag &_val)
- std::istream & gdcm::operator>> (std::istream &_is, Tag &_val)

11.176 gdcmTag.h

Go to the documentation of this file.

```
1 /*=========================================================================
2
3 Program:  GDCM (Grassroots DICOM). A DICOM library
4
5 Copyright (c) 2006-2011 Mathieu Malaterre
6 All rights reserved.
```

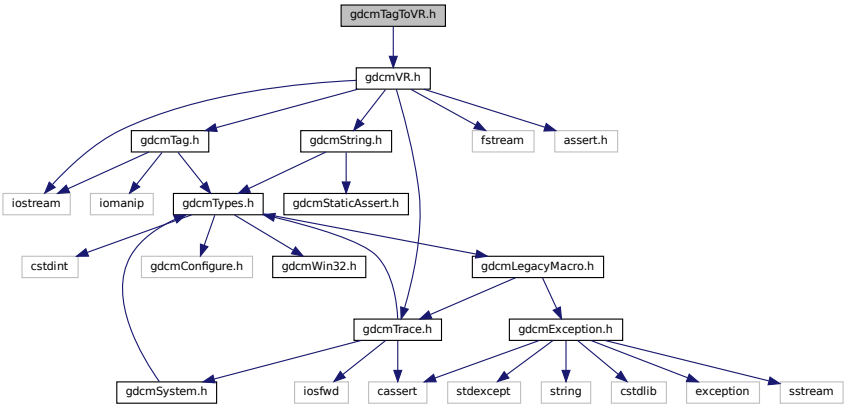
```
7 See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.
8
9 This software is distributed WITHOUT ANY WARRANTY; without even
10 the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
11 PURPOSE.  See the above copyright notice for more information.
12
13 =========================================================================*/
14 #ifndef GDCMTAG_H
15 #define GDCMTAG_H
16
17 #include "gdcmTypes.h"
18
19 #include <iostream>
20 #include <iomanip>
21
22 namespace gdcm
23 {
24
38 class GDCM_EXPORT Tag
39 {
40 public:
42   Tag(uint16_t group, uint16_t element) {
43     ElementTag.tags[0] = group; ElementTag.tags[1] = element;
44   }
47   Tag(uint32_t tag = 0) {
48     SetElementTag(tag);
49   }
50
51   friend std::ostream& operator<<(std::ostream &_os, const Tag &_val);
52   friend std::istream& operator>>(std::istream &_is, Tag &_val);
53
55   uint16_t GetGroup()const { return ElementTag.tags[0]; }
57   uint16_t GetElement()const { return ElementTag.tags[1]; }
59   void SetGroup(uint16_t group) { ElementTag.tags[0] = group; }
61   void SetElement(uint16_t element) { ElementTag.tags[1] = element; }
63   void SetElementTag(uint16_t group, uint16_t element) {
64     ElementTag.tags[0] = group; ElementTag.tags[1] = element;
65   }
66
68   uint32_t GetElementTag()const {
69 #ifndef GDCM_WORDS_BIGENDIAN
70     return (ElementTag.tag<<16) | (ElementTag.tag>>16);
71 #else
72     return ElementTag.tag;
73 #endif
74   }
75
77   void SetElementTag(uint32_t tag) {
78 #ifndef GDCM_WORDS_BIGENDIAN
79     tag = ( (tag<<16) | (tag>>16) );
80 #endif
81     ElementTag.tag = tag;
82   }
83
85   const uint16_t &operator[](const unsigned int &_id)const
86 {
87     assert(_id<2);
88     return ElementTag.tags[_id];
89     }
91   uint16_t &operator[](const unsigned int &_id)
92     {
93     assert(_id<2);
94     return ElementTag.tags[_id];
95     }
96
97   Tag &operator=(const Tag &_val)
98     {
99     ElementTag.tag = _val.ElementTag.tag;
100     return *this;
101     }
102
103   bool operator==(const Tag &_val)const
104 {
105     return ElementTag.tag == _val.ElementTag.tag;
106     }
107   bool operator!=(const Tag &_val)const
108 {
109     return ElementTag.tag != _val.ElementTag.tag;
110     }
111
114   // FIXME FIXME FIXME TODO
```

```
115   // the following is pretty dumb.  Since we have control over who is group
116   // and who is element, we should reverse them in little endian and big endian case
117   // since what we really want is fast comparison and not guarantee that group is in #0
118   // ...
119   bool operator<(const Tag &_val)const
120 {
121 #ifndef GDCM_WORDS_BIGENDIAN
122     if( ElementTag.tags[0] < _val.ElementTag.tags[0] )
123       return true;
124     if( ElementTag.tags[0] == _val.ElementTag.tags[0]
125       && ElementTag.tags[1] <  _val.ElementTag.tags[1] )
126       return true;
127     return false;
128 #else
129     // Plain comparison is enough!
130     return ( ElementTag.tag < _val.ElementTag.tag );
131 #endif
132     }
133   bool operator<=(const Tag &t2)const
134 {
135     const Tag &t1 = *this;
136     return t1 == t2 || t1 < t2;
137     }
138
139   Tag(const Tag &_val)
140     {
141     ElementTag.tag = _val.ElementTag.tag;
142     }
144   uint32_t GetLength()const { return 4; }
145
150   bool IsPublic()const { return !(ElementTag.tags[0] % 2); }
151
155   bool IsPrivate()const { return !IsPublic(); }
156
157   //-----------------------------------------------------------------------------
159   template <typename TSwap>
160   std::istream &Read(std::istream &is)
161     {
162     if( is.read(ElementTag.bytes, 4) )
163       TSwap::SwapArray(ElementTag.tags, 2);
164     return is;
165     }
166
168   template <typename TSwap>
169   const std::ostream &Write(std::ostream &os)const
170 {
171     uint16_t copy[2];
172     copy[0]= ElementTag.tags[0];
173     copy[1]= ElementTag.tags[1];
174     TSwap::SwapArray(copy, 2);
175     return os.write((char*)(&copy), 4);
176     }
177
179   Tag GetPrivateCreator()const
180 {
181     // See PS 3.5 - 7.8.1 PRIVATE DATA ELEMENT TAGS
182     // eg:  0x0123,0x1425 -> 0x0123,0x0014
183     if( IsPrivate() && !IsPrivateCreator() )
184       {
185       Tag r = *this;
186       r.SetElement( (uint16_t)(GetElement() >> 8) );
187       return r;
188       }
189     if( IsPrivateCreator() ) return *this;
190     return Tag(0x0,0x0);
191     }
193   void SetPrivateCreator(Tag const &t)
194     {
195     // See PS 3.5 - 7.8.1 PRIVATE DATA ELEMENT TAGS
196     // eg:  0x0123,0x0045 -> 0x0123,0x4567
197     assert( t.IsPrivate() /*&& t.IsPrivateCreator()*/ );
198     const uint16_t element = (uint16_t)(t.GetElement() << 8);
199     const uint16_t base = (uint16_t)(GetElement() << 8);
200     SetElement( (uint16_t)((base >> 8) + element) );
201     SetGroup( t.GetGroup() );
202     }
203
206   bool IsPrivateCreator()const
207 {
208     return IsPrivate() && (GetElement() <= 0xFF && GetElement() >= 0x10);
209     }
```

```
210
212   bool IsIllegal()const
213 {
214     // DICOM reserved those groups:
215     return GetGroup() == 0x0001 || GetGroup() == 0x0003 || GetGroup() == 0x0005 || GetGroup() == 0x0007
216     // This is a very special case, in private group, one cannot use element [0x01,0x09] ...
217 //      || (IsPrivate() && !IsPrivateCreator() && !IsGroupLength());
218       || (IsPrivate() && GetElement() > 0x0 && GetElement() < 0x10 );
219     }
220
222   bool IsGroupLength()const
223 {
224     return GetElement() == 0x0;
225     }
226
228   bool IsGroupXX(const Tag &t)const
229 {
230     if( t.GetElement() == GetElement() )
231       {
232       if( t.IsPrivate() ) return false;
233       uint16_t group = (uint16_t)((GetGroup() » 8 ) « 8);
234       return group == t.GetGroup();
235       }
236     return false;
237     }
238
244   bool ReadFromCommaSeparatedString(const char *str);
245
248   bool ReadFromContinuousString(const char *str);
249
252   std::string PrintAsContinuousString() const;
253
255   std::string PrintAsContinuousUpperCaseString() const;
256
259   bool ReadFromPipeSeparatedString(const char *str);
260
263   std::string PrintAsPipeSeparatedString() const;
264
265 private:
266   union { uint32_t tag; uint16_t tags[2]; char bytes[4]; } ElementTag;
267 };
268 //-----------------------------------------------------------------------------
269 inline std::istream& operator»(std::istream &_is, Tag &_val)
270 {
271   char c;
272   _is » c;
273   uint16_t a, b;
274   _is » std::hex » a;
275   //_is » std::hex » _val[0];
276   //_is » std::hex » _val.ElementTag.tags[0];
277   _is » c;
278   //_is » _val[1];
279   //_is » std::hex » _val.ElementTag.tags[1];
280   _is » std::hex » b;
281   _is » c;
282   _val.SetGroup( a );
283   _val.SetElement( b );
284   return _is;
285 }
286
287 inline std::ostream& operator«(std::ostream &_os, const Tag &_val)
288 {
289   _os.setf( std::ios::right);
290   _os « std::hex « '(' « std::setw( 4 ) « std::setfill( '0' )
291     « _val[0] « ',' « std::setw( 4 ) « std::setfill( '0' )
292     « _val[1] « ')' « std::setfill( ' ' ) « std::dec;
293   return _os;
294 }
295
296 } // end namespace gdcm
297
298 #endif //GDCMTAG_H
```

11.177 gdcmTagToVR.h File Reference

`#include "gdcmVR.h"`
Include dependency graph for gdcmTagToVR.h:



Namespaces

- namespace gdcm

Functions

- VR::VRType gdcm::GetVRFromTag (Tag const &tag)

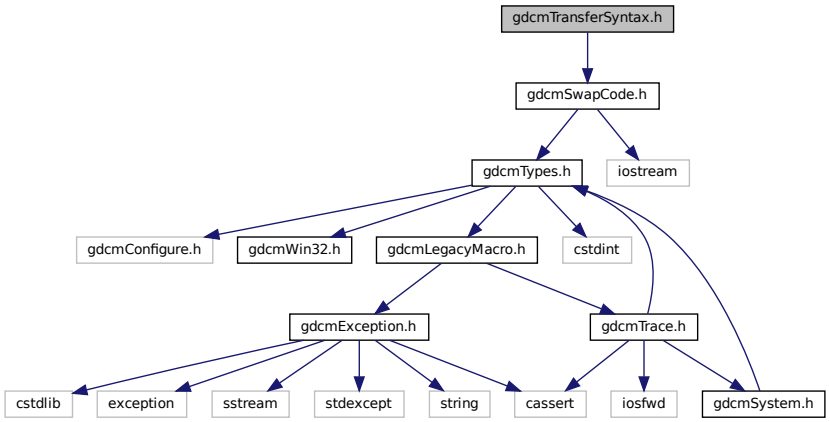
11.178 gdcmTagToVR.h

Go to the documentation of this file.

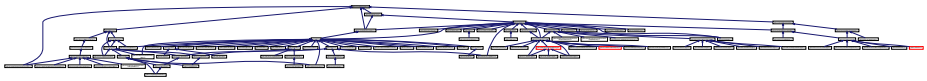
```
1 /*=========================================================================
2
3 Program:  GDCM (Grassroots DICOM). A DICOM library
4
5 Copyright (c) 2006-2011 Mathieu Malaterre
6 All rights reserved.
7 See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.
8
9 This software is distributed WITHOUT ANY WARRANTY; without even
10 the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
11 PURPOSE.  See the above copyright notice for more information.
12
13 =========================================================================*/
14 #ifndef GDCMTAGTOVR_H
15 #define GDCMTAGTOVR_H
16
17 #include "gdcmVR.h"
18
19 namespace gdcm
20 {
21   class Tag;
22   VR::VRType GetVRFromTag( Tag const & tag );
23 }
24
25 #endif // GDCMTAGTOVR_H
```

11.179 gdcmTransferSyntax.h File Reference

`#include "gdcmSwapCode.h"`
Include dependency graph for gdcmTransferSyntax.h:



This graph shows which files directly or indirectly include this file:



Classes

- class gdcm::TransferSyntax

  Class to manipulate Transfer Syntax.

Namespaces

- namespace gdcm

Functions

- std::ostream & gdcm::operator<< (std::ostream &_os, const TransferSyntax &ts)

11.180 gdcmTransferSyntax.h

Go to the documentation of this file.

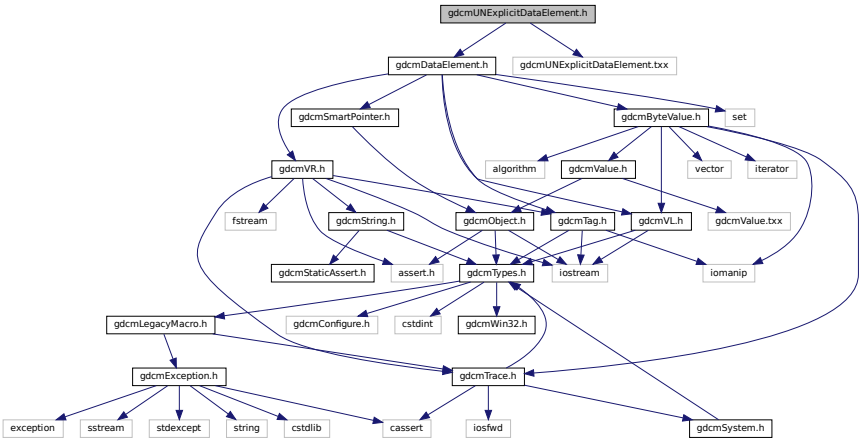
```
1 /*=========================================================================
2
3 Program:  GDCM (Grassroots DICOM). A DICOM library
4
5 Copyright (c) 2006-2011 Mathieu Malaterre
6 All rights reserved.
7 See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.
8
9 This software is distributed WITHOUT ANY WARRANTY; without even
10 the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
11 PURPOSE.  See the above copyright notice for more information.
12
13 =========================================================================*/
14 #ifndef GDCMTRANSFERSYNTAX_H
15 #define GDCMTRANSFERSYNTAX_H
16
17 #include "gdcmSwapCode.h"
18
19 namespace gdcm
20 {
21
39 class GDCM_EXPORT TransferSyntax
40 {
41 public:
42   typedef enum {
43     Unknown = 0,
44     Explicit,
45     Implicit
46   } NegociatedType;
47
48 #if 0
49   //NOT FLEXIBLE, since forces user to update lib every time new module
50   //comes out...
51   // TODO
52   typedef enum {
53     NoSpacing = 0,
54     PixelSpacing,
55     ImagerPixelSpacing,
56     PixelAspectRatio
57   } ImageSpacingType;
58   ImageSpacingType GetImageSpacing();
59 #endif
60
61   typedef enum {
62     ImplicitVRLittleEndian = 0,
63     ImplicitVRBigEndianPrivateGE,
64     ExplicitVRLittleEndian,
65     DeflatedExplicitVRLittleEndian,
66     ExplicitVRBigEndian,
67     JPEGBaselineProcess1,
68     JPEGExtendedProcess2_4,
69     JPEGExtendedProcess3_5,
70     JPEGSpectralSelectionProcess6_8,
71     JPEGFullProgressionProcess10_12,
72     JPEGLosslessProcess14,
73     JPEGLosslessProcess14_1,
74     JPEGLSLossless,
75     JPEGLSNearLossless,
76     JPEG2000Lossless,
77     JPEG2000,
78     JPEG2000Part2Lossless,
79     JPEG2000Part2,
80     RLELossless,
81     MPEG2MainProfile,
82     ImplicitVRBigEndianACRNEMA,
83     WeirdPapyrus,
84     CT_private_ELE,
85     JPIPReferenced,
86     MPEG2MainProfileHighLevel,
87     MPEG4AVCH264HighProfileLevel4_1,
88     MPEG4AVCH264BDcompatibleHighProfileLevel4_1,
89     TS_END
90   } TSType;
91
92   // Return the string as written in the official DICOM dict from
93   // a custom enum type
```

```
94   static const char* GetTSString(TSType ts);
95   static TSType GetTSType(const char *str);
96
97   NegociatedType GetNegociatedType() const;
98
102   SwapCode GetSwapCode() const;
103
104   bool IsValid()const { return TSField != TS_END; }
105
106   operator TSType ()const { return TSField; }
107
108   // FIXME: ImplicitVRLittleEndian used to be the default, but nowadays
109   // this is rather the ExplicitVRLittleEndian instead...should be change the default ?
110   TransferSyntax(TSType type = ImplicitVRLittleEndian):TSField(type) {}
111
112   // return if dataset is encoded or not (Deflate Explicit VR)
113   bool IsEncoded() const;
114
115   bool IsImplicit() const;
116   bool IsExplicit() const;
117
118   bool IsEncapsulated() const;
119
121   bool IsLossy() const;
123   bool IsLossless() const;
125   bool CanStoreLossy() const;
126
127   const char *GetString()const { return TransferSyntax::GetTSString(TSField); }
128
129   friend std::ostream &operator<<(std::ostream &os, const TransferSyntax &ts);
130 private:
131   // DO NOT EXPOSE the following.  Internal details of TransferSyntax
132 bool IsImplicit(TSType ts) const;
133 bool IsExplicit(TSType ts) const;
134 bool IsLittleEndian(TSType ts) const;
135 bool IsBigEndian(TSType ts) const;
136
137   TSType TSField;
138 };
139 //-----------------------------------------------------------------------------
140 inline std::ostream &operator<<(std::ostream &_os, const TransferSyntax &ts)
141 {
142   _os << TransferSyntax::GetTSString(ts);
143   return _os;
144
145 }
146
147 } // end namespace gdcm
148
149 #endif //GDCMTRANSFERSYNTAX_H
```

11.181 gdcmUNExplicitDataElement.h File Reference

```
#include "gdcmDataElement.h"
#include "gdcmUNExplicitDataElement.txx"
```

Include dependency graph for gdcmUNExplicitDataElement.h:



Classes

- class gdcm::UNExplicitDataElement

  Class to read/write a DataElement as UNExplicit Data Element.

Namespaces

- namespace gdcm

11.182 gdcmUNExplicitDataElement.h

Go to the documentation of this file.

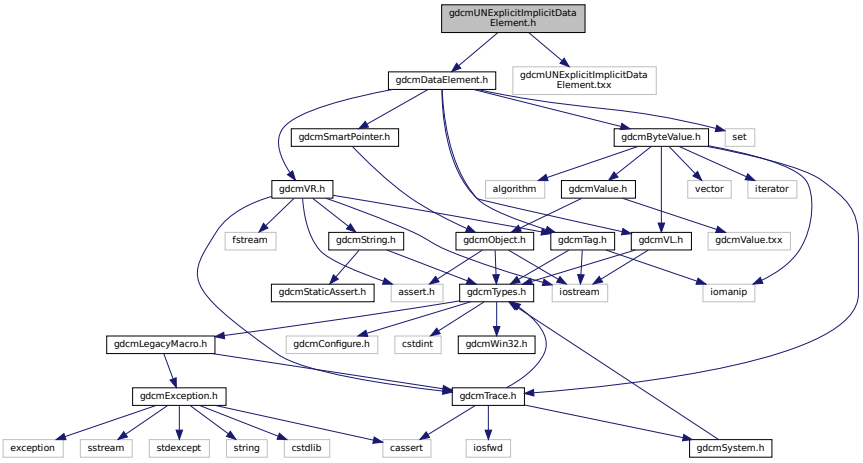
```
1 /*=========================================================================
2
3 Program:  GDCM (Grassroots DICOM). A DICOM library
4
5 Copyright (c) 2006-2011 Mathieu Malaterre
6 All rights reserved.
7 See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.
8
9 This software is distributed WITHOUT ANY WARRANTY; without even
10 the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
11 PURPOSE.  See the above copyright notice for more information.
12
13 =========================================================================*/
14 #ifndef GDCMUNEXPLICITDATAELEMENT_H
15 #define GDCMUNEXPLICITDATAELEMENT_H
16
17 #include "gdcmDataElement.h"
18
19 namespace gdcm
20 {
21 // Data Element (UNExplicit)
26 class GDCM_EXPORT UNExplicitDataElement :  public DataElement
27 {
28 public:
29   VL GetLength() const;
```

```
30
31   template <typename TSwap>
32   std::istream &Read(std::istream &is);
33
34   template <typename TSwap>
35   std::istream &ReadPreValue(std::istream &is);
36
37   template <typename TSwap>
38   std::istream &ReadValue(std::istream &is, bool readvalues = true);
39
40   template <typename TSwap>
41   std::istream &ReadWithLength(std::istream &is, VL & length);
42
43   // PURPOSELY do not provide an implementation for writing !
44   //template <typename TSwap>
45   //const std::ostream &Write(std::ostream &os) const;
46 };
47
48 } // end namespace gdcm
49
50 #include "gdcmUNExplicitDataElement.txx"
51
52 #endif //GDCMUNEXPLICITDATAELEMENT_H
```

11.183 gdcmUNExplicitImplicitDataElement.h File Reference

```
#include "gdcmDataElement.h"
#include "gdcmUNExplicitImplicitDataElement.txx"
```

Include dependency graph for gdcmUNExplicitImplicitDataElement.h:



Classes

- class gdcm::UNExplicitImplicitDataElement

  Class to read/write a DataElement as ExplicitImplicit Data Element.

Namespaces

- namespace gdcm

11.184 gdcmUNExplicitImplicitDataElement.h

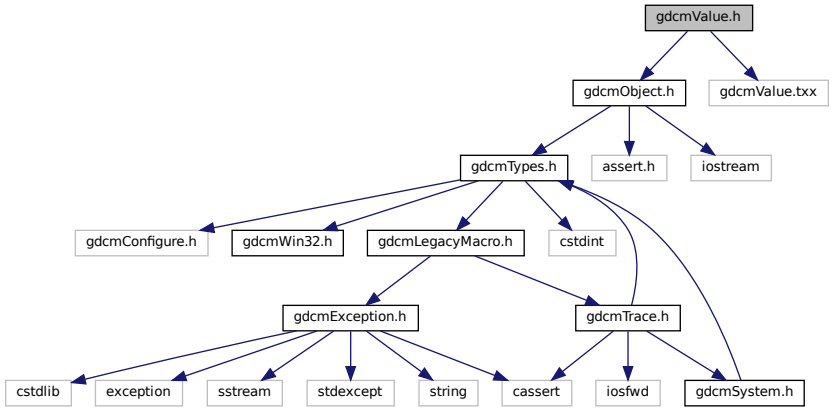
Go to the documentation of this file.

```
1 /*=========================================================================
2
3 Program:  GDCM (Grassroots DICOM). A DICOM library
4
5 Copyright (c) 2006-2011 Mathieu Malaterre
6 All rights reserved.
7 See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.
8
9 This software is distributed WITHOUT ANY WARRANTY; without even
10 the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
11 PURPOSE.  See the above copyright notice for more information.
12
13 =========================================================================*/
14 #ifndef GDCMUNEXPLICITIMPLICITDATAELEMENT_H
15 #define GDCMUNEXPLICITIMPLICITDATAELEMENT_H
16
17 #include "gdcmDataElement.h"
18
19 namespace gdcm
20 {
21 // Data Element (ExplicitImplicit)
32 class GDCM_EXPORT UNExplicitImplicitDataElement :  public DataElement
33 {
34 public:
35   VL GetLength() const;
36
37   template <typename TSwap>
38   std::istream &Read(std::istream &is);
39
40   template <typename TSwap>
41   std::istream &ReadPreValue(std::istream &is);
42
43   template <typename TSwap>
44   std::istream &ReadValue(std::istream &is);
45
46   // PURPOSELY do not provide an implementation for writing !
47   //template <typename TSwap>
48   //const std::ostream &Write(std::ostream &os) const;
49 };
50
51 } // end namespace gdcm
52
53 #include "gdcmUNExplicitImplicitDataElement.txx"
54
55 #endif //GDCMUNEXPLICITIMPLICITDATAELEMENT_H
```

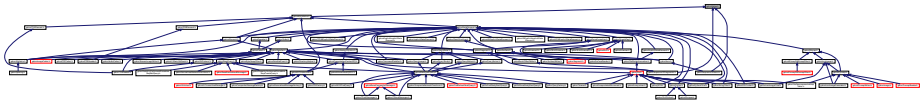
11.185 gdcmValue.h File Reference

```
#include "gdcmObject.h"
#include "gdcmValue.txx"
```

Include dependency graph for gdcmValue.h:



This graph shows which files directly or indirectly include this file:



Classes

- class gdcm::Value

  Class to represent the value of a Data Element.

Namespaces

- namespace gdcm

11.186 gdcmValue.h

Go to the documentation of this file.

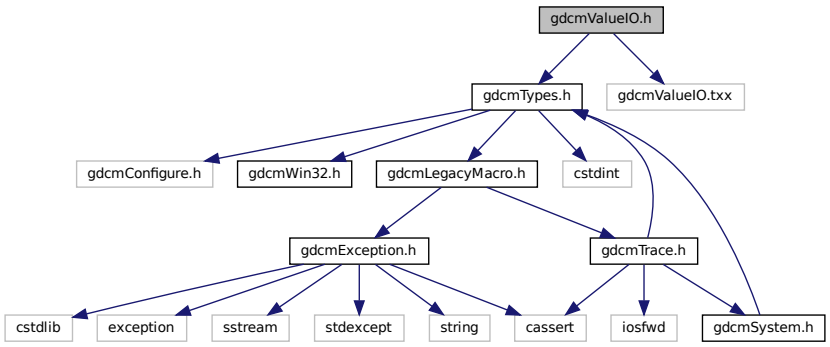
```
1 /*=========================================================================
2
3 Program:  GDCM (Grassroots DICOM). A DICOM library
4
5 Copyright (c) 2006-2011 Mathieu Malaterre
6 All rights reserved.
7 See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.
8
9 This software is distributed WITHOUT ANY WARRANTY; without even
10 the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
11 PURPOSE.  See the above copyright notice for more information.
12
```

```
13 =========================================================================*/
14 #ifndef GDCMVALUE_H
15 #define GDCMVALUE_H
16
17 #include "gdcmObject.h"
18
19 namespace gdcm { class VL; }
20 namespace gdcm_ns
21 {
22 #if !defined(SWIGPYTHON) && !defined(SWIGCSHARP) && !defined(SWIGJAVA) && !defined(SWIGPHP)
23 using namespace gdcm;
24 #endif
31 class GDCM_EXPORT Value :  public Object
32 {
33 public:
34   Value() = default;
35   ~Value() override = default;
36
37   virtual VL GetLength() const = 0;
38   virtual void SetLength(VL l) = 0;
39
40   virtual void Clear() = 0;
41
42   virtual bool operator==(const Value &val) const = 0;
43
44 protected:
45   friend class DataElement;
46   virtual void SetLengthOnly(VL l);
47 };
48
49 } // end namespace gdcm_ns
50
51 #include "gdcmValue.txx"
52
53 #endif //GDCMVALUE_H
```

11.187 gdcmValueIO.h File Reference

```
#include "gdcmTypes.h"
#include "gdcmValueIO.txx"
```

Include dependency graph for gdcmValueIO.h:



Classes

- class gdcm::ValueIO< TDE, TSwap, TType >

  Class to dispatch template calls.

Namespaces

- namespace gdcm

11.188 gdcmValueIO.h

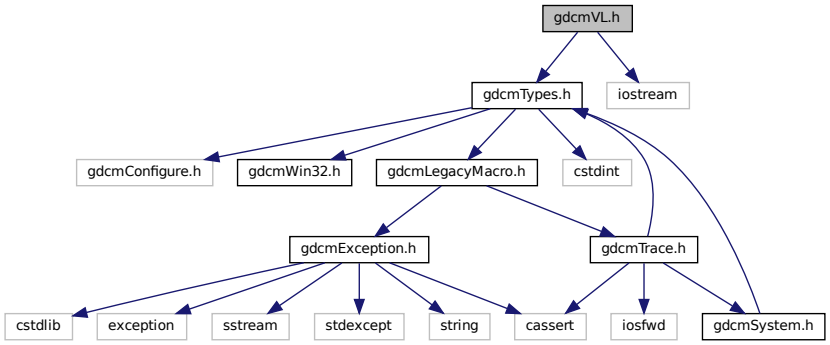
Go to the documentation of this file.

```
1 /*=========================================================================
2
3 Program:  GDCM (Grassroots DICOM). A DICOM library
4
5 Copyright (c) 2006-2011 Mathieu Malaterre
6 All rights reserved.
7 See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.
8
9 This software is distributed WITHOUT ANY WARRANTY; without even
10 the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
11 PURPOSE.  See the above copyright notice for more information.
12
13 =========================================================================*/
14 #ifndef GDCMVALUEIO_H
15 #define GDCMVALUEIO_H
16
17 #include "gdcmTypes.h"
18
19 namespace gdcm_ns
20 {
24 template <typename TDE, typename TSwap, typename TType=uint8_t>
25 class /*GDCM_EXPORT*/ ValueIO
26 {
27 public:
28   static std::istream &Read(std::istream &is, Value& v, bool readvalues);
29
30   static const std::ostream &Write(std::ostream &os, const Value& v);
31 };
32
33 } // end namespace gdcm_ns
34
35 #include "gdcmValueIO.txx"
36
37 #endif //GDCMVALUEIO_H
```

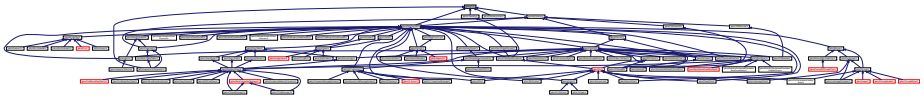
11.189 gdcmVL.h File Reference

```
#include "gdcmTypes.h"
#include <iostream>
```

Include dependency graph for gdcmVL.h:



This graph shows which files directly or indirectly include this file:



Classes

- class gdcm::VL

  Value Length.

Namespaces

- namespace gdcm

Functions

- std::ostream & gdcm::operator<< (std::ostream &os, const VL &val)

11.190 gdcmVL.h

Go to the documentation of this file.

```
1 /*=========================================================================
2
3 Program:  GDCM (Grassroots DICOM). A DICOM library
4
5 Copyright (c) 2006-2011 Mathieu Malaterre
6 All rights reserved.
7 See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.
8
9 This software is distributed WITHOUT ANY WARRANTY; without even
10 the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
11 PURPOSE.  See the above copyright notice for more information.
12
13 =========================================================================*/
14 #ifndef GDCMVL_H
15 #define GDCMVL_H
16
17 #include "gdcmTypes.h"
18
19 #include <iostream>
20
21 namespace gdcm
22 {
23
29 class GDCM_EXPORT VL
30 {
31 public:
32   typedef uint32_t Type;
33   VL(uint32_t vl = 0) :  ValueLength(vl) { }
34
35   // FIXME: ugly
36   static uint32_t GetVL32Max() { return 0xFFFFFFFF; }
37   static uint16_t GetVL16Max() { return 0xFFFF; }
38
39   bool IsUndefined()const {
40     return ValueLength == 0xFFFFFFFF;
41   }
```

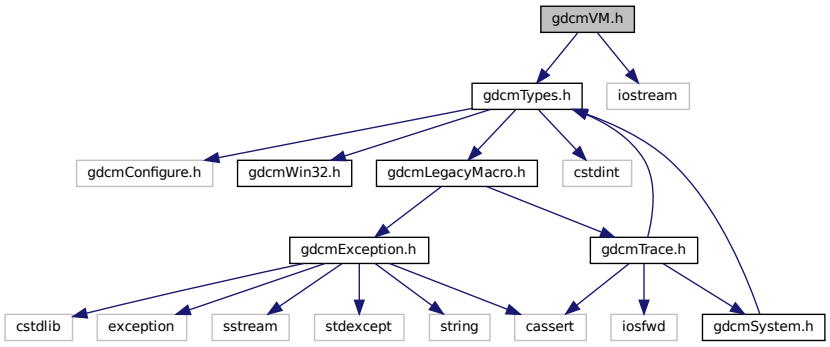
```
42   void SetToUndefined() {
43     ValueLength = 0xFFFFFFFF;
44   }
45
47   bool IsOdd()const {
48     return !IsUndefined() && ValueLength % 2;
49   }
50
52   VL& operator+=(VL const &vl) {
53     ValueLength += vl.ValueLength;
54     return *this;
55   }
56   VL& operator++() {
57     ++ValueLength;
58     return *this;
59   }
60   VL operator++(int) {
61     uint32_t tmp(ValueLength);
62     ++ValueLength;
63     return tmp;
64   }
65
66   operator uint32_t ()const { return ValueLength; }
67
68   VL GetLength()const {
69     // VL cannot know it's length...well in implicit yes...
70     // TODO: need to check we cannot call this function from an Explicit element
71     return 4;
72   }
73
74   friend std::ostream& operator<<(std::ostream& os, const VL& vl);
75
76   // PURPOSELY not implemented (could not differentiate 16bits vs 32bits VL)
77   //friend std::istream& operator>>(std::istream& is, VL& n);
78
79   template <typename TSwap>
80   std::istream &Read(std::istream &is)
81     {
82     is.read((char*)(&ValueLength), sizeof(uint32_t));
83     TSwap::SwapArray(&ValueLength,1);
84     return is;
85     }
86
87   template <typename TSwap>
88   std::istream &Read16(std::istream &is)
89     {
90     uint16_t copy;
91     is.read((char*)(&copy), sizeof(uint16_t));
92     TSwap::SwapArray(&copy,1);
93     ValueLength = copy;
94     assert( ValueLength <=  65535 /*UINT16_MAX*/ ); // ??  doh !
95     return is;
96     }
97
98   template <typename TSwap>
99   const std::ostream &Write(std::ostream &os)const
100  {
101     uint32_t copy = ValueLength;
102     if( IsOdd() )
103       {
104       ++copy;
105       }
106     TSwap::SwapArray(&copy,1);
107     return os.write((char*)(&copy), sizeof(uint32_t));
108     }
109
110   template <typename TSwap>
111   const std::ostream &Write16(std::ostream &os)const
112  {
113     assert( ValueLength <=   65535 /*UINT16_MAX*/ );
114     uint16_t copy = (uint16_t)ValueLength;
115     if( IsOdd() )
116       {
117       ++copy;
118       }
119     TSwap::SwapArray(&copy,1);
120     return os.write((char*)(&copy), sizeof(uint16_t));
121     }
122
123  private:
124    uint32_t ValueLength;
```

```
125 };
126 //-----------------------------------------------------------------------------
127 inline std::ostream& operator«(std::ostream& os, const VL& val)
128 {
129   os « /*std::hex «*/ val.ValueLength;
130   return os;
131 }
132
133 } // end namespace gdcm
134
135 #endif //GDCMVL_H
```

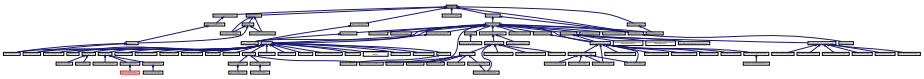
11.191 gdcmVM.h File Reference

```
#include "gdcmTypes.h"
#include <iostream>
```

Include dependency graph for gdcmVM.h:



This graph shows which files directly or indirectly include this file:



Classes

- class gdcm::VM

  Value *Multiplicity Looking at the DICOMV3 dict only there is very few cases: 1 2 3 4 5 6 8 16 24 1-2 1-3 1-8 1-32 1-99 1-n 2-2n 2-n 3-3n 3-n.*

Namespaces

- namespace gdcm

Macros

- #define TYPETOLENGTH(type, length)

Functions

- std::ostream & gdcm::operator<< (std::ostream &_os, const VM &_val)

11.191.1 Macro Definition Documentation

11.191.1.1 TYPETOLENGTH

```
#define TYPETOLENGTH(
               type,
               length )
```

Value:

```
  template<> struct VMToLength<VM::type> \
  { enum { Length = length }; };
```

11.192 gdcmVM.h

Go to the documentation of this file.

```
1 /*=========================================================================
2
3 Program:  GDCM (Grassroots DICOM). A DICOM library
4
5 Copyright (c) 2006-2011 Mathieu Malaterre
6 All rights reserved.
7 See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.
8
9 This software is distributed WITHOUT ANY WARRANTY; without even
10 the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
11 PURPOSE.  See the above copyright notice for more information.
12
13 =========================================================================*/
14 #ifndef GDCMVM_H
15 #define GDCMVM_H
16
17 #include "gdcmTypes.h"
18 #include <iostream>
19
20 namespace gdcm
21 {
22
67 class GDCM_EXPORT VM
68 {
69 public:
70   typedef enum {
71     VM0 = 0, // aka the invalid VM
72     VM1 = 1,
73     VM2 = 2,
74     VM3 = 4,
75     VM4 = 8,
76     VM5 = 16,
77     VM6 = 32,
```

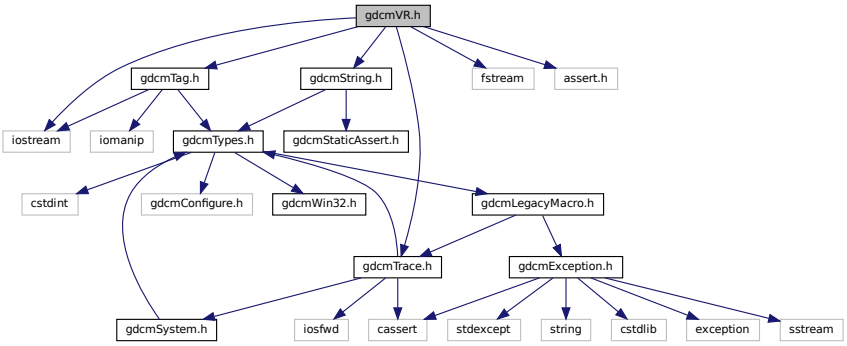
```
78     VM8 = 64,
79     VM9 = 128,
80     VM10 = 256,
81     VM12 = 512, //1024,
82     VM16 = 1024, //2048,
83     VM18 = 2048, //4096,
84     VM24 = 4096, //8192,
85     VM28 = 8192, //16384,
86     VM32 = 16384, //32768,
87     VM35 = 32768, //65536,
88     VM99 = 65536, //131072,
89     VM256 = 131072, //262144,
90     VM1_2  = VM1 | VM2,
91     VM1_3  = VM1 | VM2 | VM3,
92     VM1_4  = VM1 | VM2 | VM3 | VM4,
93     VM1_5  = VM1 | VM2 | VM3 | VM4 | VM5,
94     VM1_8  = VM1 | VM2 | VM3 | VM4 | VM5 | VM6 | VM8,
95 // The following need some work:
96     VM1_32 = VM1 | VM2 | VM3 | VM4 | VM5 | VM6 | VM8 | VM9 | VM16 | VM24 | VM32,
97     VM1_99 = VM1 | VM2 | VM3 | VM4 | VM5 | VM6 | VM8 | VM9 | VM16 | VM24 | VM32 | VM99,
98     VM1_n  = VM1 | VM2 | VM3 | VM4 | VM5 | VM6 | VM8 | VM9 | VM16 | VM24 | VM32 | VM99 | VM256,
99     VM2_2n =       VM2       | VM4       | VM6 | VM8       | VM16 | VM24 | VM32       | VM256,
100     VM2_n  =       VM2 | VM3 | VM4 | VM5 | VM6 | VM8 | VM9 | VM16 | VM24 | VM32 | VM99 | VM256,
101     VM3_4  =             VM3 | VM4,
102     VM3_3n =             VM3 |             VM6       | VM9        | VM24        | VM99 | VM256,
103     VM3_n  =             VM3 | VM4 | VM5 | VM6 | VM8 | VM9 | VM16 | VM24 | VM32 | VM99 | VM256,
104     VM4_4n =                   VM4                           | VM16 | VM24 | VM32        | VM256,
105     VM6_6n =                               VM6               | VM12 | VM18 | VM24               ,
106     VM6_n  =                               VM6 | VM8 | VM9 | VM16 | VM24 | VM32 | VM99 | VM256,
107     VM7_7n,
108     VM30_30n,
109     VM47_47n,
110     VM_END = VM1_n + 1  // Custom tag to count number of entry
111   } VMType;
112
115   static const char* GetVMString(VMType vm);
116   static VMType GetVMType(const char *vm);
117
120   static bool IsValid(int vm1, VMType vm2);
121   //bool IsValid() { return VMField != VM0 && VMField < VM_END; }
122
126   bool Compatible(VM const &vm) const;
127
129   static VMType GetVMTypeFromLength(size_t length, unsigned int size);
130   static size_t GetNumberOfElementsFromArray(const char *array, size_t length);
131
132   VM(VMType type = VM0):VMField(type) {}
133   operator VMType ()const { return VMField; }
134   unsigned int GetLength() const;
135
136   friend std::ostream &operator<<(std::ostream &os, const VM &vm);
137 protected:
138   static unsigned int GetIndex(VMType vm);
139
140 private:
141   VMType VMField;
142 };
143 //-----------------------------------------------------------------------------
144 inline std::ostream& operator<<(std::ostream& _os, const VM &_val)
145 {
146   assert( VM::GetVMString(_val) );
147   _os << VM::GetVMString(_val);
148   return _os;
149 }
150
151 //template <int TVM> struct LengthToVM;
152 //template <> struct LengthToVM<1>
153 //{ enum { TVM = VM::VM1 }; };
154
155 template<int T> struct VMToLength;
156 #define TYPETOLENGTH(type,length) \
157 template<> struct VMToLength<VM::type> \
158 { enum { Length = length }; };
159 // TODO: Could be generated from XML file
160 //TYPETOLENGTH(VM0,1)
161 TYPETOLENGTH(VM1,1)
162 TYPETOLENGTH(VM2,2)
163 TYPETOLENGTH(VM3,3)
164 TYPETOLENGTH(VM4,4)
165 TYPETOLENGTH(VM5,5)
166 TYPETOLENGTH(VM6,6)
```

```
167 TYPETOLENGTH(VM8,8)
168 TYPETOLENGTH(VM9,9)
169 TYPETOLENGTH(VM10,10)
170 TYPETOLENGTH(VM12,12)
171 TYPETOLENGTH(VM16,16)
172 TYPETOLENGTH(VM18,18)
173 TYPETOLENGTH(VM24,24)
174 TYPETOLENGTH(VM28,28)
175 TYPETOLENGTH(VM32,32)
176 TYPETOLENGTH(VM35,35)
177 TYPETOLENGTH(VM99,99)
178 TYPETOLENGTH(VM256,256)
179 //TYPETOLENGTH(VM1_2,2)
180 //TYPETOLENGTH(VM1_3,3)
181 //TYPETOLENGTH(VM1_8,8)
182 //TYPETOLENGTH(VM1_32,32)
183 //TYPETOLENGTH(VM1_99,99)
184 //TYPETOLENGTH(VM1_n,
185 //TYPETOLENGTH(VM2_2n,
186 //TYPETOLENGTH(VM2_n,
187 //TYPETOLENGTH(VM3_3n,
188 //TYPETOLENGTH(VM3_n,
189
190 } // end namespace gdcm
191
192 #endif //GDCMVM_H
```

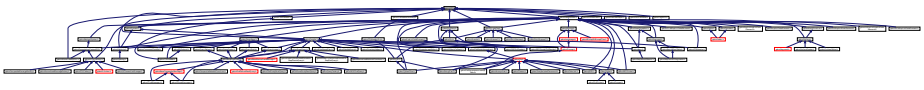
11.193 gdcmVR.h File Reference

```
#include "gdcmTag.h"
#include "gdcmTrace.h"
#include "gdcmString.h"
#include <iostream>
#include <fstream>
#include <assert.h>
```

Include dependency graph for gdcmVR.h:



This graph shows which files directly or indirectly include this file:



Classes

- struct gdcm::UI
- class gdcm::VR

  VR class.

Namespaces

- namespace gdcm

Macros

- #define TYPETOENCODING(type, rep, rtype)
- #define VRTypeTemplateCase(type)

Typedefs

- typedef String<'\\', 16 > gdcm::AEComp
- typedef String<'\\', 64 > gdcm::ASComp
- typedef String<'\\', 16 > gdcm::CSComp
- typedef String<'\\', 64 > gdcm::DAComp
- typedef String<'\\', 64 > gdcm::DTComp
- typedef String<'\\', 64 > gdcm::LOComp
- typedef String<'\\', 64 > gdcm::LTComp
- typedef String<'\\', 64 > gdcm::PNComp
- typedef String<'\\', 64 > gdcm::SHComp
- typedef String<'\\', 64 > gdcm::STComp
- typedef String<'\\', 16 > gdcm::TMComp
- typedef String<'\\', 4294967294 > gdcm::UCComp
- typedef String<'\\', 64, 0 > gdcm::UIComp
- typedef String<'\\', 4294967294 > gdcm::URComp
- typedef String<'\\', 64 > gdcm::UTComp

Functions

- std::ostream & gdcm::operator<< (std::ostream &_os, const UI &_val)
- std::ostream & gdcm::operator<< (std::ostream &_os, const VR &val)
- gdcm::TYPETOENCODING (SQ, VRBINARY, unsigned char) TYPETOENCODING(UN

Variables

- gdcm::VRBINARY

11.193.1 Macro Definition Documentation

11.193.1.1 TYPETOENCODING

```
#define TYPETOENCODING(
              type,
              rep,
              rtype )
```

Value:

```
  template<> struct VRToEncoding<VR::type>    \
  { enum:long long { Mode = VR::rep }; };                \
  template<> struct VRToType<VR::type>        \
  { typedef rtype Type; };
```

11.193.1.2 VRTypeTemplateCase

```
#define VRTypeTemplateCase(
              type )
```

Value:

```
  case VR::type:  \
    return sizeof ( VRToType<VR::type>::Type );
```

11.194 gdcmVR.h

Go to the documentation of this file.

```
1 /*=========================================================================
2
3 Program:  GDCM (Grassroots DICOM). A DICOM library
4
5 Copyright (c) 2006-2011 Mathieu Malaterre
6 All rights reserved.
7 See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.
8
9 This software is distributed WITHOUT ANY WARRANTY; without even
10 the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
11 PURPOSE.  See the above copyright notice for more information.
12
13 =========================================================================*/
14 #ifndef GDCMVR_H
15 #define GDCMVR_H
16
17 #include "gdcmTag.h"
18 #include "gdcmTrace.h"
19 #include "gdcmString.h"
20
21 #include <iostream>
22 #include <fstream>
23 #include <assert.h>
24
25 //these defines are here to ensure compilation on sunos gcc
26 #if defined (CS)
27 # undef CS
28 #endif
```

```
29 #if defined (DS)
30 # undef DS
31 #endif
32 #if defined (SS)
33 # undef SS
34 #endif
35
36
37 namespace gdcm
38 {
39
54 class GDCM_EXPORT VR
55 {
56 public:
57   enum VRType :  long long {
58     // Warning:  Do not write if ( vr & VR::INVALID ) but if ( vr == VR::INVALID )
59     INVALID = 0, // For Item/(Seq) Item Delimitation Item
60     AE = 1,
61     AS = 2,
62     AT = 4,
63     CS = 8,
64     DA = 16,
65     DS = 32,
66     DT = 64,
67     FD = 128,
68     FL = 256,
69     IS = 512,
70     LO = 1024,
71     LT = 2048,
72     OB = 4096,
73     OD = 134217728, // 2^27
74     OF = 8192,
75     OL = 268435456, // 2^28
76     OV = 2147483648, // 2^31
77     OW = 16384,
78     PN = 32768,
79     SH = 65536,
80     SL = 131072,
81     SQ = 262144,
82     SS = 524288,
83     ST = 1048576,
84     SV = 4294967296, // 2^32
85     TM = 2097152,
86     UC = 536870912, // 2^29
87     UI = 4194304,
88     UL = 8388608,
89     UN = 16777216,
90     UR = 1073741824, // 2^30
91     US = 33554432,
92     UT = 67108864,
93     UV = 8589934592, // 2^33
94     OB_OW = OB | OW,
95     US_SS = US | SS,
96     US_SS_OW = US | SS | OW,
97     US_OW = US | OW,
98     // The following do not have a VRString equivalent (ie cannot be found in PS 3.6)
99     VL16 = AE | AS | AT | CS | DA | DS | DT | FD | FL | IS | LO | LT | PN | SH | SL | SS | ST | TM | UI | UL
   | US, // if( VR & VL16 ) => VR has its VL coded over 16bits
100     VL32 = OB | OW | OD | OF | OL | OV | SQ | SV | UC | UN | UR | UT | UV, // if( VR & VL32 ) => VR has its
   VL coded over 32bits
101     VRASCII = AE | AS | CS | DA | DS | DT | IS | LO | LT | PN | SH | ST | TM | UC | UI | UR | UT,
102     VRBINARY = AT | FL | FD | OB | OD | OF | OL | OV | OW | SL | SQ | SS | SV | UL | UN | US | UV, // FIXME:
   UN ?
103     // PS 3.5:
104     // Data Elements with a VR of SQ, OD, OF, OL, OW, OB or UN shall always have a Value Multiplicity of
   one.
105     // GDCM is adding a couple more:  AS, LT, ST, UT
106     VR_VM1 = AS | LT | ST | UT | SQ | OF | OL | OV | OD | OW | OB | UN, // All those VR have a VM1
107     VRALL = VRASCII | VRBINARY,
108     VR_END = UV+1  // Invalid VR, need to be max(VRType)+1
109   };
110
111   static const char *GetVRString(VRType vr);
112
113   // This function will only look at the very first two chars nothing else
114   static VRType GetVRTypeFromFile(const char *vr);
115
116   // You need to make sure end of string is \0
117   static VRType GetVRType(const char *vr);
118   static const char *GetVRStringFromFile(VRType vr);
119
```

```
120   static bool IsValid(const char *vr);
121   // Check if vr1 is valid against vr2,
122   // Typically vr1 is read from the file and vr2 is taken from the dict
123   static bool IsValid(const char *vr1, VRType vr2);
124   //static bool IsValid(const VRType &vr1, const VRType &vr2);
125   // Find out if the string read is byte swapped
126   static bool IsSwap(const char *vr);
127
128   // Size read on disk
129   // FIXME: int ?
130   int GetLength()const {
131     return VR::GetLength(VRField);
132   }
133   unsigned int GetSizeof() const;
134   static uint32_t GetLength(VRType vr) {
135     //if( vr == VR::INVALID ) return 4;
136     if( vr & VL32 )
137       {
138       return 4;
139       }
140     else
141       return 2;
142   }
143
144   // Some use of template metaprograming with ugly macro
145   static bool IsBinary(VRType vr);
146   static bool IsASCII(VRType vr);
147   // TODO: REMOVE ME
148   static bool CanDisplay(VRType vr);
149   // TODO: REMOVE ME
150   static bool IsBinary2(VRType vr);
151   // TODO: REMOVE ME
152   static bool IsASCII2(VRType vr);
153
154   VR(VRType vr = INVALID):VRField(vr) { }
155   //VR(VR const &vr):VRField(vr.VRField) { }
156   std::istream &Read(std::istream &is)
157     {
158     char vr[2];
159     is.read(vr, 2);
160     VRField = GetVRTypeFromFile(vr);
161     assert( VRField != VR::VR_END );
162     if( VRField == VR::INVALID )
163     {
164       // \0\2 Data/TheralysGDCM120Bug.dcm
165       // \0\0
      Data/MR_Philips_Intera_PrivateSequenceExplicitVR_in_SQ_2001_e05f_item_wrong_lgt_use_NOSHADOWSEQ.dcm
166       // \0\4 Data/BugGDCM2_UndefItemWrongVL.dcm
167       // \44\0 Data/gdcm-MR-PHILIPS-16-Multi-Seq.dcm
168       // \0\20 Data/ExplicitVRforPublicElementsImplicitVRforShadowElements.dcm
169       // \0\3 Data/DMCPACS_ExplicitImplicit_BogusIOP.dcm
170       // \0\4 Data/THERALYS-12-MONO2-Uncompressed-Even_Length_Tag.dcm
171       // \0\4 Data/PrivateGEImplicitVRBigEndianTransferSyntax16Bits.dcm
172       // \0\4 Data/GE_DLX-8-MONO2-PrivateSyntax.dcm
173       throw Exception( "INVALID VR" );
174     }
175     if( VRField & VL32 )
176       {
177 #if 0
178       // For some reason this seems slower on my linux box...
179       is.seekg(2, std::ios::cur );
180 #else
181       char dum[2];
182       is.read(dum, 2);
183       if( !(dum[0] == 0 && dum[1] == 0 ))
184         {
185         // JDDICOM_Sample4.dcm
186         gdcmDebugMacro( "32bits VR contains non zero bytes.  Skipped" );
187         }
188 #endif
189       }
190     return is;
191     }
192
193   const std::ostream &Write(std::ostream &os)const
194 {
195     VRType vrfield = VRField;
196     gdcmAssertAlwaysMacro( !IsDual() );
197     if( vrfield == VR::INVALID )
198       {
199       //vrfield = VR::UN;
```

```
200       }
201     const char *vr = GetVRString(vrfield);
202     //assert( strlen( vr ) == 2 );
203     assert( vr[0] && vr[1] && vr[2] == 0 );
204     os.write(vr, 2);
205     // See PS 3.5, Data Element Structure With Explicit VR
206     if( vrfield & VL32 )
207       {
208       const char dum[2] = {0, 0};
209       os.write(dum,2);
210       }
211     return os;
212     }
213   friend std::ostream &operator«(std::ostream &os, const VR &vr);
214
215   operator VRType ()const { return VRField; }
216
217   unsigned int GetSize() const;
218
219   bool Compatible(VR const &vr) const;
220
221   bool IsVRFile() const;
222
223   bool IsDual() const;
224
225 private:
226   // Internal function that map a VRType to an index in the VRStrings table
227   static unsigned int GetIndex(VRType vr);
228   VRType VRField;
229 };
230 //-----------------------------------------------------------------------------
231 inline std::ostream &operator«(std::ostream &_os, const VR &val)
232 {
233   //_os « VR::GetVRStringFromFile(val.VRField);
234   _os « VR::GetVRString(val.VRField);
235   return _os;
236 }
237
238 // Apparently SWIG is not happy with something, somewhere below...
239 #ifndef SWIG
240
241 // Tells whether VR Type is ASCII or Binary
242 template<long long T> struct VRToEncoding;
243 // Convert from VR Type to real underlying type
244 template<long long T> struct VRToType;
245 #define TYPETOENCODING(type,rep, rtype)         \
246 template<> struct VRToEncoding<VR::type>    \
247 { enum:long long { Mode = VR::rep }; };                 \
248 template<> struct VRToType<VR::type>        \
249 { typedef rtype Type; };
250
251
252 // Do not use me
253 struct UI { char Internal[64+1];
254   friend std::ostream& operator«(std::ostream &_os, const UI &_val);
255 };
256 inline std::ostream& operator«(std::ostream &_os, const UI &_val)
257 {
258   _os « _val.Internal;
259   return _os;
260 }
261
262 typedef String<'\\',16> AEComp;
263 typedef String<'\\',64> ASComp;
264 typedef String<'\\',16> CSComp;
265 typedef String<'\\',64> DAComp;
266 typedef String<'\\',64> DTComp;
267 typedef String<'\\',64> LOComp;
268 typedef String<'\\',64> LTComp;
269 typedef String<'\\',64> PNComp;
270 typedef String<'\\',64> SHComp;
271 typedef String<'\\',64> STComp;
272 typedef String<'\\',4294967294> UCComp;
273 typedef String<'\\',4294967294> URComp;
274 typedef String<'\\',16> TMComp;
275 typedef String<'\\',64,0> UIComp;
276 typedef String<'\\',64> UTComp;
277
278
279 // TODO: Could be generated from XML file
280 TYPETOENCODING(AE,VRASCII ,AEComp)
```

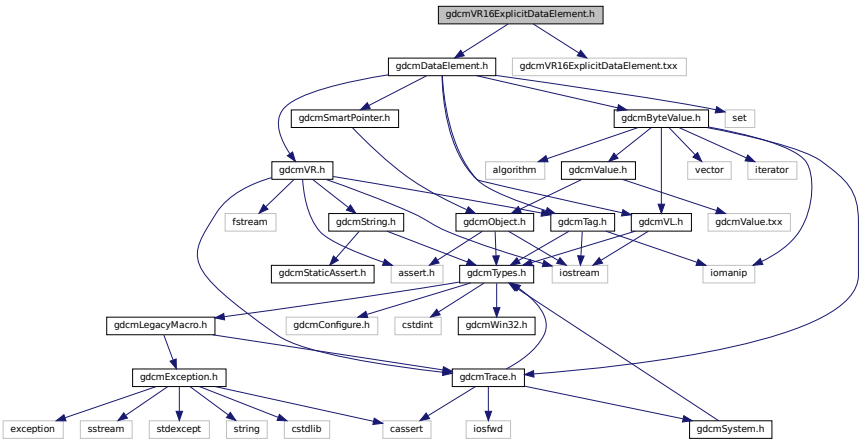
```
281 TYPETOENCODING(AS,VRASCII ,ASComp)
282 TYPETOENCODING(AT,VRBINARY,Tag)
283 TYPETOENCODING(CS,VRASCII ,CSComp)
284 TYPETOENCODING(DA,VRASCII ,DAComp)
285 TYPETOENCODING(DS,VRASCII ,double)
286 TYPETOENCODING(DT,VRASCII ,DTComp)
287 TYPETOENCODING(FL,VRBINARY,float)
288 TYPETOENCODING(FD,VRBINARY,double)
289 TYPETOENCODING(IS,VRASCII ,int32_t)
290 TYPETOENCODING(LO,VRASCII ,LOComp)
291 TYPETOENCODING(LT,VRASCII ,LTComp)
292 TYPETOENCODING(OB,VRBINARY,uint8_t)
293 TYPETOENCODING(OD,VRBINARY,double)
294 TYPETOENCODING(OF,VRBINARY,float)
295 TYPETOENCODING(OL,VRBINARY,uint32_t)
296 TYPETOENCODING(OV,VRBINARY,uint64_t)
297 TYPETOENCODING(OW,VRBINARY,uint16_t)
298 TYPETOENCODING(PN,VRASCII ,PNComp)
299 TYPETOENCODING(SH,VRASCII ,SHComp)
300 TYPETOENCODING(SL,VRBINARY,int32_t)
301 TYPETOENCODING(SQ,VRBINARY,unsigned char) // FIXME
302 TYPETOENCODING(SS,VRBINARY,int16_t)
303 TYPETOENCODING(ST,VRASCII ,STComp)
304 TYPETOENCODING(SV,VRBINARY,int64_t)
305 TYPETOENCODING(TM,VRASCII ,TMComp)
306 TYPETOENCODING(UC,VRASCII ,UCComp)
307 TYPETOENCODING(UI,VRASCII ,UIComp)
308 TYPETOENCODING(UL,VRBINARY,uint32_t)
309 TYPETOENCODING(UN,VRBINARY,uint8_t) // FIXME ?
310 TYPETOENCODING(UR,VRASCII,URComp)
311 TYPETOENCODING(US,VRBINARY,uint16_t)
312 TYPETOENCODING(UT,VRASCII ,UTComp)
313 TYPETOENCODING(UV,VRBINARY,uint64_t)
314
315 #define VRTypeTemplateCase(type) \
316 case VR::type:  \
317 return sizeof ( VRToType<VR::type>::Type );
318
319 inline unsigned int VR::GetSize()const
320 {
321   switch(VRField)
322   {
323     VRTypeTemplateCase(AE)
324     VRTypeTemplateCase(AS)
325     VRTypeTemplateCase(AT)
326     VRTypeTemplateCase(CS)
327     VRTypeTemplateCase(DA)
328     VRTypeTemplateCase(DS)
329     VRTypeTemplateCase(DT)
330     VRTypeTemplateCase(FL)
331     VRTypeTemplateCase(FD)
332     VRTypeTemplateCase(IS)
333     VRTypeTemplateCase(LO)
334     VRTypeTemplateCase(LT)
335     VRTypeTemplateCase(OB)
336     VRTypeTemplateCase(OD)
337     VRTypeTemplateCase(OF)
338     VRTypeTemplateCase(OL)
339     VRTypeTemplateCase(OV)
340     VRTypeTemplateCase(OW)
341     VRTypeTemplateCase(PN)
342     VRTypeTemplateCase(SH)
343     VRTypeTemplateCase(SL)
344     VRTypeTemplateCase(SQ)
345     VRTypeTemplateCase(SS)
346     VRTypeTemplateCase(ST)
347     VRTypeTemplateCase(SV)
348     VRTypeTemplateCase(TM)
349     VRTypeTemplateCase(UC)
350     VRTypeTemplateCase(UI)
351     VRTypeTemplateCase(UL)
352     VRTypeTemplateCase(UN)
353     VRTypeTemplateCase(UR)
354     VRTypeTemplateCase(US)
355     VRTypeTemplateCase(UT)
356     VRTypeTemplateCase(UV)
357     case VR::US_SS:
358       return 2;
359
360     case VR::INVALID:
361     case VR::OB_OW:
```

```
362     case VR::US_SS_OW:
363     case VR::US_OW:
364     case VR::VL16:
365     case VR::VL32:
366     case VR::VRASCII:
367     case VR::VRBINARY:
368     case VR::VR_VM1:
369     case VR::VRALL:
370     case VR::VR_END:
371     default:
372        assert( 0 && "should not" );
373   }
374   return 0;
375 }
376 #endif // SWIG
377
378
379 } // end namespace gdcm
380
381 #endif //GDCMVR_H
```

11.195 gdcmVR16ExplicitDataElement.h File Reference

```
#include "gdcmDataElement.h"
#include "gdcmVR16ExplicitDataElement.txx"
```

Include dependency graph for gdcmVR16ExplicitDataElement.h:



Classes

- class gdcm::VR16ExplicitDataElement

  Class to read/write a DataElement as Explicit Data Element.

Namespaces

- namespace gdcm

11.196 gdcmVR16ExplicitDataElement.h

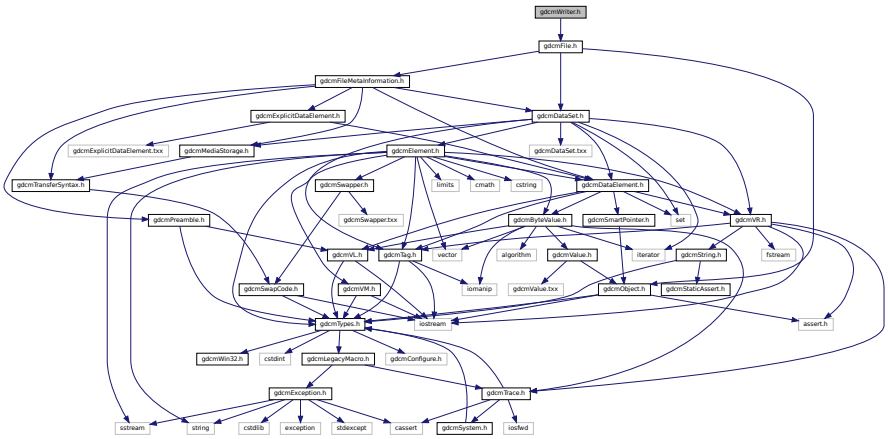
Go to the documentation of this file.

```
1 /*=========================================================================
2
3 Program:  GDCM (Grassroots DICOM). A DICOM library
4
5 Copyright (c) 2006-2011 Mathieu Malaterre
6 All rights reserved.
7 See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.
8
9 This software is distributed WITHOUT ANY WARRANTY; without even
10 the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
11 PURPOSE.  See the above copyright notice for more information.
12
13 =========================================================================*/
14 #ifndef GDCMVR16EXPLICITDATAELEMENT_H
15 #define GDCMVR16EXPLICITDATAELEMENT_H
16
17 #include "gdcmDataElement.h"
18
19 namespace gdcm
20 {
21 // Data Element (Explicit)
28 class GDCM_EXPORT VR16ExplicitDataElement :  public DataElement
29 {
30 public:
31   VL GetLength() const;
32
33   template <typename TSwap>
34   std::istream &Read(std::istream &is);
35
36   template <typename TSwap>
37   std::istream &ReadPreValue(std::istream &is);
38
39   template <typename TSwap>
40   std::istream &ReadValue(std::istream &is, bool readvalues = true);
41
42   template <typename TSwap>
43   std::istream &ReadWithLength(std::istream &is, VL & length);
44
45   // PURPOSELY do not provide an implementation for writing !
46   //template <typename TSwap>
47   //const std::ostream &Write(std::ostream &os) const;
48 };
49
50 } // end namespace gdcm
51
52 #include "gdcmVR16ExplicitDataElement.txx"
53
54 #endif //GDCMVR16EXPLICITDATAELEMENT_H
```

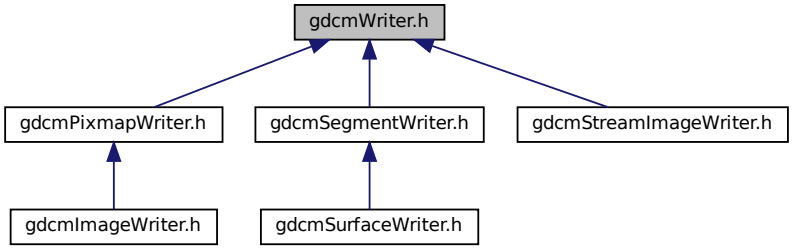
11.197 gdcmWriter.h File Reference

```
#include "gdcmFile.h"
```

Include dependency graph for gdcmWriter.h:



This graph shows which files directly or indirectly include this file:



Classes

- class gdcm::Writer

  Writer ala DOM (Document Object Model)

Namespaces

- namespace gdcm

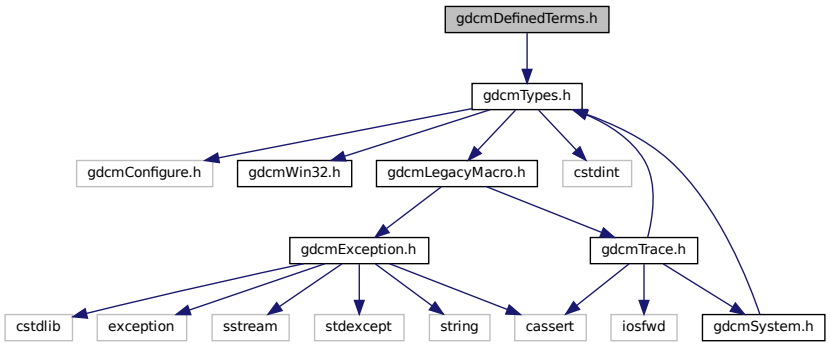
11.198 gdcmWriter.h

Go to the documentation of this file.

```
1 /*=========================================================================
2
3 Program:  GDCM (Grassroots DICOM). A DICOM library
4
5 Copyright (c) 2006-2011 Mathieu Malaterre
6 All rights reserved.
7 See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.
8
9 This software is distributed WITHOUT ANY WARRANTY; without even
10 the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
11 PURPOSE.  See the above copyright notice for more information.
12
13 =========================================================================*/
14
15 #ifndef GDCMWRITER_H
16 #define GDCMWRITER_H
17
18 #include "gdcmFile.h"
19
20 namespace gdcm
21 {
22
23 class FileMetaInformation;
48 class GDCM_EXPORT Writer
49 {
50 public:
51   Writer();
52   virtual ~Writer();
53
55   virtual bool Write(); // Execute()
56
58   void SetFileName(const char *filename_native);
59
61   void SetStream(std::ostream &output_stream) {
62     Stream = &output_stream;
63   }
64
66   void SetFile(const File& f) { F = f; }
67   File &GetFile() { return *F; }
68
70   void SetCheckFileMetaInformation(bool b) { CheckFileMetaInformation = b; }
71   void CheckFileMetaInformationOff() { CheckFileMetaInformation = false; }
72   void CheckFileMetaInformationOn() { CheckFileMetaInformation = true; }
73
74 protected:
75   void SetWriteDataSetOnly(bool b) { WriteDataSetOnly = b; }
76
77 protected:
78   friend class StreamImageWriter;
79   //this function is added for the StreamImageWriter, which needs to write
80   //up to the pixel data and then stops right before writing the pixel data.
81   //after that, for the raw codec at least, zeros are written for the length of the data
82   std::ostream* GetStreamPtr()const { return Stream; }
83
84 protected:
85   std::ostream *Stream;
86   std::ofstream *Ofstream;
87   bool GetCheckFileMetaInformation()const { return CheckFileMetaInformation; }
88
89 private:
90   SmartPointer<File> F;
91   bool CheckFileMetaInformation;
92   bool WriteDataSetOnly;
93 };
94
95 } // end namespace gdcm
96
97 #endif //GDCMWRITER_H
```

11.199 gdcmDefinedTerms.h File Reference

`#include "gdcmTypes.h"`
Include dependency graph for gdcmDefinedTerms.h:



Classes

- class gdcm::DefinedTerms

  Defined Terms are used when the specified explicit Values may be extended by implementors to include additional new Values. These new Values shall be specified in the Conformance Statement (see PS 3.2) and shall not have the same meaning as currently defined Values in this standard. A Data Element with Defined Terms that does not contain a Value equivalent to one of the Values currently specified in this standard shall not be considered to have an invalid value. Note: Interpretation Type ID (4008,0210) is an example of a Data Element having Defined Terms. It is defined to have a Value that may be one of the set of standard Values; REPORT or AMENDMENT (see PS 3.3). Because this Data Element has Defined Terms other Interpretation Type IDs may be defined by the implementor.

Namespaces

- namespace gdcm

11.200 gdcmDefinedTerms.h

Go to the documentation of this file.

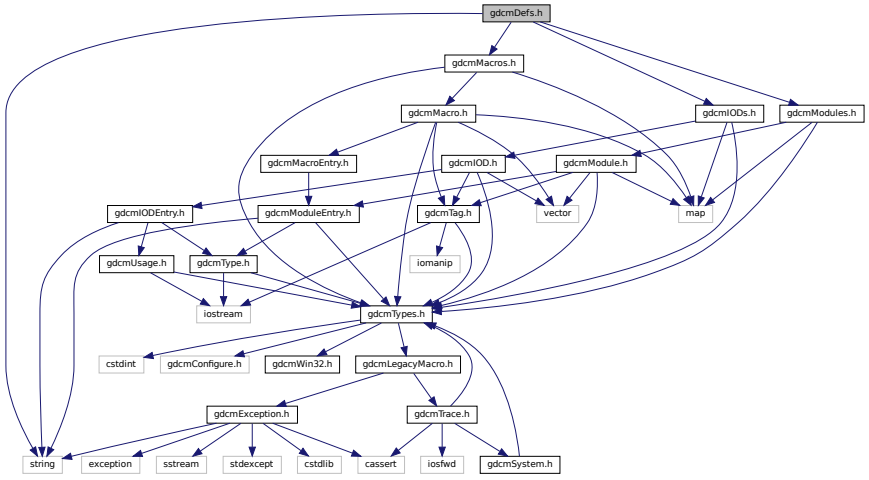
```
1 /*=========================================================================
2
3 Program:  GDCM (Grassroots DICOM). A DICOM library
4
5 Copyright (c) 2006-2011 Mathieu Malaterre
6 All rights reserved.
7 See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.
8
9 This software is distributed WITHOUT ANY WARRANTY; without even
10 the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
11 PURPOSE.  See the above copyright notice for more information.
12
13 =========================================================================*/
14 #ifndef GDCMDEFINEDTERMS_H
```

```
15 #define GDCMDEFINEDTERMS_H
16
17 #include "gdcmTypes.h"
18
19 namespace gdcm
20 {
27 class GDCM_EXPORT DefinedTerms
28 {
29 public:
30   DefinedTerms() = default;
31 private:
32 };
33
34 } // end namespace gdcm
35
36 #endif //GDCMDEFINEDTERMS_H
```

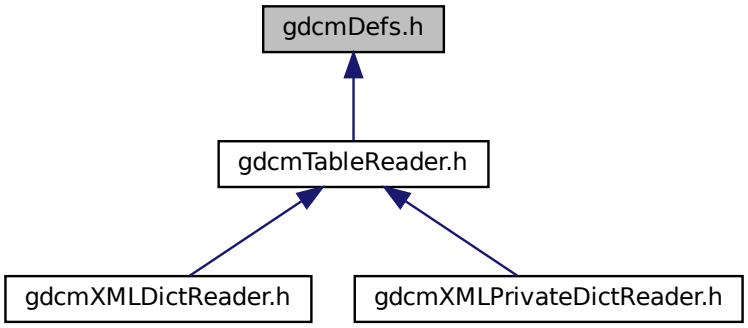
11.201 gdcmDefs.h File Reference

```
#include "gdcmModules.h"
#include "gdcmMacros.h"
#include "gdcmIODs.h"
#include <string>
```

Include dependency graph for gdcmDefs.h:



This graph shows which files directly or indirectly include this file:



Classes

- class gdcm::Defs

  FIXME I do not like the name 'Defs'.

Namespaces

- namespace gdcm

11.202 gdcmDefs.h

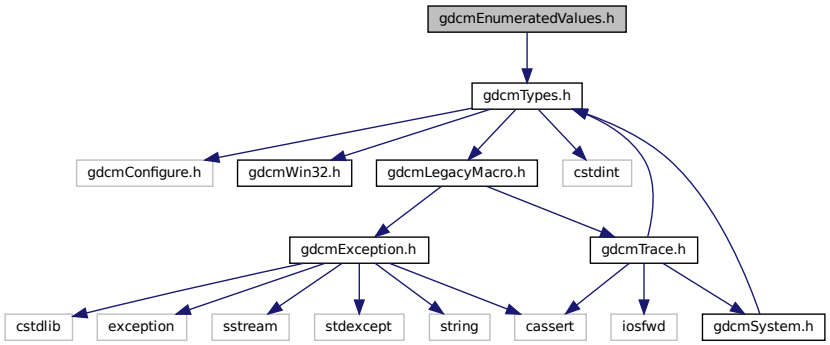
Go to the documentation of this file.

```
1 /*=========================================================================
2
3 Program:  GDCM (Grassroots DICOM). A DICOM library
4
5 Copyright (c) 2006-2011 Mathieu Malaterre
6 All rights reserved.
7 See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.
8
9 This software is distributed WITHOUT ANY WARRANTY; without even
10 the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
11 PURPOSE.  See the above copyright notice for more information.
12
13 =========================================================================*/
14 #ifndef GDCMDEFS_H
15 #define GDCMDEFS_H
16
17 #include "gdcmModules.h"
18 #include "gdcmMacros.h"
19 #include "gdcmIODs.h"
20
21 #include <string>
22
23 namespace gdcm
24 {
25 class DataSet;
```

```
26 class File;
27 class MediaStorage;
32 class GDCM_EXPORT Defs
33 {
34 public:
35   Defs();
36   ~Defs();
37   Defs &operator=(const Defs &val) = delete;
38   Defs(const Defs &val) = delete;
39
40   const Modules &GetModules()const { return Part3Modules; }
41   Modules &GetModules() { return Part3Modules; }
42
45   const Macros &GetMacros()const { return Part3Macros; }
46   Macros &GetMacros() { return Part3Macros; }
47
48   const IODs & GetIODs()const { return Part3IODs; }
49   IODs & GetIODs() { return Part3IODs; }
50
51   bool IsEmpty()const { return GetModules().IsEmpty(); }
52
53   bool Verify(const File& file) const;
54
55   // \deprecated DO NOT USE
56   bool Verify(const DataSet& ds) const;
57
58   Type GetTypeFromTag(const File& file, const Tag& tag) const;
59
60   static const char *GetIODNameFromMediaStorage(MediaStorage const &ms);
61
62   const IOD& GetIODFromFile(const File& file) const;
63
64 protected:
65   friend class Global;
66   void LoadDefaults();
67   void LoadFromFile(const char *filename);
68
69 private:
70   // Part 3 stuff:
71   Macros Part3Macros;
72   Modules Part3Modules;
73   IODs Part3IODs;
74
75 };
76
77
78 } // end namespace gdcm
79
80 #endif //GDCMDEFS_H
```

11.203 gdcmEnumeratedValues.h File Reference

`#include "gdcmTypes.h"`
Include dependency graph for gdcmEnumeratedValues.h:



Classes

- class gdcm::EnumeratedValues

  Element. A Data Element with Enumerated Values that does not have a Value equivalent to one of the Values specified in this standard has an invalid value within the scope of a specific Information Object/SOP Class definition. Note:

Namespaces

- namespace gdcm

11.204 gdcmEnumeratedValues.h

Go to the documentation of this file.

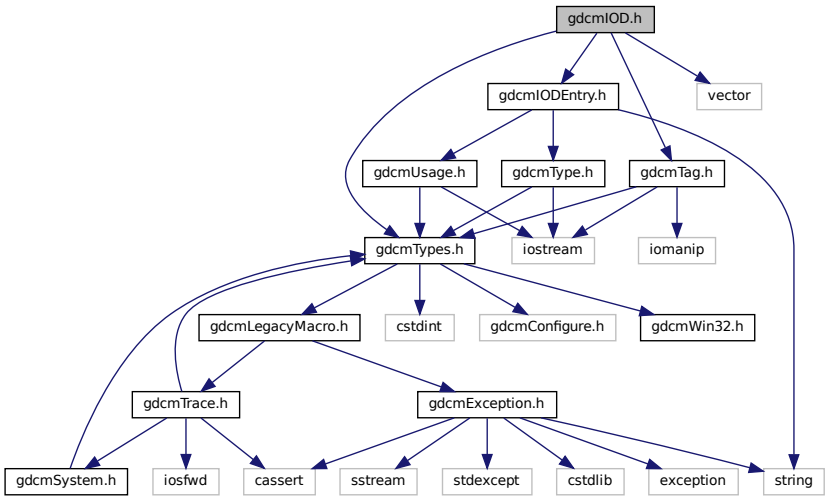
```
1 /*=========================================================================
2
3 Program:  GDCM (Grassroots DICOM). A DICOM library
4
5 Copyright (c) 2006-2011 Mathieu Malaterre
6 All rights reserved.
7 See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.
8
9 This software is distributed WITHOUT ANY WARRANTY; without even
10 the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
11 PURPOSE.  See the above copyright notice for more information.
12
13 =========================================================================*/
14 #ifndef GDCMENUMERATEDVALUES_H
15 #define GDCMENUMERATEDVALUES_H
16
17 #include "gdcmTypes.h"
18
19 namespace gdcm
20 {
34 class GDCM_EXPORT EnumeratedValues
```

```
35 {
36 public:
37   EnumeratedValues() = default;
38 private:
39 };
40
41 } // end namespace gdcm
42
43 #endif //GDCMENUMERATEDVALUES_H
```

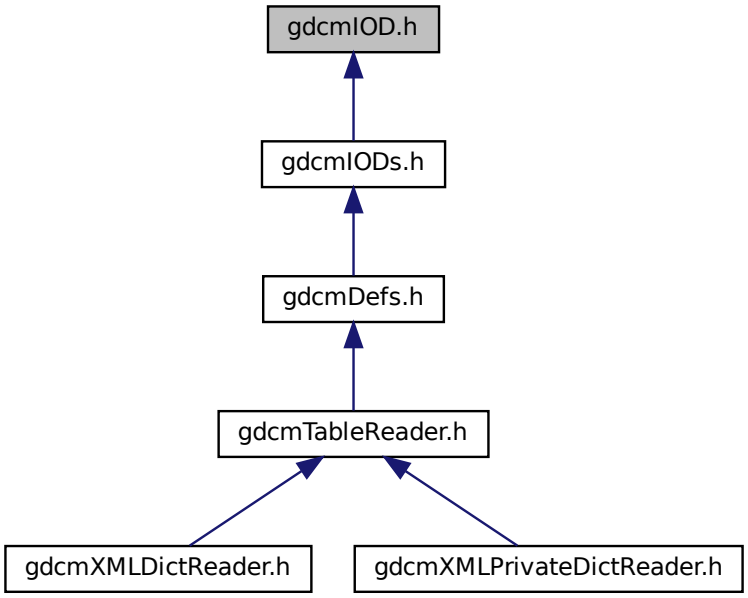
11.205 gdcmIOD.h File Reference

```
#include "gdcmTypes.h"
#include "gdcmTag.h"
#include "gdcmIODEntry.h"
#include <vector>
```

Include dependency graph for gdcmIOD.h:



This graph shows which files directly or indirectly include this file:



Classes

- class gdcm::IOD

  Class for representing a IOD.

Namespaces

- namespace gdcm

Functions

- std::ostream & gdcm::operator<< (std::ostream &_os, const IOD &_val)

11.206 gdcmIOD.h

Go to the documentation of this file.

```
1 /*=========================================================================
2
3 Program:  GDCM (Grassroots DICOM). A DICOM library
4
```

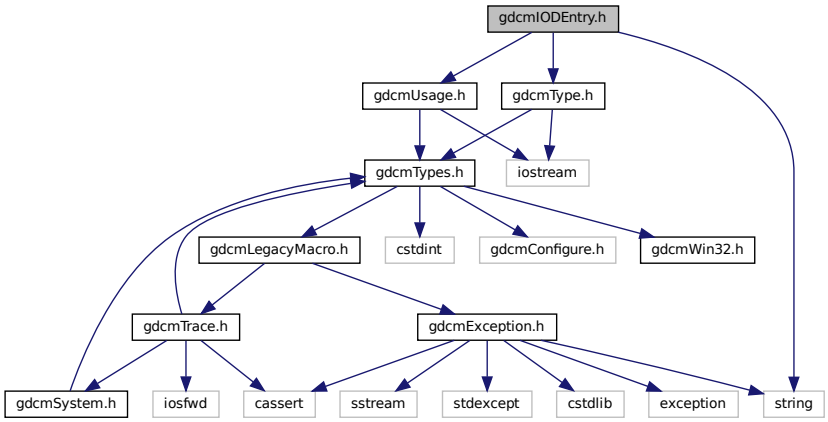
```
5 Copyright (c) 2006-2011 Mathieu Malaterre
6 All rights reserved.
7 See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.
8
9 This software is distributed WITHOUT ANY WARRANTY; without even
10 the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
11 PURPOSE.  See the above copyright notice for more information.
12
13 =========================================================================*/
14 #ifndef GDCMIOD_H
15 #define GDCMIOD_H
16
17 #include "gdcmTypes.h"
18 #include "gdcmTag.h"
19 #include "gdcmIODEntry.h"
20
21 #include <vector>
22
23 namespace gdcm
24 {
25 class DataSet;
26 class Defs;
27
34 class GDCM_EXPORT IOD
35 {
36 public:
37   typedef std::vector<IODEntry> MapIODEntry;
38   typedef MapIODEntry::size_type SizeType;
39
40   IOD() = default;
41   friend std::ostream& operator«(std::ostream& _os, const IOD &_val);
42
43   void Clear() { IODInternal.clear(); }
44
45   void AddIODEntry(const IODEntry & iode)
46     {
47     IODInternal.push_back(iode);
48     }
49
50   SizeType GetNumberOfIODs()const {
51     return IODInternal.size();
52   }
53
54   const IODEntry& GetIODEntry(SizeType idx)const
55 {
56     return IODInternal[idx];
57     }
58
59   Type GetTypeFromTag(const Defs &defs, const Tag& tag) const;
60
61 private:
62   //IOD &operator=(const IOD &_val); // purposely not implemented
63   //IOD(const IOD &_val); // purposely not implemented
64
65   MapIODEntry IODInternal;
66 };
67 //-----------------------------------------------------------------------------
68 inline std::ostream& operator«(std::ostream& _os, const IOD &_val)
69 {
70   IOD::MapIODEntry::const_iterator it = _val.IODInternal.begin();
71   for(;it != _val.IODInternal.end(); ++it)
72     {
73     _os « *it « '\n';
74     }
75
76   return _os;
77 }
78
79 } // end namespace gdcm
80
81 #endif //GDCMIOD_H
```

11.207 gdcmIODEntry.h File Reference

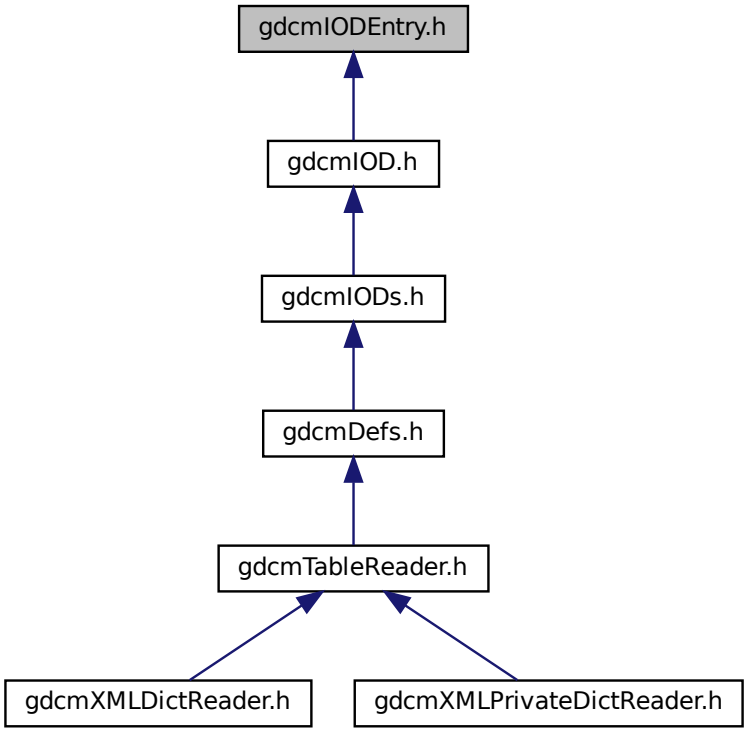
```
#include "gdcmUsage.h"
#include "gdcmType.h"
```

```
#include <string>
```

Include dependency graph for gdcmIODEntry.h:



This graph shows which files directly or indirectly include this file:



Classes

- class gdcm::IODEntry

  Class for representing a IODEntry.

Namespaces

- namespace gdcm

Functions

- std::ostream & gdcm::operator<< (std::ostream &_os, const IODEntry &_val)

11.208 gdcmIODEntry.h

Go to the documentation of this file.

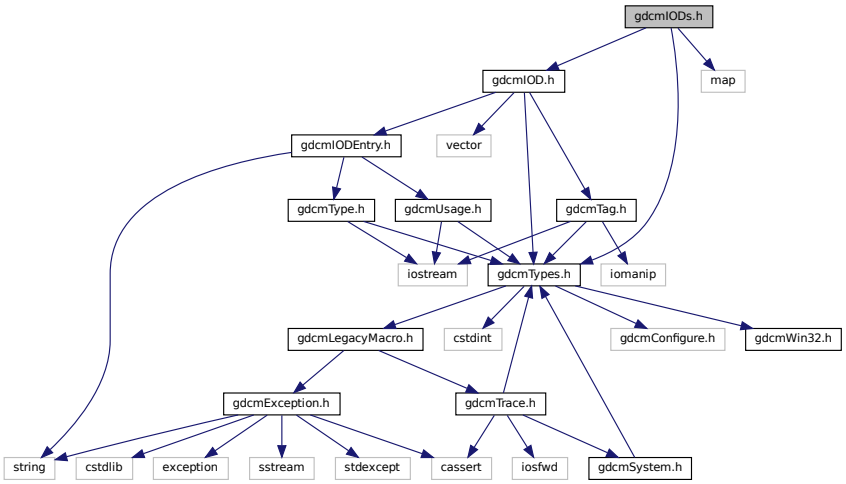
```
1 /*=========================================================================
2
3 Program:  GDCM (Grassroots DICOM). A DICOM library
4
5 Copyright (c) 2006-2011 Mathieu Malaterre
6 All rights reserved.
7 See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.
8
9 This software is distributed WITHOUT ANY WARRANTY; without even
10 the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
11 PURPOSE.  See the above copyright notice for more information.
12
13 =========================================================================*/
14 #ifndef GDCMIODENTRY_H
15 #define GDCMIODENTRY_H
16
17 #include "gdcmUsage.h"
18 #include "gdcmType.h"
19
20 #include <string>
21
22 namespace gdcm
23 {
51 class GDCM_EXPORT IODEntry
52 {
53 public:
54   IODEntry(const char *name = "", const char *ref = "", const char *usag =
       ""):Name(name),Ref(ref),usage(usag) {
55   }
56   friend std::ostream& operator<<(std::ostream& _os, const IODEntry &_val);
57
58   void SetIE(const char *ie) { IE = ie; }
59   const char *GetIE()const { return IE.c_str(); }
60
61   void SetName(const char *name) { Name = name; }
62   const char *GetName()const { return Name.c_str(); }
63
64   void SetRef(const char *ref) { Ref = ref; }
65   const char *GetRef()const { return Ref.c_str(); }
66
67   void SetUsage(const char *usag) { usage = usag; }
68   const char *GetUsage()const { return usage.c_str(); }
69   Usage::UsageType GetUsageType() const;
70
71 private:
72   std::string IE;
73
74   std::string Name;
75
```

```
76   std::string Ref;
77
78   std::string usage;
79 };
80 //-----------------------------------------------------------------------------
81 inline std::ostream& operator«(std::ostream& _os, const IODEntry &_val)
82 {
83   _os « _val.IE « "\t" « _val.Name « "\t" « _val.Ref « "\t" « _val.usage;
84   return _os;
85 }
86
87 } // end namespace gdcm
88
89 #endif //GDCMIODENTRY_H
```

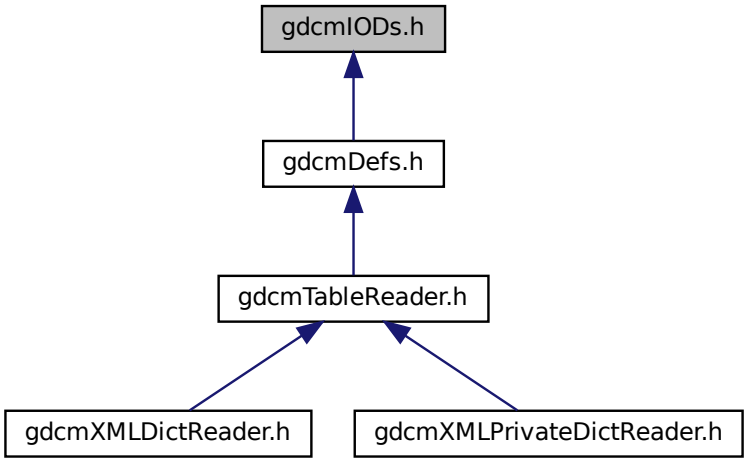
11.209 gdcmIODs.h File Reference

```
#include "gdcmTypes.h"
#include "gdcmIOD.h"
#include <map>
```

Include dependency graph for gdcmIODs.h:



This graph shows which files directly or indirectly include this file:



Classes

- class gdcm::IODs

  Class for representing a IODs.

Namespaces

- namespace gdcm

Functions

- std::ostream & gdcm::operator<< (std::ostream &_os, const IODs &_val)

11.210 gdcmIODs.h

Go to the documentation of this file.

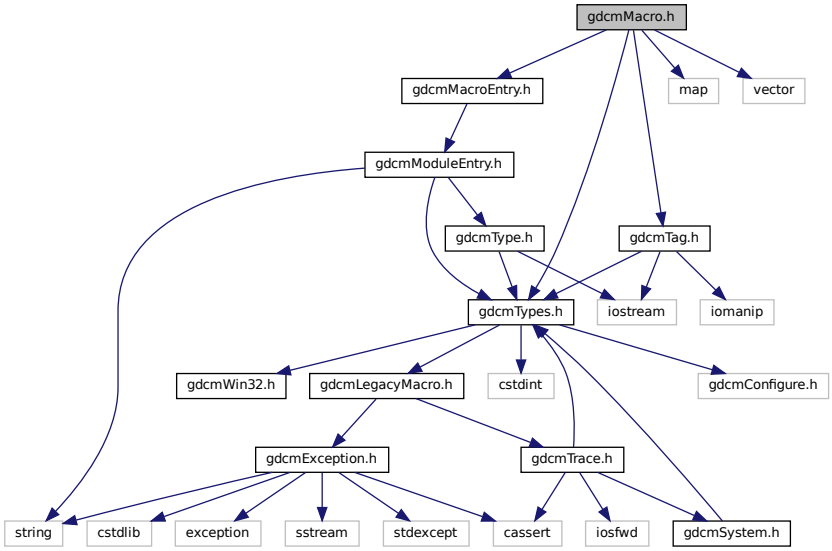
```
1 /*=========================================================================
2
3 Program:  GDCM (Grassroots DICOM). A DICOM library
4
5 Copyright (c) 2006-2011 Mathieu Malaterre
6 All rights reserved.
7 See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.
8
9 This software is distributed WITHOUT ANY WARRANTY; without even
10 the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
11 PURPOSE.  See the above copyright notice for more information.
```

```
12
13 =========================================================================*/
14 #ifndef GDCMIODS_H
15 #define GDCMIODS_H
16
17 #include "gdcmTypes.h"
18 #include "gdcmIOD.h"
19
20 #include <map>
21
22 namespace gdcm
23 {
29 class GDCM_EXPORT IODs
30 {
31 public:
32   typedef std::string IODName;
33   typedef std::map<IODName, IOD> IODMapType;
34
35   IODs() = default;
36   friend std::ostream& operator«(std::ostream& _os, const IODs &_val);
37
38   void Clear() { IODsInternal.clear(); }
39
40   void AddIOD(const char *name , const IOD & module )
41     {
42     IODsInternal.insert(
43       IODMapType::value_type(name, module));
44     }
45   const IOD &GetIOD(const char *name)const
46 {
47     //return IODsInternal[name];
48     IODMapType::const_iterator it = IODsInternal.find( name );
49     assert( it != IODsInternal.end() );
50     assert( it->first == name );
51     return it->second;
52     }
53
54   typedef IODMapType::const_iterator IODMapTypeConstIterator;
55   IODMapTypeConstIterator Begin()const { return IODsInternal.begin(); }
56   IODMapTypeConstIterator End()const { return IODsInternal.end(); }
57
58 private:
59   IODMapType IODsInternal;
60 };
61 //----------------------------------------------------------------------------
62 inline std::ostream& operator«(std::ostream& _os, const IODs &_val)
63 {
64   IODs::IODMapType::const_iterator it = _val.IODsInternal.begin();
65   for(;it != _val.IODsInternal.end(); ++it)
66     {
67     const std::string &name = it->first;
68     const IOD &m = it->second;
69     _os « name « " " « m « '\n';
70     }
71
72   return _os;
73 }
74
75
76 } // end namespace gdcm
77
78 #endif //GDCMIODS_H
```

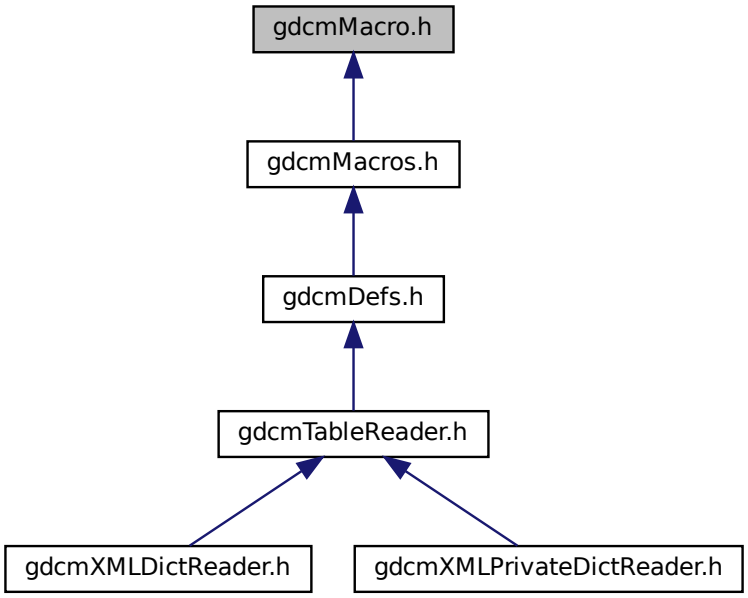
11.211 gdcmMacro.h File Reference

```
#include "gdcmTypes.h"
#include "gdcmTag.h"
#include "gdcmMacroEntry.h"
#include <map>
#include <vector>
```

Include dependency graph for gdcmMacro.h:



This graph shows which files directly or indirectly include this file:



Classes

- class gdcm::Macro

  Class for representing a Macro.

Namespaces

- namespace gdcm

Functions

- std::ostream & gdcm::operator<< (std::ostream &_os, const Macro &_val)

11.212 gdcmMacro.h

Go to the documentation of this file.

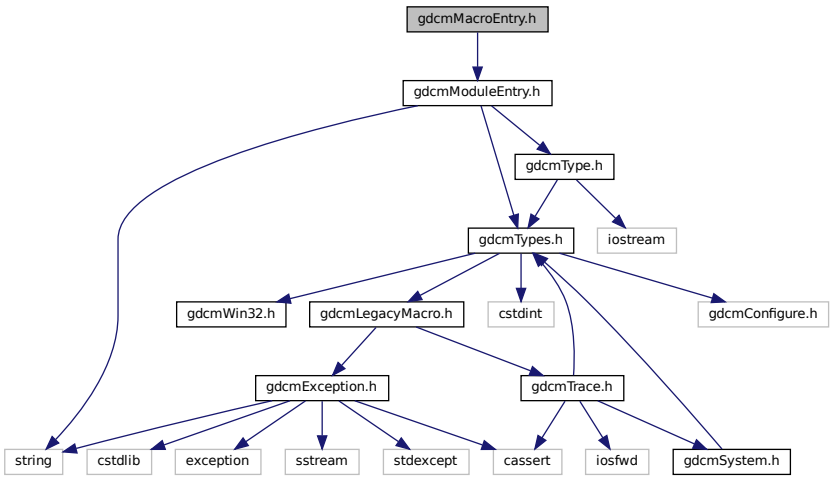
```
1 /*=========================================================================
2
3 Program:  GDCM (Grassroots DICOM). A DICOM library
4
5 Copyright (c) 2006-2011 Mathieu Malaterre
6 All rights reserved.
7 See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.
8
9 This software is distributed WITHOUT ANY WARRANTY; without even
10 the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
11 PURPOSE.  See the above copyright notice for more information.
12
13 =========================================================================*/
14 #ifndef GDCMMACRO_H
15 #define GDCMMACRO_H
16
17 #include "gdcmTypes.h"
18 #include "gdcmTag.h"
19 #include "gdcmMacroEntry.h"
20
21 #include <map>
22 #include <vector>
23
24 namespace gdcm
25 {
26
27 class DataSet;
28 class Usage;
36 class GDCM_EXPORT Macro
37 {
38 public:
39   typedef std::map<Tag, MacroEntry> MapModuleEntry;
40   typedef std::vector<std::string> ArrayIncludeMacrosType;
41
42   //typedef MapModuleEntry::const_iterator ConstIterator;
43   //typedef MapModuleEntry::iterator Iterator;
44   //ConstIterator Begin() const { return ModuleInternal.begin(); }
45   //Iterator Begin() { return ModuleInternal.begin(); }
46   //ConstIterator End() const { return ModuleInternal.end(); }
47   //Iterator End() { return ModuleInternal.end(); }
48
49   Macro() = default;
50   friend std::ostream& operator<<(std::ostream& _os, const Macro&_val);
51
52   void Clear() { ModuleInternal.clear(); }
53
55   void AddMacroEntry(const Tag& tag, const MacroEntry & module)
56     {
57     ModuleInternal.insert(
```

```
58       MapModuleEntry::value_type(tag, module));
59     }
60
63   bool FindMacroEntry(const Tag &tag) const;
64   const MacroEntry& GetMacroEntry(const Tag &tag) const;
65
66   void SetName( const char *name) { Name = name; }
67   const char *GetName()const { return Name.c_str(); }
68
69   // Verify will print on std::cerr for error
70   // Upon success will return true, false otherwise
71   bool Verify(const DataSet& ds, Usage const & usage) const;
72
73 private:
74   //Module &operator=(const Module &_val); // purposely not implemented
75   //Module(const Module &_val); // purposely not implemented
76
77   MapModuleEntry ModuleInternal;
78   std::string Name;
79 };
80 //-----------------------------------------------------------------------------
81 inline std::ostream& operator«(std::ostream& _os, const Macro &_val)
82 {
83   _os « _val.Name « '\n';
84   Macro::MapModuleEntry::const_iterator it = _val.ModuleInternal.begin();
85   for(;it != _val.ModuleInternal.end(); ++it)
86     {
87     const Tag &t = it->first;
88     const MacroEntry &de = it->second;
89     _os « t « " " « de « '\n';
90     }
91
92   return _os;
93 }
94
95 } // end namespace gdcm
96
97 #endif //GDCMMACRO_H
```

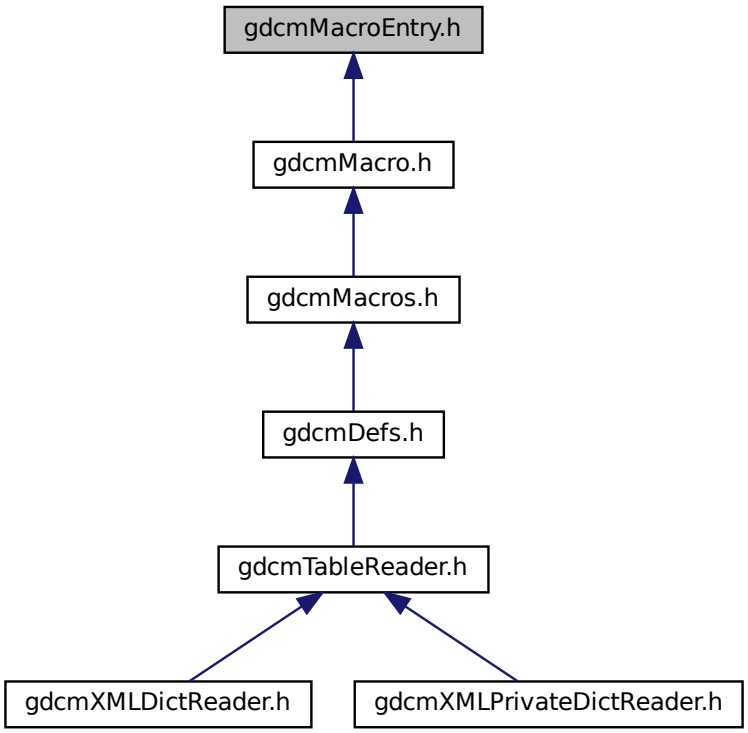
11.213 gdcmMacroEntry.h File Reference

`#include "gdcmModuleEntry.h"`

Include dependency graph for gdcmMacroEntry.h:



This graph shows which files directly or indirectly include this file:



Macros

- #define GDCMMACROENTRY_H

11.213.1 Macro Definition Documentation

11.213.1.1 GDCMMACROENTRY_H

```
#define GDCMMACROENTRY_H
```

11.214 gdcmMacroEntry.h

Go to the documentation of this file.

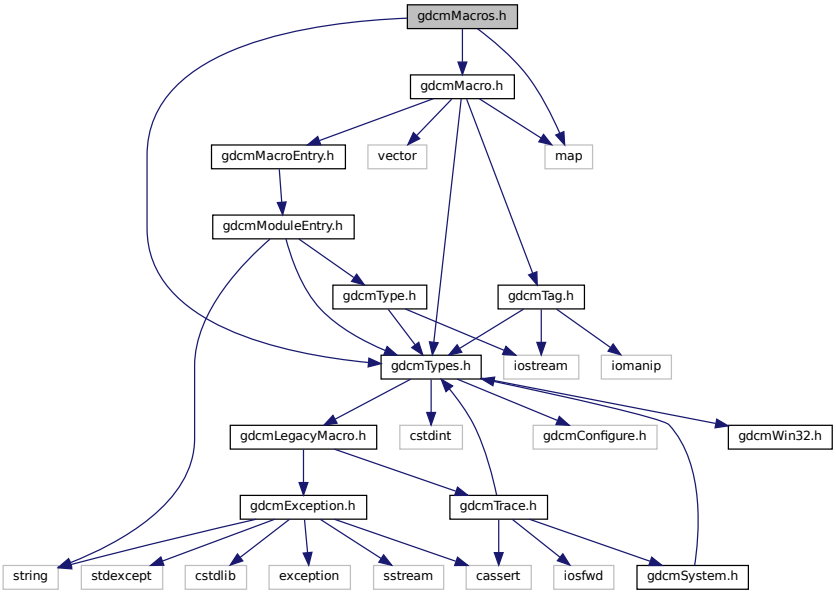
```
1 /*=========================================================================
2
3 Program:  GDCM (Grassroots DICOM). A DICOM library
4
5 Copyright (c) 2006-2011 Mathieu Malaterre
6 All rights reserved.
7 See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.
8
9 This software is distributed WITHOUT ANY WARRANTY; without even
10 the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
11 PURPOSE.  See the above copyright notice for more information.
12
13 =========================================================================*/
14 #if 0
15 #ifndef GDCMMACROENTRY_H
16 #define GDCMMACROENTRY_H
17
18 #include "gdcmTypes.h"
19 #include "gdcmType.h"
20
21 #include <string>
22
23 namespace gdcm
24 {
30 class GDCM_EXPORT MacroEntry
31 {
32 public:
33   MacroEntry(const char *name = "", const char *type = "3", const char *description =
      ""):Name(name)/*,Type(type)*/,DescriptionField(description) {
34     DataElementType = Type::GetTypeType(type);
35   }
36   virtual ~MacroEntry() {} // important
37   friend std::ostream& operator<<(std::ostream& _os, const MacroEntry &_val);
38
39   void SetName(const char *name) { Name = name; }
40   const char *GetName()const { return Name.c_str(); }
41
42   void SetType(const Type &type) { DataElementType = type; }
43   const Type &GetType()const { return DataElementType; }
44
45   /*
46 * WARNING: 'Description' is currently a std::string, but it might change in the future
47 * do not expect it to remain the same, and always use the ModuleEntry::Description typedef
48 * instead.
49 */
50   typedef std::string Description;
51   void SetDescription(const char *d) { DescriptionField = d; }
52   const Description & GetDescription()const { return DescriptionField; }
53
54 protected:
55   // PS 3.3 repeats the name of an attribute, but often contains typos
56   // for now we will not use this info, but instead access the DataDict instead
57   std::string Name;
58
59   // An attribute, encoded as a Data Element, may or may not be required in a
60   // Data Set, depending on that Attribute's Data Element Type.
61   Type DataElementType;
62
63   // TODO: for now contains the raw description (with enumerated values, defined terms...)
64   Description DescriptionField;
65 };
66 //-----------------------------------------------------------------------------
67 inline std::ostream& operator<<(std::ostream& _os, const MacroEntry &_val)
68 {
69   _os << _val.Name << "\t" << _val.DataElementType << "\t" << _val.DescriptionField;
70   return _os;
71 }
72
73
74 } // end namespace gdcm
75
76 #endif //GDCMMODULEENTRY_H
77 #endif
78
79 #ifndef GDCMMACROENTRY_H
80 #define GDCMMACROENTRY_H
```

```
81 #include "gdcmModuleEntry.h"
82 #endif
```

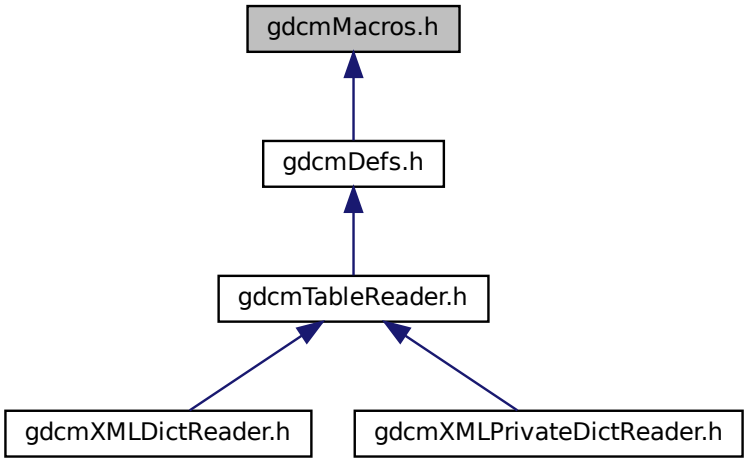
11.215 gdcmMacros.h File Reference

```
#include "gdcmTypes.h"
#include "gdcmMacro.h"
#include <map>
```

Include dependency graph for gdcmMacros.h:



This graph shows which files directly or indirectly include this file:



Classes

- class gdcm::Macros

  Class for representing a Modules.

Namespaces

- namespace gdcm

Functions

- std::ostream & gdcm::operator<< (std::ostream &_os, const Macros &_val)

11.216 gdcmMacros.h

Go to the documentation of this file.

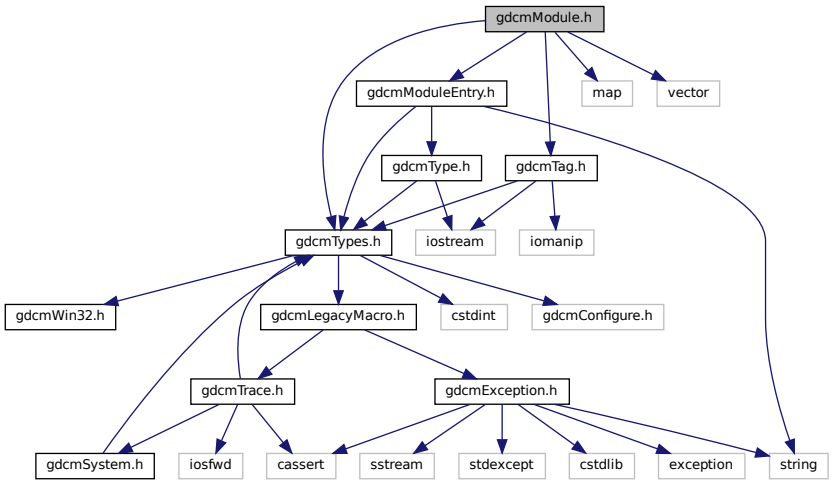
```
1 /*=========================================================================
2
3 Program:  GDCM (Grassroots DICOM). A DICOM library
4
5 Copyright (c) 2006-2011 Mathieu Malaterre
6 All rights reserved.
7 See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.
8
9 This software is distributed WITHOUT ANY WARRANTY; without even
10 the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
11 PURPOSE.  See the above copyright notice for more information.
```

```
12
13 =========================================================================*/
14 #ifndef GDCMMACROS_H
15 #define GDCMMACROS_H
16
17 #include "gdcmTypes.h"
18 #include "gdcmMacro.h"
19
20 #include <map>
21
22 namespace gdcm
23 {
29 class GDCM_EXPORT Macros
30 {
31 public:
32   typedef std::map<std::string, Macro> ModuleMapType;
33
34   Macros() = default;
35   friend std::ostream& operator«(std::ostream& _os, const Macros&_val);
36
37   void Clear() { ModulesInternal.clear(); }
38
39   // A Module is inserted based on it's ref
40   void AddMacro(const char *ref, const Macro & module )
41     {
42     assert( ref && *ref );
43     assert( ModulesInternal.find( ref ) == ModulesInternal.end() );
44     ModulesInternal.insert(
45       ModuleMapType::value_type(ref, module));
46     }
47   const Macro &GetMacro(const char *name)const
48 {
49     assert( name && *name );
50     ModuleMapType::const_iterator it = ModulesInternal.find( name );
51     assert( it != ModulesInternal.end() );
52     assert( it->first == name );
53     return it->second;
54     }
55
56   bool IsEmpty()const { return ModulesInternal.empty(); }
57
58 private:
59   ModuleMapType ModulesInternal;
60 };
61 //-----------------------------------------------------------------------------
62 inline std::ostream& operator«(std::ostream& _os, const Macros &_val)
63 {
64   Macros::ModuleMapType::const_iterator it = _val.ModulesInternal.begin();
65   for(;it != _val.ModulesInternal.end(); ++it)
66     {
67     const std::string &name = it->first;
68     const Macro &m = it->second;
69     _os « name « " " « m « '\n';
70     }
71
72   return _os;
73 }
74
75
76 } // end namespace gdcm
77
78 #endif //GDCMMODULES_H
```

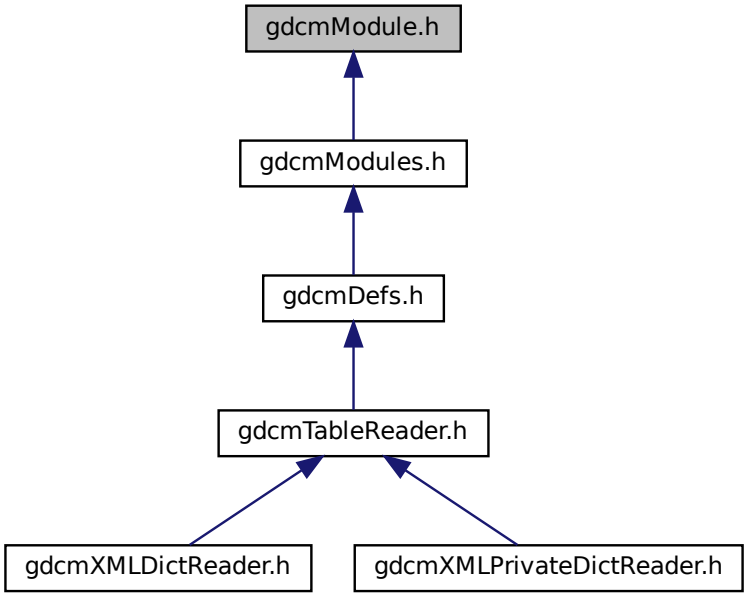
11.217 gdcmModule.h File Reference

```
#include "gdcmTypes.h"
#include "gdcmTag.h"
#include "gdcmModuleEntry.h"
#include <map>
#include <vector>
```

Include dependency graph for gdcmModule.h:



This graph shows which files directly or indirectly include this file:



Classes

- class gdcm::Module

Class for representing a Module.

Namespaces

- namespace gdcm

Functions

- std::ostream & gdcm::operator<< (std::ostream &_os, const Module &_val)

11.218 gdcmModule.h

Go to the documentation of this file.

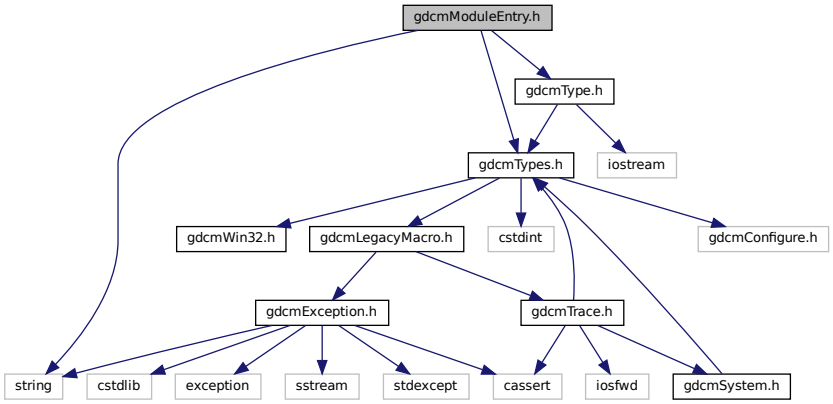
```
1 /*=========================================================================
2
3 Program:  GDCM (Grassroots DICOM). A DICOM library
4
5 Copyright (c) 2006-2011 Mathieu Malaterre
6 All rights reserved.
7 See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.
8
9 This software is distributed WITHOUT ANY WARRANTY; without even
10 the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
11 PURPOSE.  See the above copyright notice for more information.
12
13 =========================================================================*/
14 #ifndef GDCMMODULE_H
15 #define GDCMMODULE_H
16
17 #include "gdcmTypes.h"
18 #include "gdcmTag.h"
19 #include "gdcmModuleEntry.h"
20
21 #include <map>
22 #include <vector>
23
24 namespace gdcm
25 {
26
27 class DataSet;
28 class Usage;
29 class Macros;
37 class GDCM_EXPORT Module
38 {
39 public:
40   typedef std::map<Tag, ModuleEntry> MapModuleEntry;
41   typedef std::vector<std::string> ArrayIncludeMacrosType;
42
43   //typedef MapModuleEntry::const_iterator ConstIterator;
44   //typedef MapModuleEntry::iterator Iterator;
45   //ConstIterator Begin() const { return ModuleInternal.begin(); }
46   //Iterator Begin() { return ModuleInternal.begin(); }
47   //ConstIterator End() const { return ModuleInternal.end(); }
48   //Iterator End() { return ModuleInternal.end(); }
49
50   Module() = default;
51   friend std::ostream& operator<<(std::ostream& _os, const Module &_val);
52
53   void Clear() { ModuleInternal.clear(); }
54
56   void AddModuleEntry(const Tag& tag, const ModuleEntry & module)
57     {
58     ModuleInternal.insert(
59       MapModuleEntry::value_type(tag, module));
60     }
61
62   void AddMacro(const char *include)
```

```
63     {
64     ArrayIncludeMacros.push_back( include );
65     }
66
69   bool FindModuleEntryInMacros(Macros const &macros, const Tag &tag) const;
70   const ModuleEntry& GetModuleEntryInMacros(Macros const &macros, const Tag &tag) const;
71
72   void SetName( const char *name) { Name = name; }
73   const char *GetName()const { return Name.c_str(); }
74
75   // Verify will print on std::cerr for error
76   // Upon success will return true, false otherwise
77   bool Verify(const DataSet& ds, Usage const & usage) const;
78
79 private:
80   //Module &operator=(const Module &_val); // purposely not implemented
81   //Module(const Module &_val); // purposely not implemented
82
83   MapModuleEntry ModuleInternal;
84   std::string Name;
85   ArrayIncludeMacrosType ArrayIncludeMacros;
86 };
87 //-----------------------------------------------------------------------------
88 inline std::ostream& operator«(std::ostream& _os, const Module &_val)
89 {
90   _os « _val.Name « '\n';
91   Module::MapModuleEntry::const_iterator it = _val.ModuleInternal.begin();
92   for(;it != _val.ModuleInternal.end(); ++it)
93     {
94     const Tag &t = it->first;
95     const ModuleEntry &de = it->second;
96     _os « t « " " « de « '\n';
97     }
98
99   return _os;
100 }
101
102 } // end namespace gdcm
103
104 #endif //GDCMMODULE_H
```

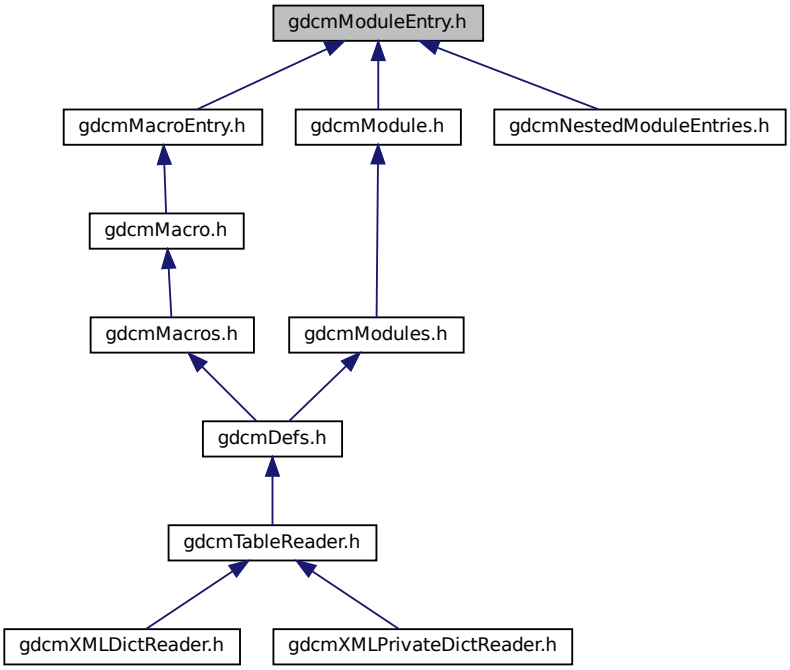
11.219 gdcmModuleEntry.h File Reference

```
#include "gdcmTypes.h"
#include "gdcmType.h"
#include <string>
```

Include dependency graph for gdcmModuleEntry.h:



This graph shows which files directly or indirectly include this file:



Classes

- class gdcm::ModuleEntry

  Class for representing a ModuleEntry.

Namespaces

- namespace gdcm

Typedefs

- typedef ModuleEntry gdcm::MacroEntry

Functions

- std::ostream & gdcm::operator<< (std::ostream &_os, const ModuleEntry &_val)

11.220 gdcmModuleEntry.h

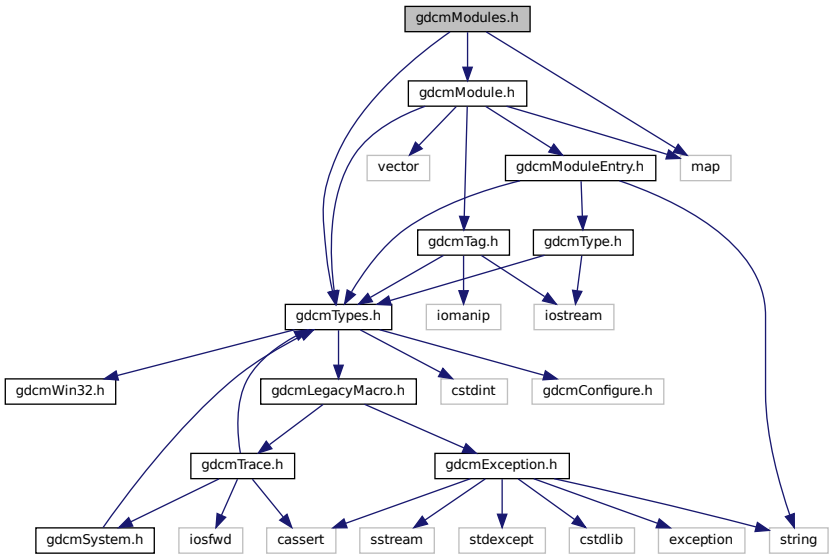
Go to the documentation of this file.

```
1 /*=========================================================================
2
3 Program:  GDCM (Grassroots DICOM). A DICOM library
4
5 Copyright (c) 2006-2011 Mathieu Malaterre
6 All rights reserved.
7 See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.
8
9 This software is distributed WITHOUT ANY WARRANTY; without even
10 the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
11 PURPOSE.  See the above copyright notice for more information.
12
13 =========================================================================*/
14 #ifndef GDCMMODULEENTRY_H
15 #define GDCMMODULEENTRY_H
16
17 #include "gdcmTypes.h"
18 #include "gdcmType.h"
19
20 #include <string>
21
22 namespace gdcm
23 {
29 class GDCM_EXPORT ModuleEntry
30 {
31 public:
32   ModuleEntry(const char *name = "", const char *type = "3", const char *description =
      "")://Name(name)/*,Type(type)*/,DescriptionField(description) {
33     DataElementType = Type::GetTypeType(type);
34   }
35   virtual ~ModuleEntry() = default; // important
36   friend std::ostream& operator<<(std::ostream& _os, const ModuleEntry &_val);
37
38   void SetName(const char *name) { Name = name; }
39   const char *GetName()const { return Name.c_str(); }
40
41   void SetType(const Type &type) { DataElementType = type; }
42   const Type &GetType()const { return DataElementType; }
43
44   /*
45 * WARNING: 'Description' is currently a std::string, but it might change in the future
46 * do not expect it to remain the same, and always use the ModuleEntry::Description typedef
47 * instead.
48 */
49   typedef std::string Description;
50   void SetDescription(const char *d) { DescriptionField = d; }
51   const Description & GetDescription()const { return DescriptionField; }
52
53 protected:
54   // PS 3.3 repeats the name of an attribute, but often contains typos
55   // for now we will not use this info, but instead access the DataDict instead
56   std::string Name;
57
58   // An attribute, encoded as a Data Element, may or may not be required in a
59   // Data Set, depending on that Attribute's Data Element Type.
60   Type DataElementType;
61
62   // TODO: for now contains the raw description (with enumerated values, defined terms...)
63   Description DescriptionField;
64 };
65 //-----------------------------------------------------------------------------
66 inline std::ostream& operator<<(std::ostream& _os, const ModuleEntry &_val)
67 {
68   _os << _val.Name << "\t" << _val.DataElementType << "\t" << _val.DescriptionField;
69   return _os;
70 }
71
72 typedef ModuleEntry MacroEntry;
73
74
75 } // end namespace gdcm
76
77 #endif //GDCMMODULEENTRY_H
```

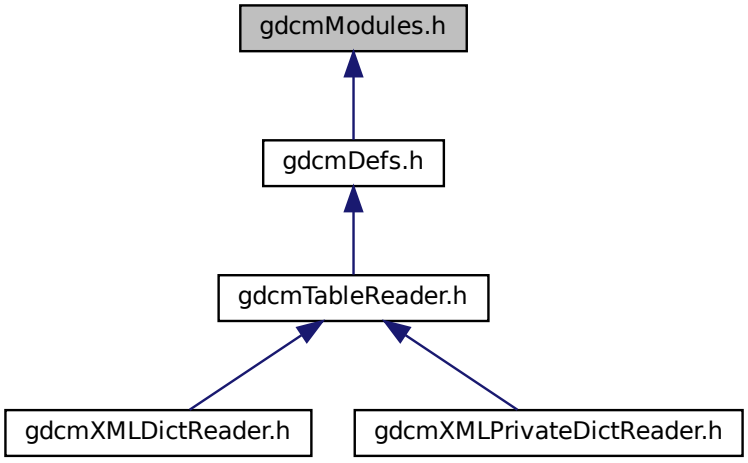
11.221 gdcmModules.h File Reference

```
#include "gdcmTypes.h"
#include "gdcmModule.h"
#include <map>
```

Include dependency graph for gdcmModules.h:



This graph shows which files directly or indirectly include this file:



Classes

- class gdcm::Modules

  Class for representing a Modules.

Namespaces

- namespace gdcm

Functions

- std::ostream & gdcm::operator<< (std::ostream &_os, const Modules &_val)

11.222 gdcmModules.h

Go to the documentation of this file.

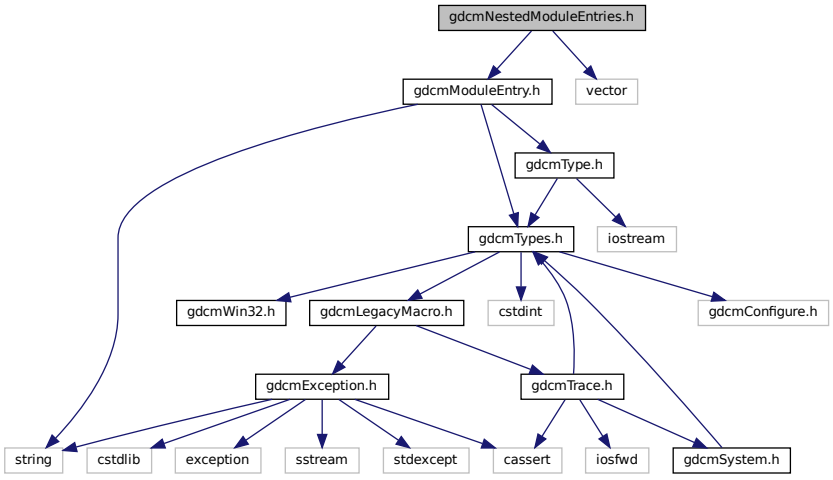
```
1 /*=========================================================================
2
3 Program:  GDCM (Grassroots DICOM). A DICOM library
4
5 Copyright (c) 2006-2011 Mathieu Malaterre
6 All rights reserved.
7 See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.
8
9 This software is distributed WITHOUT ANY WARRANTY; without even
10 the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
11 PURPOSE.  See the above copyright notice for more information.
12
13 =========================================================================*/
14 #ifndef GDCMMODULES_H
15 #define GDCMMODULES_H
16
17 #include "gdcmTypes.h"
18 #include "gdcmModule.h"
19
20 #include <map>
21
22 namespace gdcm
23 {
29 class GDCM_EXPORT Modules
30 {
31 public:
32   typedef std::map<std::string, Module> ModuleMapType;
33
34   Modules() = default;
35   friend std::ostream& operator<<(std::ostream& _os, const Modules &_val);
36
37   void Clear() { ModulesInternal.clear(); }
38
39   // A Module is inserted based on it's ref
40   void AddModule(const char *ref, const Module & module )
41     {
42     assert( ref && *ref );
43     assert( ModulesInternal.find( ref ) == ModulesInternal.end() );
44     ModulesInternal.insert(
45       ModuleMapType::value_type(ref, module));
46     }
47   const Module &GetModule(const char *name)const
48 {
49     assert( name && *name );
50     ModuleMapType::const_iterator it = ModulesInternal.find( name );
51     assert( it != ModulesInternal.end() );
52     assert( it->first == name );
53     return it->second;
54     }
```

```
55
56   bool IsEmpty()const { return ModulesInternal.empty(); }
57
58 private:
59   ModuleMapType ModulesInternal;
60 };
61 //-----------------------------------------------------------------------------
62 inline std::ostream& operator«(std::ostream& _os, const Modules &_val)
63 {
64   Modules::ModuleMapType::const_iterator it = _val.ModulesInternal.begin();
65   for(;it != _val.ModulesInternal.end(); ++it)
66     {
67     const std::string &name = it->first;
68     const Module &m = it->second;
69     _os « name « " " « m « '\n';
70     }
71
72   return _os;
73 }
74
75
76
77 } // end namespace gdcm
78
79 #endif //GDCMMODULES_H
```

11.223 gdcmNestedModuleEntries.h File Reference

```
#include "gdcmModuleEntry.h"
#include <vector>
```

Include dependency graph for gdcmNestedModuleEntries.h:



Classes

- class gdcm::NestedModuleEntries

  Class for representing a NestedModuleEntries.

Namespaces

- namespace gdcm

Typedefs

- typedef NestedModuleEntries gdcm::NestedMacroEntries

Functions

- std::ostream & gdcm::operator<< (std::ostream &_os, const NestedModuleEntries &_val)

11.224 gdcmNestedModuleEntries.h

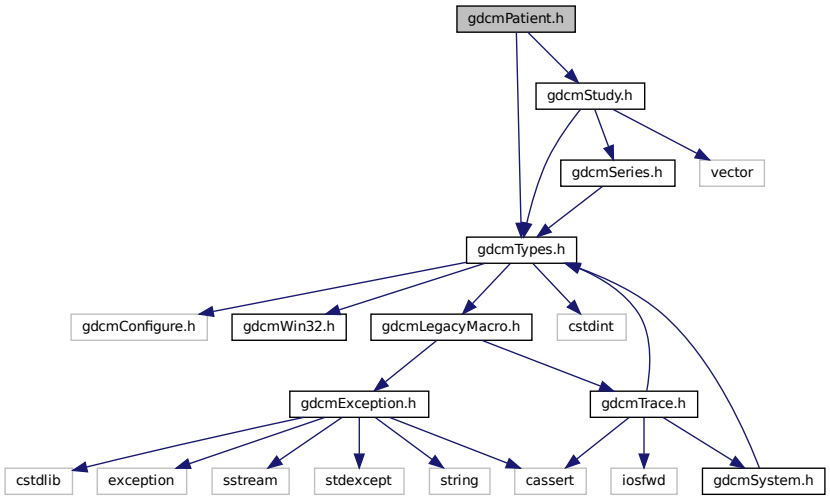
Go to the documentation of this file.

```
1 /*=========================================================================
2
3 Program:  GDCM (Grassroots DICOM). A DICOM library
4
5 Copyright (c) 2006-2011 Mathieu Malaterre
6 All rights reserved.
7 See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.
8
9 This software is distributed WITHOUT ANY WARRANTY; without even
10 the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
11 PURPOSE.  See the above copyright notice for more information.
12
13 =========================================================================*/
14 #ifndef GDCMNESTEDMODULEENTRIES_H
15 #define GDCMNESTEDMODULEENTRIES_H
16
17 #include "gdcmModuleEntry.h"
18 #include <vector>
19
20 namespace gdcm
21 {
27 class GDCM_EXPORT NestedModuleEntries :  public ModuleEntry
28 {
29 public:
30   NestedModuleEntries(const char *name = "", const char *type = "3", const char *description =
       ""):ModuleEntry(name,type,description) { }
31   friend std::ostream& operator«(std::ostream& _os, const NestedModuleEntries &_val);
32
33   typedef std::vector<ModuleEntry>::size_type SizeType;
34   SizeType GetNumberOfModuleEntries() { return ModuleEntriesList.size(); }
35
36   const ModuleEntry &GetModuleEntry(SizeType idx)const { return ModuleEntriesList[idx]; }
37   ModuleEntry &GetModuleEntry(SizeType idx) { return ModuleEntriesList[idx]; }
38
39   void AddModuleEntry(const ModuleEntry &me) { ModuleEntriesList.push_back( me ); }
40
41 private:
42   std::vector<ModuleEntry> ModuleEntriesList;
43 };
44 //-----------------------------------------------------------------------------
45 inline std::ostream& operator«(std::ostream& _os, const NestedModuleEntries &_val)
46 {
47   _os « "Nested:" « _val.Name « "\t" « _val.DataElementType « "\t" « _val.DescriptionField;
48   return _os;
49 }
50
51 typedef NestedModuleEntries NestedMacroEntries;
52
53
54 } // end namespace gdcm
55
56 #endif //GDCMNESTEDMODULEENTRIES_H
```

11.225 gdcmPatient.h File Reference

```
#include "gdcmTypes.h"
#include "gdcmStudy.h"
```

Include dependency graph for gdcmPatient.h:



Classes

- class gdcm::Patient

  See PS 3.3 - 2007 DICOM MODEL OF THE REAL-WORLD, p 54.

Namespaces

- namespace gdcm

11.226 gdcmPatient.h

Go to the documentation of this file.

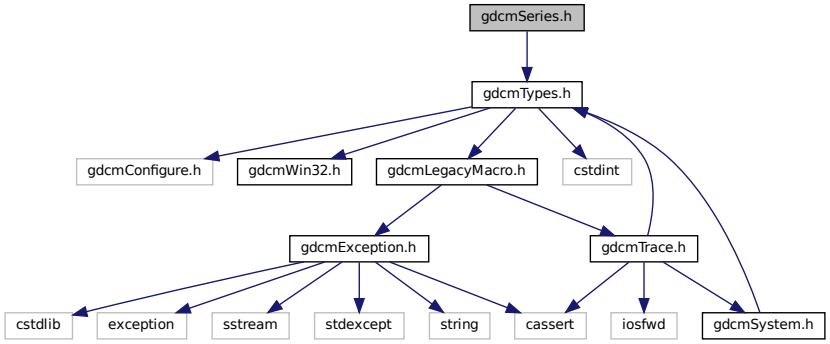
```
1 /*=========================================================================
2
3 Program:  GDCM (Grassroots DICOM). A DICOM library
4
5 Copyright (c) 2006-2011 Mathieu Malaterre
6 All rights reserved.
7 See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.
8
9 This software is distributed WITHOUT ANY WARRANTY; without even
10 the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
11 PURPOSE.  See the above copyright notice for more information.
12
13 =========================================================================*/
```

```
14 #ifndef GDCMPATIENT_H
15 #define GDCMPATIENT_H
16
17 #include "gdcmTypes.h"
18 #include "gdcmStudy.h"
19
20 namespace gdcm
21 {
27 class GDCM_EXPORT Patient
28 {
29 public:
30   Patient() = default;
31
32 private:
33   std::vector<Study> StudyList;
34 };
35
36 } // end namespace gdcm
37
38 #endif //GDCMPATIENT_H
```

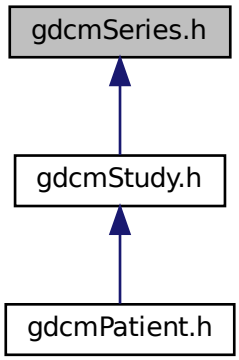
11.227 gdcmSeries.h File Reference

`#include "gdcmTypes.h"`

Include dependency graph for gdcmSeries.h:



This graph shows which files directly or indirectly include this file:



Classes

- class gdcm::Series
  Series.

Namespaces

- namespace gdcm

11.228 gdcmSeries.h

Go to the documentation of this file.

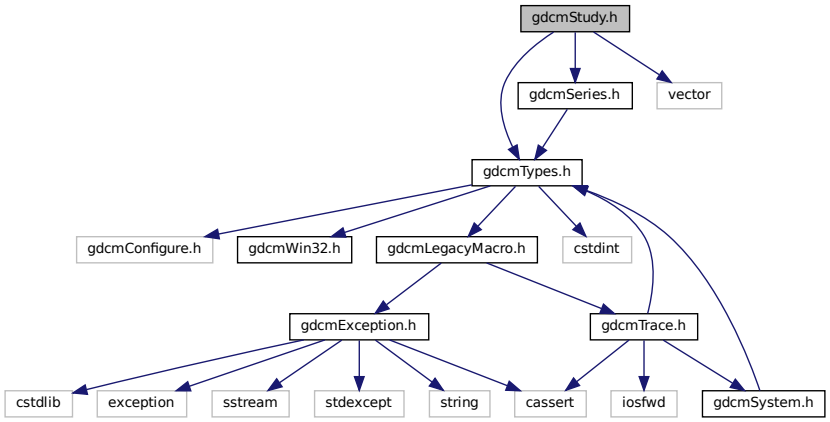
```
1 /*=========================================================================
2
3 Program:  GDCM (Grassroots DICOM). A DICOM library
4
5 Copyright (c) 2006-2011 Mathieu Malaterre
6 All rights reserved.
7 See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.
8
9 This software is distributed WITHOUT ANY WARRANTY; without even
10 the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
11 PURPOSE.  See the above copyright notice for more information.
12
13 =========================================================================*/
14 #ifndef GDCMSERIES_H
15 #define GDCMSERIES_H
16
17 #include "gdcmTypes.h"
18
19 namespace gdcm
20 {
24 class GDCM_EXPORT Series
25 {
```

```
26 public:
27   Series() = default;
28 private:
29   // Image, Waveform...
30 };
31
32 } // end namespace gdcm
33
34 #endif //GDCMSERIES_H
```

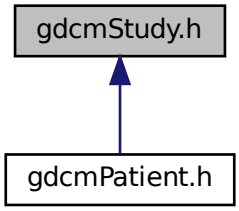
11.229 gdcmStudy.h File Reference

```
#include "gdcmTypes.h"
#include "gdcmSeries.h"
#include <vector>
```

Include dependency graph for gdcmStudy.h:



This graph shows which files directly or indirectly include this file:



Classes

- class gdcm::Study

  Study.

Namespaces

- namespace gdcm

11.230 gdcmStudy.h

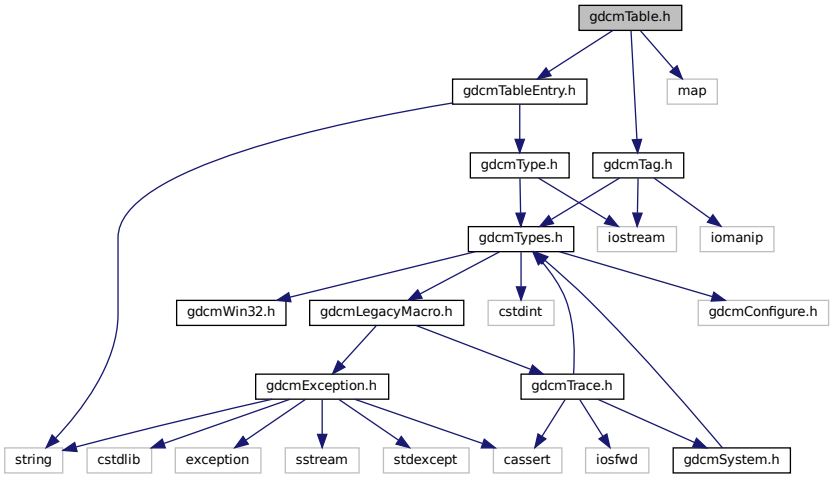
Go to the documentation of this file.

```
1 /*=========================================================================
2
3 Program:  GDCM (Grassroots DICOM). A DICOM library
4
5 Copyright (c) 2006-2011 Mathieu Malaterre
6 All rights reserved.
7 See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.
8
9 This software is distributed WITHOUT ANY WARRANTY; without even
10 the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
11 PURPOSE.  See the above copyright notice for more information.
12
13 =========================================================================*/
14 #ifndef GDCMSTUDY_H
15 #define GDCMSTUDY_H
16
17 #include "gdcmTypes.h"
18 #include "gdcmSeries.h"
19
20 #include <vector>
21
22 namespace gdcm
23 {
27 class GDCM_EXPORT Study
28 {
29 public:
30   Study() = default;
31 private:
32   std::vector<Series> SeriesList;
33 };
34
35 } // end namespace gdcm
36
37 #endif //GDCMSTUDY_H
```

11.231 gdcmTable.h File Reference

```
#include "gdcmTableEntry.h"
#include "gdcmTag.h"
```

`#include <map>`
Include dependency graph for gdcmTable.h:



Classes

- class gdcm::Table

  Table.

Namespaces

- namespace gdcm

11.232 gdcmTable.h

Go to the documentation of this file.

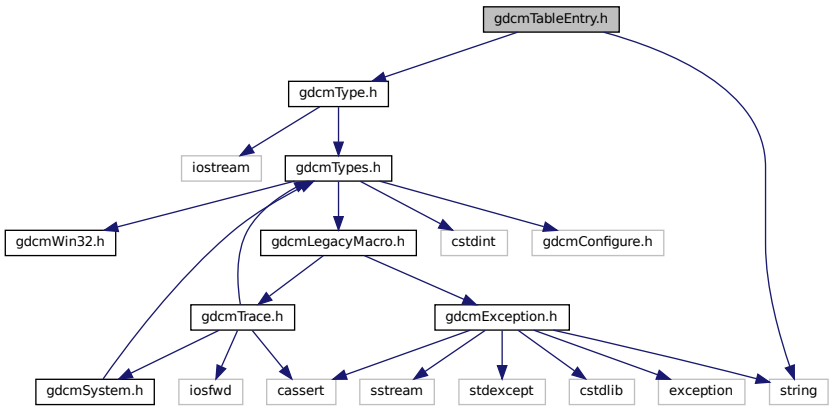
```
1 /*=========================================================================
2
3 Program:  GDCM (Grassroots DICOM). A DICOM library
4
5 Copyright (c) 2006-2011 Mathieu Malaterre
6 All rights reserved.
7 See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.
8
9 This software is distributed WITHOUT ANY WARRANTY; without even
10 the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
11 PURPOSE.  See the above copyright notice for more information.
12
13 =========================================================================*/
14 #ifndef GDCMTABLE_H
15 #define GDCMTABLE_H
16
17 #include "gdcmTableEntry.h"
18 #include "gdcmTag.h"
19
20 #include <map>
```

```
21
22 namespace gdcm
23 {
24
28 class Table
29 {
30 public:
31   typedef std::map<Tag, TableEntry> MapTableEntry;
32   Table() = default;
33   ~Table() = default;
34   Table &operator=(const Table &_val) = delete;
35   Table(const Table&_val) = delete;
36
37   friend std::ostream& operator«(std::ostream& _os, const Table &_val);
38
39   void InsertEntry(Tag const &tag, TableEntry const &te)
40     {
41 #ifndef NDEBUG
42     MapTableEntry::size_type s = TableInternal.size();
43 #endif
44     TableInternal.insert(
45       MapTableEntry::value_type(tag, te));
46     assert( s < TableInternal.size() );
47     }
48
49   const TableEntry &GetTableEntry(const Tag &tag)const
50 {
51     MapTableEntry::const_iterator it =
52       TableInternal.find(tag);
53     if (it == TableInternal.end())
54       {
55       assert( 0 && "Impossible" );
56       return GetTableEntry(Tag(0,0));
57       }
58     return it->second;
59     }
60
61   MapTableEntry TableInternal;
62 };
63
64 } // end namespace gdcm
65
66 #endif //GDCMTABLE_H
```

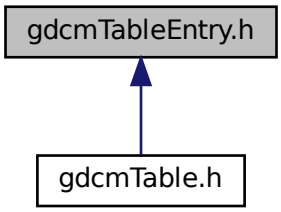
11.233 gdcmTableEntry.h File Reference

```
#include "gdcmType.h"
#include <string>
```

Include dependency graph for gdcmTableEntry.h:



This graph shows which files directly or indirectly include this file:



Classes

- class gdcm::TableEntry

  TableEntry.

Namespaces

- namespace gdcm

11.234 gdcmTableEntry.h

Go to the documentation of this file.

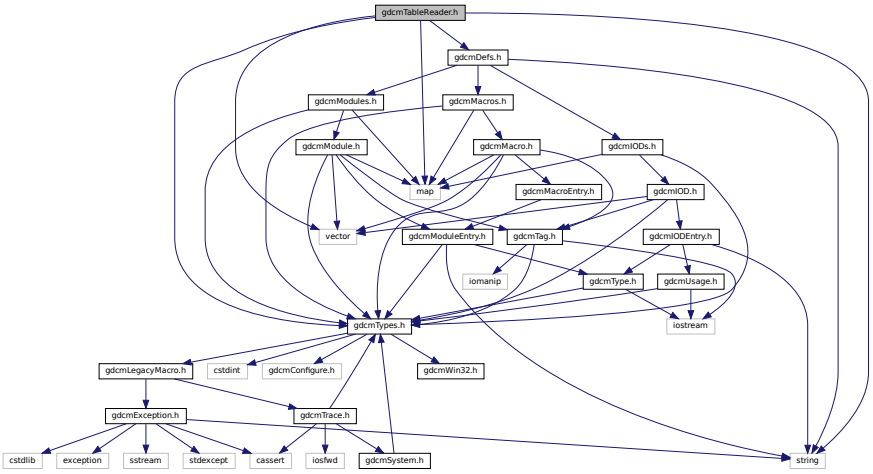
```
1 /*=========================================================================
2
3 Program:  GDCM (Grassroots DICOM). A DICOM library
4
5 Copyright (c) 2006-2011 Mathieu Malaterre
6 All rights reserved.
7 See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.
8
9 This software is distributed WITHOUT ANY WARRANTY; without even
10 the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
11 PURPOSE.  See the above copyright notice for more information.
12
13 =========================================================================*/
14 #ifndef GDCMTABLEENTRY_H
15 #define GDCMTABLEENTRY_H
16
17 #include "gdcmType.h"
18
19 #include <string>
20
21 namespace gdcm
22 {
23
27 class TableEntry
28 {
29 public:
30   TableEntry(const char *attribute = nullptr,
31     Type const &type = Type(), const char * des = nullptr ) :
32     Attribute(attribute ?  attribute :  ""),TypeField(type),Description(des ?  des :  "") {}
```

```
33   ~TableEntry() = default;
34
35 private:
36   std::string Attribute;
37   Type TypeField;
38   std::string Description;
39 };
40
41 } // end namespace gdcm
42
43 #endif //GDCMTABLEENTRY_H
```

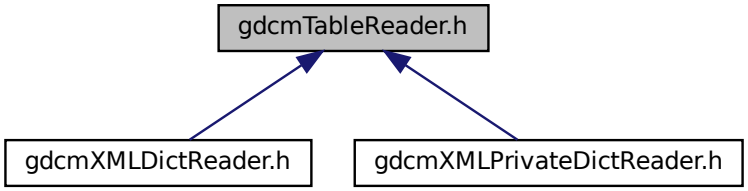
11.235 gdcmTableReader.h File Reference

```
#include "gdcmTypes.h"
#include "gdcmDefs.h"
#include <string>
#include <vector>
#include <map>
```

Include dependency graph for gdcmTableReader.h:



This graph shows which files directly or indirectly include this file:



Classes

- class gdcm::TableReader

  Class for representing a TableReader.

Namespaces

- namespace gdcm

11.236 gdcmTableReader.h

Go to the documentation of this file.

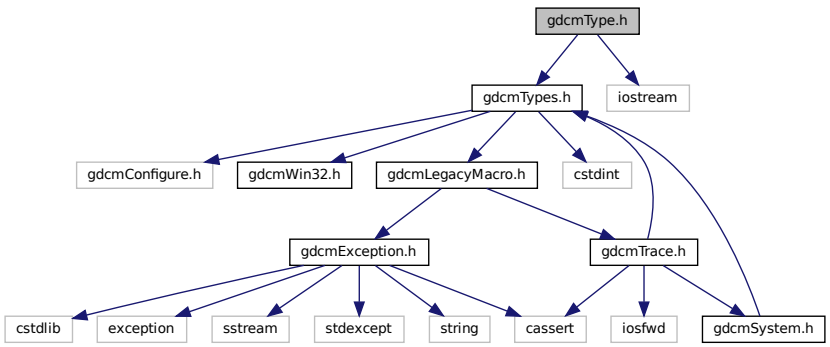
```
1 /*=========================================================================
2
3 Program:  GDCM (Grassroots DICOM). A DICOM library
4
5 Copyright (c) 2006-2011 Mathieu Malaterre
6 All rights reserved.
7 See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.
8
9 This software is distributed WITHOUT ANY WARRANTY; without even
10 the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
11 PURPOSE.  See the above copyright notice for more information.
12
13 =========================================================================*/
14 #ifndef GDCMTABLEREADER_H
15 #define GDCMTABLEREADER_H
16
17 #include "gdcmTypes.h"
18 #include "gdcmDefs.h"
19 //#include "gdcmModule.h"
20 //#include "gdcmIOD.h"
21 //#include "gdcmIODs.h"
22 //#include "gdcmModules.h"
23
24 #include <string>
25 #include <vector>
26 #include <map>
27
28 namespace gdcm
29 {
34 class GDCM_EXPORT TableReader
35 {
36 public:
37   TableReader(Defs &defs):CurrentDefs(defs),ParsingModule(false),ParsingModuleEntry(false),
38   ParsingModuleEntryDescription(false),
39   ParsingMacro(false),
40   ParsingMacroEntry(false),
41   ParsingMacroEntryDescription(false),
42   ParsingIOD(false),
43   ParsingIODEntry(false),
44   Description() {}
45   virtual ~TableReader() = default;
46
47   // Set/Get filename
48   void SetFilename(const char *filename) { Filename = filename; }
49   const char *GetFilename() { return Filename.c_str(); }
50
51   int Read();
52
53 //protected:
54   // You need to override those function in your subclasses:
55   virtual void StartElement(const char *name, const char **atts);
56   virtual void EndElement(const char *name);
57   virtual void CharacterDataHandler(const char *data, int length);
58
59 void HandleModuleEntry(const char **atts);
60 void HandleModule(const char **atts);
61 void HandleModuleEntryDescription(const char **atts);
```

```
62 void HandleMacroEntry(const char **atts);
63 void HandleMacro(const char **atts);
64 void HandleMacroEntryDescription(const char **atts);
65 void HandleModuleInclude(const char **atts);
66 void HandleIODEntry(const char **atts);
67 void HandleIOD(const char **atts);
68
69   //const Modules & GetModules() const { return CurrentModules; }
70   //const Macros & GetMacros() const { return CurrentMacros; }
71   //const IODs & GetIODs() const { return CurrentIODs; }
72   const Defs & GetDefs()const { return CurrentDefs; }
73
74 private:
75   std::string Filename;
76   Defs &CurrentDefs;
77   //Macros CurrentMacros;
78   //Modules CurrentModules;
79   //IODs CurrentIODs;
80   Macro CurrentMacro;
81   Module CurrentModule;
82   IOD CurrentIOD;
83   MacroEntry CurrentMacroEntry;
84   ModuleEntry CurrentModuleEntry;
85   IODEntry CurrentIODEntry;
86   std::string CurrentModuleName;
87   std::string CurrentModuleRef;
88   std::string CurrentMacroRef;
89   bool ParsingModule;
90   bool ParsingModuleEntry;
91   bool ParsingModuleEntryDescription;
92   bool ParsingMacro;
93   bool ParsingMacroEntry;
94   bool ParsingMacroEntryDescription;
95   bool ParsingIOD;
96   bool ParsingIODEntry;
97   Tag CurrentTag;
98   std::string Description;
99 };
100
101 } // end namespace gdcm
102
103 #endif //GDCMTABLEREADER_H
```

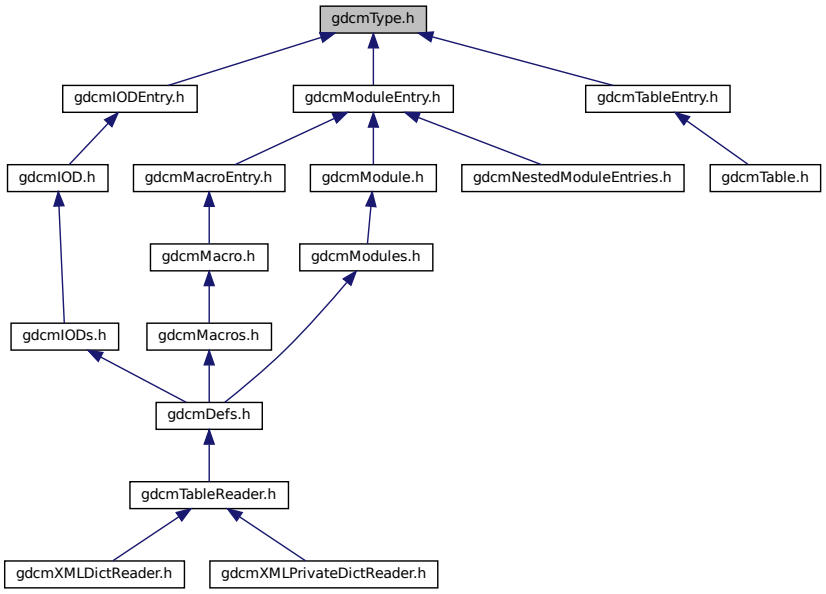
11.237 gdcmType.h File Reference

```
#include "gdcmTypes.h"
#include <iostream>
```

Include dependency graph for gdcmType.h:



This graph shows which files directly or indirectly include this file:



Classes

- class gdcm::Type

  Type.

Namespaces

- namespace gdcm

Functions

- std::ostream & gdcm::operator<< (std::ostream &_os, const Type &val)

11.238 gdcmType.h

Go to the documentation of this file.

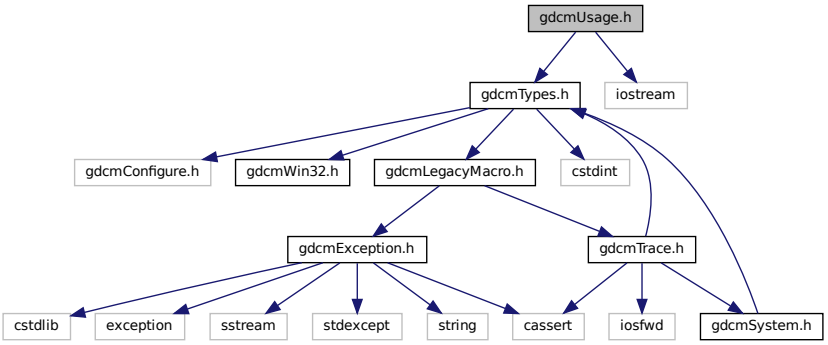
```
1 /*=========================================================================
2
3 Program:  GDCM (Grassroots DICOM). A DICOM library
4
5 Copyright (c) 2006-2011 Mathieu Malaterre
6 All rights reserved.
7 See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.
8
```

```
9 This software is distributed WITHOUT ANY WARRANTY; without even
10 the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
11 PURPOSE.  See the above copyright notice for more information.
12
13 =========================================================================*/
14
15 #ifndef GDCMTYPE_H
16 #define GDCMTYPE_H
17
18 #include "gdcmTypes.h"
19
20 #include <iostream>
21
22 namespace gdcm
23 {
24
41 class GDCM_EXPORT Type
42 {
43 public:
44   typedef enum {
45     T1 = 0,
46     T1C,
47     T2,
48     T2C,
49     T3,
50     UNKNOWN
51   } TypeType;
52
53   Type(TypeType type = UNKNOWN) : TypeField(type) { }
54
55   operator TypeType ()const { return TypeField; }
56   friend std::ostream &operator«(std::ostream &os, const Type &vr);
57
58   static const char *GetTypeString(TypeType type);
59   static TypeType GetTypeType(const char *type);
60
61 private:
62   TypeType TypeField;
63 };
64 //-----------------------------------------------------------------------------
65 inline std::ostream &operator«(std::ostream &_os, const Type &val)
66 {
67   _os « Type::GetTypeString(val.TypeField);
68   return _os;
69 }
70
71 } // end namespace gdcm
72
73 #endif //GDCMTYPE_H
```

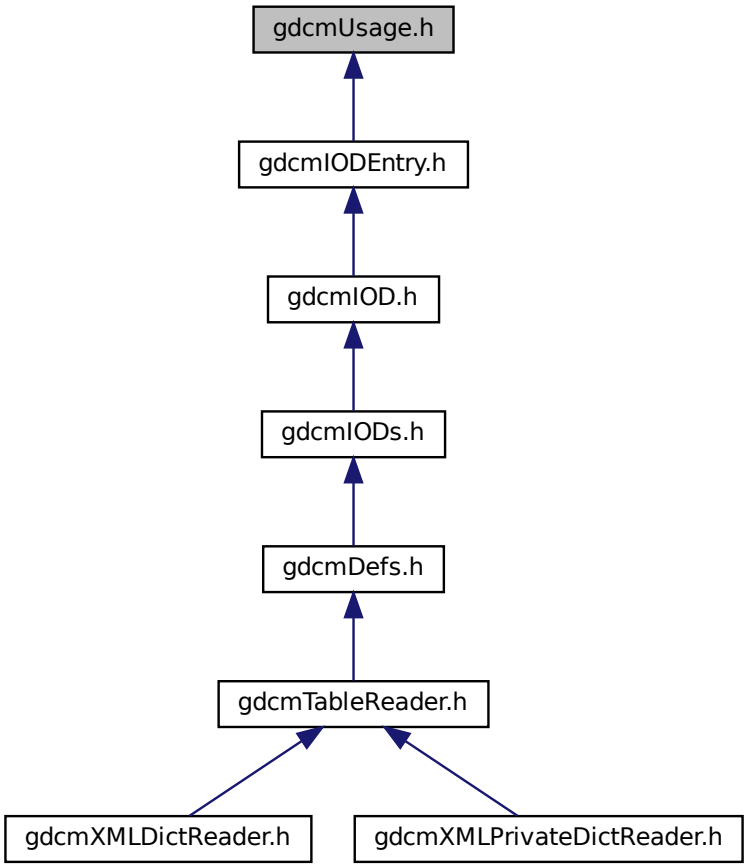
11.239 gdcmUsage.h File Reference

```
#include "gdcmTypes.h"
#include <iostream>
```

Include dependency graph for gdcmUsage.h:



This graph shows which files directly or indirectly include this file:



Classes

- class gdcm::Usage

  Usage.

Namespaces

- namespace gdcm

Functions

- std::ostream & gdcm::operator<< (std::ostream &_os, const Usage &val)

11.240 gdcmUsage.h

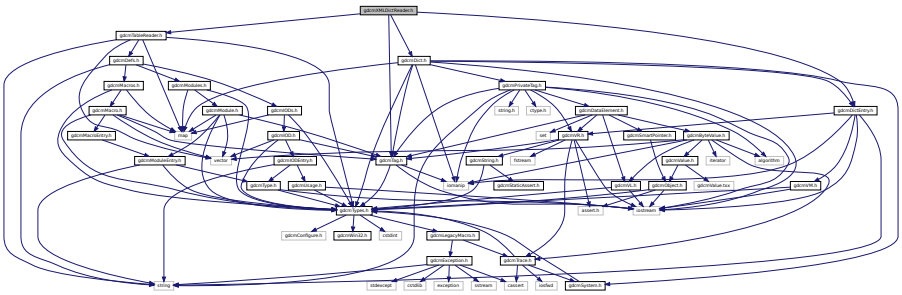
Go to the documentation of this file.

```
1 /*=========================================================================
2
3 Program:  GDCM (Grassroots DICOM). A DICOM library
4
5 Copyright (c) 2006-2011 Mathieu Malaterre
6 All rights reserved.
7 See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.
8
9 This software is distributed WITHOUT ANY WARRANTY; without even
10 the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
11 PURPOSE.  See the above copyright notice for more information.
12
13 =========================================================================*/
14 #ifndef GDCMUSAGE_H
15 #define GDCMUSAGE_H
16
17 #include "gdcmTypes.h"
18
19 #include <iostream>
20
21 namespace gdcm
22 {
23
48 class GDCM_EXPORT Usage
49 {
50 public:
51   typedef enum {
52     Mandatory, // (see A.1.3.1) , abbreviated M
53     Conditional, // (see A.1.3.2) , abbreviated C
54     UserOption, // (see A.1.3.3) , abbreviated U
55     Invalid
56   } UsageType;
57
58   Usage(UsageType type = Invalid) :  UsageField(type) { }
59
60   operator UsageType ()const { return UsageField; }
61   friend std::ostream &operator«(std::ostream &os, const Usage &vr);
62
63   static const char *GetUsageString(UsageType type);
64   static UsageType GetUsageType(const char *type);
65
66 private:
67   UsageType UsageField;
68 };
69 //-----------------------------------------------------------------------------
70 inline std::ostream &operator«(std::ostream &_os, const Usage &val)
71 {
72   _os « Usage::GetUsageString(val.UsageField);
73   return _os;
74 }
75
76 } // end namespace gdcm
77
78 #endif //GDCMUSAGE_H
```

11.241 gdcmXMLDictReader.h File Reference

```
#include "gdcmTableReader.h"
#include "gdcmDict.h"
#include "gdcmDictEntry.h"
#include "gdcmTag.h"
```

Include dependency graph for gdcmXMLDictReader.h:



Classes

- class gdcm::XMLDictReader

  Class for representing a XMLDictReader.

Namespaces

- namespace gdcm

11.242 gdcmXMLDictReader.h

Go to the documentation of this file.

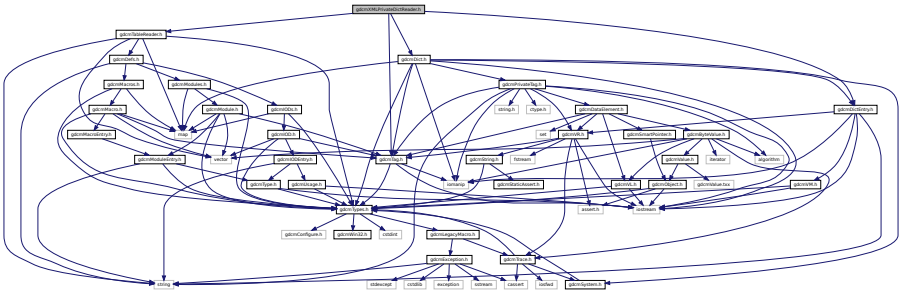
```
1 /*=========================================================================
2
3 Program:  GDCM (Grassroots DICOM). A DICOM library
4
5 Copyright (c) 2006-2011 Mathieu Malaterre
6 All rights reserved.
7 See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.
8
9 This software is distributed WITHOUT ANY WARRANTY; without even
10 the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
11 PURPOSE.  See the above copyright notice for more information.
12
13 =========================================================================*/
14 #ifndef GDCMXMLDICTREADER_H
15 #define GDCMXMLDICTREADER_H
16
17 #include "gdcmTableReader.h"
18 #include "gdcmDict.h"
19 #include "gdcmDictEntry.h"
20 #include "gdcmTag.h"
21
22 namespace gdcm
23 {
29 class GDCM_EXPORT XMLDictReader :  public TableReader
```

```
30 {
31 public:
32   XMLDictReader();
33   ~XMLDictReader() {}
34
35   void StartElement(const char *name, const char **atts);
36   void EndElement(const char *name);
37   void CharacterDataHandler(const char *data, int length);
38
39   const Dict & GetDict() { return DICOMDict; }
40
41 protected:
42   void HandleEntry(const char **atts);
43   void HandleDescription(const char **atts);
44
45 private:
46   Dict DICOMDict;
47   Tag CurrentTag;
48   DictEntry CurrentDE;
49   bool ParsingDescription;
50   std::string Description;
51 };
52
53 } // end namespace gdcm
54
55 #endif //GDCMXMLDICTREADER_H
```

11.243 gdcmXMLPrivateDictReader.h File Reference

```
#include "gdcmTableReader.h"
#include "gdcmDict.h"
#include "gdcmDictEntry.h"
#include "gdcmTag.h"
```

Include dependency graph for gdcmXMLPrivateDictReader.h:



Classes

- class gdcm::XMLPrivateDictReader

  Class for representing a XMLPrivateDictReader.

Namespaces

- namespace gdcm

11.244 gdcmXMLPrivateDictReader.h

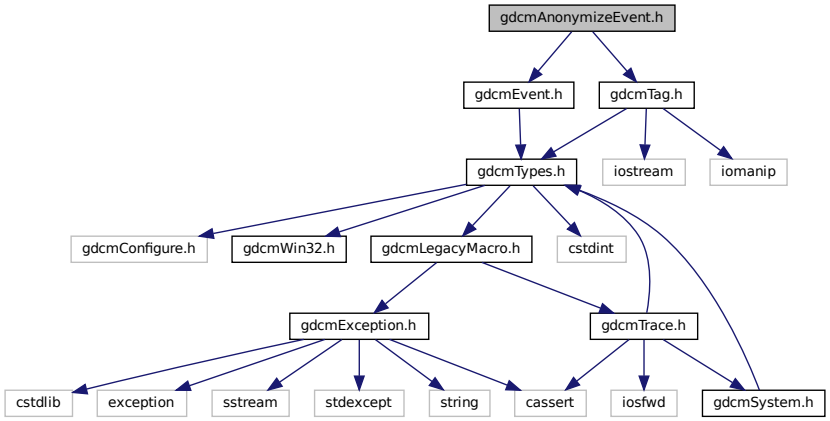
Go to the documentation of this file.

```
1 /*=========================================================================
2
3 Program:  GDCM (Grassroots DICOM). A DICOM library
4
5 Copyright (c) 2006-2011 Mathieu Malaterre
6 All rights reserved.
7 See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.
8
9 This software is distributed WITHOUT ANY WARRANTY; without even
10 the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
11 PURPOSE.  See the above copyright notice for more information.
12
13 =========================================================================*/
14 #ifndef GDCMXMLPRIVATEDICTREADER_H
15 #define GDCMXMLPRIVATEDICTREADER_H
16
17 #include "gdcmTableReader.h"
18 #include "gdcmDict.h"
19 #include "gdcmDictEntry.h"
20 #include "gdcmTag.h"
21
22 namespace gdcm
23 {
29 class GDCM_EXPORT XMLPrivateDictReader :  public TableReader
30 {
31 public:
32   XMLPrivateDictReader();
33   ~XMLPrivateDictReader() {}
34
35   void StartElement(const char *name, const char **atts);
36   void EndElement(const char *name);
37   void CharacterDataHandler(const char *data, int length);
38
39   const PrivateDict & GetPrivateDict() { return PDict; }
40
41 protected:
42   void HandleEntry(const char **atts);
43   void HandleDescription(const char **atts);
44
45 private:
46   PrivateDict PDict;
47   PrivateTag CurrentTag;
48   DictEntry CurrentDE;
49   bool ParsingDescription;
50   std::string Description;
51 };
52
53 } // end namespace gdcm
54
55 #endif //GDCMXMLPRIVATEDICTREADER_H
```

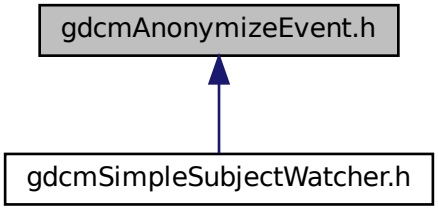
11.245 gdcmAnonymizeEvent.h File Reference

```
#include "gdcmEvent.h"
#include "gdcmTag.h"
```

Include dependency graph for gdcmAnonymizeEvent.h:



This graph shows which files directly or indirectly include this file:



Classes

- class gdcm::AnonymizeEvent

  AnonymizeEvent.

Namespaces

- namespace gdcm

11.246 gdcmAnonymizeEvent.h

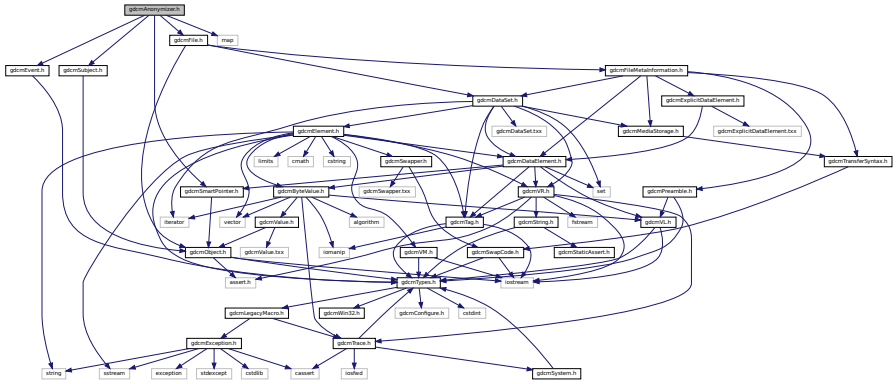
Go to the documentation of this file.

```
1 /*=========================================================================
2
3 Program:  GDCM (Grassroots DICOM). A DICOM library
4
5 Copyright (c) 2006-2011 Mathieu Malaterre
6 All rights reserved.
7 See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.
8
9 This software is distributed WITHOUT ANY WARRANTY; without even
10 the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
11 PURPOSE.  See the above copyright notice for more information.
12
13 =========================================================================*/
14 #ifndef GDCMANONYMIZEEVENT_H
15 #define GDCMANONYMIZEEVENT_H
16
17 #include "gdcmEvent.h"
18 #include "gdcmTag.h"
19
20 namespace gdcm
21 {
22
29 class AnonymizeEvent :  public AnyEvent
30 {
31 public:
32   typedef AnonymizeEvent Self;
33   typedef AnyEvent Superclass;
34   AnonymizeEvent(Tag const &tag = 0):m_Tag(tag) {}
35   ~AnonymizeEvent() override = default;
36   AnonymizeEvent(const Self&s) :  AnyEvent(s){};
37   void operator=(const Self&) = delete;
38
39   const char * GetEventName()const override { return "AnonymizeEvent"; }
40   bool CheckEvent(const ::gdcm::Event* e)const override
41 { return (dynamic_cast<const Self*>(e) == nullptr ?  false :  true) ; }
42   ::gdcm::Event* MakeObject()const override
43 { return new Self; }
44
45   void SetTag(const Tag& t ) { m_Tag = t; }
46   Tag const & GetTag()const { return m_Tag; }
47 private:
48   Tag m_Tag;
49 };
50
51
52 } // end namespace gdcm
53
54 #endif //GDCMANONYMIZEEVENT_H
```

11.247 gdcmAnonymizer.h File Reference

```
#include "gdcmFile.h"
#include "gdcmSubject.h"
#include "gdcmEvent.h"
#include "gdcmSmartPointer.h"
#include <map>
```

Include dependency graph for gdcmAnonymizer.h:



Classes

- class gdcm::Anonymizer

  Anonymizer.

Namespaces

- namespace gdcm

11.248 gdcmAnonymizer.h

Go to the documentation of this file.

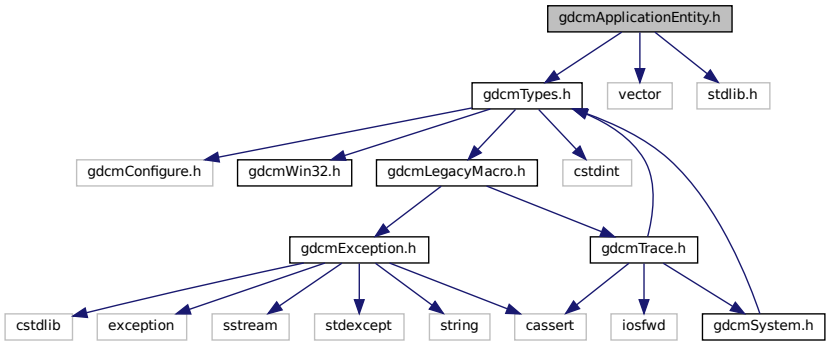
```
1 /*=========================================================================
2
3 Program:  GDCM (Grassroots DICOM). A DICOM library
4
5 Copyright (c) 2006-2011 Mathieu Malaterre
6 All rights reserved.
7 See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.
8
9 This software is distributed WITHOUT ANY WARRANTY; without even
10 the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
11 PURPOSE.  See the above copyright notice for more information.
12
13 =========================================================================*/
14 #ifndef GDCMANONYMIZER_H
15 #define GDCMANONYMIZER_H
16
17 #include "gdcmFile.h"
18 #include "gdcmSubject.h"
19 #include "gdcmEvent.h"
20 #include "gdcmSmartPointer.h"
21
22 #include <map>
23
24 namespace gdcm
25 {
26 class TagPath;
27 class IOD;
28 class CryptographicMessageSyntax;
29
```

```
77 class GDCM_EXPORT Anonymizer :  public Subject
78 {
79 public:
80   Anonymizer():F(new File),CMS(nullptr) {}
81   ~Anonymizer() override;
82
84   bool Empty( Tag const &t );
85
90   bool Empty( PrivateTag const &pt );
91
93   bool Clear( Tag const &t );
94   bool Clear( PrivateTag const &pt );
95
97   bool Remove( Tag const &t );
98
104   bool Remove( PrivateTag const &pt );
105
108   bool Replace( Tag const &t, const char *value );
109   bool Replace( PrivateTag const &t, const char *value );
110
113   bool Replace( Tag const &t, const char *value, VL const & vl );
114   bool Replace( PrivateTag const &t, const char *value, VL const & vl );
115
117   bool RemovePrivateTags();
118
120   bool RemoveGroupLength();
121
123   bool RemoveRetired();
124
126   void SetFile(const File& f) { F = f; }
127   //const File &GetFile() const { return *F; }
128   File &GetFile() { return *F; }
129
134   bool BasicApplicationLevelConfidentialityProfile(bool deidentify = true);
135
137   void SetCryptographicMessageSyntax( CryptographicMessageSyntax *cms );
138   const CryptographicMessageSyntax *GetCryptographicMessageSyntax() const;
139
141   static SmartPointer<Anonymizer> New() { return new Anonymizer; }
142
144   static std::vector<Tag> GetBasicApplicationLevelConfidentialityProfileAttributes();
145
148   static void ClearInternalUIDs();
149
150 protected:
151   // Internal function used to either empty a tag or set it's value to a dummy value (Type 1 vs Type 2)
152   bool BALCPProtect(DataSet &ds, Tag const & tag, const IOD &iod);
153   bool CanEmptyTag(Tag const &tag, const IOD &iod) const;
154   void RecurseDataSet( DataSet & ds );
155
156 private:
157   bool BasicApplicationLevelConfidentialityProfile1();
158   bool BasicApplicationLevelConfidentialityProfile2();
159   bool CheckIfSequenceContainsAttributeToAnonymize(File const &file, SequenceOfItems* sqi) const;
160
161 private:
162   // I would prefer to have a smart pointer to DataSet but DataSet does not derive from Object...
163   SmartPointer<File> F;
164   CryptographicMessageSyntax *CMS;
165
166   typedef std::pair< Tag, std::string > TagValueKey;
167   typedef std::map< TagValueKey, std::string > DummyMapNonUIDTags;
168   typedef std::map< std::string, std::string > DummyMapUIDTags;
169   static DummyMapNonUIDTags dummyMapNonUIDTags;
170   static DummyMapUIDTags dummyMapUIDTags;
171 };
172
179 } // end namespace gdcm
180
181 #endif //GDCMANONYMIZER_H
```

11.249 gdcmApplicationEntity.h File Reference

```
#include "gdcmTypes.h"
#include <vector>
```

`#include <stdlib.h>`
Include dependency graph for gdcmApplicationEntity.h:



Classes

- class gdcm::ApplicationEntity
  ApplicationEntity.

Namespaces

- namespace gdcm

11.250 gdcmApplicationEntity.h

Go to the documentation of this file.

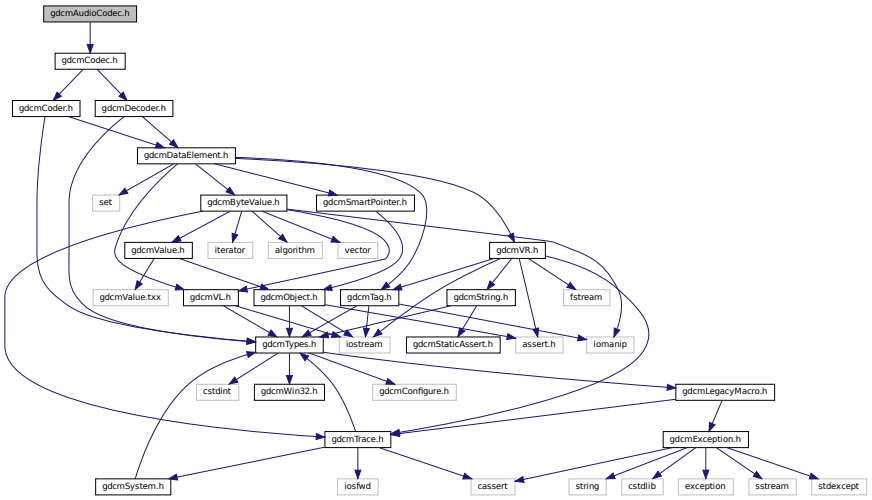
```
1 /*=========================================================================
2
3 Program:  GDCM (Grassroots DICOM). A DICOM library
4
5 Copyright (c) 2006-2011 Mathieu Malaterre
6 All rights reserved.
7 See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.
8
9 This software is distributed WITHOUT ANY WARRANTY; without even
10 the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
11 PURPOSE.  See the above copyright notice for more information.
12
13 =========================================================================*/
14 #ifndef GDCMAPPLICATIONENTITY_H
15 #define GDCMAPPLICATIONENTITY_H
16
17 #include "gdcmTypes.h"
18 #include <vector>
19 #include <stdlib.h> // abort
20
21 namespace gdcm
22 {
23
35 class GDCM_EXPORT ApplicationEntity
36 {
37 public:
```

```
38   static const unsigned int MaxNumberOfComponents = 1;
39   static const unsigned int MaxLength = 16;
40   std::string Internal;
41   static const char Separator = ' ';
42   static const char Padding   = ' ';
43   //static const char Excluded[5] = { '\\' /* 5CH */, '\n' /* LF */, '\f', /* FF */, '\r' /* CR */, 0x1b /*
      ESC */};
44
45   bool IsValid()const {
46     return true;
47   }
48   void Squeeze() {
49     // trim leading and trailing white spaces
50   }
51   void SetBlob(const std::vector<char>& v) {
52     (void)v;
53     assert(0); //TODO
54   }
55   void Print(std::ostream &os)const {
56   (void)os;
57     assert(0); //TODO
58   }
59 };
60
61 } // end namespace gdcm
62
63 #endif //GDCMAPPLICATIONENTITY_H
```

11.251 gdcmAudioCodec.h File Reference

`#include "gdcmCodec.h"`

Include dependency graph for gdcmAudioCodec.h:



Classes

- class gdcm::AudioCodec

  AudioCodec.

Namespaces

- namespace gdcm

11.252 gdcmAudioCodec.h

Go to the documentation of this file.

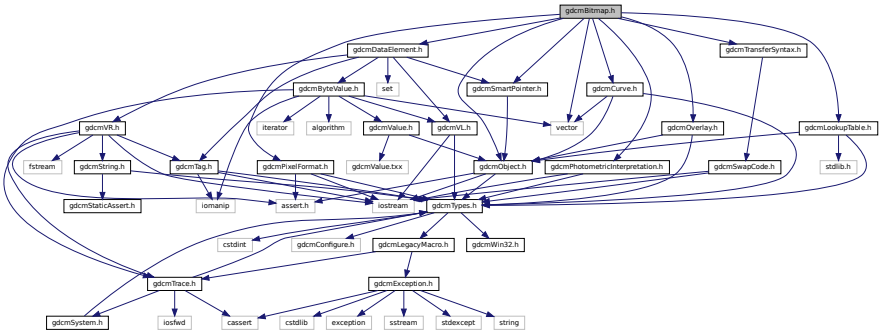
```
1 /*=========================================================================
2
3 Program:  GDCM (Grassroots DICOM). A DICOM library
4
5 Copyright (c) 2006-2011 Mathieu Malaterre
6 All rights reserved.
7 See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.
8
9 This software is distributed WITHOUT ANY WARRANTY; without even
10 the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
11 PURPOSE.  See the above copyright notice for more information.
12
13 =========================================================================*/
14 #ifndef GDCMAUDIOCODEC_H
15 #define GDCMAUDIOCODEC_H
16
17 #include "gdcmCodec.h"
18
19 namespace gdcm
20 {
21
25 class GDCM_EXPORT AudioCodec :  public Codec
26 {
27 public:
28   AudioCodec();
29   ~AudioCodec() override;
30   bool CanCode(TransferSyntax const &)const override { return false; }
31   bool CanDecode(TransferSyntax const &)const override { return false; }
32   bool Decode(DataElement const &is, DataElement &os) override;
33 };
34
35 } // end namespace gdcm
36
37 #endif //GDCMAUDIOCODEC_H
```

11.253 gdcmBitmap.h File Reference

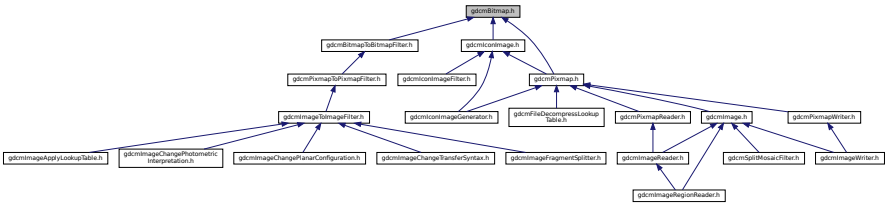
```
#include "gdcmObject.h"
#include "gdcmCurve.h"
#include "gdcmDataElement.h"
#include "gdcmLookupTable.h"
#include "gdcmOverlay.h"
#include "gdcmPhotometricInterpretation.h"
#include "gdcmPixelFormat.h"
#include "gdcmSmartPointer.h"
#include "gdcmTransferSyntax.h"
```

```
#include <vector>
```

Include dependency graph for gdcmBitmap.h:



This graph shows which files directly or indirectly include this file:



Classes

- class gdcm::Bitmap

  Bitmap class.

Namespaces

- namespace gdcm

11.254 gdcmBitmap.h

Go to the documentation of this file.

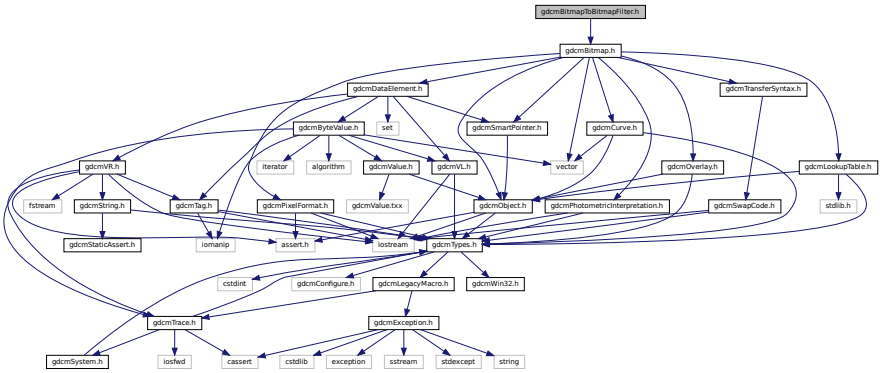
```
1 /*=========================================================================
2
3 Program:  GDCM (Grassroots DICOM). A DICOM library
4
5 Copyright (c) 2006-2011 Mathieu Malaterre
6 All rights reserved.
7 See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.
8
9 This software is distributed WITHOUT ANY WARRANTY; without even
10 the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
```

```
11 PURPOSE.  See the above copyright notice for more information.
12
13 =========================================================================*/
14 #ifndef GDCMBITMAP_H
15 #define GDCMBITMAP_H
16
17 #include "gdcmObject.h"
18 #include "gdcmCurve.h"
19 #include "gdcmDataElement.h"
20 //#include "gdcmIconImage.h"
21 #include "gdcmLookupTable.h"
22 #include "gdcmOverlay.h"
23 #include "gdcmPhotometricInterpretation.h"
24 #include "gdcmPixelFormat.h"
25 #include "gdcmSmartPointer.h"
26 #include "gdcmTransferSyntax.h"
27
28 #include <vector>
29
30 namespace gdcm
31 {
32
38 class GDCM_EXPORT Bitmap :  public Object
39 {
40 public:
41   Bitmap();
42   ~Bitmap() override;
43   void Print(std::ostream &) const override;
44
45   virtual bool AreOverlaysInPixelData()const { return false; }
46   virtual bool UnusedBitsPresentInPixelData()const { return false; }
47
49   unsigned int GetNumberOfDimensions() const;
50   void SetNumberOfDimensions(unsigned int dim);
51
53   unsigned int GetPlanarConfiguration() const;
55   void SetPlanarConfiguration(unsigned int pc);
56
58   bool GetNeedByteSwap()const
59 {
60     return NeedByteSwap;
61     }
62   void SetNeedByteSwap(bool b)
63     {
64     NeedByteSwap = b;
65     }
66
67
69   void SetTransferSyntax(TransferSyntax const &ts) {
70     TS = ts;
71   }
72   const TransferSyntax &GetTransferSyntax()const {
73     return TS;
74   }
75   bool IsTransferSyntaxCompatible( TransferSyntax const & ts ) const;
76   void SetDataElement(DataElement const &de) {
77     PixelData = de;
78   }
79   const DataElement& GetDataElement()const { return PixelData; }
80   DataElement& GetDataElement() { return PixelData; }
81
83   void SetLUT(LookupTable const &lut)
84     {
85     LUT = SmartPointer<LookupTable>( const_cast<LookupTable*>(&lut) );
86     }
87   const LookupTable &GetLUT()const
88 {
89     return *LUT;
90     }
91   LookupTable &GetLUT()
92     {
93     return *LUT;
94     }
95
97   const unsigned int *GetDimensions() const;
98   unsigned int GetDimension(unsigned int idx) const;
99
100   void SetColumns(unsigned int col) { SetDimension(0,col); }
101   unsigned int GetColumns()const { return GetDimension(0); }
102   void SetRows(unsigned int rows) { SetDimension(1,rows); }
103   unsigned int GetRows()const { return GetDimension(1); }
```

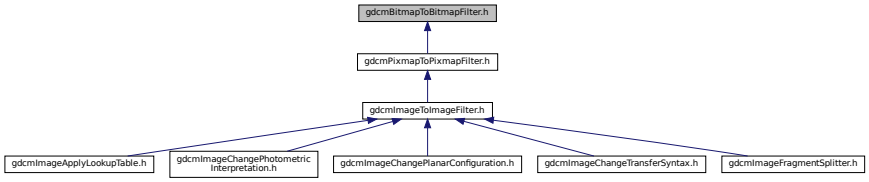
```
104   void SetDimensions(const unsigned int dims[3]);
105   void SetDimension(unsigned int idx, unsigned int dim);
107   const PixelFormat &GetPixelFormat()const
108 {
109     return PF;
110     }
111   PixelFormat &GetPixelFormat()
112     {
113     return PF;
114     }
115   void SetPixelFormat(PixelFormat const &pf)
116     {
117     PF = pf;
118     PF.Validate();
119     }
120
122   const PhotometricInterpretation &GetPhotometricInterpretation() const;
123   void SetPhotometricInterpretation(PhotometricInterpretation const &pi);
124
125   bool IsEmpty()const { return Dimensions.empty(); }
126   void Clear();
127
131   unsigned long GetBufferLength() const;
132
134   bool GetBuffer(char *buffer) const;
135
137   bool IsLossy() const;
138
140   void SetLossyFlag(bool f) { LossyFlag = f; }
141
142 protected:
143   bool TryRAWCodec(char *buffer, bool &lossyflag) const;
144   bool TryJPEGCodec(char *buffer, bool &lossyflag) const;
145   bool TryPVRGCodec(char *buffer, bool &lossyflag) const;
146   bool TryKAKADUCodec(char *buffer, bool &lossyflag) const;
147   bool TryJPEGLSCodec(char *buffer, bool &lossyflag) const;
148   bool TryJPEG2000Codec(char *buffer, bool &lossyflag) const;
149   bool TryRLECodec(char *buffer, bool &lossyflag) const;
150
151   bool TryJPEGCodec2(std::ostream &os) const;
152   bool TryJPEG2000Codec2(std::ostream &os) const;
153
154   bool GetBuffer2(std::ostream &os) const;
155
156   friend class PixmapReader;
157   friend class ImageChangeTransferSyntax;
158   // Function to compute the lossy flag based only on the image buffer.
159   // Watch out that image can be lossy but in implicit little endian format...
160   bool ComputeLossyFlag();
161
162 //private:
163 protected:
164   unsigned int PlanarConfiguration;
165   unsigned int NumberOfDimensions;
166   TransferSyntax TS;
167   PixelFormat PF; // SamplesPerPixel, BitsAllocated, BitsStored, HighBit, PixelRepresentation
168   PhotometricInterpretation PI;
169   // Mind dump:  unsigned int is required here, since we are reading (0028,0008) Number Of Frames
170   // which is VR::IS, so I cannot simply assumed that unsigned short is enough...  :(
171   std::vector<unsigned int> Dimensions; // Col/Row
172   DataElement PixelData; // copied from 7fe0,0010
173
174   typedef SmartPointer<LookupTable> LUTPtr;
175   LUTPtr LUT;
176   // I believe the following 3 ivars can be derived from TS ...
177   bool NeedByteSwap; // FIXME: remove me
178   bool LossyFlag;
179
180 private:
181   bool GetBufferInternal(char *buffer, bool &lossyflag) const;
182 };
183
184 } // end namespace gdcm
185
186 #endif //GDCMBITMAP_H
```

11.255 gdcmBitmapToBitmapFilter.h File Reference

`#include "gdcmBitmap.h"`
Include dependency graph for gdcmBitmapToBitmapFilter.h:



This graph shows which files directly or indirectly include this file:



Classes

- class gdcm::BitmapToBitmapFilter

  BitmapToBitmapFilter class.

Namespaces

- namespace gdcm

11.256 gdcmBitmapToBitmapFilter.h

Go to the documentation of this file.

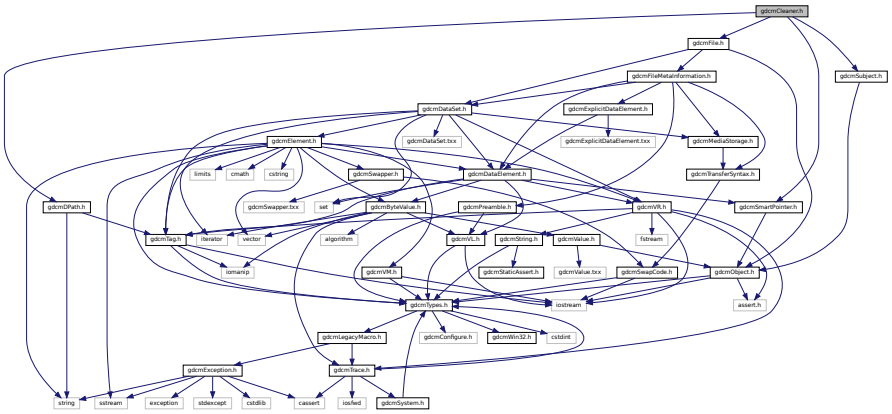
```
1 /*=========================================================================
2
3 Program:  GDCM (Grassroots DICOM). A DICOM library
4
```

```
5 Copyright (c) 2006-2011 Mathieu Malaterre
6 All rights reserved.
7 See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.
8
9 This software is distributed WITHOUT ANY WARRANTY; without even
10 the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
11 PURPOSE.  See the above copyright notice for more information.
12
13 =========================================================================*/
14 #ifndef GDCMBITMAPTOBITMAPFILTER_H
15 #define GDCMBITMAPTOBITMAPFILTER_H
16
17 #include "gdcmBitmap.h"
18
19 namespace gdcm
20 {
21
26 class GDCM_EXPORT BitmapToBitmapFilter
27 {
28 public:
29   BitmapToBitmapFilter();
30   ~BitmapToBitmapFilter() = default;
31
33   void SetInput(const Bitmap& image);
34
36   const Bitmap &GetOutput()const { return *Output; }
37
38   // SWIG/Java hack:
39   const Bitmap &GetOutputAsBitmap() const;
40
41 protected:
42   SmartPointer<Bitmap> Input;
43   SmartPointer<Bitmap> Output;
44 };
45
46 } // end namespace gdcm
47
48 #endif //GDCMBITMAPTOBITMAPFILTER_H
```

11.257 gdcmCleaner.h File Reference

```
#include "gdcmDPath.h"
#include "gdcmFile.h"
#include "gdcmSmartPointer.h"
#include "gdcmSubject.h"
```

Include dependency graph for gdcmCleaner.h:



Classes

- class gdcm::Cleaner

  Cleaner.

Namespaces

- namespace gdcm

11.258 gdcmCleaner.h

Go to the documentation of this file.

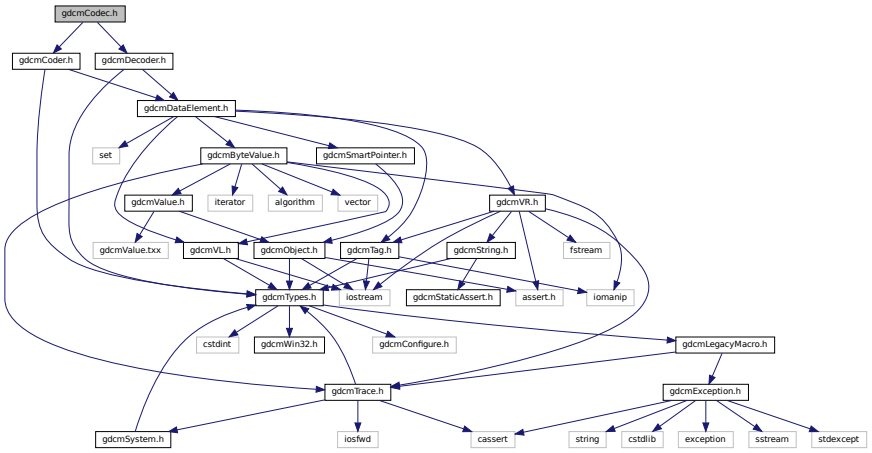
```
1 /*=========================================================================
2
3 Program:  GDCM (Grassroots DICOM). A DICOM library
4
5 Copyright (c) 2006-2011 Mathieu Malaterre
6 All rights reserved.
7 See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.
8
9 This software is distributed WITHOUT ANY WARRANTY; without even
10 the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
11 PURPOSE.  See the above copyright notice for more information.
12
13 =========================================================================*/
14 #ifndef GDCMCLEANER_H
15 #define GDCMCLEANER_H
16
17 #include "gdcmDPath.h"
18 #include "gdcmFile.h"
19 #include "gdcmSmartPointer.h"
20 #include "gdcmSubject.h"
21
22 namespace gdcm {
30 class GDCM_EXPORT Cleaner :  public Subject {
31  public:
32   Cleaner();
33   ~Cleaner();
34
36   bool Empty(Tag const &t);
37   bool Empty(PrivateTag const &pt);
38   bool Empty(DPath const &dpath);
39   bool Empty(VR const &vr);
40
41   bool Remove(Tag const &t);
42   bool Remove(PrivateTag const &pt);
43   bool Remove(DPath const &dpath);
44   bool Remove(VR const &vr);
45
47   bool Scrub(Tag const &t);
48   bool Scrub(PrivateTag const &pt);
49   bool Scrub(DPath const &dpath);
50   bool Scrub(VR const &vr);
51
52   bool Preserve(DPath const &dpath);
53
56   void RemoveAllMissingPrivateCreator(bool remove);
57
60   bool RemoveMissingPrivateCreator(Tag const &t);
61
63   void RemoveAllGroupLength(bool remove);
64
66   void RemoveAllIllegal(bool remove);
67
69   bool Clean();
70
72   void SetFile(const File &f) { F = f; }
73   // const File &GetFile() const { return *F; }
74   File &GetFile() { return *F; }
```

```
75
77   static SmartPointer<Cleaner> New() { return new Cleaner; }
78
79  private:
80   // I would prefer to have a smart pointer to DataSet but DataSet does not
81   // derive from Object...
82   SmartPointer<File> F;
83   struct impl;
84   // PIMPL idiom
85   impl *pimpl;
86 };
87
88 }  // end namespace gdcm
89
90 #endif  // GDCMCLEANER_H
```

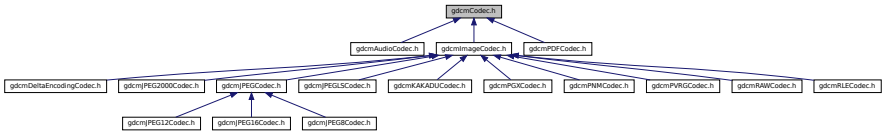
11.259 gdcmCodec.h File Reference

```
#include "gdcmCoder.h"
#include "gdcmDecoder.h"
```

Include dependency graph for gdcmCodec.h:



This graph shows which files directly or indirectly include this file:



Classes

- class gdcm::Codec

  Codec class.

Namespaces

- namespace gdcm

11.260 gdcmCodec.h

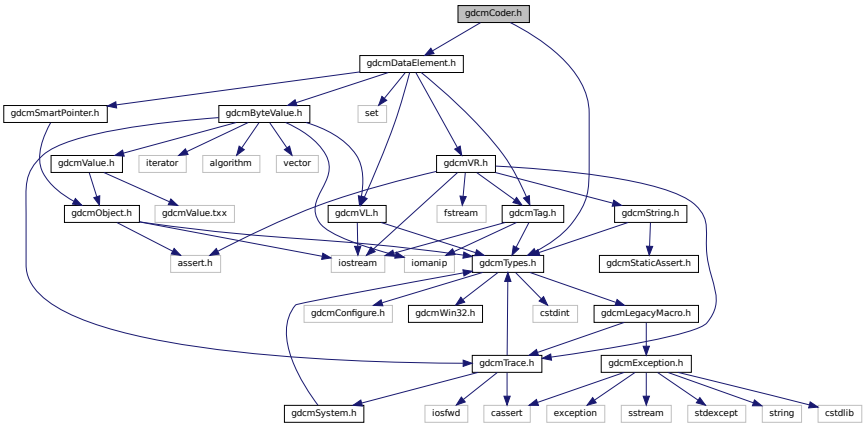
Go to the documentation of this file.

```
1 /*=========================================================================
2
3 Program:  GDCM (Grassroots DICOM). A DICOM library
4
5 Copyright (c) 2006-2011 Mathieu Malaterre
6 All rights reserved.
7 See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.
8
9 This software is distributed WITHOUT ANY WARRANTY; without even
10 the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
11 PURPOSE.  See the above copyright notice for more information.
12
13 =========================================================================*/
14 #ifndef GDCMCODEC_H
15 #define GDCMCODEC_H
16
17 #include "gdcmCoder.h"
18 #include "gdcmDecoder.h"
19
20 namespace gdcm
21 {
22
26 class GDCM_EXPORT Codec :  public Coder, public Decoder
27 {
28 };
29
30 } // end namespace gdcm
31
32 #endif //GDCMCODEC_H
```

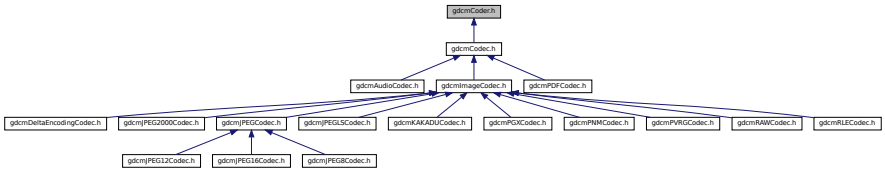
11.261 gdcmCoder.h File Reference

```
#include "gdcmTypes.h"
#include "gdcmDataElement.h"
```

Include dependency graph for gdcmCoder.h:



This graph shows which files directly or indirectly include this file:



Classes

- class gdcm::Coder

  Coder.

Namespaces

- namespace gdcm

11.262 gdcmCoder.h

Go to the documentation of this file.

```
1 /*=========================================================================
2
3 Program:  GDCM (Grassroots DICOM). A DICOM library
4
5 Copyright (c) 2006-2011 Mathieu Malaterre
6 All rights reserved.
7 See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.
8
9 This software is distributed WITHOUT ANY WARRANTY; without even
10 the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
11 PURPOSE.  See the above copyright notice for more information.
12
13 =========================================================================*/
14 #ifndef GDCMCODER_H
15 #define GDCMCODER_H
16
17 #include "gdcmTypes.h"
18 #include "gdcmDataElement.h" // FIXME
19
20 namespace gdcm
21 {
22
23 class TransferSyntax;
24 class DataElement;
28 class GDCM_EXPORT Coder
29 {
30 public:
31   virtual ~Coder() = default;
32
34   virtual bool CanCode(TransferSyntax const &) const = 0;
35
36   // Note:  in / out are reserved keyword in C#.  Change to in_ / out_
37
39   virtual bool Code(DataElement const &in_, DataElement &out_) { (void)in_; (void)out_; return false; }
40 protected:
41   virtual bool InternalCode(const char *bv, unsigned long len, std::ostream &os) {
      (void)bv;(void)os;(void)len;return false; }
42 };
43
44 } // end namespace gdcm
45
46 #endif //GDCMCODER_H
```

11.263 gdcmConstCharWrapper.h File Reference

Classes

- class gdcm::ConstCharWrapper

  Do not use me.

Namespaces

- namespace gdcm

11.264 gdcmConstCharWrapper.h

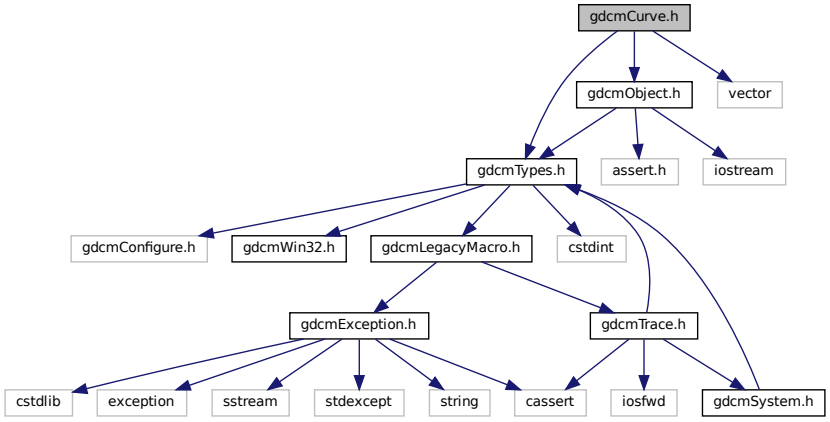
Go to the documentation of this file.

```
1 /*=========================================================================
2
3 Program:  GDCM (Grassroots DICOM). A DICOM library
4
5 Copyright (c) 2006-2011 Mathieu Malaterre
6 All rights reserved.
7 See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.
8
9 This software is distributed WITHOUT ANY WARRANTY; without even
10 the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
11 PURPOSE.  See the above copyright notice for more information.
12
13 =========================================================================*/
14 #ifndef GDCMCONSTCHARWRAPPER_H
15 #define GDCMCONSTCHARWRAPPER_H
16
17 namespace gdcm
18 {
19
20 #error
21
22 /*
23 * This class is a pure hack.  Its only goal is to work around a bad bug in :
24 *  $ swig -version
25 *  SWIG Version 1.3.31
26 *
27 *  See
28 * -
     http://sourceforge.net/mailarchive/forum.php?thread_name=bf0c3b3f0802290552y5163989t76572b80a044ce28%40mail.gmail.com&forum
29 *
30 *  As a side note there is also a problem with const reference to enum type:
31 * -
     http://sourceforge.net/mailarchive/forum.php?thread_name=bf0c3b3f0802290552y5163989t76572b80a044ce28%40mail.gmail.com&forum
32 *
33 *   And to keep a track of isse with swig here is the last one:
34 *
35 * -
     http://sourceforge.net/mailarchive/forum.php?thread_name=bf0c3b3f0802290552y5163989t76572b80a044ce28%40mail.gmail.com&forum
36 */
37
38
42 class ConstCharWrapper
43 {
44 public:
45   ConstCharWrapper(const char *i=0):Internal(i) {}
46   operator const char * () const { return Internal; }
47 private:
48   const char *Internal;
49 };
50
51 } // end namespace gdcm
52
53 #endif //GDCMCONSTCHARWRAPPER_H
```

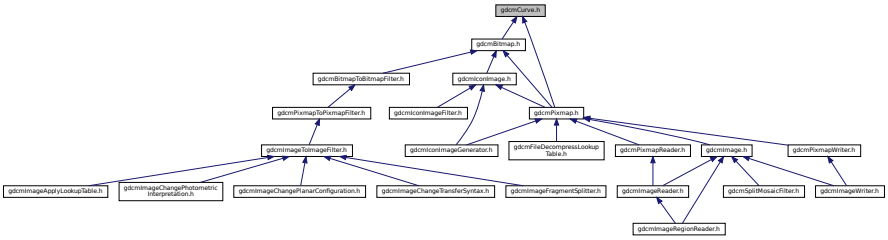
11.265 gdcmCurve.h File Reference

```
#include "gdcmTypes.h"
#include "gdcmObject.h"
#include <vector>
```

Include dependency graph for gdcmCurve.h:



This graph shows which files directly or indirectly include this file:



Classes

- class gdcm::Curve

  Curve class to handle element 50xx,3000 Curve Data.

Namespaces

- namespace gdcm

11.266 gdcmCurve.h

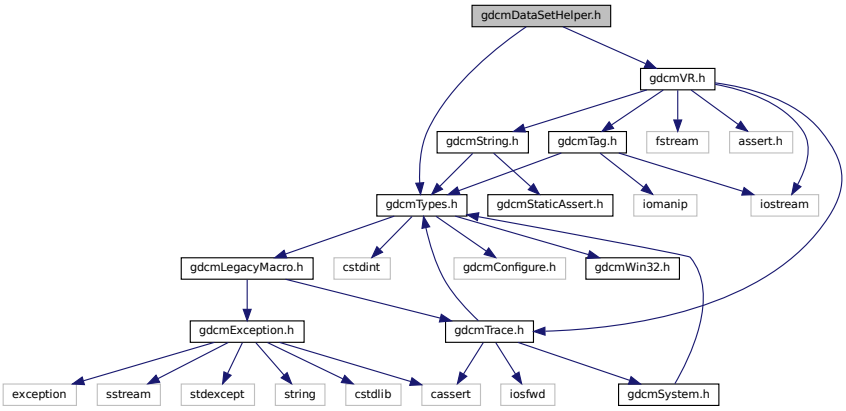
Go to the documentation of this file.

```
1 /*=========================================================================
2
3 Program:  GDCM (Grassroots DICOM). A DICOM library
4
5 Copyright (c) 2006-2011 Mathieu Malaterre
6 All rights reserved.
7 See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.
8
9 This software is distributed WITHOUT ANY WARRANTY; without even
10 the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
11 PURPOSE.  See the above copyright notice for more information.
12
13 =========================================================================*/
14 #ifndef GDCMCURVE_H
15 #define GDCMCURVE_H
16
17 #include "gdcmTypes.h"
18 #include "gdcmObject.h"
19
20 #include <vector>
21
22 namespace gdcm
23 {
24
25 class CurveInternal;
26 class ByteValue;
27 class DataSet;
28 class DataElement;
40 class GDCM_EXPORT Curve :  public Object
41 {
42 public:
43   Curve();
44   ~Curve() override;
45   void Print(std::ostream &) const override;
46
47   void GetAsPoints(float *array) const;
48
49   static unsigned int GetNumberOfCurves(DataSet const & ds);
50
51   // Update curve data from dataelmenet de:
52   void Update(const DataElement & de);
53
54   void SetGroup(unsigned short group);
55   unsigned short GetGroup() const;
56   void SetDimensions(unsigned short dimensions);
57   unsigned short GetDimensions() const;
58   void SetNumberOfPoints(unsigned short numberofpoints);
59   unsigned short GetNumberOfPoints() const;
60   void SetTypeOfData(const char *typeofdata);
61   const char *GetTypeOfData() const;
62   // See PS 3.3 - 2004 - C.10.2.1.1 Type of data
63   const char *GetTypeOfDataDescription() const;
64   void SetCurveDescription(const char *curvedescription);
65   void SetDataValueRepresentation(unsigned short datavaluerepresentation);
66   unsigned short GetDataValueRepresentation() const;
67   void SetCurveDataDescriptor(const uint16_t * values, size_t num);
68   std::vector<unsigned short> const &GetCurveDataDescriptor() const;
69   void SetCoordinateStartValue( unsigned short v );
70   void SetCoordinateStepValue( unsigned short v );
71
72   void SetCurve(const char *array, unsigned int length);
73
74   bool IsEmpty() const;
75
76   void Decode(std::istream &is, std::ostream &os);
77
78   Curve(Curve const &ov);
79 private:
80   double ComputeValueFromStartAndStep(unsigned int idx) const;
81   CurveInternal *Internal;
82 };
83
84 } // end namespace gdcm
85
86 #endif //GDCMCURVE_H
```

11.267 gdcmDataSetHelper.h File Reference

```
#include "gdcmTypes.h"
#include "gdcmVR.h"
```

Include dependency graph for gdcmDataSetHelper.h:



Classes

- class gdcm::DataSetHelper

  DataSetHelper (internal class, not intended for user level)

Namespaces

- namespace gdcm

11.268 gdcmDataSetHelper.h

Go to the documentation of this file.

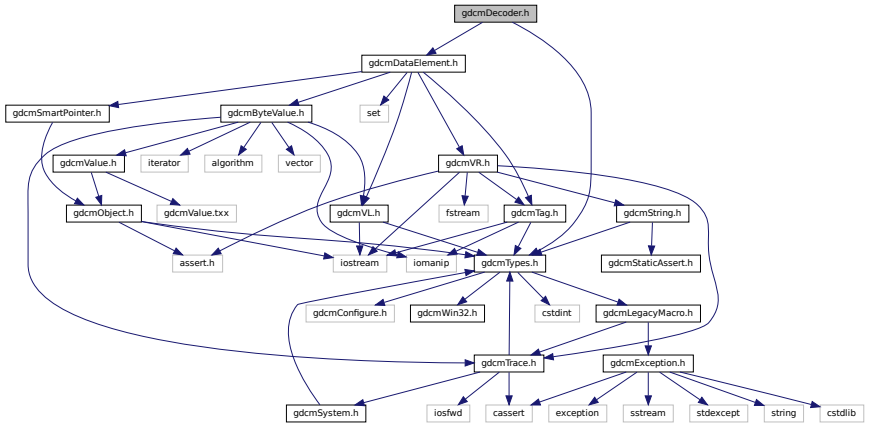
```
1 /*=========================================================================
2
3 Program:  GDCM (Grassroots DICOM). A DICOM library
4
5 Copyright (c) 2006-2011 Mathieu Malaterre
6 All rights reserved.
7 See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.
8
9 This software is distributed WITHOUT ANY WARRANTY; without even
10 the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
11 PURPOSE.  See the above copyright notice for more information.
12
13 =========================================================================*/
14 #ifndef GDCMDATASETHELPER_H
15 #define GDCMDATASETHELPER_H
16
17 #include "gdcmTypes.h"
18 #include "gdcmVR.h"
```

```
19
20 namespace gdcm
21 {
22 class DataSet;
23 class File;
24 class Tag;
25 class SequenceOfItems;
26
32 class GDCM_EXPORT DataSetHelper
33 {
34 public:
37   static VR ComputeVR(File const & file, DataSet const &ds, const Tag& tag);
38
39   //static SequenceOfItems* ComputeSQFromByteValue(File const & file, DataSet const &ds, const Tag &tag);
40
41 protected:
42 };
43
44 } // end namespace gdcm
45
46 #endif // GDCMDATASETHELPER_H
```

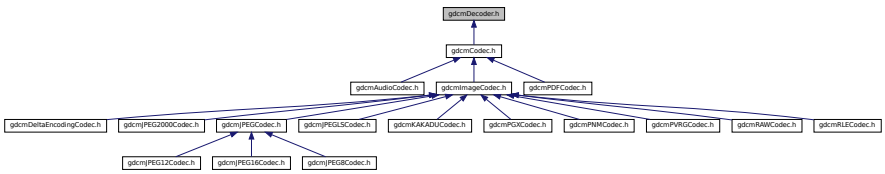
11.269 gdcmDecoder.h File Reference

```
#include "gdcmTypes.h"
#include "gdcmDataElement.h"
```

Include dependency graph for gdcmDecoder.h:



This graph shows which files directly or indirectly include this file:



Classes

- class gdcm::Decoder

  Decoder.

Namespaces

- namespace gdcm

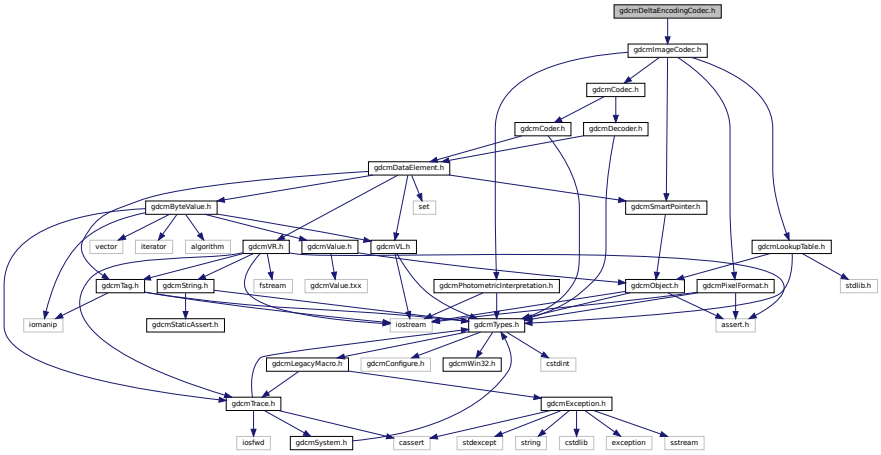
11.270 gdcmDecoder.h

Go to the documentation of this file.

```
1 /*=========================================================================
2
3 Program:  GDCM (Grassroots DICOM). A DICOM library
4
5 Copyright (c) 2006-2011 Mathieu Malaterre
6 All rights reserved.
7 See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.
8
9 This software is distributed WITHOUT ANY WARRANTY; without even
10 the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
11 PURPOSE.  See the above copyright notice for more information.
12
13 =========================================================================*/
14
15 #ifndef GDCMDECODER_H
16 #define GDCMDECODER_H
17
18 #include "gdcmTypes.h"
19 #include "gdcmDataElement.h" // FIXME
20
21 namespace gdcm
22 {
23
24 class TransferSyntax;
25 class DataElement;
29 class GDCM_EXPORT Decoder
30 {
31 public:
32   virtual ~Decoder() = default;
33
35   virtual bool CanDecode(TransferSyntax const &) const = 0;
36
38   virtual bool Decode(DataElement const &, DataElement &) { return false; }
39 protected:
40   virtual bool DecodeByStreams(std::istream &, std::ostream &) { return false; }
41 };
42
43 } // end namespace gdcm
44
45 #endif //GDCMDECODER_H
```

11.271 gdcmDeltaEncodingCodec.h File Reference

`#include "gdcmImageCodec.h"`
Include dependency graph for gdcmDeltaEncodingCodec.h:



Classes

- class gdcm::DeltaEncodingCodec

  DeltaEncodingCodec compression used by some private vendor.

Namespaces

- namespace gdcm

11.272 gdcmDeltaEncodingCodec.h

Go to the documentation of this file.

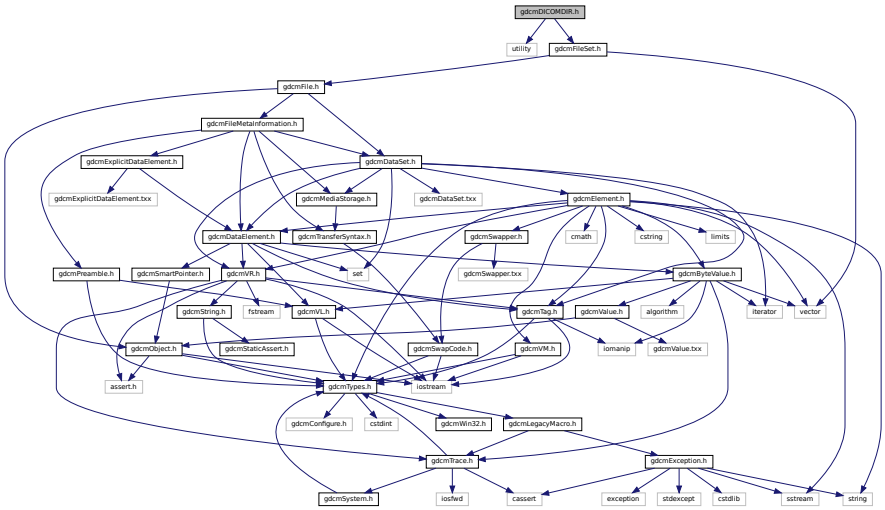
```
1 /*=========================================================================
2
3 Program:  GDCM (Grassroots DICOM). A DICOM library
4
5 Copyright (c) 2006-2011 Mathieu Malaterre
6 All rights reserved.
7 See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.
8
9 This software is distributed WITHOUT ANY WARRANTY; without even
10 the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
11 PURPOSE.  See the above copyright notice for more information.
12
13 =========================================================================*/
14 #ifndef GDCMDELTAENCODINGCODEC_H
15 #define GDCMDELTAENCODINGCODEC_H
16
17 #include "gdcmImageCodec.h"
18 #error do not use
19
20 namespace gdcm
```

```
21 {
22
27 class DeltaEncodingCodec :  public ImageCodec
28 {
29 public:
30   DeltaEncodingCodec();
31   ~DeltaEncodingCodec();
32   bool CanDecode(TransferSyntax const &ts);
33   bool Decode(DataElement const &is, DataElement &os);
34 protected:
35   bool Decode(std::istream &is, std::ostream &os);
36 };
37
38 } // end namespace gdcm
39
40 #endif //GDCMDELTAENCODINGCODEC_H
```

11.273 gdcmDICOMDIR.h File Reference

```
#include <utility>
#include "gdcmFileSet.h"
```

Include dependency graph for gdcmDICOMDIR.h:



Classes

- class gdcm::DICOMDIR

  DICOMDIR class.

Namespaces

- namespace gdcm

11.274 gdcmDICOMDIR.h

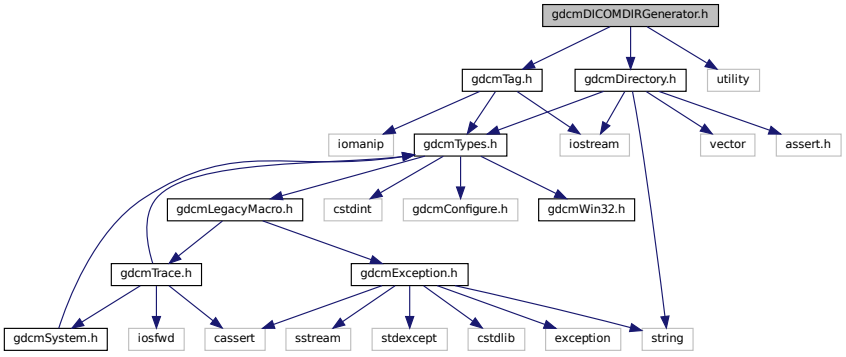
Go to the documentation of this file.

```
1 /*=========================================================================
2
3 Program:  GDCM (Grassroots DICOM). A DICOM library
4
5 Copyright (c) 2006-2011 Mathieu Malaterre
6 All rights reserved.
7 See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.
8
9 This software is distributed WITHOUT ANY WARRANTY; without even
10 the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
11 PURPOSE.  See the above copyright notice for more information.
12
13 =========================================================================*/
14 #ifndef GDCMDICOMDIR_H
15 #define GDCMDICOMDIR_H
16
17 #include <utility>
18 #include "gdcmFileSet.h"
19
20 namespace gdcm
21 {
27 class GDCM_EXPORT DICOMDIR
28 {
29 public:
30   DICOMDIR() = default;
31   DICOMDIR(FileSet  fs):_FS(std::move(std::move(fs))) {}
32
33 private:
34   FileSet _FS;
35   //13 sept 2010 mmr-- added the underscore to FS to compile under Sunos gcc
36 };
37
38 } // end namespace gdcm
39
40 #endif //GDCMDICOMDIR_H
```

11.275 gdcmDICOMDIRGenerator.h File Reference

```
#include "gdcmDirectory.h"
#include "gdcmTag.h"
#include <utility>
```

Include dependency graph for gdcmDICOMDIRGenerator.h:



Classes

- class gdcm::DICOMDIRGenerator

  DICOMDIRGenerator class.

Namespaces

- namespace gdcm

11.276 gdcmDICOMDIRGenerator.h

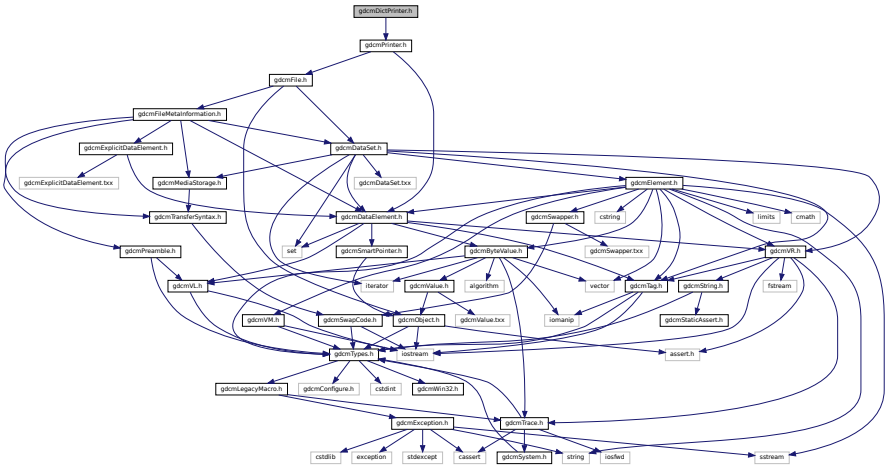
Go to the documentation of this file.

```
1 /*=========================================================================
2
3 Program:  GDCM (Grassroots DICOM). A DICOM library
4
5 Copyright (c) 2006-2011 Mathieu Malaterre
6 All rights reserved.
7 See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.
8
9 This software is distributed WITHOUT ANY WARRANTY; without even
10 the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
11 PURPOSE.  See the above copyright notice for more information.
12
13 =========================================================================*/
14 #ifndef GDCMDICOMDIRGENERATOR_H
15 #define GDCMDICOMDIRGENERATOR_H
16
17 #include "gdcmDirectory.h"
18 #include "gdcmTag.h"
19 #include <utility> // std::pair
20
21 namespace gdcm
22 {
23 class File;
24 class Scanner;
25 class SequenceOfItems;
26 class VL;
27 class DICOMDIRGeneratorInternal;
28
56 class GDCM_EXPORT DICOMDIRGenerator
57 {
58 public:
59   typedef Directory::FilenamesType  FilenamesType;
60   typedef Directory::FilenameType  FilenameType;
61   DICOMDIRGenerator();
62   ~DICOMDIRGenerator();
63
65   void SetFilenames( FilenamesType const & fns );
66
68   void SetRootDirectory( FilenameType const & root );
69
72   void SetDescriptor( const char *d );
73
75   bool Generate();
76
78   void SetFile(const File& f);
79   File &GetFile();
80
81 protected:
82   Scanner &GetScanner();
83   bool AddPatientDirectoryRecord();
84   bool AddStudyDirectoryRecord();
85   bool AddSeriesDirectoryRecord();
86   bool AddImageDirectoryRecord();
87
88 private:
89   const char *ComputeFileID(const char *);
90   bool TraverseDirectoryRecords(VL start );
```

```
91   bool ComputeDirectoryRecordsOffset(const SequenceOfItems *sqi, VL start);
92   size_t FindNextDirectoryRecord( size_t item1, const char *directorytype );
93   SequenceOfItems *GetDirectoryRecordSequence();
94   size_t FindLowerLevelDirectoryRecord( size_t item1, const char *directorytype );
95   typedef std::pair< std::string, Tag> MyPair;
96   MyPair GetReferenceValueForDirectoryType(size_t item);
97   bool SeriesBelongToStudy(const char *seriesuid, const char *studyuid);
98   bool ImageBelongToSeries(const char *sopuid, const char *seriesuid, Tag const &t1, Tag const &t2);
99   bool ImageBelongToSameSeries(const char *sopuid, const char *seriesuid, Tag const &t);
100
101   DICOMDIRGeneratorInternal * Internals;
102 };
103
109 } // end namespace gdcm
110
111 #endif //GDCMDICOMDIRGENERATOR_H
```

11.277 gdcmDictPrinter.h File Reference

`#include "gdcmPrinter.h"`
Include dependency graph for gdcmDictPrinter.h:



Classes

- class gdcm::DictPrinter

  DictPrinter class.

Namespaces

- namespace gdcm

11.278 gdcmDictPrinter.h

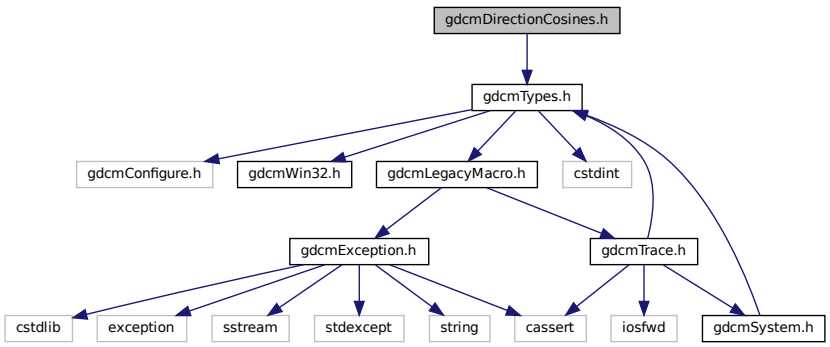
Go to the documentation of this file.

```
1 /*=========================================================================
2
3 Program:  GDCM (Grassroots DICOM). A DICOM library
4
5 Copyright (c) 2006-2011 Mathieu Malaterre
6 All rights reserved.
7 See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.
8
9 This software is distributed WITHOUT ANY WARRANTY; without even
10 the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
11 PURPOSE.  See the above copyright notice for more information.
12
13 =========================================================================*/
14 #ifndef GDCMDICTPRINTER_H
15 #define GDCMDICTPRINTER_H
16
17 #include "gdcmPrinter.h"
18
19 namespace gdcm
20 {
21
25 // It's a sink there is no output
26 class GDCM_EXPORT DictPrinter :  public Printer
27 {
28 public:
29   DictPrinter();
30   ~DictPrinter();
31
32   void Print(std::ostream& os);
33
34 protected:
35   void PrintDataElement2(std::ostream& os, const DataSet &ds, const DataElement &ide);
36   void PrintDataSet2(std::ostream& os, const DataSet &ds);
37 };
38
39 } // end namespace gdcm
40
41 #endif //GDCMDICTPRINTER_H
```

11.279 gdcmDirectionCosines.h File Reference

`#include "gdcmTypes.h"`

Include dependency graph for gdcmDirectionCosines.h:



Classes

- class gdcm::DirectionCosines

  class to handle DirectionCosines

Namespaces

- namespace gdcm

11.280 gdcmDirectionCosines.h

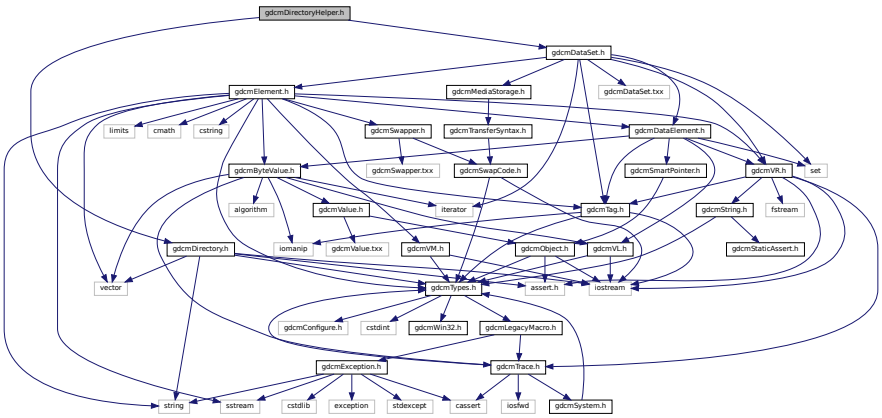
Go to the documentation of this file.

```
1 /*=========================================================================
2
3 Program:  GDCM (Grassroots DICOM). A DICOM library
4
5 Copyright (c) 2006-2011 Mathieu Malaterre
6 All rights reserved.
7 See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.
8
9 This software is distributed WITHOUT ANY WARRANTY; without even
10 the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
11 PURPOSE.  See the above copyright notice for more information.
12
13 =========================================================================*/
14 #ifndef GDCMDIRECTIONCOSINES_H
15 #define GDCMDIRECTIONCOSINES_H
16
17 #include "gdcmTypes.h"
18
19 namespace gdcm
20 {
21
25 class GDCM_EXPORT DirectionCosines
26 {
27 public:
28   DirectionCosines();
29   DirectionCosines(const double dircos[6]);
30   // Cannot get the following signature to be wrapped with swig...
31   //DirectionCosines(const double *dircos = 0 );
32   ~DirectionCosines();
33
35   void Print(std::ostream &) const;
36
38   void Cross(double z[3]) const;
39
41   double Dot() const;
42
44   static double Dot(const double x[3], const double y[3]);
45
47   void Normalize();
48
50   static void Normalize(double v[3]);
51
53   operator const double* () const { return Values; }
54
56   bool IsValid() const;
57
60   bool SetFromString(const char *str);
61
63   double CrossDot(DirectionCosines const &dc) const;
64
66   double ComputeDistAlongNormal(const double ipp[3]) const;
67
68 private:
69   double Values[6];
70 };
71
72 } // end namespace gdcm
73
74 #endif //GDCMDIRECTIONCOSINES_H
```

11.281 gdcmDirectoryHelper.h File Reference

```
#include "gdcmDirectory.h"
#include "gdcmDataSet.h"
```

Include dependency graph for gdcmDirectoryHelper.h:



Classes

- class gdcm::DirectoryHelper

  DirectoryHelper.

Namespaces

- namespace gdcm

11.282 gdcmDirectoryHelper.h

Go to the documentation of this file.

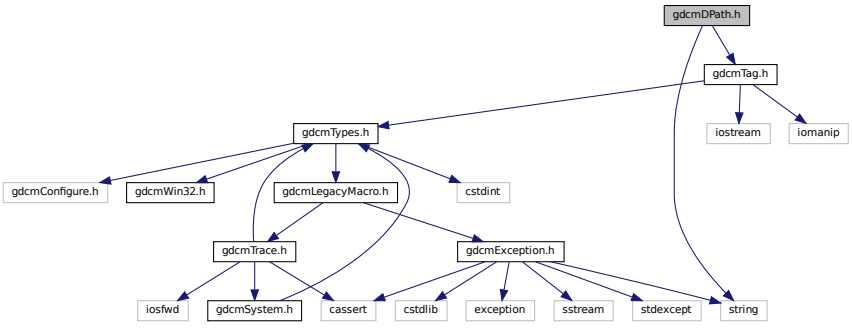
```
1 /*=========================================================================
2
3 Program:  GDCM (Grassroots DICOM). A DICOM library
4
5 Copyright (c) 2006-2011 Mathieu Malaterre
6 All rights reserved.
7 See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.
8
9 This software is distributed WITHOUT ANY WARRANTY; without even
10 the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
11 PURPOSE.  See the above copyright notice for more information.
12
13 =========================================================================*/
14
15 #include "gdcmDirectory.h"
16 #include "gdcmDataSet.h"
17
18 namespace gdcm
19 {
20
```

```
35 class GDCM_EXPORT DirectoryHelper
36 {
37 public:
38   //returns all series UIDs in a given directory that match a particular SOP Instance UID
39   static Directory::FilenamesType GetSeriesUIDsBySOPClassUID(const std::string& inDirectory,
40     const std::string& inSOPClassUID);
41
42   //specific implementations of the SOPClassUID grabber, so you don't have to
43   //remember the SOP Class UIDs of CT or MR images.
44   static Directory::FilenamesType GetCTImageSeriesUIDs(const std::string& inDirectory);
45   static Directory::FilenamesType GetMRImageSeriesUIDs(const std::string& inDirectory);
46   static Directory::FilenamesType GetRTStructSeriesUIDs(const std::string& inDirectory);
47
48   //given a directory and a series UID, provide all filenames with that series UID.
49   static Directory::FilenamesType GetFilenamesFromSeriesUIDs(const std::string& inDirectory,
50     const std::string& inSeriesUID);
51
52   //given a series UID, load all the images associated with that series UID
53   //these images will be IPP sorted, so that they can be used for gathering all
54   //the necessary information for generating an RTStruct
55   //this function should be called by the writer once, if the writer's dataset
56   //vector is empty.    Make sure to have a new writer for new rtstructs.
57   static std::vector<DataSet> LoadImageFromFiles(const std::string& inDirectory,
58     const std::string& inSeriesUID);
59
60   //When writing RTStructs, each contour will have z position defined.
61   //use that z position to determine the SOPInstanceUID for that plane.
62   static std::string RetrieveSOPInstanceUIDFromZPosition(double inZPos,
63     const std::vector<DataSet>& inDS);
64
65   //When writing RTStructs, the frame of reference is done by planes to start with
66   static std::string RetrieveSOPInstanceUIDFromIndex(int inIndex,
67    const std::vector<DataSet>& inDS);
68
69   //each plane needs to know the SOPClassUID, and that won't change from image to image
70   //so, retrieve this once at the start of writing.
71   static std::string GetSOPClassUID(const std::vector<DataSet>& inDS);
72
73   //retrieve the frame of reference from the set of datasets
74   static std::string GetFrameOfReference(const std::vector<DataSet>& inDS);
75
76   //both the image and polydata readers use these functions to get std::strings
77   static std::string GetStringValueFromTag(const Tag& t, const DataSet& ds);
78 };
79
80 }
```

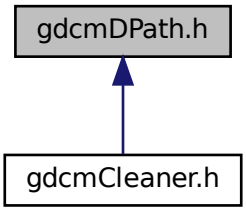
11.283 gdcmDPath.h File Reference

```
#include "gdcmTag.h"
#include <string>
```

Include dependency graph for gdcmDPath.h:



This graph shows which files directly or indirectly include this file:



Classes

- class gdcm::DPath

  class to handle a DICOM path While supp 118 did introduced a notion of XPath for XML Native model this convention is too XML-centric. Instead prefer DCMTK style notation `https://groups.google.com/g/comp.protocols.dicom/c/IyIH0IOBMPA`

Namespaces

- namespace gdcm

Functions

- std::ostream & gdcm::operator<< (std::ostream &os, const DPath &val)

11.284 gdcmDPath.h

Go to the documentation of this file.

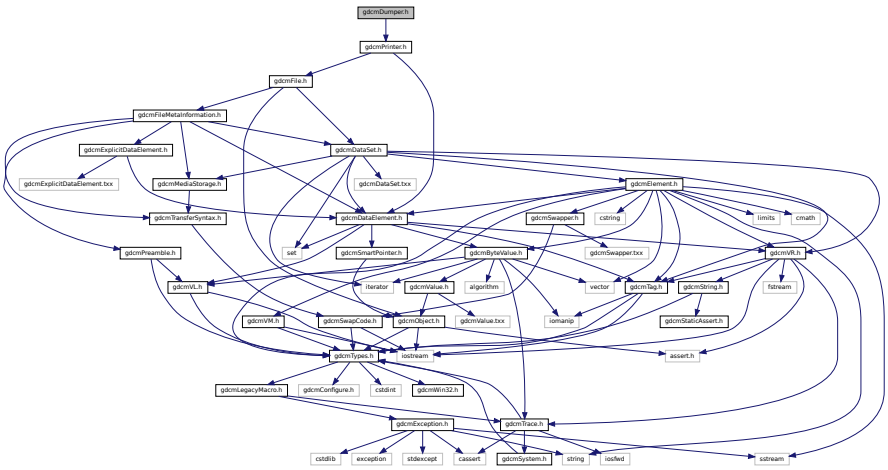
```
1 /*=========================================================================
2
3 Program:  GDCM (Grassroots DICOM). A DICOM library
4
5 Copyright (c) 2006-2011 Mathieu Malaterre
6 All rights reserved.
7 See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.
8
9 This software is distributed WITHOUT ANY WARRANTY; without even
10 the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
11 PURPOSE.  See the above copyright notice for more information.
12
13 =========================================================================*/
14 #ifndef GDCMDPATH_H
15 #define GDCMDPATH_H
16
17 #include "gdcmTag.h"
18 #include <string>
19
```

```
20 namespace gdcm {
21
28 class GDCM_EXPORT DPath {
29   friend std::ostream &operator«(std::ostream &_os, const DPath &_val);
30
31  public:
32   DPath();
33   ~DPath();
34   void Print(std::ostream &) const;
35   bool operator<(const DPath &rhs) const;
36
37   bool ConstructFromString(const char *path);
38
40   bool Match(DPath const &other) const;
41
43   static bool IsValid(const char *path);
44
45  private:
46   std::string Path;
47 };
48
49 inline std::ostream &operator«(std::ostream &os, const DPath &val) {
50   os « val.Path;
51   return os;
52 }
53
54 }  // end namespace gdcm
55
56 #endif  // GDCMDPATH_H
```

11.285 gdcmDumper.h File Reference

`#include "gdcmPrinter.h"`

Include dependency graph for gdcmDumper.h:



Classes

- class gdcm::Dumper

  Codec *class.*

Namespaces

- namespace gdcm

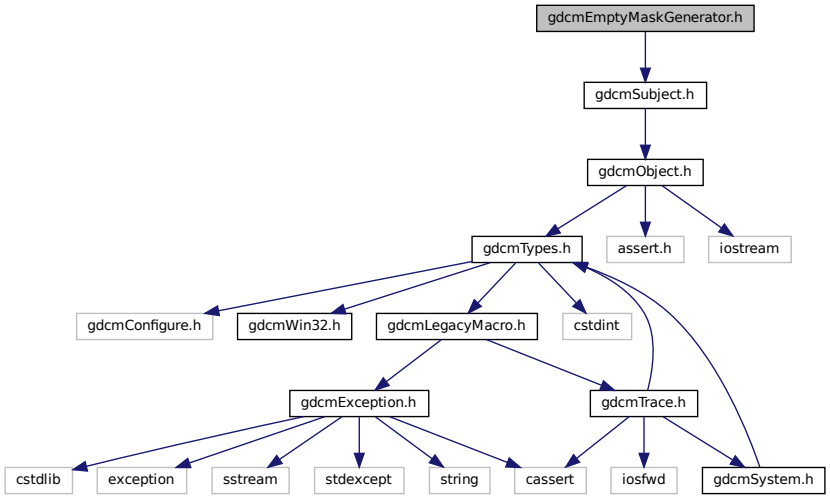
11.286 gdcmDumper.h

Go to the documentation of this file.

```
1 /*=========================================================================
2
3 Program:  GDCM (Grassroots DICOM). A DICOM library
4
5 Copyright (c) 2006-2011 Mathieu Malaterre
6 All rights reserved.
7 See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.
8
9 This software is distributed WITHOUT ANY WARRANTY; without even
10 the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
11 PURPOSE.  See the above copyright notice for more information.
12
13 =========================================================================*/
14 #ifndef GDCMDUMPER_H
15 #define GDCMDUMPER_H
16
17 #include "gdcmPrinter.h"
18
19 namespace gdcm
20 {
21
22 // It's a sink there is no output
30 class GDCM_EXPORT Dumper :  public Printer
31 {
32 public:
33   Dumper() { PrintStyle = CONDENSED_STYLE; }
34   ~Dumper() = default;
35 };
36
37 } // end namespace gdcm
38
39 #endif //GDCMDUMPER_H
```

11.287 gdcmEmptyMaskGenerator.h File Reference

`#include "gdcmSubject.h"`
Include dependency graph for gdcmEmptyMaskGenerator.h:



Classes

- class gdcm::EmptyMaskGenerator

  EmptyMaskGenerator Main class to generate a Empty Mask Series from an input Series. This class takes an input folder and generates a series of DICOM files in the specified output directory. This class handles multiples DICOM Series within the same input directory.

Namespaces

- namespace gdcm

11.288 gdcmEmptyMaskGenerator.h

Go to the documentation of this file.

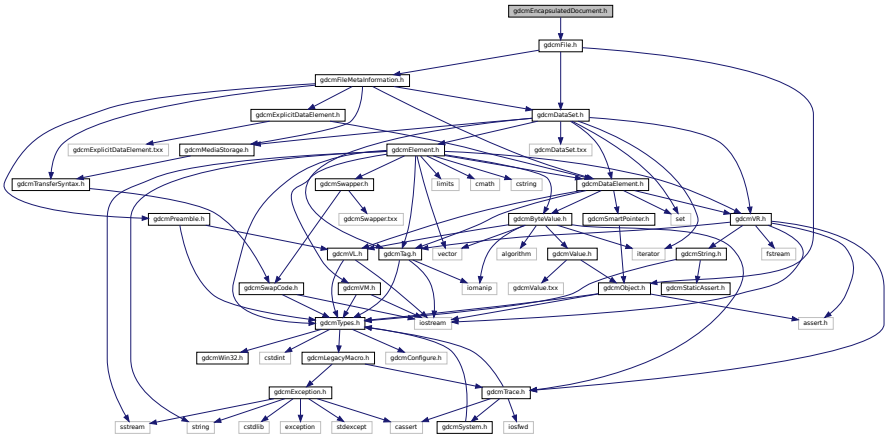
```
1 /*=========================================================================
2
3 Program:  GDCM (Grassroots DICOM). A DICOM library
4
5 Copyright (c) 2006-2011 Mathieu Malaterre
6 All rights reserved.
7 See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.
8
9 This software is distributed WITHOUT ANY WARRANTY; without even
10 the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
11 PURPOSE.  See the above copyright notice for more information.
12
```

```
13 =========================================================================*/
14 #ifndef GDCMEMPTYMASKGENERATOR_H
15 #define GDCMEMPTYMASKGENERATOR_H
16
17 #include "gdcmSubject.h"
18
19 namespace gdcm {
50 class GDCM_EXPORT EmptyMaskGenerator
51 {
52 public:
53   EmptyMaskGenerator();
54   ~EmptyMaskGenerator();
55
56   enum SOPClassUIDMode {
57     UseOriginalSOPClassUID = 0, // default
58     UseGrayscaleSecondaryImageStorage
59   };
60
63   void SetSOPClassUIDMode( SOPClassUIDMode mode );
64
66   void SetInputDirectory( const char * dirname );
67
69   void SetOutputDirectory( const char * dirname );
70
72   bool Execute();
73
74 private:
75   struct impl;
76   // PIMPL idiom
77   impl* pimpl;
78 };
79 } // end namespace gdcm
80 #endif //GDCMEMPTYMASKGENERATOR_H
```

11.289 gdcmEncapsulatedDocument.h File Reference

`#include "gdcmFile.h"`

Include dependency graph for gdcmEncapsulatedDocument.h:



Classes

- class gdcm::EncapsulatedDocument

  EncapsulatedDocument.

Namespaces

- namespace gdcm

11.290 gdcmEncapsulatedDocument.h

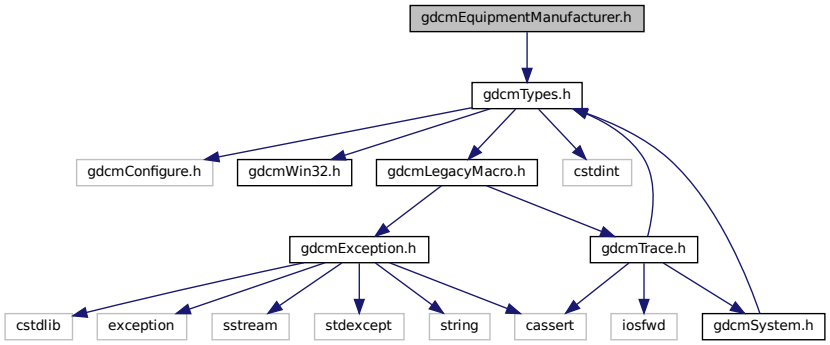
Go to the documentation of this file.

```
1 /*=========================================================================
2
3 Program:  GDCM (Grassroots DICOM). A DICOM library
4
5 Copyright (c) 2006-2011 Mathieu Malaterre
6 All rights reserved.
7 See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.
8
9 This software is distributed WITHOUT ANY WARRANTY; without even
10 the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
11 PURPOSE.  See the above copyright notice for more information.
12
13 =========================================================================*/
14 #ifndef GDCMENCAPSULATEDDOCUMENT_H
15 #define GDCMENCAPSULATEDDOCUMENT_H
16
17 #include "gdcmFile.h"
18
19 namespace gdcm
20 {
24 class GDCM_EXPORT EncapsulatedDocument
25 {
26 public:
27   EncapsulatedDocument() = default;
28
29 private:
30 };
31
32 } // end namespace gdcm
33
34 #endif //GDCMENCAPSULATEDDOCUMENT_H
```

11.291 gdcmEquipmentManufacturer.h File Reference

`#include "gdcmTypes.h"`

Include dependency graph for gdcmEquipmentManufacturer.h:



Classes

- class gdcm::EquipmentManufacturer

Namespaces

- namespace gdcm

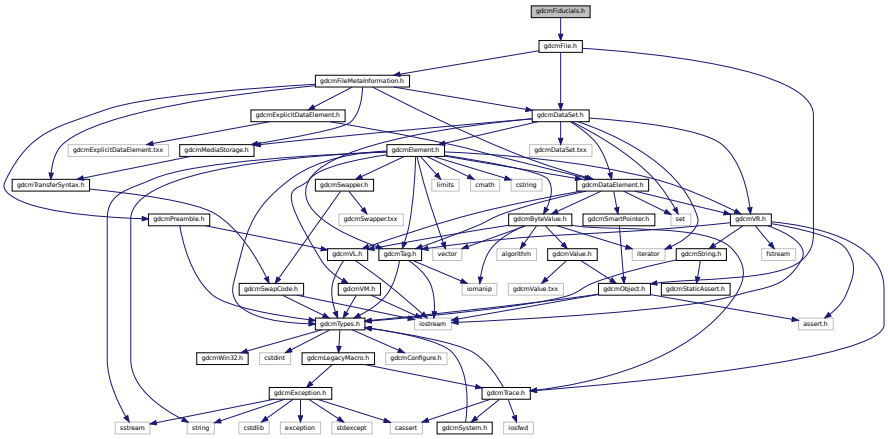
11.292 gdcmEquipmentManufacturer.h

Go to the documentation of this file.

```
1 /*=========================================================================
2
3 Program:  GDCM (Grassroots DICOM). A DICOM library
4
5 Copyright (c) 2006-2011 Mathieu Malaterre
6 All rights reserved.
7 See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.
8
9 This software is distributed WITHOUT ANY WARRANTY; without even
10 the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
11 PURPOSE.  See the above copyright notice for more information.
12
13 =========================================================================*/
14 #ifndef GDCMEQUIPMENTMANUFACTURER_H
15 #define GDCMEQUIPMENTMANUFACTURER_H
16
17 #include "gdcmTypes.h"
18
19 namespace gdcm
20 {
21
22 class DataSet;
30 class GDCM_EXPORT EquipmentManufacturer
31 {
32 public:
33
34   typedef enum {
35     UNKNOWN = 0,
36     FUJI,
37     GEMS,
38     HITACHI,
39     KODAK,
40     MARCONI,
41     PMS,
42     SIEMENS,
43     TOSHIBA
44   } Type;
45
46   static Type Compute( DataSet const & ds );
47
48   static const char *TypeToString( Type type );
49
50 private:
51   static EquipmentManufacturer::Type GuessFromPrivateAttributes( DataSet const & ds );
52 };
53
54 } // end namespace gdcm
55
56 #endif // GDCMEQUIPMENTMANUFACTURER_H
```

11.293 gdcmFiducials.h File Reference

`#include "gdcmFile.h"`
Include dependency graph for gdcmFiducials.h:



Classes

- class gdcm::Fiducials

  Fiducials.

Namespaces

- namespace gdcm

11.294 gdcmFiducials.h

Go to the documentation of this file.

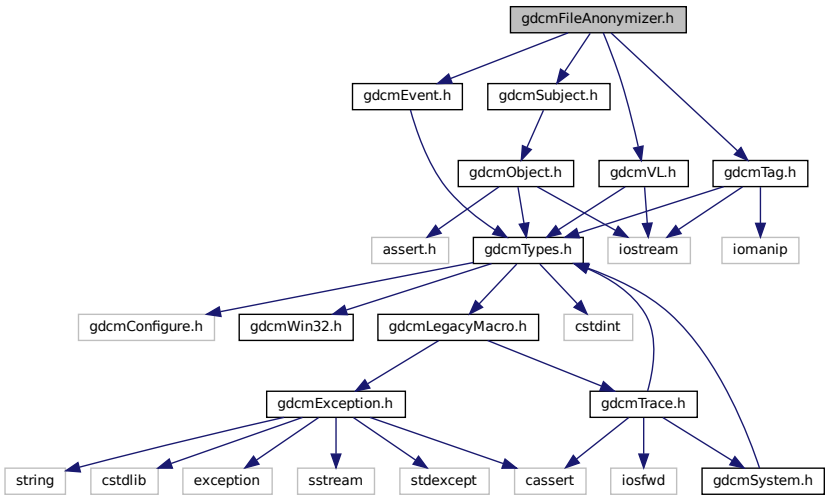
```
1 /*=========================================================================
2
3 Program:  GDCM (Grassroots DICOM). A DICOM library
4
5 Copyright (c) 2006-2011 Mathieu Malaterre
6 All rights reserved.
7 See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.
8
9 This software is distributed WITHOUT ANY WARRANTY; without even
10 the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
11 PURPOSE.  See the above copyright notice for more information.
12
13 =========================================================================*/
14 #ifndef GDCMFIDUCIALS_H
15 #define GDCMFIDUCIALS_H
16
17 #include "gdcmFile.h"
18
19 namespace gdcm
20 {
24 class GDCM_EXPORT Fiducials
```

```
25 {
26 public:
27   Fiducials() = default;
28
29 private:
30 };
31
32 } // end namespace gdcm
33
34 #endif //GDCMFIDUCIALS_H
```

11.295 gdcmFileAnonymizer.h File Reference

```
#include "gdcmSubject.h"
#include "gdcmEvent.h"
#include "gdcmTag.h"
#include "gdcmVL.h"
```

Include dependency graph for gdcmFileAnonymizer.h:



Classes

- class gdcm::FileAnonymizer

  FileAnonymizer.

Namespaces

- namespace gdcm

11.296 gdcmFileAnonymizer.h

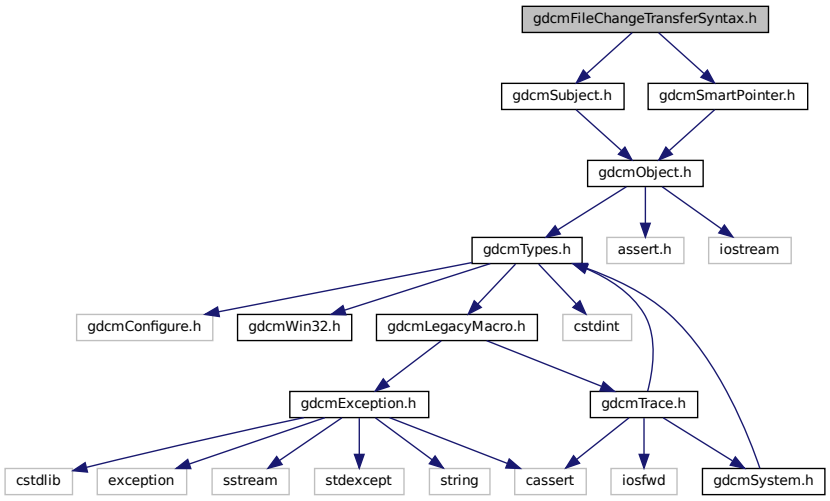
Go to the documentation of this file.

```
1 /*=========================================================================
2
3 Program:  GDCM (Grassroots DICOM). A DICOM library
4
5 Copyright (c) 2006-2011 Mathieu Malaterre
6 All rights reserved.
7 See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.
8
9 This software is distributed WITHOUT ANY WARRANTY; without even
10 the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
11 PURPOSE.  See the above copyright notice for more information.
12
13 =========================================================================*/
14 #ifndef GDCMFILEANONYMIZER_H
15 #define GDCMFILEANONYMIZER_H
16
17 #include "gdcmSubject.h"
18 #include "gdcmEvent.h"
19 #include "gdcmTag.h"
20 #include "gdcmVL.h"
21
22 namespace gdcm
23 {
24 class FileAnonymizerInternals;
25
47 class GDCM_EXPORT FileAnonymizer :  public Subject
48 {
49 public:
50   FileAnonymizer();
51   ~FileAnonymizer() override;
52
55   void Empty( Tag const &t );
56
58   void Remove( Tag const &t );
59
63   void Replace( Tag const &t, const char *value_str );
64
67   void Replace( Tag const &t, const char *value_data, VL const & vl );
68
70   void SetInputFileName(const char *filename_native);
71
73   void SetOutputFileName(const char *filename_native);
74
76   bool Write();
77
78 private:
79   bool ComputeEmptyTagPosition();
80   bool ComputeRemoveTagPosition();
81   bool ComputeReplaceTagPosition();
82   FileAnonymizerInternals *Internals;
83 };
84
85 } // end namespace gdcm
86
87 #endif //GDCMFILEANONYMIZER_H
```

11.297 gdcmFileChangeTransferSyntax.h File Reference

```
#include "gdcmSubject.h"
#include "gdcmSmartPointer.h"
```

Include dependency graph for gdcmFileChangeTransferSyntax.h:



Classes

- class gdcm::FileChangeTransferSyntax

  FileChangeTransferSyntax.

Namespaces

- namespace gdcm

11.298 gdcmFileChangeTransferSyntax.h

Go to the documentation of this file.

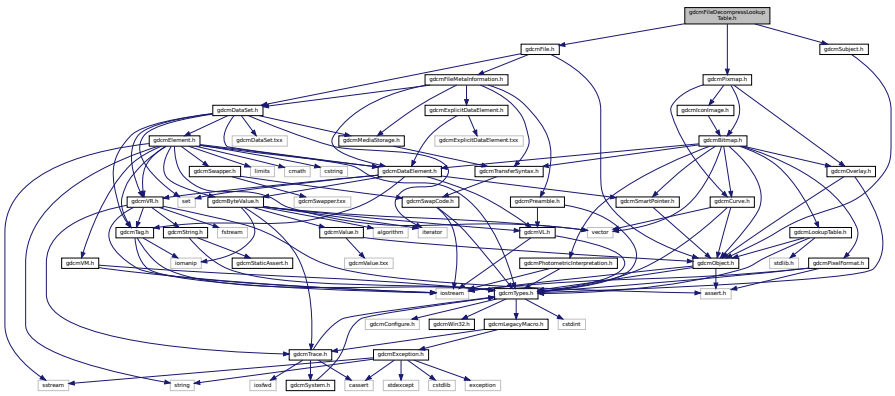
```
1 /*=========================================================================
2
3 Program:  GDCM (Grassroots DICOM). A DICOM library
4
5 Copyright (c) 2006-2011 Mathieu Malaterre
6 All rights reserved.
7 See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.
8
9 This software is distributed WITHOUT ANY WARRANTY; without even
10 the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
11 PURPOSE.  See the above copyright notice for more information.
12
13 =========================================================================*/
14 #ifndef GDCMFILECHANGETRANSFERSYNTAX_H
15 #define GDCMFILECHANGETRANSFERSYNTAX_H
16
17 #include "gdcmSubject.h"
18 #include "gdcmSmartPointer.h"
19
20 namespace gdcm
21 {
```

```
22 class FileChangeTransferSyntaxInternals;
23 class ImageCodec;
24 class TransferSyntax;
25
39 class GDCM_EXPORT FileChangeTransferSyntax :  public Subject
40 {
41 public:
42   FileChangeTransferSyntax();
43   ~FileChangeTransferSyntax() override;
44
46   void SetInputFileName(const char *filename_native);
47
49   void SetOutputFileName(const char *filename_native);
50
52   bool Change();
53
55   void SetTransferSyntax( TransferSyntax const & ts );
56
59   ImageCodec * GetCodec();
60
62   static SmartPointer<FileChangeTransferSyntax> New() { return new FileChangeTransferSyntax; }
63
64 private:
65   bool InitializeCopy();
66   bool UpdateCompressionLevel(double level);
67   FileChangeTransferSyntaxInternals *Internals;
68 };
69
70 } // end namespace gdcm
71
72 #endif //GDCMFILEANONYMIZER_H
```

11.299 gdcmFileDecompressLookupTable.h File Reference

```
#include "gdcmSubject.h"
#include "gdcmFile.h"
#include "gdcmPixmap.h"
```

Include dependency graph for gdcmFileDecompressLookupTable.h:



Classes

- class gdcm::FileDecompressLookupTable

  FileDecompressLookupTable class.

Namespaces

- namespace gdcm

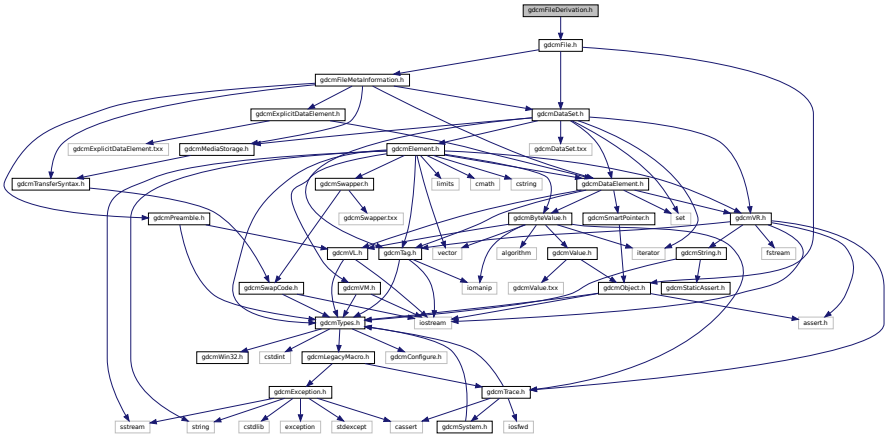
11.300 gdcmFileDecompressLookupTable.h

Go to the documentation of this file.

```
1 /*=========================================================================
2
3 Program:  GDCM (Grassroots DICOM). A DICOM library
4
5 Copyright (c) 2006-2011 Mathieu Malaterre
6 All rights reserved.
7 See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.
8
9 This software is distributed WITHOUT ANY WARRANTY; without even
10 the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
11 PURPOSE.  See the above copyright notice for more information.
12
13 =========================================================================*/
14 #ifndef GDCMFILEDECOMPRESSLOOKUPTABLE_H
15 #define GDCMFILEDECOMPRESSLOOKUPTABLE_H
16
17 #include "gdcmSubject.h"
18 #include "gdcmFile.h"
19 #include "gdcmPixmap.h"
20
21 namespace gdcm
22 {
23
24 class DataElement;
30 class GDCM_EXPORT FileDecompressLookupTable :  public Subject
31 {
32 public:
33   FileDecompressLookupTable() = default;
34   ~FileDecompressLookupTable() override = default;
35
37   bool Change();
38
40   void SetFile(const File& f) { F = f; }
41   File &GetFile() { return *F; }
42
43   const Pixmap& GetPixmap()const { return *PixelData; }
44   Pixmap& GetPixmap() { return *PixelData; }
45   void SetPixmap(Pixmap const &img) { PixelData = img; }
46
47 protected:
48
49 private:
50   SmartPointer<File> F;
51   SmartPointer<Pixmap> PixelData;
52 };
53
54 } // end namespace gdcm
55
56 #endif //GDCMFILEDECOMPRESSLOOKUPTABLE_H
```

11.301 gdcmFileDerivation.h File Reference

`#include "gdcmFile.h"`
Include dependency graph for gdcmFileDerivation.h:



Classes

- class gdcm::FileDerivation

  FileDerivation class.

Namespaces

- namespace gdcm

11.302 gdcmFileDerivation.h

Go to the documentation of this file.

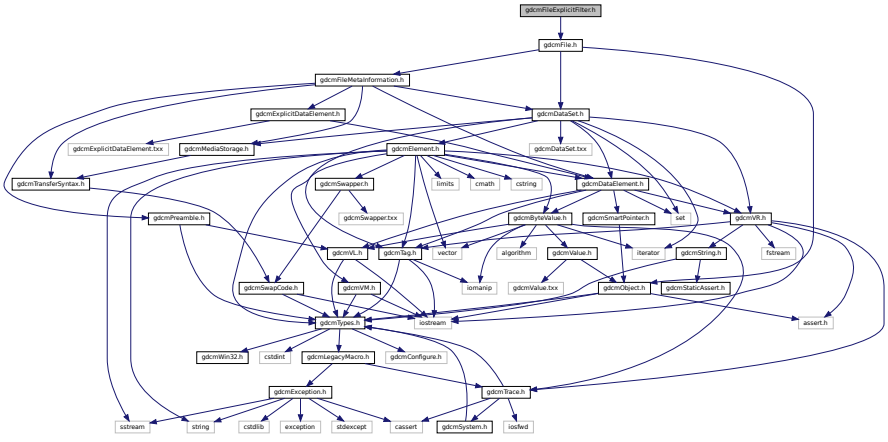
```
1 /*=========================================================================
2
3 Program:  GDCM (Grassroots DICOM). A DICOM library
4
5 Copyright (c) 2006-2011 Mathieu Malaterre
6 All rights reserved.
7 See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.
8
9 This software is distributed WITHOUT ANY WARRANTY; without even
10 the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
11 PURPOSE.  See the above copyright notice for more information.
12
13 =========================================================================*/
14 #ifndef GDCMFILEDERIVATION_H
15 #define GDCMFILEDERIVATION_H
16
17 #include "gdcmFile.h"
18
19 namespace gdcm
20 {
21
```

```
22 class FileDerivationInternals;
23 class DataSet;
39 class GDCM_EXPORT FileDerivation
40 {
41 public:
42   FileDerivation();
43   ~FileDerivation();
44
49   bool AddReference(const char *referencedsopclassuid, const char *referencedsopinstanceuid);
50
51   // CID 7202 Source Image Purposes of Reference
52   // {"DCM",121320,"Uncompressed predecessor"},
53
55   void SetPurposeOfReferenceCodeSequenceCodeValue(unsigned int codevalue);
56
57   // CID 7203 Image Derivation
58   // { "DCM",113040,"Lossy Compression" },
59
61   void SetDerivationCodeSequenceCodeValue(unsigned int codevalue);
62
64   void SetDerivationDescription( const char *dd );
65
69   void SetAppendDerivationHistory(bool b);
70
72   bool Derive();
73
75   void SetFile(const File& f) { F = f; }
76   File &GetFile() { return *F; }
77   const File &GetFile()const { return *F; }
78
79 protected:
80   bool AddDerivationDescription();
81   bool AddSourceImageSequence();
82   bool AddPurposeOfReferenceCodeSequence(DataSet &ds);
83
84 private:
85   SmartPointer<File> F;
86   FileDerivationInternals *Internals;
87 };
88
96 } // end namespace gdcm
97
98 #endif //GDCMFILEDERIVATION_H
```

11.303 gdcmFileExplicitFilter.h File Reference

`#include "gdcmFile.h"`

Include dependency graph for gdcmFileExplicitFilter.h:



Classes

- class gdcm::FileExplicitFilter

  FileExplicitFilter class.

Namespaces

- namespace gdcm

11.304 gdcmFileExplicitFilter.h

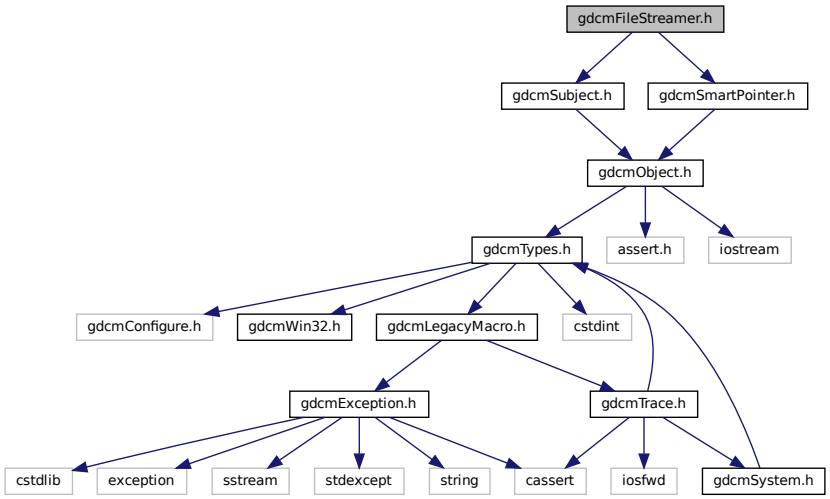
Go to the documentation of this file.

```
1 /*=========================================================================
2
3 Program:  GDCM (Grassroots DICOM). A DICOM library
4
5 Copyright (c) 2006-2011 Mathieu Malaterre
6 All rights reserved.
7 See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.
8
9 This software is distributed WITHOUT ANY WARRANTY; without even
10 the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
11 PURPOSE.  See the above copyright notice for more information.
12
13 =========================================================================*/
14 #ifndef GDCMFILEEXPLICITFILTER_H
15 #define GDCMFILEEXPLICITFILTER_H
16
17 #include "gdcmFile.h"
18
19 namespace gdcm
20 {
21 class Dicts;
22
38 class GDCM_EXPORT FileExplicitFilter
39 {
40 public:
41   FileExplicitFilter():F(new
       File),ChangePrivateTags(false),UseVRUN(true),RecomputeItemLength(false),RecomputeSequenceLength(false) {}
42   ~FileExplicitFilter() = default;
43
45   void SetChangePrivateTags(bool b) { ChangePrivateTags = b;}
46
48   void SetUseVRUN(bool b) { UseVRUN = b; }
49
51   void SetRecomputeItemLength(bool b);
52   void SetRecomputeSequenceLength(bool b);
53
55
57   bool Change();
58
60   void SetFile(const File& f) { F = f; }
61   File &GetFile() { return *F; }
62
63 protected:
64   bool ProcessDataSet(DataSet &ds, Dicts const & dicts);
65   bool ChangeFMI();
66
67 private:
68   SmartPointer<File> F;
69   bool ChangePrivateTags;
70   bool UseVRUN;
71   bool RecomputeItemLength;
72   bool RecomputeSequenceLength;
73 };
74
75
76 } // end namespace gdcm
77
78 #endif //GDCMFILEEXPLICITFILTER_H
```

11.305 gdcmFileStreamer.h File Reference

```
#include "gdcmSubject.h"
#include "gdcmSmartPointer.h"
```

Include dependency graph for gdcmFileStreamer.h:



Classes

- class gdcm::FileStreamer

  FileStreamer.

Namespaces

- namespace gdcm

11.306 gdcmFileStreamer.h

Go to the documentation of this file.

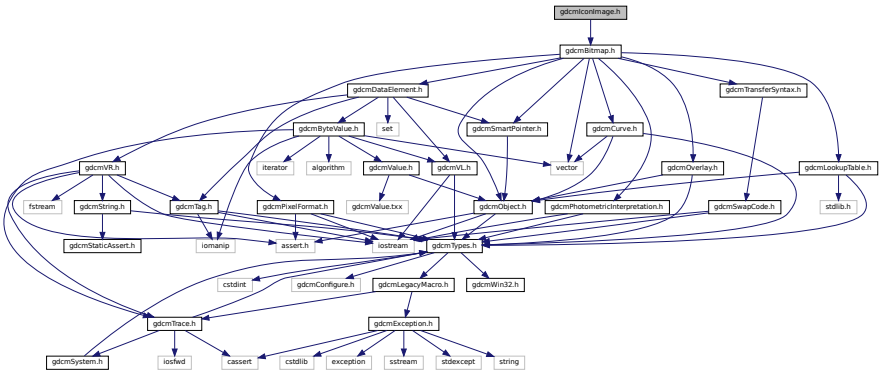
```
1 /*=========================================================================
2
3 Program:  GDCM (Grassroots DICOM). A DICOM library
4
5 Copyright (c) 2006-2011 Mathieu Malaterre
6 All rights reserved.
7 See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.
8
9 This software is distributed WITHOUT ANY WARRANTY; without even
10 the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
11 PURPOSE.  See the above copyright notice for more information.
12
13 =========================================================================*/
```

```
14 #ifndef GDCMFILESTREAMER_H
15 #define GDCMFILESTREAMER_H
16
17 #include "gdcmSubject.h"
18 #include "gdcmSmartPointer.h"
19
20 namespace gdcm
21 {
22 class FileStreamerInternals;
23
24 class Tag;
25 class PrivateTag;
41 class GDCM_EXPORT FileStreamer :  public Subject
42 {
43 public:
44   FileStreamer();
45   ~FileStreamer() override;
46
48   void SetTemplateFileName(const char *filename_native);
49
50   // Decide to check template or not (default:  false)
55   void CheckTemplateFileName(bool check);
56
58   void SetOutputFileName(const char *filename_native);
59
64   bool CheckDataElement( const Tag & t );
65
68   bool StartDataElement( const Tag & t );
70   bool AppendToDataElement( const Tag & t, const char *array, size_t len );
72   bool StopDataElement( const Tag & t );
76   bool ReserveDataElement( size_t len );
77
85   bool StartGroupDataElement( const PrivateTag & pt, size_t maxsizede = 0, uint8_t startoffset = 0 );
87   bool AppendToGroupDataElement( const PrivateTag & pt, const char *array, size_t len );
89   bool StopGroupDataElement( const PrivateTag & pt );
92   bool ReserveGroupDataElement( unsigned short ndataelement );
93
95   static SmartPointer<FileStreamer> New() { return new FileStreamer; }
96
97 private:
98   bool InitializeCopy();
99   FileStreamerInternals *Internals;
100 };
101
102 } // end namespace gdcm
103
104 #endif //GDCMFILESTREAMER_H
```

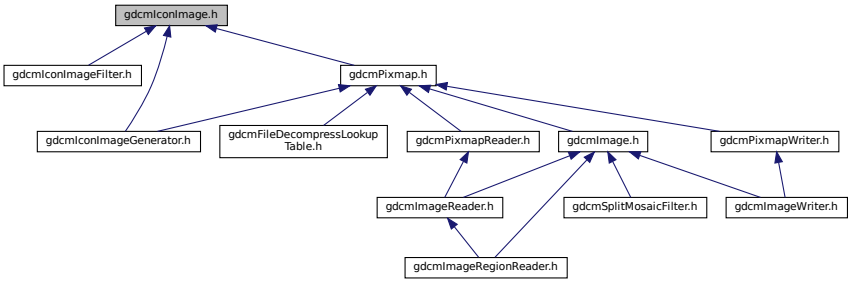
11.307 gdcmIconImage.h File Reference

`#include "gdcmBitmap.h"`

Include dependency graph for gdcmIconImage.h:



This graph shows which files directly or indirectly include this file:



Namespaces

- namespace gdcm

Typedefs

- typedef Bitmap gdcm::IconImage

11.308 gdcmIconImage.h

Go to the documentation of this file.

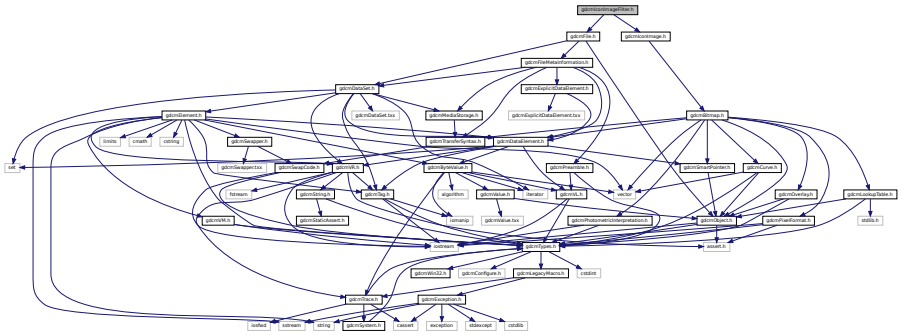
```
1 /*=========================================================================
2
3 Program:  GDCM (Grassroots DICOM). A DICOM library
4
5 Copyright (c) 2006-2011 Mathieu Malaterre
6 All rights reserved.
7 See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.
8
9 This software is distributed WITHOUT ANY WARRANTY; without even
10 the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
11 PURPOSE.  See the above copyright notice for more information.
12
13 =========================================================================*/
14 #ifndef GDCMICONIMAGE_H
15 #define GDCMICONIMAGE_H
16
17 #if 0
18 #include "gdcmObject.h"
19 #include "gdcmDataElement.h"
20 #include "gdcmPhotometricInterpretation.h"
21 #include "gdcmPixelFormat.h"
22 #include "gdcmTransferSyntax.h"
23
24 #include <vector>
25
26 namespace gdcm
27 {
28
32 class GDCM_EXPORT IconImage :  public Object
33 {
34 public:
35   IconImage();
36   ~IconImage();
37   void Print(std::ostream &)const {}
```

```
38
40   void SetTransferSyntax(TransferSyntax const &ts) {
41     TS = ts;
42   }
43   const TransferSyntax &GetTransferSyntax()const {
44     return TS;
45   }
46   void SetDataElement(DataElement const &de) {
47     PixelData = de;
48   }
49   const DataElement& GetDataElement()const { return PixelData; }
50
51   void SetColumns(unsigned int col) { SetDimension(0,col); }
52   void SetRows(unsigned int rows) { SetDimension(1,rows); }
53   void SetDimension(unsigned int idx, unsigned int dim);
54   int GetColumns()const { return Dimensions[0]; }
55   int GetRows()const { return Dimensions[1]; }
56   // Get/Set PixelFormat
57   const PixelFormat &GetPixelFormat()const
58 {
59     return PF;
60     }
61   void SetPixelFormat(PixelFormat const &pf)
62     {
63     PF = pf;
64     }
65
66   const PhotometricInterpretation &GetPhotometricInterpretation() const;
67   void SetPhotometricInterpretation(PhotometricInterpretation const &pi);
68
69   bool IsEmpty()const { return Dimensions.size() == 0; }
70   void Clear();
71
72   bool GetBuffer(char *buffer) const;
73
74 private:
75   TransferSyntax TS;
76   PixelFormat PF; // SamplesPerPixel, BitsAllocated, BitsStored, HighBit, PixelRepresentation
77   PhotometricInterpretation PI;
78   std::vector<unsigned int> Dimensions; // Col/Row
79   std::vector<double> Spacing; // PixelAspectRatio ?
80   DataElement PixelData; // copied from 7fe0,0010
81   static const unsigned int NumberOfDimensions = 2;
82 };
83
84 } // end namespace gdcm
85 #endif
86 #include "gdcmBitmap.h"
87
88 namespace gdcm
89 {
90   //class GDCM_EXPORT IconImage :  public Pixmap {};
91   typedef Bitmap IconImage;
92 }
93
94 #endif //GDCMICONIMAGE_H
```

11.309 gdcmIconImageFilter.h File Reference

```
#include "gdcmFile.h"
#include "gdcmIconImage.h"
```

Include dependency graph for gdcmIconImageFilter.h:



Classes

- class gdcm::IconImageFilter

  IconImageFilter.

Namespaces

- namespace gdcm

11.310 gdcmIconImageFilter.h

Go to the documentation of this file.

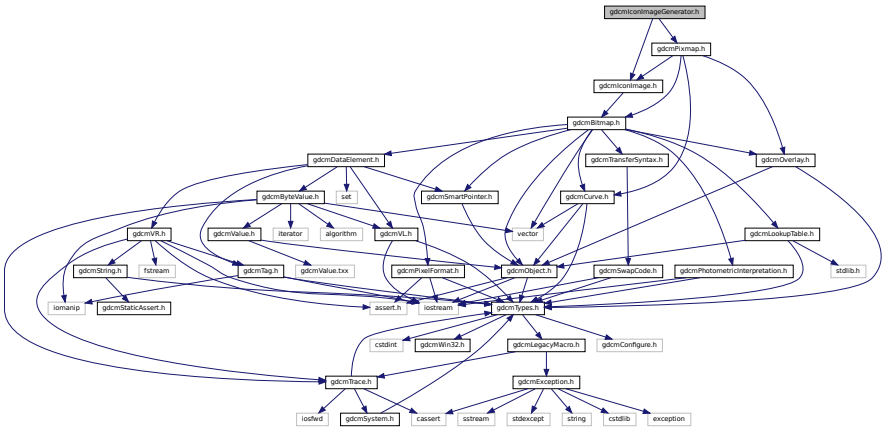
```
1 /*=========================================================================
2
3 Program:  GDCM (Grassroots DICOM). A DICOM library
4
5 Copyright (c) 2006-2011 Mathieu Malaterre
6 All rights reserved.
7 See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.
8
9 This software is distributed WITHOUT ANY WARRANTY; without even
10 the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
11 PURPOSE.  See the above copyright notice for more information.
12
13 =========================================================================*/
14 #ifndef GDCMICONIMAGEFILTER_H
15 #define GDCMICONIMAGEFILTER_H
16
17 #include "gdcmFile.h"
18 #include "gdcmIconImage.h"
19
20 namespace gdcm
21 {
22 class IconImageFilterInternals;
23
49 class GDCM_EXPORT IconImageFilter
50 {
51 public:
52   IconImageFilter();
53   ~IconImageFilter();
54
56   void SetFile(const File& f) { F = f; }
57   File &GetFile() { return *F; }
58   const File &GetFile()const { return *F; }
```

```
59
61   bool Extract();
62
64   unsigned int GetNumberOfIconImages() const;
65   IconImage& GetIconImage( unsigned int i ) const;
66
67 protected:
68   void ExtractIconImages();
69   void ExtractVeproIconImages();
70
71 private:
72   SmartPointer<File> F;
73   IconImageFilterInternals *Internals;
74 };
75
76 } // end namespace gdcm
77
78 #endif //GDCMICONIMAGEFILTER_H
```

11.311 gdcmIconImageGenerator.h File Reference

```
#include "gdcmPixmap.h"
#include "gdcmIconImage.h"
```

Include dependency graph for gdcmIconImageGenerator.h:



Classes

- class gdcm::IconImageGenerator

  IconImageGenerator.

Namespaces

- namespace gdcm

11.312 gdcmIconImageGenerator.h

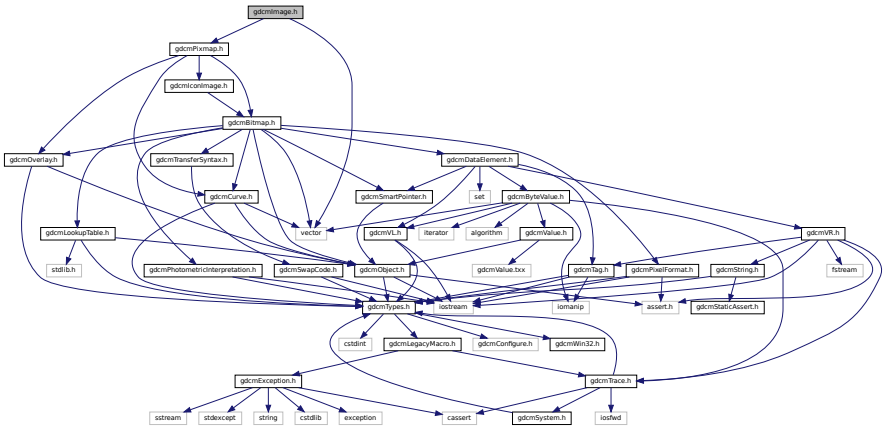
Go to the documentation of this file.

```
1 /*=========================================================================
2
3 Program:  GDCM (Grassroots DICOM). A DICOM library
4
5 Copyright (c) 2006-2011 Mathieu Malaterre
6 All rights reserved.
7 See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.
8
9 This software is distributed WITHOUT ANY WARRANTY; without even
10 the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
11 PURPOSE.  See the above copyright notice for more information.
12
13 =========================================================================*/
14 #ifndef GDCMICONIMAGEGENERATOR_H
15 #define GDCMICONIMAGEGENERATOR_H
16
17 #include "gdcmPixmap.h"
18 #include "gdcmIconImage.h"
19
20 namespace gdcm
21 {
22 class IconImageGeneratorInternals;
41 class GDCM_EXPORT IconImageGenerator
42 {
43 public:
44   IconImageGenerator();
45   ~IconImageGenerator();
46
48   void SetPixmap(const Pixmap& p) { P = p; }
49   Pixmap &GetPixmap() { return *P; }
50   const Pixmap &GetPixmap()const { return *P; }
51
53   void SetOutputDimensions(const unsigned int dims[2]);
54
58   void SetPixelMinMax(double min, double max);
59
63   void AutoPixelMinMax(bool b);
64
69   void ConvertRGBToPaletteColor(bool b);
70
74   void SetOutsideValuePixel(double v);
75
77   bool Generate();
78
80   const IconImage& GetIconImage()const { return *I; }
81
82 protected:
83
84 private:
85   void BuildLUT( Bitmap & bitmap, unsigned int maxcolor );
86
87   SmartPointer<Pixmap> P;
88   SmartPointer<IconImage> I;
89   IconImageGeneratorInternals *Internals;
90 };
91
92 } // end namespace gdcm
93
94 #endif //GDCMICONIMAGEGENERATOR_H
```

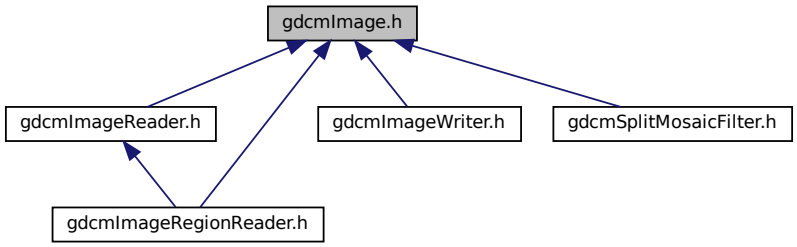
11.313 gdcmImage.h File Reference

```
#include "gdcmPixmap.h"
#include <vector>
```

Include dependency graph for gdcmImage.h:



This graph shows which files directly or indirectly include this file:



Classes

- class gdcm::Image

  Image.

Namespaces

- namespace gdcm

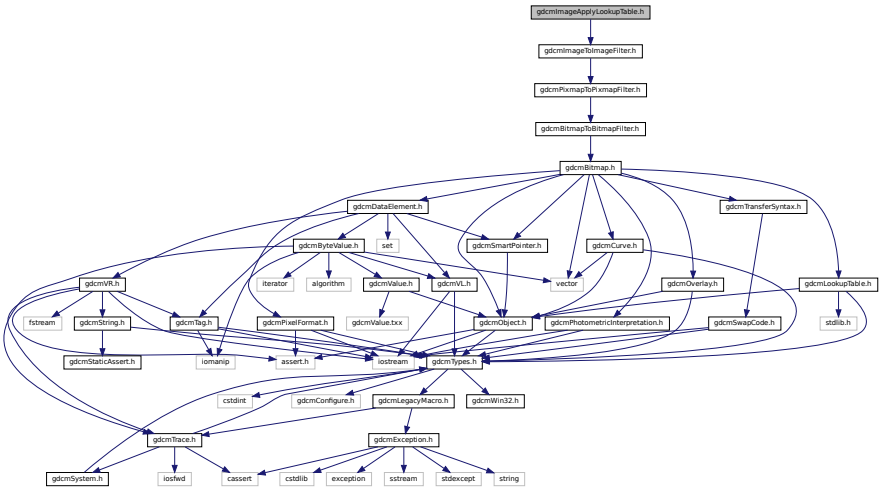
11.314 gdcmImage.h

Go to the documentation of this file.

```
1 /*=========================================================================
2
3 Program:  GDCM (Grassroots DICOM). A DICOM library
4
5 Copyright (c) 2006-2011 Mathieu Malaterre
6 All rights reserved.
7 See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.
8
9 This software is distributed WITHOUT ANY WARRANTY; without even
10 the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
11 PURPOSE.  See the above copyright notice for more information.
12
13 =========================================================================*/
14 #ifndef GDCMIMAGE_H
15 #define GDCMIMAGE_H
16
17 #include "gdcmPixmap.h"
18
19 #include <vector>
20
21 namespace gdcm
22 {
23
46 class GDCM_EXPORT Image :  public Pixmap
47 {
48 public:
49   Image ():Spacing(),SC(),Intercept(0),Slope(1) {
50     //DirectionCosines.resize(6);
51   Origin.resize( 3 /*NumberOfDimensions*/ ); // fill with 0
52   DirectionCosines.resize( 6 ); // fill with 0
53   DirectionCosines[0] = 1;
54   DirectionCosines[4] = 1;
55   Spacing.resize( 3 /*NumberOfDimensions*/, 1 ); // fill with 1
56
57   }
58   ~Image() override = default;
59
63   const double *GetSpacing() const;
64   double GetSpacing(unsigned int idx) const;
65   void SetSpacing(const double spacing[3]);
66   void SetSpacing(unsigned int idx, double spacing);
67
70   const double *GetOrigin() const;
71   double GetOrigin(unsigned int idx) const;
72   void SetOrigin(const float origin[3]);
73   void SetOrigin(const double origin[3]);
74   void SetOrigin(unsigned int idx, double ori);
75
78   const double *GetDirectionCosines() const;
79   double GetDirectionCosines(unsigned int idx) const;
80   void SetDirectionCosines(const float dircos[6]);
81   void SetDirectionCosines(const double dircos[6]);
82   void SetDirectionCosines(unsigned int idx, double dircos);
83
85   void Print(std::ostream &os) const override;
86
88   void SetIntercept(double intercept) { Intercept = intercept; }
89   double GetIntercept()const { return Intercept; }
90
92   void SetSlope(double slope) { Slope = slope; }
93   double GetSlope()const { return Slope; }
94
95 private:
96   std::vector<double> Spacing;
97   std::vector<double> Origin;
98   std::vector<double> DirectionCosines;
99
100   // I believe the following 3 ivars can be derived from TS ...
101   SwapCode SC;
102   double Intercept;
103   double Slope;
104 };
105
111 } // end namespace gdcm
112
113 #endif //GDCMIMAGE_H
```

11.315 gdcmImageApplyLookupTable.h File Reference

`#include "gdcmImageToImageFilter.h"`
Include dependency graph for gdcmImageApplyLookupTable.h:



Classes

- class gdcm::ImageApplyLookupTable

  ImageApplyLookupTable class.

Namespaces

- namespace gdcm

11.316 gdcmImageApplyLookupTable.h

Go to the documentation of this file.

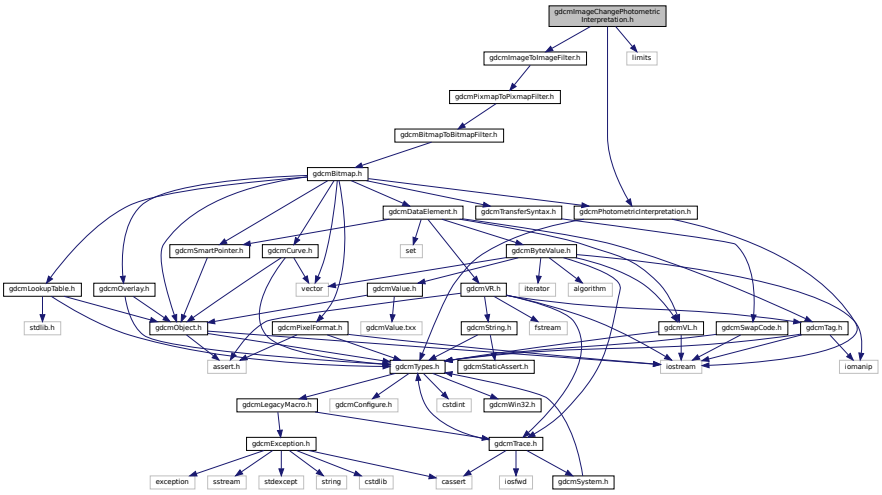
```
1 /*=========================================================================
2
3 Program:  GDCM (Grassroots DICOM). A DICOM library
4
5 Copyright (c) 2006-2011 Mathieu Malaterre
6 All rights reserved.
7 See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.
8
9 This software is distributed WITHOUT ANY WARRANTY; without even
10 the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
11 PURPOSE.  See the above copyright notice for more information.
12
13 =========================================================================*/
14 #ifndef GDCMIMAGEAPPLYLOOKUPTABLE_H
15 #define GDCMIMAGEAPPLYLOOKUPTABLE_H
16
17 #include "gdcmImageToImageFilter.h"
18
```

```
19 namespace gdcm
20 {
21
22 class DataElement;
28 class GDCM_EXPORT ImageApplyLookupTable :  public ImageToImageFilter
29 {
30 public:
31   ImageApplyLookupTable();
32   ~ImageApplyLookupTable();
33
35   bool Apply();
36
38   void SetRGB8(bool b);
39
40 protected:
41
42 private:
43   struct impl;
44   // PIMPL idiom
45   impl* pimpl;
46 };
47
48 } // end namespace gdcm
49
50 #endif //GDCMIMAGEAPPLYLOOKUPTABLE_H
```

11.317 gdcmImageChangePhotometricInterpretation.h File Reference

```
#include "gdcmImageToImageFilter.h"
#include "gdcmPhotometricInterpretation.h"
#include <limits>
```

Include dependency graph for gdcmImageChangePhotometricInterpretation.h:



Classes

- class gdcm::ImageChangePhotometricInterpretation

  ImageChangePhotometricInterpretation class.

Namespaces

- namespace gdcm

Functions

- template<typename T >
  static T gdcm::Clamp (int v)
- template<typename T >
  static int gdcm::Round (T x)

11.318 gdcmImageChangePhotometricInterpretation.h

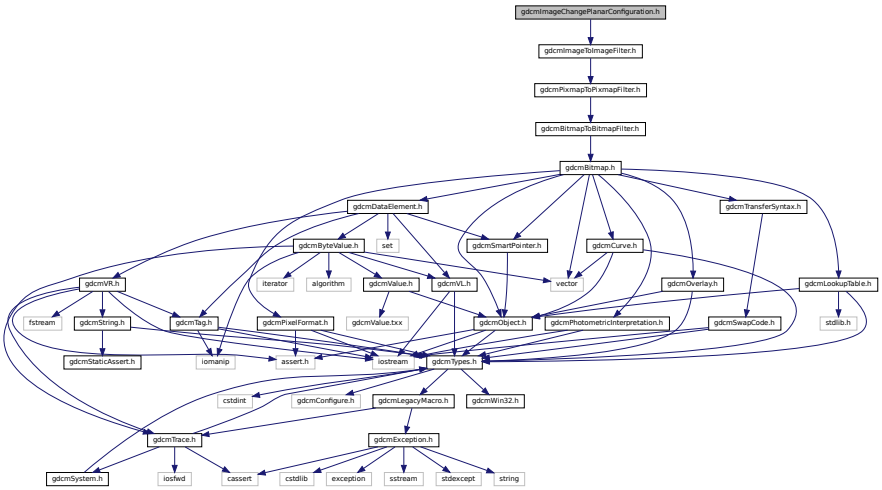
Go to the documentation of this file.

```
1 /*=========================================================================
2
3 Program:  GDCM (Grassroots DICOM). A DICOM library
4
5 Copyright (c) 2006-2011 Mathieu Malaterre
6 All rights reserved.
7 See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.
8
9 This software is distributed WITHOUT ANY WARRANTY; without even
10 the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
11 PURPOSE.  See the above copyright notice for more information.
12
13 =========================================================================*/
14 #ifndef GDCMIMAGECHANGEPHOTOMETRICINTERPRETATION_H
15 #define GDCMIMAGECHANGEPHOTOMETRICINTERPRETATION_H
16
17 #include "gdcmImageToImageFilter.h"
18 #include "gdcmPhotometricInterpretation.h"
19 #include <limits>
20
21 namespace gdcm
22 {
23
24 class DataElement;
29 class GDCM_EXPORT ImageChangePhotometricInterpretation :  public ImageToImageFilter
30 {
31 public:
32   ImageChangePhotometricInterpretation():PI() {}
33   ~ImageChangePhotometricInterpretation() = default;
34
36   void SetPhotometricInterpretation(PhotometricInterpretation const &pi) { PI = pi; }
37   const PhotometricInterpretation &GetPhotometricInterpretation()const { return PI; }
38
40   bool Change();
41
44   template <typename T>
45   static void RGB2YBR(T ybr[3], const T rgb[3], unsigned short storedbits = 8);
46   template <typename T>
47   static void YBR2RGB(T rgb[3], const T ybr[3], unsigned short storedbits = 8);
48
49 protected:
50   bool ChangeMonochrome();
51   bool ChangeYBR2RGB();
52   bool ChangeRGB2YBR();
53
54 private:
55   PhotometricInterpretation PI;
56 };
57
58 template <typename T>
59 static inline int Round(T x)
60 {
61   return (int)(x+0.5);
62 }
```

```
63
64 template <typename T>
65 static inline T Clamp(int v)
66 {
67   assert( std::numeric_limits<T>::min() == 0 );
68   return v < 0 ?  0 :  (v > std::numeric_limits<T>::max() ?  std::numeric_limits<T>::max() :  v);
69 }
70
71
72 template <typename T>
73 void ImageChangePhotometricInterpretation::RGB2YBR(T ybr[3], const T rgb[3], unsigned short storedbits)
74 {
75   // Implementation details, since the equations from:
76   // http://dicom.nema.org/medical/dicom/current/output/chtml/part03/sect_C.7.6.3.html#sect_C.7.6.3.1.2
77   // are rounded to the 4th decimal precision, prefer the exact equation from the original document at:
78   // CCIR Recommendation 601-2, also found in T.871 (Section §7, page 4)
79   const double R = rgb[0];
80   const double G = rgb[1];
81   const double B = rgb[2];
82   assert( storedbits <= sizeof(T) * 8 );
83   const int halffullscale = 1 « (storedbits - 1);
84   const int Y  = Round(  0.299 * R + 0.587 * G + 0.114 * B                      );
85   const int CB = Round((-0.299 * R - 0.587 * G + 0.886 * B)/1.772 + halffullscale);
86   const int CR = Round(( 0.701 * R - 0.587 * G - 0.114 * B)/1.402 + halffullscale);
87   ybr[0] = Clamp<T>(Y );
88   ybr[1] = Clamp<T>(CB);
89   ybr[2] = Clamp<T>(CR);
90 }
91
92 template <typename T>
93 void ImageChangePhotometricInterpretation::YBR2RGB(T rgb[3], const T ybr[3], unsigned short storedbits)
94 {
95   const double Y  = ybr[0];
96   const double Cb = ybr[1];
97   const double Cr = ybr[2];
98   assert( storedbits <= sizeof(T) * 8 );
99   const int halffullscale = 1 « (storedbits - 1);
100   const int R = Round(Y                                          + 1.402 * (Cr-halffullscale)               );
101   const int G = Round(Y -( 0.114 * 1.772 * (Cb-halffullscale) + 0.299 * 1.402 * (Cr-halffullscale))/0.587);
102   const int B = Round(Y          + 1.772 * (Cb-halffullscale)                                               );
103   rgb[0] = Clamp<T>(R);
104   rgb[1] = Clamp<T>(G);
105   rgb[2] = Clamp<T>(B);
106 }
107
108 } // end namespace gdcm
109
110 #endif //GDCMIMAGECHANGEPHOTOMETRICINTERPRETATION_H
```

11.319 gdcmImageChangePlanarConfiguration.h File Reference

`#include "gdcmImageToImageFilter.h"`
Include dependency graph for gdcmImageChangePlanarConfiguration.h:



Classes

- class gdcm::ImageChangePlanarConfiguration

  ImageChangePlanarConfiguration class.

Namespaces

- namespace gdcm

11.320 gdcmImageChangePlanarConfiguration.h

Go to the documentation of this file.

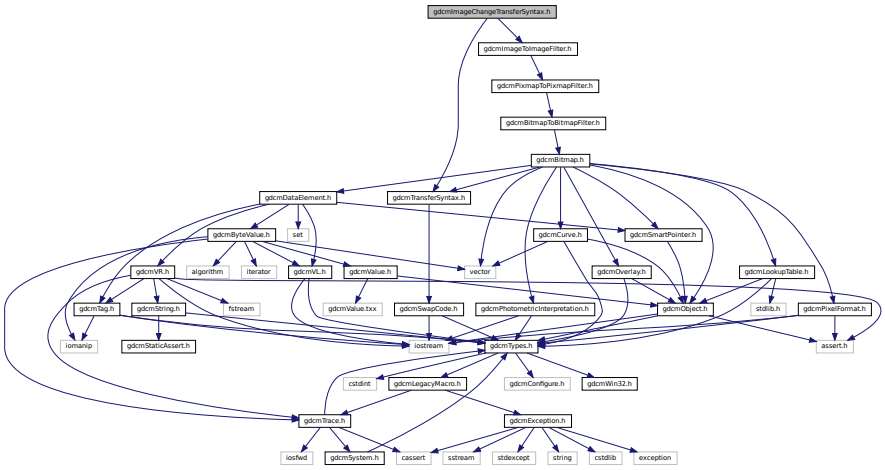
```
1 /*=========================================================================
2
3 Program:  GDCM (Grassroots DICOM). A DICOM library
4
5 Copyright (c) 2006-2011 Mathieu Malaterre
6 All rights reserved.
7 See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.
8
9 This software is distributed WITHOUT ANY WARRANTY; without even
10 the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
11 PURPOSE.  See the above copyright notice for more information.
12
13 =========================================================================*/
14 #ifndef GDCMIMAGECHANGEPLANARCONFIGURATION_H
15 #define GDCMIMAGECHANGEPLANARCONFIGURATION_H
16
17 #include "gdcmImageToImageFilter.h"
18
```

```
19 namespace gdcm
20 {
21
22 class DataElement;
28 class GDCM_EXPORT ImageChangePlanarConfiguration :  public ImageToImageFilter
29 {
30 public:
31   ImageChangePlanarConfiguration():PlanarConfiguration(0) {}
32   ~ImageChangePlanarConfiguration() = default;
33
35   void SetPlanarConfiguration(unsigned int pc) { PlanarConfiguration = pc; }
36   unsigned int GetPlanarConfiguration()const { return PlanarConfiguration; }
37
40   template <typename T>
41   static size_t RGBPlanesToRGBPixels(T *out, const T *r, const T *g, const T *b, size_t s);
42
46   template <typename T>
47   static size_t RGBPixelsToRGBPlanes(T *r, T *g, T *b, const T* rgb, size_t s);
48
50   bool Change();
51
52 protected:
53
54 private:
55   unsigned int PlanarConfiguration;
56 };
57
58 template <typename T>
59 size_t ImageChangePlanarConfiguration::RGBPlanesToRGBPixels(T *out, const T *r, const T *g, const T *b,
     size_t s)
60 {
61   T *pout = out;
62   for(size_t i = 0; i < s; ++i )
63     {
64     *pout++ = *r++;
65     *pout++ = *g++;
66     *pout++ = *b++;
67     }
68
69   assert( (size_t)(pout - out) == 3 * s );
70   return pout - out;
71 }
72
73 template <typename T>
74 size_t ImageChangePlanarConfiguration::RGBPixelsToRGBPlanes(T *r, T *g, T *b, const T *rgb, size_t s)
75 {
76   const T *prgb = rgb;
77   for(size_t i = 0; i < s; ++i )
78     {
79     *r++ = *prgb++;
80     *g++ = *prgb++;
81     *b++ = *prgb++;
82     }
83   assert( (size_t)(prgb - rgb) == 3 * s );
84   return prgb - rgb;
85 }
86
87
88 } // end namespace gdcm
89
90 #endif //GDCMIMAGECHANGEPLANARCONFIGURATION_H
```

11.321 gdcmImageChangeTransferSyntax.h File Reference

```
#include "gdcmImageToImageFilter.h"
#include "gdcmTransferSyntax.h"
```

Include dependency graph for gdcmImageChangeTransferSyntax.h:



Classes

- class gdcm::ImageChangeTransferSyntax

  ImageChangeTransferSyntax class.

Namespaces

- namespace gdcm

11.322 gdcmImageChangeTransferSyntax.h

Go to the documentation of this file.

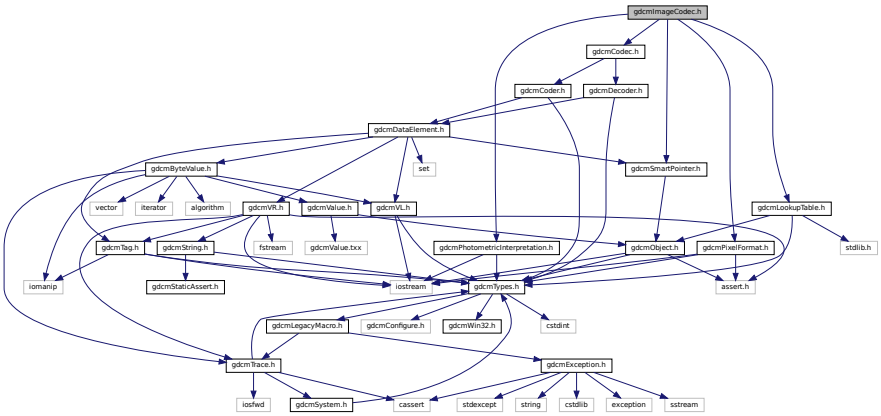
```
1 /*=========================================================================
2
3 Program:  GDCM (Grassroots DICOM). A DICOM library
4
5 Copyright (c) 2006-2011 Mathieu Malaterre
6 All rights reserved.
7 See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.
8
9 This software is distributed WITHOUT ANY WARRANTY; without even
10 the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
11 PURPOSE.  See the above copyright notice for more information.
12
13 =========================================================================*/
14 #ifndef GDCMIMAGECHANGETRANSFERSYNTAX_H
15 #define GDCMIMAGECHANGETRANSFERSYNTAX_H
16
17 #include "gdcmImageToImageFilter.h"
18 #include "gdcmTransferSyntax.h"
19
20 namespace gdcm
21 {
22
23 class DataElement;
24 class ImageCodec;
39 class GDCM_EXPORT ImageChangeTransferSyntax :  public ImageToImageFilter
```

```
40 {
41 public:
42
      ImageChangeTransferSyntax():TS(TransferSyntax::TS_END),Force(false),CompressIconImage(false),UserCodec(nullptr)
      {}
43   ~ImageChangeTransferSyntax() = default;
44
46   void SetTransferSyntax(const TransferSyntax &ts) { TS = ts; }
48   const TransferSyntax &GetTransferSyntax()const { return TS; }
49
51   bool Change();
52
55   void SetCompressIconImage(bool b) { CompressIconImage = b; }
56
61   void SetForce( bool f ) { Force = f; }
62
68   void SetUserCodec(ImageCodec *ic) { UserCodec = ic; }
69
70 protected:
71   bool TryJPEGCodec(const DataElement &pixelde, Bitmap const &input, Bitmap &output);
72   bool TryJPEG2000Codec(const DataElement &pixelde, Bitmap const &input, Bitmap &output);
73   bool TryJPEGLSCodec(const DataElement &pixelde, Bitmap const &input, Bitmap &output);
74   bool TryRAWCodec(const DataElement &pixelde, Bitmap const &input, Bitmap &output);
75   bool TryRLECodec(const DataElement &pixelde, Bitmap const &input, Bitmap &output);
76
77 private:
78   TransferSyntax TS;
79   bool Force;
80   bool CompressIconImage;
81
82   ImageCodec *UserCodec;
83 };
84
90 } // end namespace gdcm
91
92 #endif //GDCMIMAGECHANGETRANSFERSYNTAX_H
```

11.323 gdcmImageCodec.h File Reference

```
#include "gdcmCodec.h"
#include "gdcmPhotometricInterpretation.h"
#include "gdcmLookupTable.h"
#include "gdcmSmartPointer.h"
#include "gdcmPixelFormat.h"
```

Include dependency graph for gdcmImageCodec.h:



This graph shows which files directly or indirectly include this file:

Classes

- class gdcm::ImageCodec

  ImageCodec.

Namespaces

- namespace gdcm

11.324 gdcmImageCodec.h

Go to the documentation of this file.

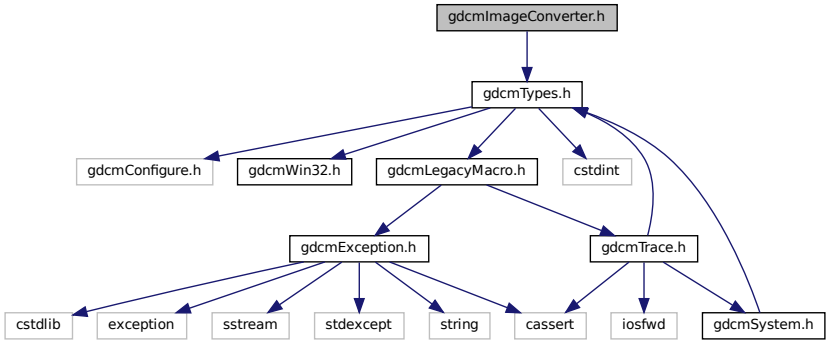
```
1 /*=========================================================================
2
3 Program:  GDCM (Grassroots DICOM). A DICOM library
4
5 Copyright (c) 2006-2011 Mathieu Malaterre
6 All rights reserved.
7 See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.
8
9 This software is distributed WITHOUT ANY WARRANTY; without even
10 the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
11 PURPOSE.  See the above copyright notice for more information.
12
13 =========================================================================*/
14 #ifndef GDCMIMAGECODEC_H
15 #define GDCMIMAGECODEC_H
16
17 #include "gdcmCodec.h"
18 #include "gdcmPhotometricInterpretation.h"
19 #include "gdcmLookupTable.h"
20 #include "gdcmSmartPointer.h"
21 #include "gdcmPixelFormat.h"
22
23 namespace gdcm
24 {
25
30 class GDCM_EXPORT ImageCodec :  public Codec
31 {
32   friend class ImageChangePhotometricInterpretation;
33 public:
34   ImageCodec();
35   ~ImageCodec() override;
36   bool CanCode(TransferSyntax const &)const override { return false; }
37   bool CanDecode(TransferSyntax const &)const override { return false; }
38   bool Decode(DataElement const &is_, DataElement &os) override;
39   bool IsLossy() const;
40   void SetLossyFlag(bool l);
41   bool GetLossyFlag() const;
42
43   virtual bool GetHeaderInfo(std::istream &is_, TransferSyntax &ts);
44
45   virtual ImageCodec * Clone() const = 0;
46
```

```
47 protected:
48   bool DecodeByStreams(std::istream &is_, std::ostream &os) override;
49   virtual bool IsValid(PhotometricInterpretation const &pi);
50 public:
51
52   unsigned int GetPlanarConfiguration()const
53 {
54     return PlanarConfiguration;
55     }
56   void SetPlanarConfiguration(unsigned int pc)
57     {
58     assert( pc == 0 || pc == 1 );
59     PlanarConfiguration = pc;
60     }
61
62   PixelFormat &GetPixelFormat()
63     {
64     return PF;
65     }
66   const PixelFormat &GetPixelFormat()const
67 {
68     return PF;
69     }
70   virtual void SetPixelFormat(PixelFormat const &pf)
71     {
72     PF = pf;
73     }
74   const PhotometricInterpretation &GetPhotometricInterpretation() const;
75   void SetPhotometricInterpretation(PhotometricInterpretation const &pi);
76
77   bool GetNeedByteSwap()const
78 {
79     return NeedByteSwap;
80     }
81   void SetNeedByteSwap(bool b)
82     {
83     NeedByteSwap = b;
84     }
85   void SetNeedOverlayCleanup(bool b)
86     {
87     NeedOverlayCleanup = b;
88     }
89   void SetLUT(LookupTable const &lut)
90     {
91     LUT = SmartPointer<LookupTable>( const_cast<LookupTable*>(&lut) );
92     }
93   const LookupTable &GetLUT()const
94 {
95     return *LUT;
96     }
97
98   void SetDimensions(const unsigned int d[3]);
99   void SetDimensions(const std::vector<unsigned int> & d);
100   const unsigned int *GetDimensions()const { return Dimensions; }
101   void SetNumberOfDimensions(unsigned int dim);
102   unsigned int GetNumberOfDimensions() const;
103
104   bool CleanupUnusedBits(char * data, size_t datalen);
105
106 protected:
107   // Streaming (write) API:
114   friend class FileChangeTransferSyntax;
115   virtual bool StartEncode( std::ostream & os );
116   virtual bool IsRowEncoder();
117   virtual bool IsFrameEncoder();
118   virtual bool AppendRowEncode( std::ostream & out, const char * data, size_t datalen );
119   virtual bool AppendFrameEncode( std::ostream & out, const char * data, size_t datalen );
120   virtual bool StopEncode( std::ostream & os);
121
122 protected:
123   bool RequestPlanarConfiguration;
124   bool RequestPaddedCompositePixelCode;
125 //private:
126   unsigned int PlanarConfiguration;
127   PhotometricInterpretation PI;
128   PixelFormat PF;
129   bool NeedByteSwap;
130   bool NeedOverlayCleanup;
131
132   typedef SmartPointer<LookupTable> LUTPtr;
133   LUTPtr LUT;
```

```
134   unsigned int Dimensions[3]; // FIXME
135   unsigned int NumberOfDimensions;
136   bool LossyFlag;
137
138   bool DoOverlayCleanup(std::istream &is_, std::ostream &os);
139   bool DoByteSwap(std::istream &is_, std::ostream &os);
140   bool DoYBR(std::istream &is_, std::ostream &os);
141   bool DoYBRFull422(std::istream &is_, std::ostream &os);
142   bool DoPlanarConfiguration(std::istream &is_, std::ostream &os);
143   bool DoSimpleCopy(std::istream &is_, std::ostream &os);
144   bool DoPaddedCompositePixelCode(std::istream &is_, std::ostream &os);
145   bool DoInvertMonochrome(std::istream &is_, std::ostream &os);
146
147   //template <typename T>
148   //bool DoInvertPlanarConfiguration(T *output, const T *input, uint32_t length);
149 };
150
151 } // end namespace gdcm
152
153 #endif //GDCMIMAGECODEC_H
```

11.325 gdcmImageConverter.h File Reference

`#include "gdcmTypes.h"`
Include dependency graph for gdcmImageConverter.h:



Classes

- class gdcm::ImageConverter

  Image Converter.

Namespaces

- namespace gdcm

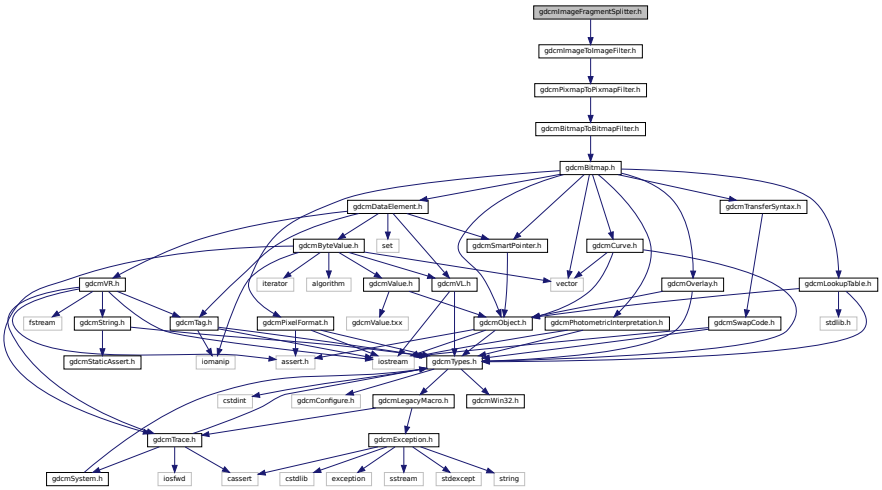
11.326 gdcmImageConverter.h

Go to the documentation of this file.

```
1 /*=========================================================================
2
3 Program:  GDCM (Grassroots DICOM). A DICOM library
4
5 Copyright (c) 2006-2011 Mathieu Malaterre
6 All rights reserved.
7 See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.
8
9 This software is distributed WITHOUT ANY WARRANTY; without even
10 the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
11 PURPOSE.  See the above copyright notice for more information.
12
13 =========================================================================*/
14
15 #ifndef GDCMIMAGECONVERTER_H
16 #define GDCMIMAGECONVERTER_H
17
18 #include "gdcmTypes.h"
19
20 namespace gdcm
21 {
22
23 class Image;
33 class GDCM_EXPORT ImageConverter
34 {
35 public:
36   ImageConverter();
37   ~ImageConverter();
38
39   void SetInput(Image const &input);
40   const Image& GetOuput() const;
41
42   void Convert();
43
44 private:
45   Image *Input;
46   Image *Output;
47 };
48
49 } // end namespace gdcm
50
51 #endif //GDCMIMAGECONVERTER_H
```

11.327 gdcmImageFragmentSplitter.h File Reference

`#include "gdcmImageToImageFilter.h"`
Include dependency graph for gdcmImageFragmentSplitter.h:



Classes

- class gdcm::ImageFragmentSplitter

  ImageFragmentSplitter class.

Namespaces

- namespace gdcm

11.328 gdcmImageFragmentSplitter.h

Go to the documentation of this file.

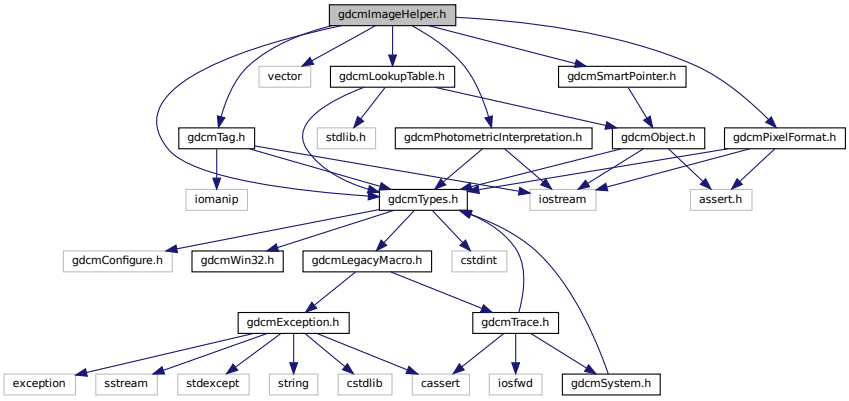
```
1 /*=========================================================================
2
3 Program:  GDCM (Grassroots DICOM). A DICOM library
4
5 Copyright (c) 2006-2011 Mathieu Malaterre
6 All rights reserved.
7 See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.
8
9 This software is distributed WITHOUT ANY WARRANTY; without even
10 the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
11 PURPOSE.  See the above copyright notice for more information.
12
13 =========================================================================*/
14 #ifndef GDCMIMAGEFRAGMENTSPLITTER_H
15 #define GDCMIMAGEFRAGMENTSPLITTER_H
16
17 #include "gdcmImageToImageFilter.h"
18
```

```
19 namespace gdcm
20 {
21
22 class DataElement;
27 class GDCM_EXPORT ImageFragmentSplitter :  public ImageToImageFilter
28 {
29 public:
30   ImageFragmentSplitter():FragmentSizeMax(0),Force(false) {}
31   ~ImageFragmentSplitter() = default;
32
34   bool Split();
35
37   void SetFragmentSizeMax(unsigned int fragsize);
38   unsigned int GetFragmentSizeMax()const { return FragmentSizeMax; }
39
42   void SetForce( bool f ) { Force = f; }
43
44 protected:
45
46 private:
47   unsigned int FragmentSizeMax;
48   bool Force;
49 };
50
51 } // end namespace gdcm
52
53 #endif //GDCMIMAGEFRAGMENTSPLITTER_H
```

11.329 gdcmImageHelper.h File Reference

```
#include "gdcmTypes.h"
#include "gdcmTag.h"
#include <vector>
#include "gdcmPixelFormat.h"
#include "gdcmPhotometricInterpretation.h"
#include "gdcmSmartPointer.h"
#include "gdcmLookupTable.h"
```

Include dependency graph for gdcmImageHelper.h:



Classes

- class gdcm::ImageHelper

  ImageHelper (internal class, not intended for user level)
- struct gdcm::RealWorldValueMappingContent

Namespaces

- namespace gdcm

11.330 gdcmImageHelper.h

Go to the documentation of this file.

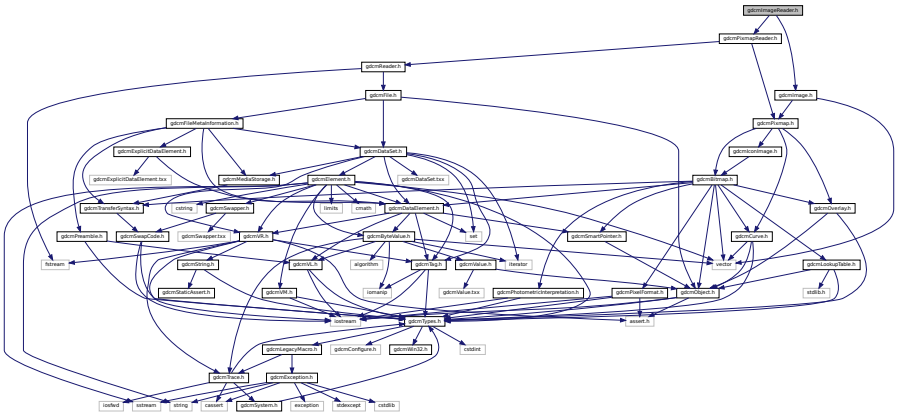
```
1 /*=========================================================================
2
3 Program:  GDCM (Grassroots DICOM). A DICOM library
4
5 Copyright (c) 2006-2011 Mathieu Malaterre
6 All rights reserved.
7 See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.
8
9 This software is distributed WITHOUT ANY WARRANTY; without even
10 the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
11 PURPOSE.  See the above copyright notice for more information.
12
13 =========================================================================*/
14 #ifndef GDCMIMAGEHELPER_H
15 #define GDCMIMAGEHELPER_H
16
17 #include "gdcmTypes.h"
18 #include "gdcmTag.h"
19 #include <vector>
20 #include "gdcmPixelFormat.h"
21 #include "gdcmPhotometricInterpretation.h"
22 #include "gdcmSmartPointer.h"
23 #include "gdcmLookupTable.h"
24
25 namespace gdcm
26 {
27
28 class MediaStorage;
29 class DataSet;
30 class File;
31 class Image;
32 class Pixmap;
33 class ByteValue;
34
35 // minimal struct:
36 struct RealWorldValueMappingContent {
37   double RealWorldValueIntercept;
38   double RealWorldValueSlope;
39   // http://dicom.nema.org/MEDICAL/DICOM/2014c/output/chtml/part16/sect_CID_7181.html
40   std::string CodeValue;
41   std::string CodeMeaning;
42 };
43
59 class GDCM_EXPORT ImageHelper
60 {
61 public:
68   static void SetForceRescaleInterceptSlope(bool);
69   static bool GetForceRescaleInterceptSlope();
70
76   static void SetPMSRescaleInterceptSlope(bool);
77   static bool GetPMSRescaleInterceptSlope();
78
85   static void SetForcePixelSpacing(bool);
86   static bool GetForcePixelSpacing();
87
91   static std::vector<unsigned int> GetDimensionsValue(const File& f);
92   static void SetDimensionsValue(File& f, const Pixmap & img);
93
96   static PixelFormat GetPixelFormatValue(const File& f);
97
102   static std::vector<double> GetRescaleInterceptSlopeValue(File const & f);
103   static void SetRescaleInterceptSlopeValue(File & f, const Image & img);
104
105   // read only for now
106   static bool GetRealWorldValueMappingContent(File const & f, RealWorldValueMappingContent & rwvmc);
107
109   static std::vector<double> GetOriginValue(File const & f);
```

```
110   static void SetOriginValue(DataSet & ds, const Image & img);
111
114   static std::vector<double> GetDirectionCosinesValue(File const & f);
120   // FIXME: There is a major issue for image with multiple IOP (eg.  Enhanced * Image Storage).
121   static void SetDirectionCosinesValue(DataSet & ds, const std::vector<double> & dircos);
122
124   static std::vector<double> GetSpacingValue(File const & f);
125   static void SetSpacingValue(DataSet & ds, const std::vector<double> & spacing);
126
128   static bool ComputeSpacingFromImagePositionPatient(const std::vector<double> &imageposition,
      std::vector<double> & spacing);
129
130   static bool GetDirectionCosinesFromDataSet(DataSet const & ds, std::vector<double> & dircos);
131
132   //functions to get more information from a file
133   //useful for the stream image reader, which fills in necessary image information
134   //distinctly from the reader-style data input
135   static PhotometricInterpretation GetPhotometricInterpretationValue(File const& f);
136   //returns the configuration of colors in a plane, either RGB RGB RGB or RRR GGG BBB
137   static unsigned int GetPlanarConfigurationValue(const File& f);
138
140   static SmartPointer<LookupTable> GetLUT(File const& f);
141
142   // Moved from PixampReader to here.  Generally used for photometric interpretation.
143   static const ByteValue* GetPointerFromElement(Tag const &tag, File const& f);
144
146   static MediaStorage ComputeMediaStorageFromModality(const char *modality,
147     unsigned int dimension = 2, PixelFormat const & pf = PixelFormat(),
148     PhotometricInterpretation const & pi = PhotometricInterpretation(),
149     double rescaleintercept = 0, double rescaleslope = 1 );
150
151 protected:
152   static Tag GetSpacingTagFromMediaStorage(MediaStorage const &ms);
153   static Tag GetZSpacingTagFromMediaStorage(MediaStorage const &ms);
154
155 private:
156   static bool ForceRescaleInterceptSlope;
157   static bool PMSRescaleInterceptSlope;
158   static bool ForcePixelSpacing;
159 };
160
161 } // end namespace gdcm
162
163 #endif // GDCMIMAGEHELPER_H
```

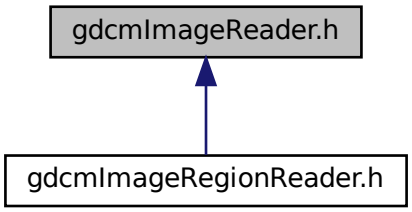
11.331 gdcmImageReader.h File Reference

```
#include "gdcmPixmapReader.h"
#include "gdcmImage.h"
```

Include dependency graph for gdcmImageReader.h:



This graph shows which files directly or indirectly include this file:



Classes

- class gdcm::ImageReader

  ImageReader.

Namespaces

- namespace gdcm

11.332 gdcmImageReader.h

Go to the documentation of this file.

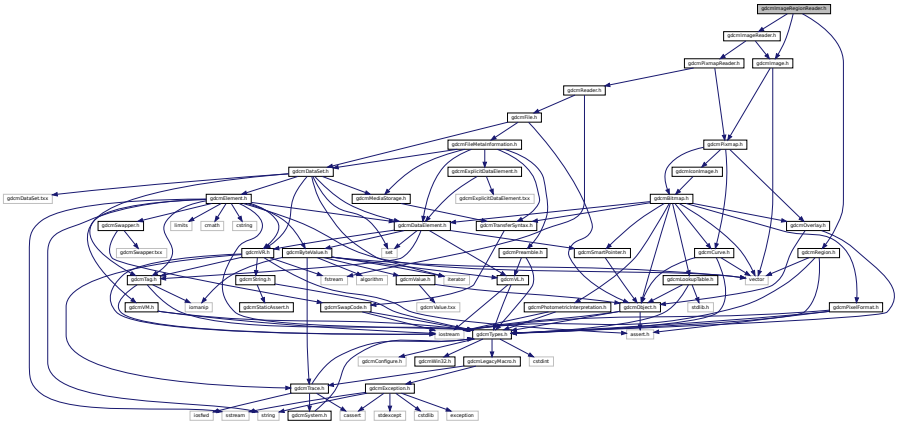
```
1 /*=========================================================================
2
3 Program:  GDCM (Grassroots DICOM). A DICOM library
4
5 Copyright (c) 2006-2011 Mathieu Malaterre
6 All rights reserved.
7 See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.
8
9 This software is distributed WITHOUT ANY WARRANTY; without even
10 the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
11 PURPOSE.  See the above copyright notice for more information.
12
13 =========================================================================*/
14 #ifndef GDCMIMAGEREADER_H
15 #define GDCMIMAGEREADER_H
16
17 #include "gdcmPixmapReader.h"
18 #include "gdcmImage.h"
19
20 namespace gdcm
21 {
22
23 class MediaStorage;
33 class GDCM_EXPORT ImageReader :  public PixmapReader
34 {
35 public:
36   ImageReader();
37   ~ImageReader() override;//needs to be virtual to ensure lack of memory leaks
38
```

```
42
43   bool Read() override;
44
45   // Following methods are valid only after a call to 'Read'
46
48   const Image& GetImage() const;
49   Image& GetImage();
50   //void SetImage(Image const &img);
51
52 protected:
53   bool ReadImage(MediaStorage const &ms) override;
54   bool ReadACRNEMAImage() override;
55 };
56
57 } // end namespace gdcm
58
59 #endif //GDCMIMAGEREADER_H
```

11.333 gdcmImageRegionReader.h File Reference

```
#include "gdcmImageReader.h"
#include "gdcmImage.h"
#include "gdcmRegion.h"
```

Include dependency graph for gdcmImageRegionReader.h:



Classes

- class gdcm::ImageRegionReader

  ImageRegionReader.

Namespaces

- namespace gdcm

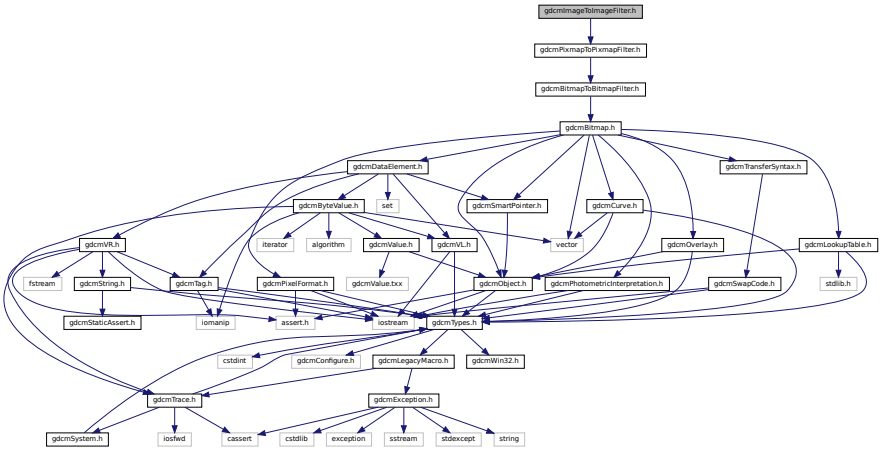
11.334 gdcmImageRegionReader.h

Go to the documentation of this file.

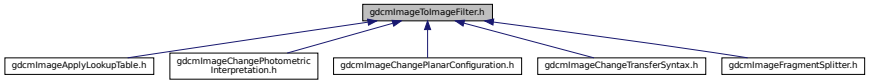
```
1 /*=========================================================================
2
3 Program:  GDCM (Grassroots DICOM). A DICOM library
4
5 Copyright (c) 2006-2011 Mathieu Malaterre
6 All rights reserved.
7 See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.
8
9 This software is distributed WITHOUT ANY WARRANTY; without even
10 the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
11 PURPOSE.  See the above copyright notice for more information.
12
13 =========================================================================*/
14 #ifndef GDCMIMAGEEXTENTREADER_H
15 #define GDCMIMAGEEXTENTREADER_H
16
17 #include "gdcmImageReader.h"
18 #include "gdcmImage.h"
19 #include "gdcmRegion.h"
20
21 namespace gdcm
22 {
23
24 class ImageRegionReaderInternals;
34 class GDCM_EXPORT ImageRegionReader :  public ImageReader
35 {
36 public:
37   ImageRegionReader();
38   ~ImageRegionReader() override;
39
41   void SetRegion(Region const & region);
42   Region const &GetRegion() const;
43
47   size_t ComputeBufferLength() const;
48
51   bool ReadInformation();
52
57   bool ReadIntoBuffer(char *inreadbuffer, size_t buflen);
58
59 protected:
61   bool Read() override;
62
63 private:
64   BoxRegion ComputeBoundingBox();
65   bool ReadRAWIntoBuffer(char *buffer, size_t buflen);
66   bool ReadRLEIntoBuffer(char *buffer, size_t buflen);
67   bool ReadJPEG2000IntoBuffer(char *buffer, size_t buflen);
68   bool ReadJPEGIntoBuffer(char *buffer, size_t buflen);
69   bool ReadJPEGLSIntoBuffer(char *buffer, size_t buflen);
70   ImageRegionReaderInternals *Internals;
71 };
72
73 } // end namespace gdcm
74
75 #endif //GDCMIMAGEEXTENTREADER_H
```

11.335 gdcmImageToImageFilter.h File Reference

`#include "gdcmPixmapToPixmapFilter.h"`
Include dependency graph for gdcmImageToImageFilter.h:



This graph shows which files directly or indirectly include this file:



Classes

- class gdcm::ImageToImageFilter
  ImageToImageFilter class.

Namespaces

- namespace gdcm

11.336 gdcmImageToImageFilter.h

Go to the documentation of this file.

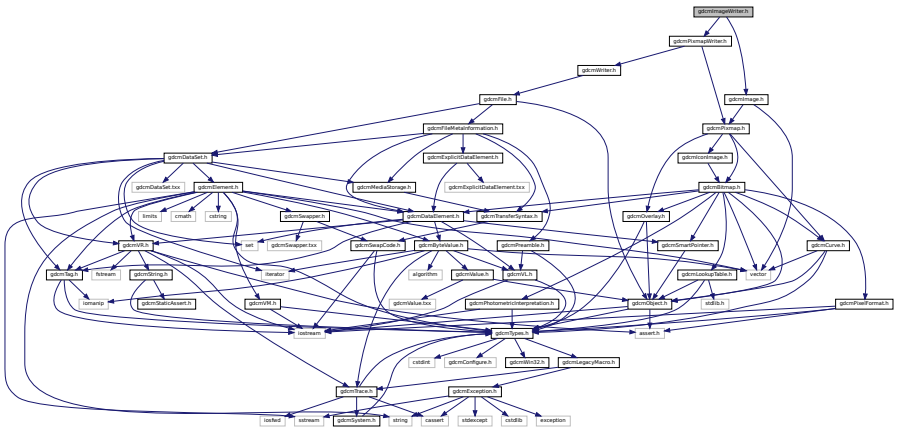
```
1 /*=========================================================================
2
3 Program:  GDCM (Grassroots DICOM). A DICOM library
4
5 Copyright (c) 2006-2011 Mathieu Malaterre
```

```
6 All rights reserved.
7 See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.
8
9 This software is distributed WITHOUT ANY WARRANTY; without even
10 the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
11 PURPOSE.  See the above copyright notice for more information.
12
13 =========================================================================*/
14 #ifndef GDCMIMAGETOIMAGEFILTER_H
15 #define GDCMIMAGETOIMAGEFILTER_H
16
17 #include "gdcmPixmapToPixmapFilter.h"
18
19 namespace gdcm
20 {
21
22 class Image;
27 class GDCM_EXPORT ImageToImageFilter :  public PixmapToPixmapFilter
28 {
29 public:
30   ImageToImageFilter();
31   ~ImageToImageFilter() = default;
32
33   Image &GetInput();
34
35   // NOTE: covariant return-type to preserve backward compatible API
37   const Image &GetOutput() const;
38
39 protected:
40 };
41
42 } // end namespace gdcm
43
44 #endif //GDCMIMAGETOIMAGEFILTER_H
```

11.337 gdcmImageWriter.h File Reference

```
#include "gdcmPixmapWriter.h"
#include "gdcmImage.h"
```

Include dependency graph for gdcmImageWriter.h:



Classes

- class gdcm::ImageWriter

  ImageWriter.

Namespaces

- namespace gdcm

11.338 gdcmImageWriter.h

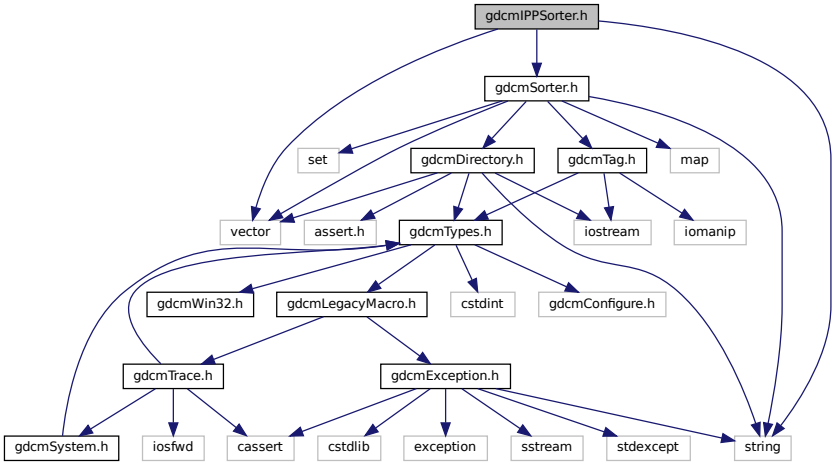
Go to the documentation of this file.

```
1 /*=========================================================================
2
3 Program:  GDCM (Grassroots DICOM). A DICOM library
4
5 Copyright (c) 2006-2011 Mathieu Malaterre
6 All rights reserved.
7 See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.
8
9 This software is distributed WITHOUT ANY WARRANTY; without even
10 the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
11 PURPOSE.  See the above copyright notice for more information.
12
13 =========================================================================*/
14 #ifndef GDCMIMAGEWRITER_H
15 #define GDCMIMAGEWRITER_H
16
17 #include "gdcmPixmapWriter.h"
18 #include "gdcmImage.h"
19
20 namespace gdcm
21 {
22
23 class Image;
32 class GDCM_EXPORT ImageWriter :  public PixmapWriter
33 {
34 public:
35   ImageWriter();
36   ~ImageWriter() override;
37
41   const Image& GetImage()const override { return dynamic_cast<const Image&>(*PixelData); }
42   Image& GetImage()override { return dynamic_cast<Image&>(*PixelData); } // FIXME
43   //void SetImage(Image const &img);
44
46   bool Write() override; // Execute()
47
50   MediaStorage ComputeTargetMediaStorage();
51 protected:
52
53 private:
54 };
55
56 } // end namespace gdcm
57
58 #endif //GDCMIMAGEWRITER_H
```

11.339 gdcmIPPSorter.h File Reference

```
#include "gdcmSorter.h"
#include <vector>
#include <string>
```

Include dependency graph for gdcmIPPSorter.h:



Classes

- class gdcm::IPPSorter

  IPPSorter.

Namespaces

- namespace gdcm

11.340 gdcmIPPSorter.h

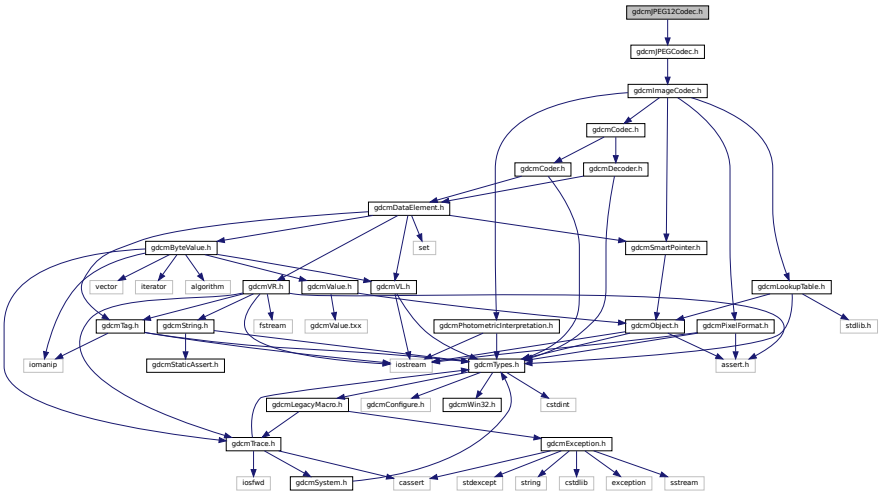
Go to the documentation of this file.

```
1 /*=========================================================================
2
3 Program:  GDCM (Grassroots DICOM). A DICOM library
4
5 Copyright (c) 2006-2011 Mathieu Malaterre
6 All rights reserved.
7 See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.
8
9 This software is distributed WITHOUT ANY WARRANTY; without even
10 the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
11 PURPOSE.  See the above copyright notice for more information.
12
13 =========================================================================*/
14 #ifndef GDCMIPPSORTER_H
15 #define GDCMIPPSORTER_H
16
17 #include "gdcmSorter.h"
18
19 #include <vector>
20 #include <string>
21
22 namespace gdcm
23 {
```

```
43 class GDCM_EXPORT IPPSorter :  public Sorter
44 {
45 public:
46   IPPSorter();
47
48   // FIXME: I do not like public virtual function...
55   bool Sort(std::vector<std::string> const & filenames) override;
56
64   void SetComputeZSpacing(bool b) { ComputeZSpacing = b; }
68   void SetZSpacingTolerance(double tol) { ZTolerance = tol; }
69   double GetZSpacingTolerance()const { return ZTolerance; }
70
80   void SetDirectionCosinesTolerance(double tol) { DirCosTolerance = tol; }
81   double GetDirectionCosinesTolerance()const { return DirCosTolerance; }
82
86   void SetDropDuplicatePositions(bool b) { DropDuplicatePositions = b; }
87
94   double GetZSpacing()const { return ZSpacing; }
95
96 protected:
97   bool ComputeZSpacing;
98   bool DropDuplicatePositions;
99   double ZSpacing;
100   double ZTolerance;
101   double DirCosTolerance;
102
103 private:
104   GDCM_LEGACY(bool ComputeSpacing(std::vector<std::string> const & filenames))
105 };
106
107
108 } // end namespace gdcm
109
110 #endif //GDCMIPPSORTER_H
```

11.341 gdcmJPEG12Codec.h File Reference

`#include "gdcmJPEGCodec.h"`
Include dependency graph for gdcmJPEG12Codec.h:



Classes

- class gdcm::JPEG12Codec

  Class to do JPEG 12bits (lossy & lossless)

Namespaces

- namespace gdcm

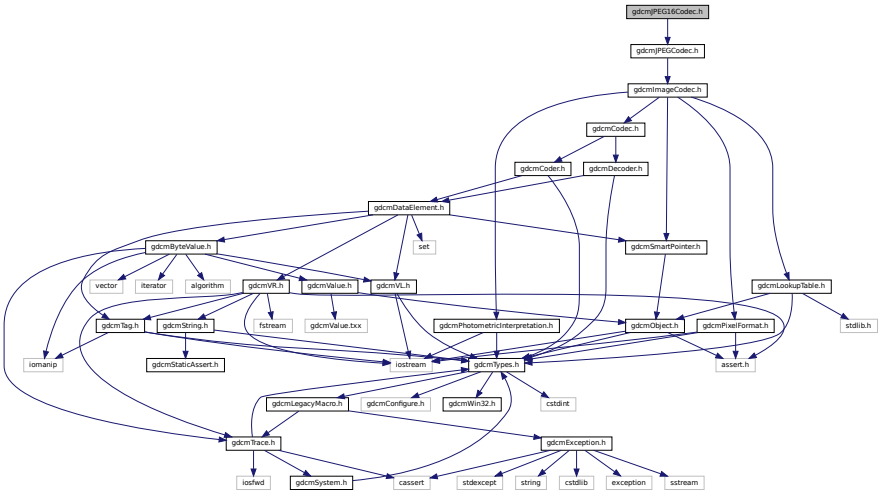
11.342 gdcmJPEG12Codec.h

Go to the documentation of this file.

```
1 /*=========================================================================
2
3 Program:  GDCM (Grassroots DICOM). A DICOM library
4
5 Copyright (c) 2006-2011 Mathieu Malaterre
6 All rights reserved.
7 See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.
8
9 This software is distributed WITHOUT ANY WARRANTY; without even
10 the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
11 PURPOSE.  See the above copyright notice for more information.
12
13 =========================================================================*/
14 #ifndef GDCMJPEG12CODEC_H
15 #define GDCMJPEG12CODEC_H
16
17 #include "gdcmJPEGCodec.h"
18
19 namespace gdcm
20 {
21
22 class JPEGInternals_12BIT;
23 class ByteValue;
28 class JPEG12Codec :  public JPEGCodec
29 {
30 public:
31   JPEG12Codec();
32   ~JPEG12Codec() override;
33
34   bool DecodeByStreams(std::istream &is, std::ostream &os) override;
35   bool InternalCode(const char *input, unsigned long len, std::ostream &os) override;
36
37   bool GetHeaderInfo(std::istream &is, TransferSyntax &ts) override;
38
39 protected:
40   bool IsStateSuspension() const override;
41   bool EncodeBuffer(std::ostream &os, const char *data, size_t datalen) override;
42
43 private:
44   JPEGInternals_12BIT *Internals;
45 };
46
47 } // end namespace gdcm
48
49 #endif //GDCMJPEG12CODEC_H
```

11.343 gdcmJPEG16Codec.h File Reference

`#include "gdcmJPEGCodec.h"`
Include dependency graph for gdcmJPEG16Codec.h:



Classes

- class gdcm::JPEG16Codec

  Class to do JPEG 16bits (lossless)

Namespaces

- namespace gdcm

11.344 gdcmJPEG16Codec.h

Go to the documentation of this file.

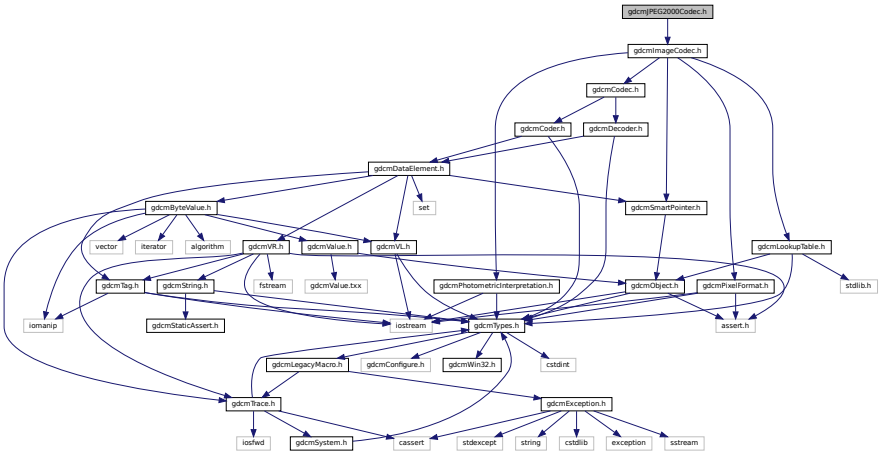
```
1 /*=========================================================================
2
3 Program:  GDCM (Grassroots DICOM). A DICOM library
4
5 Copyright (c) 2006-2011 Mathieu Malaterre
6 All rights reserved.
7 See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.
8
9 This software is distributed WITHOUT ANY WARRANTY; without even
10 the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
11 PURPOSE.  See the above copyright notice for more information.
12
13 =========================================================================*/
14 #ifndef GDCMJPEG16CODEC_H
15 #define GDCMJPEG16CODEC_H
16
17 #include "gdcmJPEGCodec.h"
18
```

```
19 namespace gdcm
20 {
21
22 class JPEGInternals_16BIT;
23 class ByteValue;
28 class JPEG16Codec :  public JPEGCodec
29 {
30 public:
31   JPEG16Codec();
32   ~JPEG16Codec() override;
33
34   bool DecodeByStreams(std::istream &is, std::ostream &os) override;
35   bool InternalCode(const char *input, unsigned long len, std::ostream &os) override;
36
37   bool GetHeaderInfo(std::istream &is, TransferSyntax &ts) override;
38
39 protected:
40   bool IsStateSuspension() const override;
41   bool EncodeBuffer(std::ostream &os, const char *data, size_t datalen) override;
42
43 private:
44   JPEGInternals_16BIT *Internals;
45 };
46
47 } // end namespace gdcm
48
49 #endif //GDCMJPEG16CODEC_H
```

11.345 gdcmJPEG2000Codec.h File Reference

`#include "gdcmImageCodec.h"`

Include dependency graph for gdcmJPEG2000Codec.h:



Classes

- class gdcm::JPEG2000Codec

  Class to do JPEG 2000.

Namespaces

- namespace gdcm

11.346 gdcmJPEG2000Codec.h

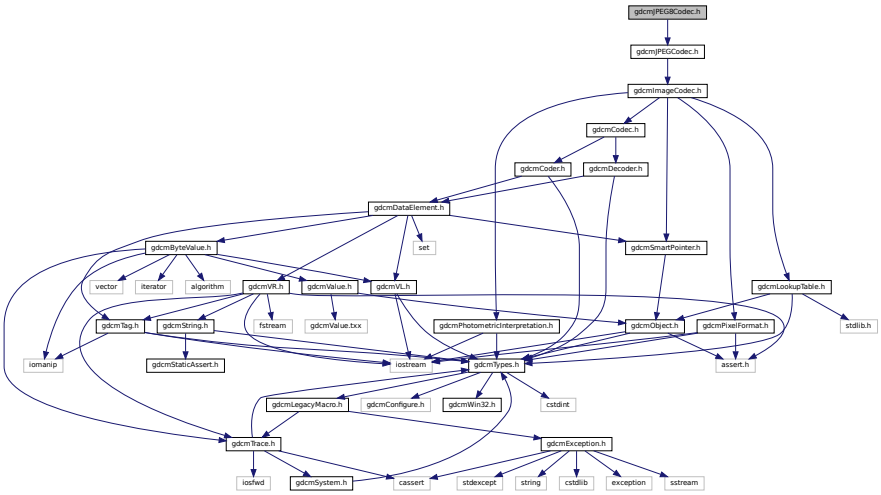
Go to the documentation of this file.

```
1 /*=========================================================================
2
3 Program:  GDCM (Grassroots DICOM). A DICOM library
4
5 Copyright (c) 2006-2011 Mathieu Malaterre
6 All rights reserved.
7 See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.
8
9 This software is distributed WITHOUT ANY WARRANTY; without even
10 the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
11 PURPOSE.  See the above copyright notice for more information.
12
13 =========================================================================*/
14 #ifndef GDCMJPEG2000CODEC_H
15 #define GDCMJPEG2000CODEC_H
16
17 #include "gdcmImageCodec.h"
18
19 namespace gdcm
20 {
21
22 class JPEG2000Internals;
30 class GDCM_EXPORT JPEG2000Codec :  public ImageCodec
31 {
32 friend class ImageRegionReader;
33   friend class Bitmap;
34 public:
35   JPEG2000Codec();
36   ~JPEG2000Codec() override;
37
38   bool CanDecode(TransferSyntax const &ts) const override;
39   bool CanCode(TransferSyntax const &ts) const override;
40
41   bool Decode(DataElement const &is, DataElement &os) override;
42   bool Code(DataElement const &in, DataElement &out) override;
43
44   bool GetHeaderInfo(std::istream &is, TransferSyntax &ts) override;
45   ImageCodec * Clone() const override;
46
47   // JPEG-2000 / OpenJPEG specific way of encoding lossy-ness
48   // ref:  http://www.openjpeg.org/index.php?menu=doc#encoder
49   void SetRate(unsigned int idx, double rate);
50   double GetRate(unsigned int idx = 0) const;
51
52   void SetQuality(unsigned int idx, double q);
53   double GetQuality(unsigned int idx = 0) const;
54
55   void SetTileSize(unsigned int tx, unsigned int ty);
56
57   void SetNumberOfResolutions(unsigned int nres);
58
61   void SetNumberOfThreadsForDecompression(int nThreads);
62
63   void SetReversible(bool res);
64   void SetMCT(unsigned int mct);
65
66 protected:
67   bool DecodeExtent(
68     char *buffer,
69     unsigned int xmin, unsigned int xmax,
70     unsigned int ymin, unsigned int ymax,
71     unsigned int zmin, unsigned int zmax,
72     std::istream & is
73   );
74
75   bool DecodeByStreams(std::istream &is, std::ostream &os) override;
76
77   bool StartEncode( std::ostream & ) override;
78   bool IsRowEncoder() override;
79   bool IsFrameEncoder() override;
80   bool AppendRowEncode( std::ostream & out, const char * data, size_t datalen ) override;
81   bool AppendFrameEncode( std::ostream & out, const char * data, size_t datalen ) override;
82   bool StopEncode( std::ostream & ) override;
83
84 private:
85   std::pair<char *, size_t> DecodeByStreamsCommon(char *dummy_buffer, size_t buf_size);
```

```
86   bool CodeFrameIntoBuffer(char * outdata, size_t outlen, size_t & complen, const char * indata, size_t inlen
      );
87   bool GetHeaderInfo(const char * dummy_buffer, size_t len, TransferSyntax &ts);
88   JPEG2000Internals *Internals;
89 };
90
91 } // end namespace gdcm
92
93 #endif //GDCMJPEG2000CODEC_H
```

11.347 gdcmJPEG8Codec.h File Reference

`#include "gdcmJPEGCodec.h"`
Include dependency graph for gdcmJPEG8Codec.h:



Classes

- class gdcm::JPEG8Codec

  Class to do JPEG 8bits (lossy & lossless)

Namespaces

- namespace gdcm

11.348 gdcmJPEG8Codec.h

Go to the documentation of this file.

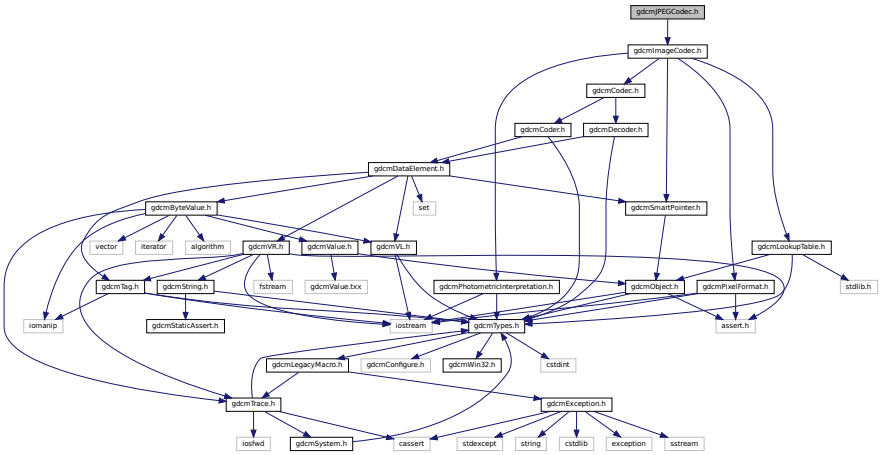
```
1 /*=========================================================================
2
3 Program:  GDCM (Grassroots DICOM). A DICOM library
4
```

```
5 Copyright (c) 2006-2011 Mathieu Malaterre
6 All rights reserved.
7 See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.
8
9 This software is distributed WITHOUT ANY WARRANTY; without even
10 the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
11 PURPOSE.  See the above copyright notice for more information.
12
13 =========================================================================*/
14 #ifndef GDCMJPEG8CODEC_H
15 #define GDCMJPEG8CODEC_H
16
17 #include "gdcmJPEGCodec.h"
18
19 namespace gdcm
20 {
21
22 class JPEGInternals_8BIT;
23 class ByteValue;
28 class JPEG8Codec :  public JPEGCodec
29 {
30 public:
31   JPEG8Codec();
32   ~JPEG8Codec() override;
33
34   bool DecodeByStreams(std::istream &is, std::ostream &os) override;
35   bool InternalCode(const char *input, unsigned long len, std::ostream &os) override;
36
37   bool GetHeaderInfo(std::istream &is, TransferSyntax &ts) override;
38
39 protected:
40   bool IsStateSuspension() const override;
41   bool EncodeBuffer(std::ostream &os, const char *data, size_t datalen) override;
42
43 private:
44   JPEGInternals_8BIT *Internals;
45 };
46
47 } // end namespace gdcm
48
49 #endif //GDCMJPEG8CODEC_H
```

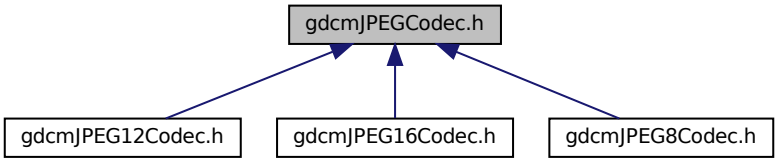
11.349 gdcmJPEGCodec.h File Reference

`#include "gdcmImageCodec.h"`

Include dependency graph for gdcmJPEGCodec.h:



This graph shows which files directly or indirectly include this file:



Classes

- class gdcm::JPEGCodec

  JPEG codec.

Namespaces

- namespace gdcm

11.350 gdcmJPEGCodec.h

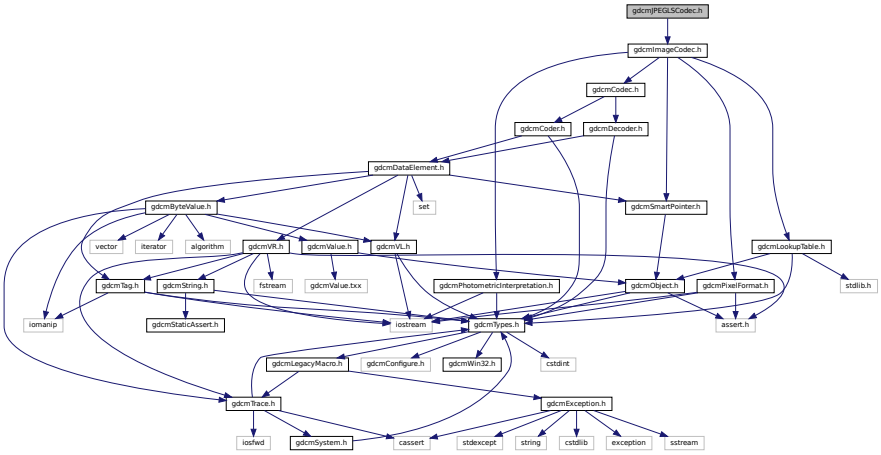
Go to the documentation of this file.

```
1 /*=========================================================================
2
3 Program:  GDCM (Grassroots DICOM). A DICOM library
4
5 Copyright (c) 2006-2011 Mathieu Malaterre
6 All rights reserved.
7 See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.
8
9 This software is distributed WITHOUT ANY WARRANTY; without even
10 the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
11 PURPOSE.  See the above copyright notice for more information.
12
13 =========================================================================*/
14 #ifndef GDCMJPEGCODEC_H
15 #define GDCMJPEGCODEC_H
16
17 #include "gdcmImageCodec.h"
18
19 namespace gdcm
20 {
21
22 class PixelFormat;
23 class TransferSyntax;
40 class GDCM_EXPORT JPEGCodec :  public ImageCodec
41 {
42 friend class ImageRegionReader;
43 public:
44   JPEGCodec();
45   ~JPEGCodec() override;
46   bool CanDecode(TransferSyntax const &ts) const override;
47   bool CanCode(TransferSyntax const &ts) const override;
48   bool Decode(DataElement const &is, DataElement &os) override;
49   void SetPixelFormat(PixelFormat const &pf) override;
50
```

```
52   void ComputeOffsetTable(bool b);
53
55   bool Code(DataElement const &in, DataElement &out) override;
56
57   bool GetHeaderInfo(std::istream &is, TransferSyntax &ts) override;
58   ImageCodec * Clone() const override;
59
60   //void SetReversible(bool res);
61
62   void SetQuality(double q);
63   double GetQuality() const;
64
65   void SetLossless(bool l);
66   bool GetLossless() const;
67
68   virtual bool EncodeBuffer( std::ostream & out,
69     const char *inbuffer, size_t inlen);
70
71 protected:
72   bool DecodeExtent(
73     char *buffer,
74     unsigned int xmin, unsigned int xmax,
75     unsigned int ymin, unsigned int ymax,
76     unsigned int zmin, unsigned int zmax,
77     std::istream & is
78   );
79
80   bool DecodeByStreams(std::istream &is, std::ostream &os) override;
81   bool IsValid(PhotometricInterpretation const &pi) override;
82
83   bool StartEncode( std::ostream & ) override;
84   bool IsRowEncoder() override;
85   bool IsFrameEncoder() override;
86   bool AppendRowEncode( std::ostream & out, const char * data, size_t datalen ) override;
87   bool AppendFrameEncode( std::ostream & out, const char * data, size_t datalen ) override;
88   bool StopEncode( std::ostream & ) override;
89
90 protected:
91   // Internal method called by SetPixelFormat
92   // Instantiate the right jpeg codec (8, 12 or 16)
93   void SetBitSample(int bit);
94
95   virtual bool IsStateSuspension() const;
96
97 protected:
98   int BitSample;
99   //bool Lossless;
100   int Quality;
101
102 private:
103   void SetupJPEGBitCodec(int bit);
104   JPEGCodec *Internal;
105 };
106
107 } // end namespace gdcm
108
109 #endif //GDCMJPEGCODEC_H
```

11.351 gdcmJPEGLSCodec.h File Reference

`#include "gdcmImageCodec.h"`
Include dependency graph for gdcmJPEGLSCodec.h:



Classes

- class gdcm::JPEGLSCodec

  JPEG-LS.

Namespaces

- namespace gdcm

11.352 gdcmJPEGLSCodec.h

Go to the documentation of this file.

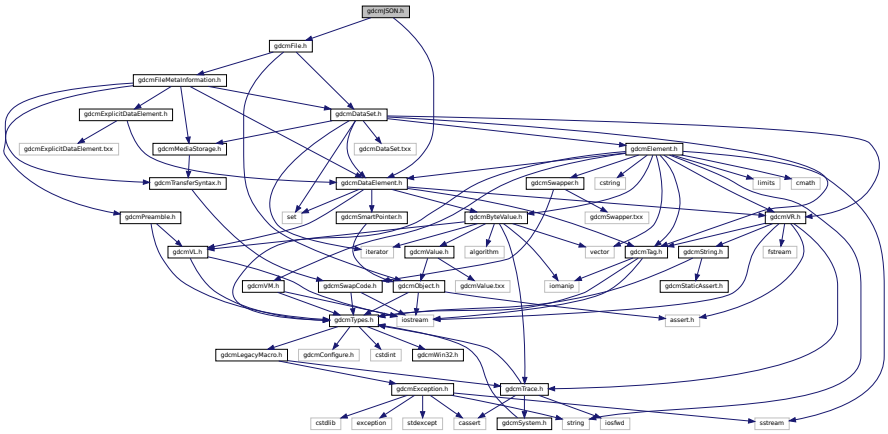
```
1 /*=========================================================================
2
3 Program:  GDCM (Grassroots DICOM). A DICOM library
4
5 Copyright (c) 2006-2011 Mathieu Malaterre
6 All rights reserved.
7 See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.
8
9 This software is distributed WITHOUT ANY WARRANTY; without even
10 the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
11 PURPOSE.  See the above copyright notice for more information.
12
13 =========================================================================*/
14 #ifndef GDCMJPEGLSCODEC_H
15 #define GDCMJPEGLSCODEC_H
16
17 #include "gdcmImageCodec.h"
18
19 namespace gdcm
20 {
```

```
21
22 class JPEGLSInternals;
30 class GDCM_EXPORT JPEGLSCodec :  public ImageCodec
31 {
32 friend class ImageRegionReader;
33 public:
34   JPEGLSCodec();
35   ~JPEGLSCodec() override;
36   bool CanDecode(TransferSyntax const &ts) const override;
37   bool CanCode(TransferSyntax const &ts) const override;
38
39   unsigned long GetBufferLength()const { return BufferLength; }
40   void SetBufferLength(unsigned long l) { BufferLength = l; }
41
42   bool Decode(DataElement const &is, DataElement &os) override;
43   bool Decode(DataElement const &in, char* outBuffer, size_t inBufferLength,
44               uint32_t inXMin, uint32_t inXMax, uint32_t inYMin,
45               uint32_t inYMax, uint32_t inZMin, uint32_t inZMax);
46   bool Code(DataElement const &in, DataElement &out) override;
47
48   bool GetHeaderInfo(std::istream &is, TransferSyntax &ts) override;
49   ImageCodec * Clone() const override;
50
51   void SetLossless(bool l);
52   bool GetLossless() const;
53
54 /*
55 * test.acr can look pretty bad, even with a lossy error of 2.  Explanation follows:
56 * I agree that the test image looks ugly.  In this particular case I can
57 * explain though.
58 *
59 * The image is 8 bit, but it does not use the full 8 bit dynamic range.  The
60 * black pixels have value 234 and the white 255.  If you set allowed lossy
61 * error to 2, you allow an error of about 10% of the actual dynamic range.
62 * That is of course very visible.
63 */
65   void SetLossyError(int error);
66
67 protected:
68   bool DecodeExtent(
69     char *buffer,
70     unsigned int xmin, unsigned int xmax,
71     unsigned int ymin, unsigned int ymax,
72     unsigned int zmin, unsigned int zmax,
73     std::istream & is
74   );
75
76   bool StartEncode( std::ostream & ) override;
77   bool IsRowEncoder() override;
78   bool IsFrameEncoder() override;
79   bool AppendRowEncode( std::ostream & out, const char * data, size_t datalen ) override;
80   bool AppendFrameEncode( std::ostream & out, const char * data, size_t datalen ) override;
81   bool StopEncode( std::ostream & ) override;
82
83 private:
84   bool DecodeByStreamsCommon(const char *buffer, size_t totalLen, std::vector<unsigned char> &rgbyteOut);
85   bool CodeFrameIntoBuffer(char * outdata, size_t outlen, size_t & complen, const char * indata, size_t inlen
      );
86
87   unsigned long BufferLength;
88   int LossyError;
89 };
90
91 } // end namespace gdcm
92
93 #endif //GDCMJPEGLSCODEC_H
```

11.353 gdcmJSON.h File Reference

```
#include "gdcmFile.h"
#include "gdcmDataElement.h"
```

Include dependency graph for gdcmJSON.h:



Classes

- class gdcm::JSON

Namespaces

- namespace gdcm

11.354 gdcmJSON.h

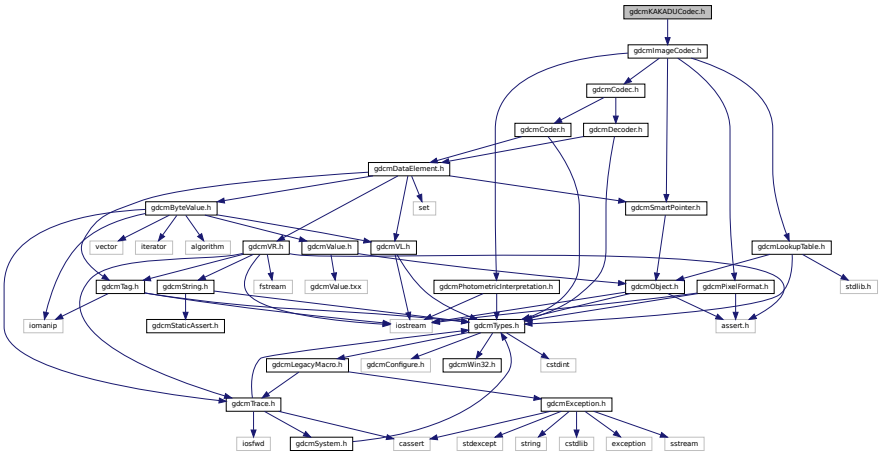
Go to the documentation of this file.

```
1 /*=========================================================================
2
3 Program:  GDCM (Grassroots DICOM). A DICOM library
4
5 Copyright (c) 2006-2011 Mathieu Malaterre
6 All rights reserved.
7 See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.
8
9 This software is distributed WITHOUT ANY WARRANTY; without even
10 the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
11 PURPOSE.  See the above copyright notice for more information.
12
13 =========================================================================*/
14 #ifndef GDCMJSON_H
15 #define GDCMJSON_H
16
17 /*
18 See Sup 166 (QIDO-RS)
19 http://www.dclunie.com/dicom-status/status.html#Supplement166
20 */
21
22 #include "gdcmFile.h"
23 #include "gdcmDataElement.h"
24
25 namespace gdcm
26 {
27
28 class JSONInternal;
```

```
29 class GDCM_EXPORT JSON
30 {
31 public:
32   JSON();
33   ~JSON();
34
35   bool GetPrettyPrint() const;
36   void SetPrettyPrint(bool onoff);
37   void PrettyPrintOn();
38   void PrettyPrintOff();
39
40   bool Code(DataSet const & in, std::ostream & os);
41   bool Decode(std::istream & is, DataSet & out);
42
43 private:
44   JSONInternal *Internals;
45 };
46
47 } // end namespace gdcm
48
49 #endif //GDCMXMLPRINTER_H
```

11.355 gdcmKAKADUCodec.h File Reference

`#include "gdcmImageCodec.h"`
Include dependency graph for gdcmKAKADUCodec.h:



Classes

- class gdcm::KAKADUCodec
  KAKADUCodec.

Namespaces

- namespace gdcm

11.356 gdcmKAKADUCodec.h

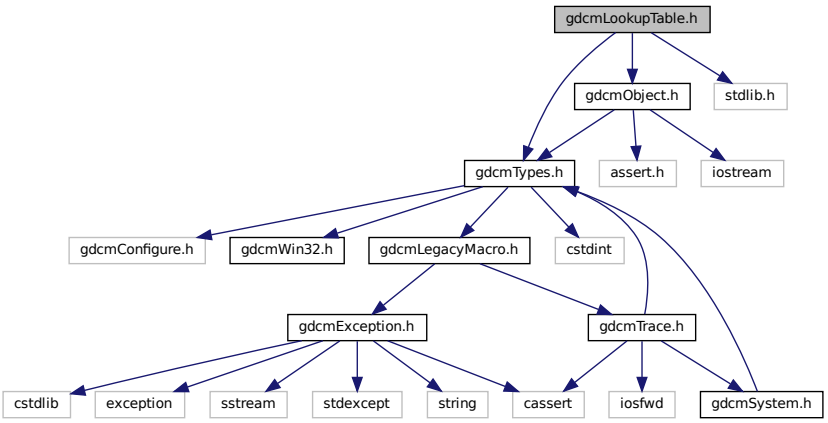
Go to the documentation of this file.

```
1 /*=========================================================================
2
3 Program:  GDCM (Grassroots DICOM). A DICOM library
4
5 Copyright (c) 2006-2011 Mathieu Malaterre
6 All rights reserved.
7 See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.
8
9 This software is distributed WITHOUT ANY WARRANTY; without even
10 the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
11 PURPOSE.  See the above copyright notice for more information.
12
13 =========================================================================*/
14 #ifndef GDCMKAKADUCODEC_H
15 #define GDCMKAKADUCODEC_H
16
17 #include "gdcmImageCodec.h"
18
19 namespace gdcm
20 {
21
25 class KAKADUCodec :  public ImageCodec
26 {
27 public:
28   KAKADUCodec();
29   ~KAKADUCodec() override;
30   bool CanDecode(TransferSyntax const &ts) const override;
31   bool CanCode(TransferSyntax const &ts) const override;
32
33   bool Decode(DataElement const &is, DataElement &os) override;
34   bool Code(DataElement const &in, DataElement &out) override;
35
36   ImageCodec * Clone() const override;
37 private:
38 };
39
40 } // end namespace gdcm
41
42 #endif //GDCMKAKADUCODEC_H
```

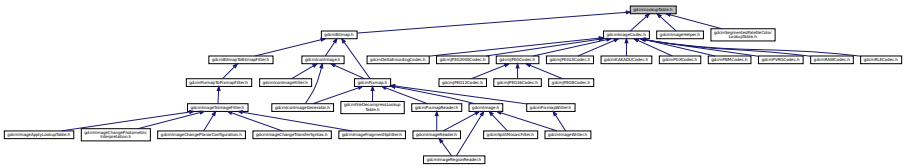
11.357 gdcmLookupTable.h File Reference

```
#include "gdcmTypes.h"
#include "gdcmObject.h"
#include <stdlib.h>
```

Include dependency graph for gdcmLookupTable.h:



This graph shows which files directly or indirectly include this file:



Classes

- class gdcm::LookupTable

  LookupTable class.

Namespaces

- namespace gdcm

11.358 gdcmLookupTable.h

Go to the documentation of this file.

```
1 /*=========================================================================
2
3 Program:  GDCM (Grassroots DICOM). A DICOM library
4
5 Copyright (c) 2006-2011 Mathieu Malaterre
6 All rights reserved.
7 See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.
8
```

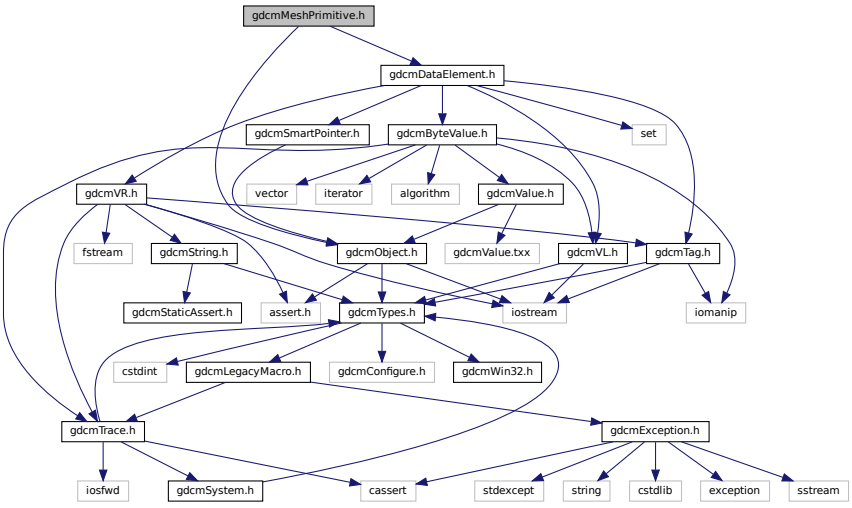
```
9 This software is distributed WITHOUT ANY WARRANTY; without even
10 the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
11 PURPOSE.  See the above copyright notice for more information.
12
13 =========================================================================*/
14
15 #ifndef GDCMLOOKUPTABLE_H
16 #define GDCMLOOKUPTABLE_H
17
18 #include "gdcmTypes.h"
19 #include "gdcmObject.h"
20 #include <stdlib.h>
21
22 namespace gdcm
23 {
24
25 class LookupTableInternal;
29 class GDCM_EXPORT LookupTable :  public Object
30 {
31 public:
32   typedef enum {
33     RED = 0,  // Keep RED == 0
34     GREEN,
35     BLUE,
36     GRAY,
37     UNKNOWN
38   } LookupTableType;
39
40   LookupTable();
41   ~LookupTable() override;
42   void Print(std::ostream &) const override;
43
45   void Allocate( unsigned short bitsample = 8 );
47   //TODO: check to see if length should be unsigned short, unsigned int, or whatever
48   void InitializeLUT(LookupTableType type, unsigned short length,
49     unsigned short subscript, unsigned short bitsize);
50   unsigned int GetLUTLength(LookupTableType type) const;
51   virtual void SetLUT(LookupTableType type, const unsigned char *array,
52     unsigned int length);
53   void GetLUT(LookupTableType type, unsigned char *array, unsigned int &length) const;
54   void GetLUTDescriptor(LookupTableType type, unsigned short &length,
55     unsigned short &subscript, unsigned short &bitsize) const;
56
58   void InitializeRedLUT(unsigned short length, unsigned short subscript,
59     unsigned short bitsize);
60   void SetRedLUT(const unsigned char *red, unsigned int length);
61   void InitializeGreenLUT(unsigned short length, unsigned short subscript,
62     unsigned short bitsize);
63   void SetGreenLUT(const unsigned char *green, unsigned int length);
64   void InitializeBlueLUT(unsigned short length, unsigned short subscript,
65     unsigned short bitsize);
66   void SetBlueLUT(const unsigned char *blue, unsigned int length);
67
69   void Clear();
70
72   void Decode(std::istream &is, std::ostream &os) const;
73
77   bool Decode(char *outputbuffer, size_t outlen, const char *inputbuffer, size_t inlen) const;
78
80   bool IsRGB8() const;
81
83   bool Decode8(char *outputbuffer, size_t outlen, const char *inputbuffer, size_t inlen) const;
84
85   LookupTable(LookupTable const &lut):Object(lut)
86     {
87     assert(0);
88     }
89
91   bool GetBufferAsRGBA(unsigned char *rgba) const;
92
94   const unsigned char *GetPointer() const;
95
97   bool WriteBufferAsRGBA(const unsigned char *rgba);
98
100    unsigned short GetBitSample()const { return BitSample; }
101
103    bool Initialized() const;
104
105 private:
107    void Encode(std::istream &is, std::ostream &os);
108
```

```
109 protected:
110   LookupTableInternal *Internal;
111   unsigned short BitSample; // refer to the pixel type (not the bit size of LUT)
112   bool IncompleteLUT:1;
113 };
114
115 } // end namespace gdcm
116
117 #endif //GDCMLOOKUPTABLE_H
```

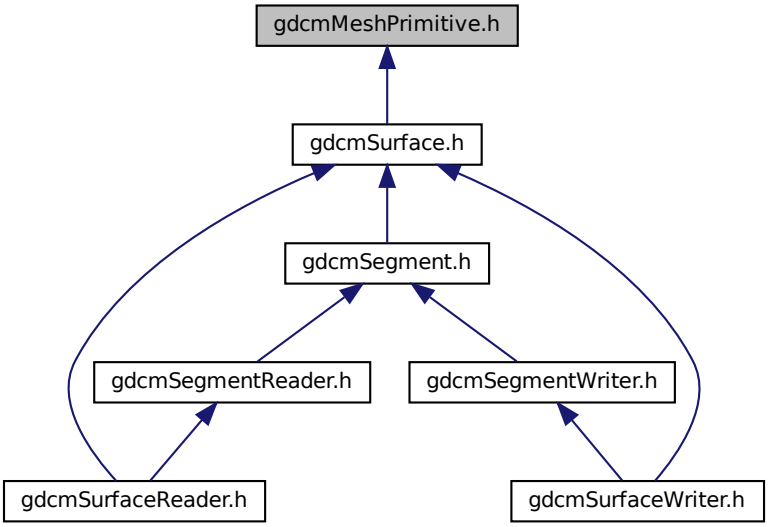
11.359 gdcmMeshPrimitive.h File Reference

```
#include <gdcmObject.h>
#include <gdcmDataElement.h>
```

Include dependency graph for gdcmMeshPrimitive.h:



This graph shows which files directly or indirectly include this file:



Classes

- class gdcm::MeshPrimitive

  This class defines surface mesh primitives.

Namespaces

- namespace gdcm

11.360 gdcmMeshPrimitive.h

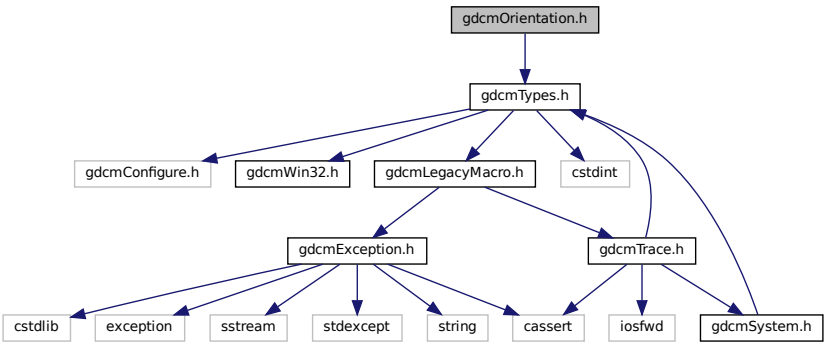
Go to the documentation of this file.

```
1 /*=========================================================================
2
3 Program:  GDCM (Grassroots DICOM). A DICOM library
4
5 Copyright (c) 2006-2011 Mathieu Malaterre
6 All rights reserved.
7 See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.
8
9 This software is distributed WITHOUT ANY WARRANTY; without even
10 the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
11 PURPOSE.  See the above copyright notice for more information.
12
13 =========================================================================*/
14
15 #ifndef GDCMMESHPRIMITIVE_H
16 #define GDCMMESHPRIMITIVE_H
17
```

```
18 #include <gdcmObject.h>
19 #include <gdcmDataElement.h>
20
21 namespace gdcm
22 {
23
30 class GDCM_EXPORT MeshPrimitive :  public Object
31 {
32 public:
33
34   typedef std::vector< DataElement > PrimitivesData;
35
41     typedef enum {
42         VERTEX = 0,
43         EDGE,
44         TRIANGLE,
45         TRIANGLE_STRIP,
46         TRIANGLE_FAN,
47         LINE,
48         FACET,
49         MPType_END
50         } MPType;
51
52     static const char * GetMPTypeString(const MPType type);
53
54     static MPType GetMPType(const char * type);
55
56     MeshPrimitive();
57
58     ~MeshPrimitive() override;
59
60     MPType GetPrimitiveType() const;
61     void SetPrimitiveType(const MPType type);
62
63     const DataElement & GetPrimitiveData() const;
64     DataElement & GetPrimitiveData();
65     void SetPrimitiveData(DataElement const & de);
66
67     const PrimitivesData & GetPrimitivesData() const;
68     PrimitivesData & GetPrimitivesData();
69     void SetPrimitivesData(PrimitivesData const & DEs);
70
71     const DataElement & GetPrimitiveData(const unsigned int idx) const;
72     DataElement & GetPrimitiveData(const unsigned int idx);
73     void SetPrimitiveData(const unsigned int idx, DataElement const & de);
74     void AddPrimitiveData(DataElement const & de);
75
76     unsigned int GetNumberOfPrimitivesData() const;
77
78 protected:
79
80     // Use to define tag where PrimitiveData will be put.
81     MPType          PrimitiveType;
82
83     // PrimitiveData contains point index list.
84     // It shall have 1 or 1-n DataElement following PrimitiveType.
85     PrimitivesData  PrimitiveData;
86 };
87
88 }
89
90 #endif // GDCMMESHPRIMITIVE_H
```

11.361 gdcmOrientation.h File Reference

`#include "gdcmTypes.h"`
Include dependency graph for gdcmOrientation.h:



Classes

- class gdcm::Orientation

  class to handle Orientation

Namespaces

- namespace gdcm

Functions

- std::ostream & gdcm::operator<< (std::ostream &os, const Orientation &o)

11.362 gdcmOrientation.h

Go to the documentation of this file.

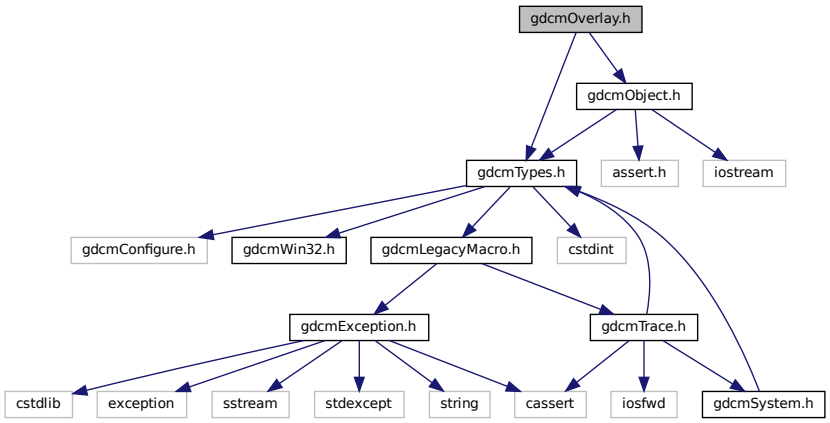
```
1 /*=========================================================================
2
3 Program:  GDCM (Grassroots DICOM). A DICOM library
4
5 Copyright (c) 2006-2011 Mathieu Malaterre
6 All rights reserved.
7 See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.
8
9 This software is distributed WITHOUT ANY WARRANTY; without even
10 the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
11 PURPOSE.  See the above copyright notice for more information.
12
13 =========================================================================*/
14 #ifndef GDCMORIENTATION_H
```

```
15 #define GDCMORIENTATION_H
16
17 #include "gdcmTypes.h"
18
19 namespace gdcm
20 {
21
25 class GDCM_EXPORT Orientation
26 {
27   friend std::ostream& operator<<(std::ostream &_os, const Orientation &o);
28 public:
29   Orientation();
30   ~Orientation();
31
33   void Print(std::ostream &) const;
34
35   typedef enum {
36     UNKNOWN,
37     AXIAL,
38     CORONAL,
39     SAGITTAL,
40     OBLIQUE
41   } OrientationType;
42
45   static OrientationType GetType(const double dircos[6]);
46
48   static void SetObliquityThresholdCosineValue(double val);
49   static double GetObliquityThresholdCosineValue();
50
52   static const char *GetLabel(OrientationType type);
53
54 protected:
55   static char GetMajorAxisFromPatientRelativeDirectionCosine(double x, double y, double z);
56
57 private:
58   static double ObliquityThresholdCosineValue;
59 };
60 //-----------------------------------------------------------------------------
61 inline std::ostream& operator<<(std::ostream &os, const Orientation &o)
62 {
63   o.Print( os );
64   return os;
65 }
66
67 } // end namespace gdcm
68
69 #endif //GDCMORIENTATION_H
```

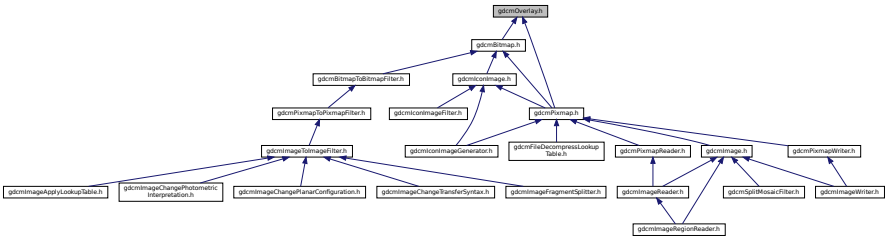
11.363 gdcmOverlay.h File Reference

```
#include "gdcmTypes.h"
#include "gdcmObject.h"
```

Include dependency graph for gdcmOverlay.h:



This graph shows which files directly or indirectly include this file:



Classes

- class gdcm::Overlay
  Overlay class.

Namespaces

- namespace gdcm

11.364 gdcmOverlay.h

Go to the documentation of this file.

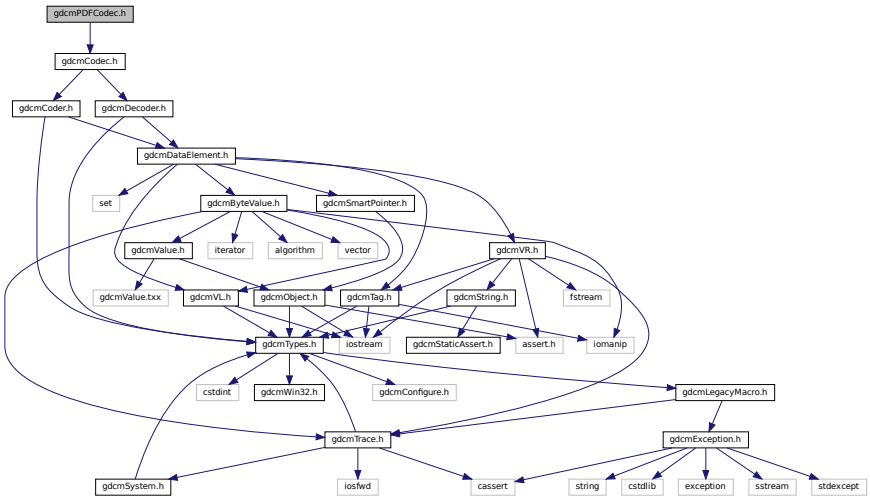
```
1 /*=========================================================================
2
3 Program:  GDCM (Grassroots DICOM). A DICOM library
4
```

```
5 Copyright (c) 2006-2011 Mathieu Malaterre
6 All rights reserved.
7 See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.
8
9 This software is distributed WITHOUT ANY WARRANTY; without even
10 the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
11 PURPOSE.  See the above copyright notice for more information.
12
13 =========================================================================*/
14 #ifndef GDCMOVERLAY_H
15 #define GDCMOVERLAY_H
16
17 #include "gdcmTypes.h"
18 #include "gdcmObject.h"
19
20 namespace gdcm
21 {
22
23 class OverlayInternal;
24 class ByteValue;
25 class DataSet;
26 class DataElement;
38 class GDCM_EXPORT Overlay :  public Object
39 {
40 public:
41   Overlay();
42   ~Overlay() override;
44   void Print(std::ostream &) const override;
45
47   void Update(const DataElement & de);
48
50   void SetGroup(unsigned short group);
52   unsigned short GetGroup() const;
54   void SetRows(unsigned short rows);
56   unsigned short GetRows() const;
58   void SetColumns(unsigned short columns);
60   unsigned short GetColumns() const;
62   void SetNumberOfFrames(unsigned int numberofframes);
64   void SetDescription(const char* description);
66   const char *GetDescription() const;
67   typedef enum {
68     Invalid  = 0,
69     Graphics = 1,
70     ROI      = 2
71   } OverlayType;
73   void SetType(const char* type);
75   const char *GetType() const;
76   OverlayType GetTypeAsEnum() const;
77   static const char *GetOverlayTypeAsString(OverlayType ot);
78   static OverlayType GetOverlayTypeFromString(const char *);
80   void SetOrigin(const signed short origin[2]);
82   const signed short * GetOrigin() const;
84   void SetFrameOrigin(unsigned short frameorigin);
86   void SetBitsAllocated(unsigned short bitsallocated);
88   unsigned short GetBitsAllocated() const;
90   void SetBitPosition(unsigned short bitposition);
92   unsigned short GetBitPosition() const;
93
95   void SetOverlay(const char *array, size_t length);
97   bool GrabOverlayFromPixelData(DataSet const &ds);
98
101   const ByteValue &GetOverlayData() const;
102
104   bool IsEmpty() const;
105
107   bool IsZero() const;
108
110   bool IsInPixelData() const;
111
113   void IsInPixelData(bool b);
114
116   void Decompress(std::ostream &os) const;
117
120   size_t GetUnpackBufferLength() const;
121
124   bool GetUnpackBuffer(char *buffer, size_t len) const;
125
126   Overlay(Overlay const &ov);
127   Overlay &operator=(Overlay const &ov);
128
129 private:
```

```
130   OverlayInternal *Internal;
131 };
132
133 } // end namespace gdcm
134
135 #endif //GDCMOVERLAY_H
```

11.365 gdcmPDFCodec.h File Reference

`#include "gdcmCodec.h"`
Include dependency graph for gdcmPDFCodec.h:



Classes

- class gdcm::PDFCodec

  PDFCodec class.

Namespaces

- namespace gdcm

11.366 gdcmPDFCodec.h

Go to the documentation of this file.

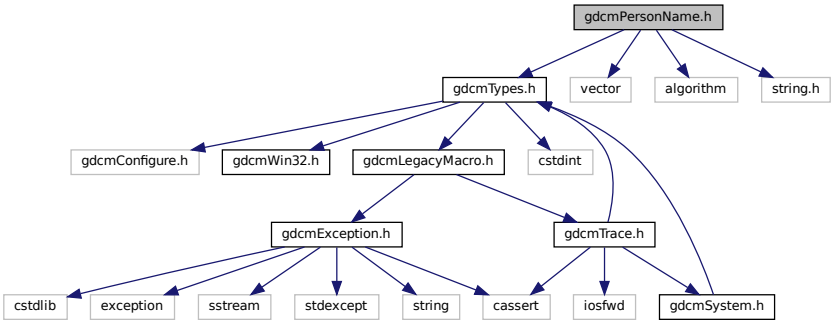
```
1 /*=========================================================================
2
3 Program:  GDCM (Grassroots DICOM). A DICOM library
4
5 Copyright (c) 2006-2011 Mathieu Malaterre
6 All rights reserved.
```

```
7 See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.
8
9 This software is distributed WITHOUT ANY WARRANTY; without even
10 the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
11 PURPOSE.  See the above copyright notice for more information.
12
13 =========================================================================*/
14 #ifndef GDCMPDFCODEC_H
15 #define GDCMPDFCODEC_H
16
17 #include "gdcmCodec.h"
18
19 namespace gdcm
20 {
21
25 class GDCM_EXPORT PDFCodec :  public Codec
26 {
27 public:
28   PDFCodec();
29   ~PDFCodec() override;
30   bool CanCode(TransferSyntax const &)const override { return false; }
31   bool CanDecode(TransferSyntax const &)const override { return false; }
32   bool Decode(DataElement const &is, DataElement &os) override;
33 };
34
35 } // end namespace gdcm
36
37 #endif //GDCMPDFCODEC_H
```

11.367 gdcmPersonName.h File Reference

```
#include "gdcmTypes.h"
#include <vector>
#include <algorithm>
#include <string.h>
```

Include dependency graph for gdcmPersonName.h:



Classes

- class gdcm::PersonName

  PersonName class.

Namespaces

- namespace gdcm

11.368 gdcmPersonName.h

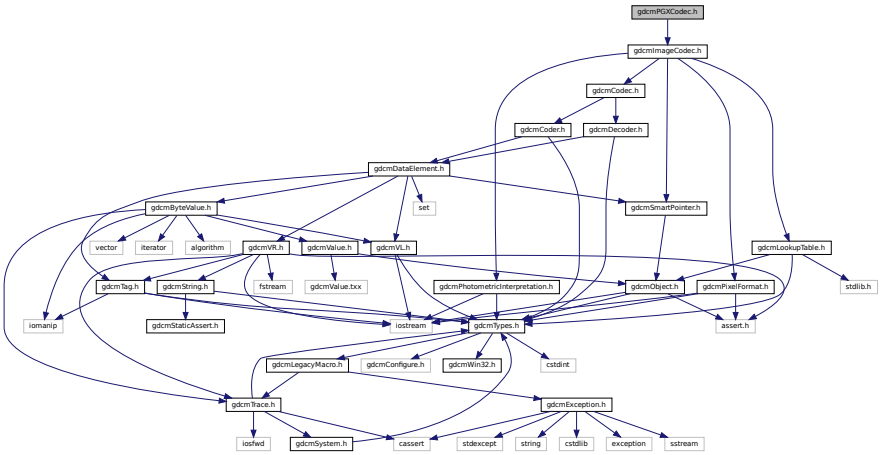
Go to the documentation of this file.

```
1 /*=========================================================================
2
3 Program:  GDCM (Grassroots DICOM). A DICOM library
4
5 Copyright (c) 2006-2011 Mathieu Malaterre
6 All rights reserved.
7 See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.
8
9 This software is distributed WITHOUT ANY WARRANTY; without even
10 the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
11 PURPOSE.  See the above copyright notice for more information.
12
13 =========================================================================*/
14
15 #ifndef GDCMPERSONNAME_H
16 #define GDCMPERSONNAME_H
17
18 #include "gdcmTypes.h"
19 #include <vector>
20 #include <algorithm> // std::min
21 #include <string.h> // strlen
22
23 namespace gdcm
24 {
25
29 class GDCM_EXPORT PersonName
30 {
31 public:
32   static const unsigned int MaxNumberOfComponents = 5;
33   static const unsigned int MaxLength = 64;
34   char Component[MaxNumberOfComponents][MaxLength+1];
35   static const char Separator = '^';
36   static const char Padding   = ' ';
37
38   unsigned int GetNumberOfComponents()const {
39     unsigned int r = 0;
40     for(unsigned int i = 0; i < 5; ++i) {
41       if( *Component[i] != '\0' ) r = i;
42       }
43     return r+1;
44   }
45   unsigned int GetMaxLength()const { return MaxLength; };
46   void SetBlob(const std::vector<char>& v) {
47   (void)v;
48     //assert(0); //TODO
49   }
50   void SetComponents(const char *comp1 = "",
51     const char *comp2 = "",
52     const char *comp3 = "",
53     const char *comp4 = "",
54     const char *comp5 = "") {
55     const char *components[5] = { comp1, comp2, comp3, comp4, comp5 };
56     SetComponents( components );
57   }
58   void SetComponents(const char *components[]) {
59     if( components )
60       for(unsigned int i = 0; i < 5; ++i) {
61         if( components[i] && strlen(components[i]) < GetMaxLength() )
62           strcpy(Component[i], components[i]);
63         assert( strlen(Component[i]) < GetMaxLength() );
64       }
65   }
66   void Print(std::ostream &os)const
67 {
68     //os « "Family Name Complex:  " « Component[0] « std::endl;
69     //os « "Given  Name Complex:  " « Component[1] « std::endl;
70     //os « "Middle Name          :  " « Component[2] « std::endl;
71     //os « "Name Suffix          :  " « Component[3] « std::endl;
72     //os « "Name Prefix          :  " « Component[4] « std::endl;
73     os « Component[0] « '^';
74     os « Component[1] « '^';
75     os « Component[2] « '^';
76     os « Component[3] « '^';
77     os « Component[4];
78   }
79 };
```

```
80
81 } // end namespace gdcm
82
83 #endif //GDCMPERSONNAME_H
```

11.369 gdcmPGXCodec.h File Reference

`#include "gdcmImageCodec.h"`
Include dependency graph for gdcmPGXCodec.h:



Classes

- class gdcm::PGXCodec

  Class to do PGX.

Namespaces

- namespace gdcm

11.370 gdcmPGXCodec.h

Go to the documentation of this file.

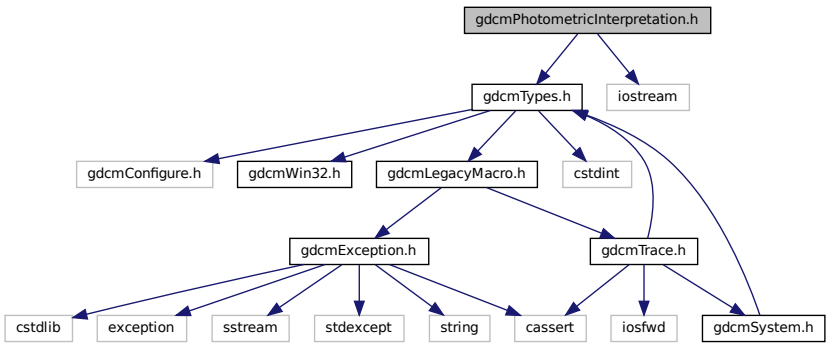
```
1 /*=========================================================================
2
3 Program:  GDCM (Grassroots DICOM). A DICOM library
4
5 Copyright (c) 2006-2011 Mathieu Malaterre
6 All rights reserved.
7 See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.
8
9 This software is distributed WITHOUT ANY WARRANTY; without even
10 the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
11 PURPOSE.  See the above copyright notice for more information.
```

```
12
13 =========================================================================*/
14 #ifndef GDCMPGXCODEC_H
15 #define GDCMPGXCODEC_H
16
17 #include "gdcmImageCodec.h"
18
19 namespace gdcm
20 {
21
26 class GDCM_EXPORT PGXCodec :  public ImageCodec
27 {
28 public:
29   PGXCodec();
30   ~PGXCodec() override;
31   bool CanDecode(TransferSyntax const &ts) const override;
32   bool CanCode(TransferSyntax const &ts) const override;
33
34   bool GetHeaderInfo(std::istream &is, TransferSyntax &ts) override;
35   ImageCodec * Clone() const override;
36
37   bool Read(const char *filename, DataElement &out) const;
38   bool Write(const char *filename, const DataElement &out) const;
39 private:
40 };
41
42 } // end namespace gdcm
43
44 #endif //GDCMPGXCODEC_H
```

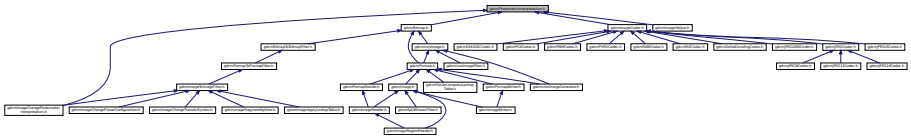
11.371 gdcmPhotometricInterpretation.h File Reference

```
#include "gdcmTypes.h"
#include <iostream>
```

Include dependency graph for gdcmPhotometricInterpretation.h:



This graph shows which files directly or indirectly include this file:



Classes

- class gdcm::PhotometricInterpretation

  Class to represent an PhotometricInterpretation.

Namespaces

- namespace gdcm

Functions

- std::ostream & gdcm::operator<< (std::ostream &os, const PhotometricInterpretation &val)

11.372 gdcmPhotometricInterpretation.h

Go to the documentation of this file.

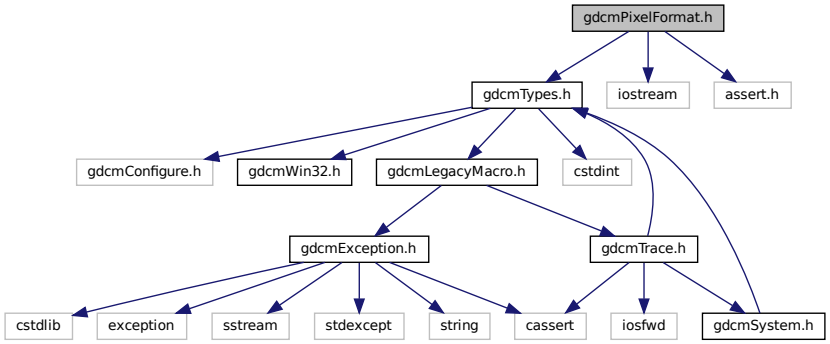
```
1 /*=========================================================================
2
3 Program:  GDCM (Grassroots DICOM). A DICOM library
4
5 Copyright (c) 2006-2011 Mathieu Malaterre
6 All rights reserved.
7 See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.
8
9 This software is distributed WITHOUT ANY WARRANTY; without even
10 the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
11 PURPOSE.  See the above copyright notice for more information.
12
13 =========================================================================*/
14
15 #ifndef GDCMPHOTOMETRICINTERPRETATION_H
16 #define GDCMPHOTOMETRICINTERPRETATION_H
17
18 #include "gdcmTypes.h"
19 #include <iostream>
20
21 namespace gdcm
22 {
23
24 class TransferSyntax;
28 class GDCM_EXPORT PhotometricInterpretation
29 {
30 public:
31   typedef enum {
32     UNKNOWN = 0,
33     MONOCHROME1,
34     MONOCHROME2,
35     PALETTE_COLOR,
36     RGB,
37     HSV,
38     ARGB, // retired
39     CMYK,
40     YBR_FULL,
41     YBR_FULL_422,
42     YBR_PARTIAL_422,
43     YBR_PARTIAL_420,
44     YBR_ICT,
45     YBR_RCT,
46     // PALETTE_COLOR ?
47     //MONOCHROME = MONOCHROME1 | MONOCHROME2,
48     //COLOR      = RGB | HSV | ARGB | CMYK | YBR_FULL | YBR_FULL_422 | YBR_PARTIAL_422 | YBR_PARTIAL_420 |
    YBR_ICT | YBR_RCT,
49     PI_END  // Helpful for internal implementation
50   } PIType; // PhotometricInterpretationType
51
```

```
52   PhotometricInterpretation(PIType pi = UNKNOWN):PIField(pi) {}
53
54   static const char *GetPIString(PIType pi);
55
56   const char *GetString() const;
57
58   // You need to make sure end of string is \0
59   static PIType GetPIType(const char *pi);
60
61   static bool IsRetired(PIType pi);
62
63   bool IsLossy() const;
64   bool IsLossless() const;
65
67   unsigned short GetSamplesPerPixel() const;
68
69   // TODO
70   // not all PhotometricInterpretation are allowed for compressed Transfer
71   // syntax
72   // static bool IsAllowedForCompressedTS(PIType pi);
73
74   friend std::ostream& operator<<(std::ostream& os, const PhotometricInterpretation& pi);
75
76   operator PIType ()const { return PIField; }
77
78   PIType GetType ()const { return PIField; }
79
80   // Will return whether current PhotometricInterpretation is the same Color Space as input:
81   // eg.  RGB and YBR_RCT are
82   bool IsSameColorSpace( PhotometricInterpretation const &pi ) const;
83
84   //static PIType GetEquivalent(TransferSyntax const &ts);
85
86 private:
87   PIType PIField;
88 };
89 //-----------------------------------------------------------------------------
90 inline std::ostream& operator<<(std::ostream& os, const PhotometricInterpretation &val)
91 {
92   const char *s = PhotometricInterpretation::GetPIString(val.PIField);
93   os << (s ?  s :  "");
94   return os;
95 }
96
97
98 } // end namespace gdcm
99
100 #endif //GDCMPHOTOMETRICINTERPRETATION_H
```

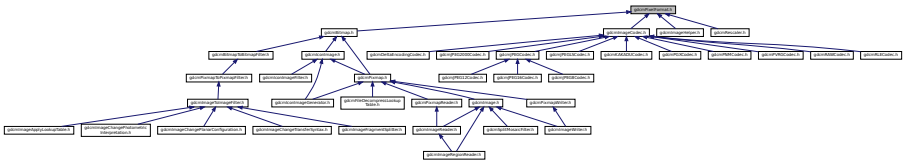
11.373 gdcmPixelFormat.h File Reference

```
#include "gdcmTypes.h"
#include <iostream>
#include <assert.h>
```

Include dependency graph for gdcmPixelFormat.h:



This graph shows which files directly or indirectly include this file:



Classes

- class gdcm::PixelFormat

  PixelFormat.

Namespaces

- namespace gdcm

Functions

- std::ostream & gdcm::operator<< (std::ostream &os, const PixelFormat &pf)

11.374 gdcmPixelFormat.h

Go to the documentation of this file.

```
1 /*=========================================================================
2
3 Program:  GDCM (Grassroots DICOM). A DICOM library
4
5 Copyright (c) 2006-2011 Mathieu Malaterre
6 All rights reserved.
7 See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.
8
9 This software is distributed WITHOUT ANY WARRANTY; without even
10 the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
11 PURPOSE.  See the above copyright notice for more information.
12
13 =========================================================================*/
14
15 #ifndef GDCMPIXELFORMAT_H
16 #define GDCMPIXELFORMAT_H
17
18 #include "gdcmTypes.h"
19 #include <iostream>
20 #include <assert.h>
21
22 namespace gdcm
23 {
24
25 class TransferSyntax;
26
45 class GDCM_EXPORT PixelFormat
46 {
47   friend class Bitmap;
48   friend std::ostream& operator<<(std::ostream &_os, const PixelFormat &pf);
49 public:
50   // When adding a type please add its dual type (its unsigned conterpart)
51   typedef enum {
52     UINT8,
53     INT8,
54     UINT12,
55     INT12,
56     UINT16,
57     INT16,
58     UINT32,  // For some DICOM files (RT or SC)
59     INT32,   //                       "   "
60     UINT64,  // Needed when input is 32bits + intercept/slope (incomplete support)
61     INT64,   //                       "   "
62     FLOAT16, // sure why not...
63     FLOAT32, // good ol' 'float'
64     FLOAT64, // aka 'double'
65     SINGLEBIT, // bool / monochrome
66     UNKNOWN // aka BitsAllocated == 0 && PixelRepresentation == 0
67   } ScalarType;
68
69   // default cstor:
70   PixelFormat () :  PixelFormat(1, 8, 8, 7, 0) {}
71
72   explicit PixelFormat (
73     unsigned short samplesperpixel,
74     unsigned short bitsallocated = 8,
75     unsigned short bitsstored = 8,
76     unsigned short highbit = 7,
77     unsigned short pixelrepresentation = 0 ) :
78   SamplesPerPixel(samplesperpixel),
79   BitsAllocated(bitsallocated),
80   BitsStored(bitsstored),
81   HighBit(highbit),
82   PixelRepresentation(pixelrepresentation) {}
83   // helper, for the common case
84   PixelFormat(ScalarType st);
85
86   // For transparency of use
87   operator ScalarType()const { return GetScalarType(); }
88
91   unsigned short GetSamplesPerPixel() const;
92   void SetSamplesPerPixel(unsigned short spp)
93     {
94     gdcmAssertMacro( spp <= 4 );
95     SamplesPerPixel = spp;
96     assert( SamplesPerPixel == 1 || SamplesPerPixel == 3 || SamplesPerPixel == 4 );
```

```
97     }
98
100   unsigned short GetBitsAllocated()const
101 {
102     return BitsAllocated;
103     }
104   void SetBitsAllocated(unsigned short ba)
105     {
106     if( ba )
107       {
108       switch( ba )
109         {
110         /* some devices (FUJIFILM CR + MONO1) incorrectly set BitsAllocated/BitsStored
111 * as bitmask instead of value.  Do what they mean instead of what they say.
112 */
113         case 0xffff:  ba = 16; break;
114         case 0x0fff:  ba = 12; break;
115         case 0x00ff:  ba =  8; break;
116         }
117       BitsAllocated = ba;
118       BitsStored = ba;
119       HighBit = (unsigned short)(ba - 1);
120       }
121     else // Make the PixelFormat as UNKNOWN
122       {
123       BitsAllocated = 0;
124       PixelRepresentation = 0;
125       }
126     }
127
129   unsigned short GetBitsStored()const
130 {
131     assert( BitsStored <= BitsAllocated );
132     return BitsStored;
133     }
134   void SetBitsStored(unsigned short bs)
135     {
136     switch( bs )
137       {
138       /* see SetBitsAllocated for explanation
139 */
140       case 0xffff:  bs = 16; break;
141       case 0x0fff:  bs = 12; break;
142       case 0x00ff:  bs =  8; break;
143       }
144     if( bs <= BitsAllocated && bs )
145       {
146       BitsStored = bs;
147       SetHighBit( (unsigned short) (bs - 1) );
148       }
149     }
150
152   unsigned short GetHighBit()const
153 {
154     assert( HighBit < BitsStored );
155     return HighBit;
156     }
157   void SetHighBit(unsigned short hb)
158     {
159     switch( hb )
160       {
161       /* broken implementations that use bitmask for BitsAllocated/Stored
162 * nonetheless use (BitsStored-1) for HighBit.  correct for this here.
163 */
164       case 0xfffe:  hb = 15; break;
165       case 0x0ffe:  hb = 11; break;
166       case 0x00fe:  hb =  7; break;
167       }
168     if( hb < BitsStored )
169       HighBit = hb;
170     }
171
173   unsigned short GetPixelRepresentation()const
174 {
175     return (unsigned short)(PixelRepresentation ?  1 :  0);
176     }
177   void SetPixelRepresentation(unsigned short pr)
178     {
179     PixelRepresentation = (unsigned short)(pr ?  1 :  0);
180     }
181
```

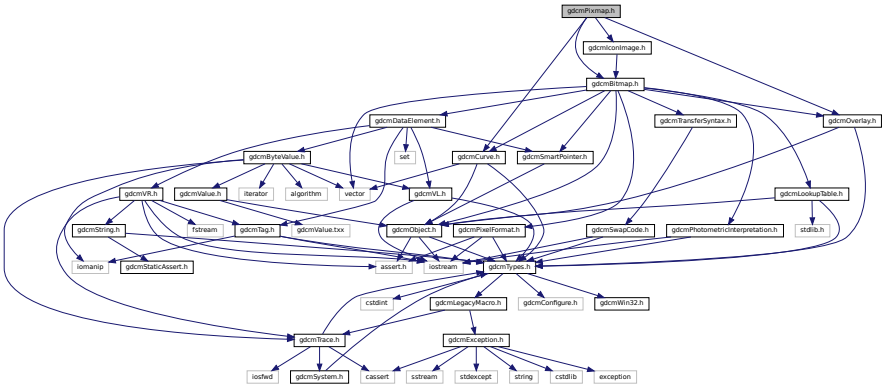
```
183   ScalarType GetScalarType() const;
184
187   void SetScalarType(ScalarType st);
188   const char *GetScalarTypeAsString() const;
189
195   uint8_t GetPixelSize() const;
196
198   void Print(std::ostream &os) const;
199
201   int64_t GetMin() const;
202
204   int64_t GetMax() const;
205
207   bool IsValid() const;
208
209   bool operator==(ScalarType st)const
210 {
211     return GetScalarType() == st;
212     }
213   bool operator!=(ScalarType st)const
214 {
215     return GetScalarType() != st;
216     }
217   bool operator==(const PixelFormat &pf)const
218 {
219     return
220       SamplesPerPixel     == pf.SamplesPerPixel &&
221       BitsAllocated       == pf.BitsAllocated &&
222       BitsStored          == pf.BitsStored &&
223       HighBit             == pf.HighBit &&
224       PixelRepresentation == pf.PixelRepresentation;
225     }
226   bool operator!=(const PixelFormat &pf)const
227 {
228     return
229       SamplesPerPixel     != pf.SamplesPerPixel ||
230       BitsAllocated       != pf.BitsAllocated ||
231       BitsStored          != pf.BitsStored ||
232       HighBit             != pf.HighBit ||
233       PixelRepresentation != pf.PixelRepresentation;
234     }
235
236   bool IsCompatible(const TransferSyntax & ts ) const;
237 protected:
239   bool Validate();
240
241 private:
242   // D 0028|0002 [US] [Samples per Pixel] [1]
243   unsigned short SamplesPerPixel;
244   // D 0028|0100 [US] [Bits Allocated] [8]
245   unsigned short BitsAllocated;
246   // D 0028|0101 [US] [Bits Stored] [8]
247   unsigned short BitsStored;
248   // D 0028|0102 [US] [High Bit] [7]
249   unsigned short HighBit;
250   // D 0028|0103 [US] [Pixel Representation] [0]
251   unsigned short PixelRepresentation;
252 };
253 //-----------------------------------------------------------------------------
254 inline std::ostream& operator<<(std::ostream &os, const PixelFormat &pf)
255 {
256   pf.Print( os );
257   return os;
258 }
259
260 } // end namespace gdcm
261
262 #endif //GDCMPIXELFORMAT_H
```

11.375 gdcmPixmap.h File Reference

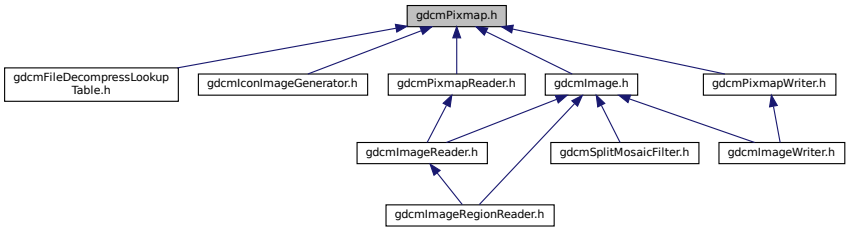
```
#include "gdcmBitmap.h"
#include "gdcmCurve.h"
#include "gdcmIconImage.h"
```

```
#include "gdcmOverlay.h"
```

Include dependency graph for gdcmPixmap.h:



This graph shows which files directly or indirectly include this file:



Classes

- class gdcm::Pixmap

  Pixmap class.

Namespaces

- namespace gdcm

11.376 gdcmPixmap.h

Go to the documentation of this file.

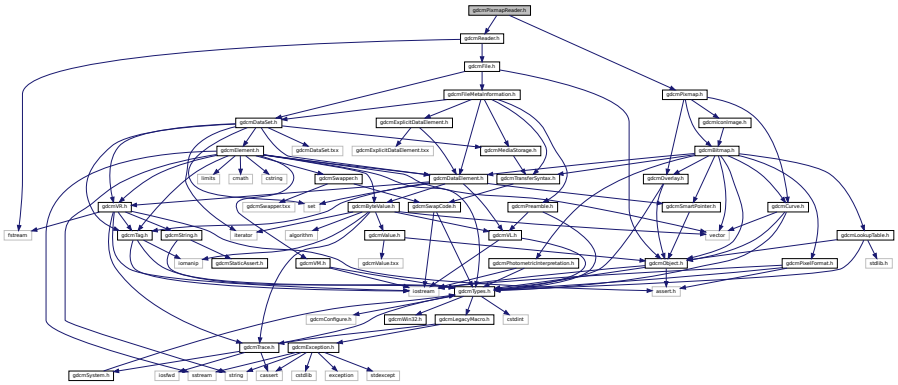
```
1 /*=========================================================================
2
3 Program:  GDCM (Grassroots DICOM). A DICOM library
4
5 Copyright (c) 2006-2011 Mathieu Malaterre
```

```
6 All rights reserved.
7 See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.
8
9 This software is distributed WITHOUT ANY WARRANTY; without even
10 the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
11 PURPOSE.  See the above copyright notice for more information.
12
13 =========================================================================*/
14 #ifndef GDCMPIXMAP_H
15 #define GDCMPIXMAP_H
16
17 #include "gdcmBitmap.h"
18 #include "gdcmCurve.h"
19 #include "gdcmIconImage.h"
20 #include "gdcmOverlay.h"
21
22 namespace gdcm
23 {
24
32 class GDCM_EXPORT Pixmap :  public Bitmap
33 {
34 public:
35   Pixmap();
36   ~Pixmap() override;
37   void Print(std::ostream &) const override;
38
40   bool AreOverlaysInPixelData() const override;
42   bool UnusedBitsPresentInPixelData() const override;
43
45   Curve& GetCurve(size_t i = 0) {
46     assert( i < Curves.size() );
47     return Curves[i];
48   }
49   const Curve& GetCurve(size_t i = 0)const {
50     assert( i < Curves.size() );
51     return Curves[i];
52   }
53   size_t GetNumberOfCurves()const { return Curves.size(); }
54   void SetNumberOfCurves(size_t n) { Curves.resize(n); }
55
57   Overlay& GetOverlay(size_t i = 0) {
58     assert( i < Overlays.size() );
59     return Overlays[i];
60   }
61   const Overlay& GetOverlay(size_t i = 0)const {
62     assert( i < Overlays.size() );
63     return Overlays[i];
64   }
65   size_t GetNumberOfOverlays()const { return Overlays.size(); }
66   void SetNumberOfOverlays(size_t n) { Overlays.resize(n); }
67   void RemoveOverlay(size_t i) {
68     assert( i < Overlays.size() );
69     Overlays.erase( Overlays.begin() + i );
70   }
71
73   const IconImage &GetIconImage()const { return *Icon; }
74   IconImage &GetIconImage() { return *Icon; }
75   void SetIconImage(IconImage const &ii) { Icon = ii; }
76
77 //private:
78 protected:
79   std::vector<Overlay>  Overlays;
80   std::vector<Curve>  Curves;
81   SmartPointer<IconImage> Icon;
82 };
83
84 } // end namespace gdcm
85
86 #endif //GDCMPIXMAP_H
```

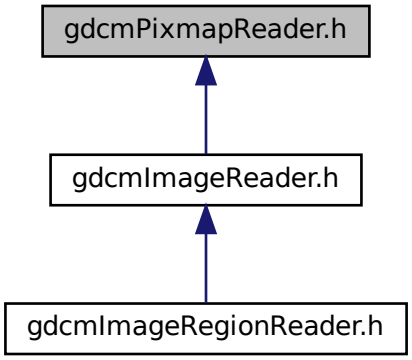
11.377 gdcmPixmapReader.h File Reference

```
#include "gdcmReader.h"
#include "gdcmPixmap.h"
```

Include dependency graph for gdcmPixmapReader.h:



This graph shows which files directly or indirectly include this file:



Classes

- class gdcm::PixmapReader
  PixmapReader.

Namespaces

- namespace gdcm

11.378 gdcmPixmapReader.h

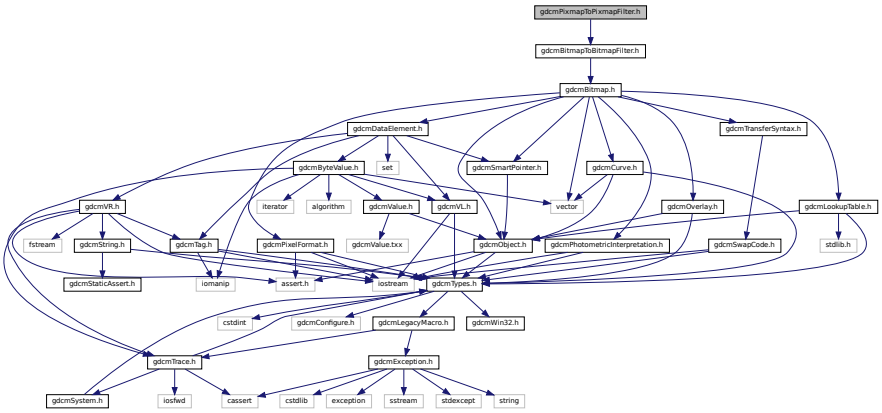
Go to the documentation of this file.

```
1 /*=========================================================================
2
3 Program:  GDCM (Grassroots DICOM). A DICOM library
4
5 Copyright (c) 2006-2011 Mathieu Malaterre
6 All rights reserved.
7 See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.
8
9 This software is distributed WITHOUT ANY WARRANTY; without even
10 the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
11 PURPOSE.  See the above copyright notice for more information.
12
13 =========================================================================*/
14 #ifndef GDCMPIXMAPREADER_H
15 #define GDCMPIXMAPREADER_H
16
17 #include "gdcmReader.h"
18 #include "gdcmPixmap.h"
19
20 namespace gdcm
21 {
22
23 class ByteValue;
24 class MediaStorage;
40 class GDCM_EXPORT PixmapReader :  public Reader
41 {
42 public:
43   PixmapReader();
44   ~PixmapReader() override; //needs to be virtual to ensure lack of memory leaks
45
49
50   bool Read() override;
51
52   // Following methods are valid only after a call to 'Read'
53
55   const Pixmap& GetPixmap() const;
56   Pixmap& GetPixmap();
57   //void SetPixamp(Pixmap const &pix);
58
59 protected:
60   bool ReadImageInternal(MediaStorage const &ms, bool handlepixeldata = true);
61   virtual bool ReadImage(MediaStorage const &ms);
62   virtual bool ReadACRNEMAImage();
63
64   SmartPointer<Pixmap> PixelData;
65 };
66
72 } // end namespace gdcm
73
74 #endif //GDCMPIXMAPREADER_H
```

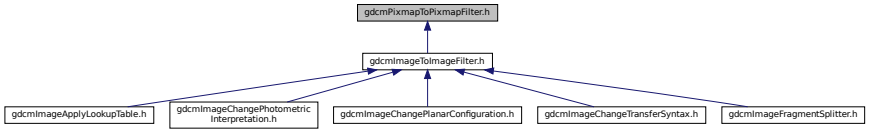
11.379 gdcmPixmapToPixmapFilter.h File Reference

```
#include "gdcmBitmapToBitmapFilter.h"
```

Include dependency graph for gdcmPixmapToPixmapFilter.h:



This graph shows which files directly or indirectly include this file:



Classes

- class gdcm::PixmapToPixmapFilter

  PixmapToPixmapFilter class.

Namespaces

- namespace gdcm

11.380 gdcmPixmapToPixmapFilter.h

Go to the documentation of this file.

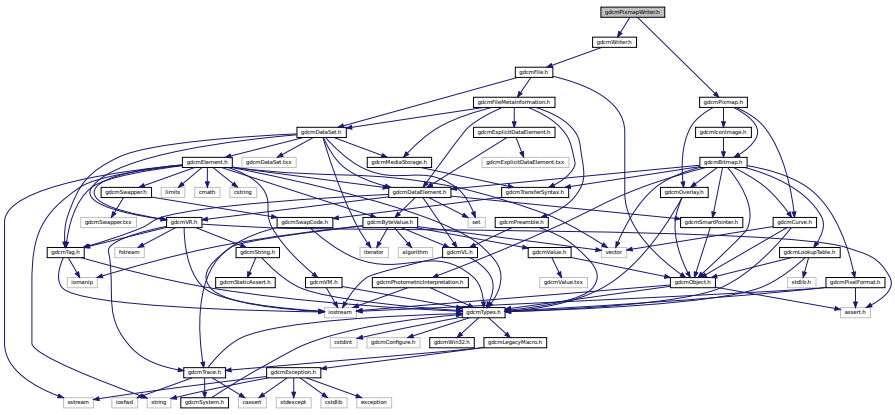
```
1 /*=========================================================================
2
3 Program:  GDCM (Grassroots DICOM). A DICOM library
4
5 Copyright (c) 2006-2011 Mathieu Malaterre
```

```
6 All rights reserved.
7 See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.
8
9 This software is distributed WITHOUT ANY WARRANTY; without even
10 the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
11 PURPOSE.  See the above copyright notice for more information.
12
13 =========================================================================*/
14 #ifndef GDCMPIXMAPTOPIXMAPFILTER_H
15 #define GDCMPIXMAPTOPIXMAPFILTER_H
16
17 #include "gdcmBitmapToBitmapFilter.h"
18
19 namespace gdcm
20 {
21
22 class Pixmap;
27 class GDCM_EXPORT PixmapToPixmapFilter :  public BitmapToBitmapFilter
28 {
29 public:
30   PixmapToPixmapFilter();
31   ~PixmapToPixmapFilter() = default;
32
33   Pixmap &GetInput();
34
36   const Pixmap &GetOutput() const;
37
38   // SWIG/Java hack:
39   const Pixmap &GetOutputAsPixmap() const;
40 };
41
42 } // end namespace gdcm
43
44 #endif //GDCMPIXMAPTOPIXMAPFILTER_H
```

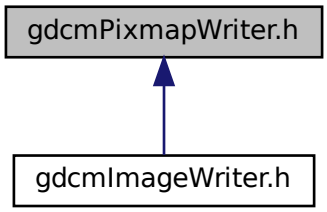
11.381 gdcmPixmapWriter.h File Reference

```
#include "gdcmWriter.h"
#include "gdcmPixmap.h"
```

Include dependency graph for gdcmPixmapWriter.h:



This graph shows which files directly or indirectly include this file:



Classes

- class gdcm::PixmapWriter

  PixmapWriter.

Namespaces

- namespace gdcm

11.382 gdcmPixmapWriter.h

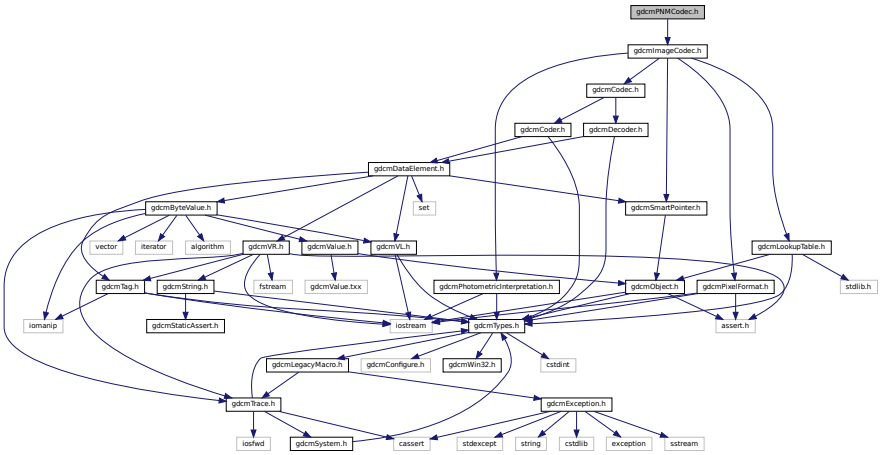
Go to the documentation of this file.

```
1 /*=========================================================================
2
3 Program:  GDCM (Grassroots DICOM). A DICOM library
4
5 Copyright (c) 2006-2011 Mathieu Malaterre
6 All rights reserved.
7 See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.
8
9 This software is distributed WITHOUT ANY WARRANTY; without even
10 the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
11 PURPOSE.  See the above copyright notice for more information.
12
13 =========================================================================*/
14 #ifndef GDCMPIXMAPWRITER_H
15 #define GDCMPIXMAPWRITER_H
16
17 #include "gdcmWriter.h"
18 #include "gdcmPixmap.h"
19
20 namespace gdcm
21 {
22
23 class StreamImageWriter;
24 class Pixmap;
36 class GDCM_EXPORT PixmapWriter :  public Writer
37 {
38 public:
39   PixmapWriter();
40   ~PixmapWriter() override;
```

```
41
42   const Pixmap& GetPixmap()const { return *PixelData; }
43   Pixmap& GetPixmap() { return *PixelData; } // FIXME
44   void SetPixmap(Pixmap const &img);
45
49   virtual const Pixmap& GetImage()const { return *PixelData; }
50   virtual Pixmap& GetImage() { return *PixelData; } // FIXME
51   virtual void SetImage(Pixmap const &img);
52
54   bool Write() override; // Execute()
55
56 protected:
57   void DoIconImage(DataSet & ds, Pixmap const & image);
58   bool PrepareWrite( MediaStorage const & refms );
59
60   SmartPointer<Pixmap> PixelData;
61 };
62
68 } // end namespace gdcm
69
70 #endif //GDCMPIXMAPWRITER_H
```

11.383 gdcmPNMCodec.h File Reference

`#include "gdcmImageCodec.h"`
Include dependency graph for gdcmPNMCodec.h:



Classes

- class gdcm::PNMCodec

  Class to do PNM.

Namespaces

- namespace gdcm

11.384 gdcmPNMCodec.h

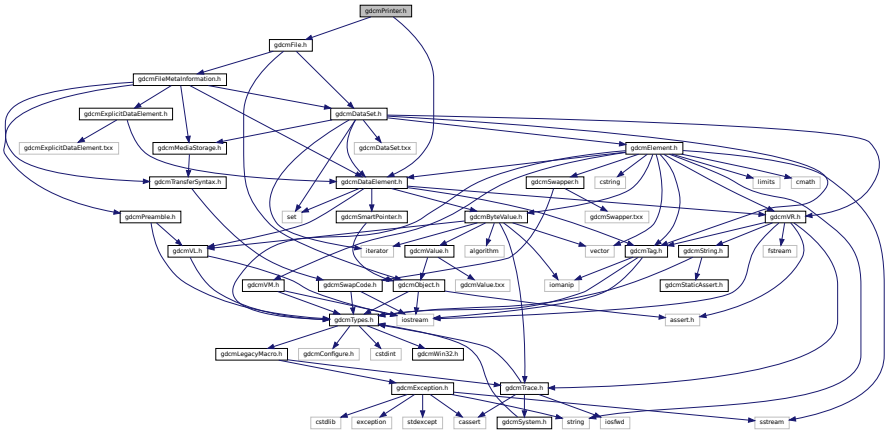
Go to the documentation of this file.

```
1 /*=========================================================================
2
3 Program:  GDCM (Grassroots DICOM). A DICOM library
4
5 Copyright (c) 2006-2011 Mathieu Malaterre
6 All rights reserved.
7 See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.
8
9 This software is distributed WITHOUT ANY WARRANTY; without even
10 the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
11 PURPOSE.  See the above copyright notice for more information.
12
13 =========================================================================*/
14 #ifndef GDCMPNMCODEC_H
15 #define GDCMPNMCODEC_H
16
17 #include "gdcmImageCodec.h"
18
19 namespace gdcm
20 {
21
29 class GDCM_EXPORT PNMCodec :  public ImageCodec
30 {
31 public:
32   PNMCodec();
33   ~PNMCodec() override;
34   bool CanDecode(TransferSyntax const &ts) const override;
35   bool CanCode(TransferSyntax const &ts) const override;
36
37   unsigned long GetBufferLength()const { return BufferLength; }
38   void SetBufferLength(unsigned long l) { BufferLength = l; }
39
40   bool GetHeaderInfo(std::istream &is, TransferSyntax &ts) override;
41   ImageCodec * Clone() const override;
42
43   bool Read(const char *filename, DataElement &out) const;
44   bool Write(const char *filename, const DataElement &out) const;
45   //bool Write(const char *filename);
46 private:
47   unsigned long BufferLength;
48 };
49
50 } // end namespace gdcm
51
52 #endif //GDCMPNMCODEC_H
```

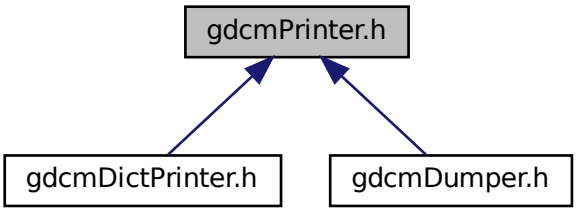
11.385 gdcmPrinter.h File Reference

```
#include "gdcmFile.h"
#include "gdcmDataElement.h"
```

Include dependency graph for gdcmPrinter.h:



This graph shows which files directly or indirectly include this file:



Classes

- class gdcm::Printer

  Printer class.

Namespaces

- namespace gdcm

11.386 gdcmPrinter.h

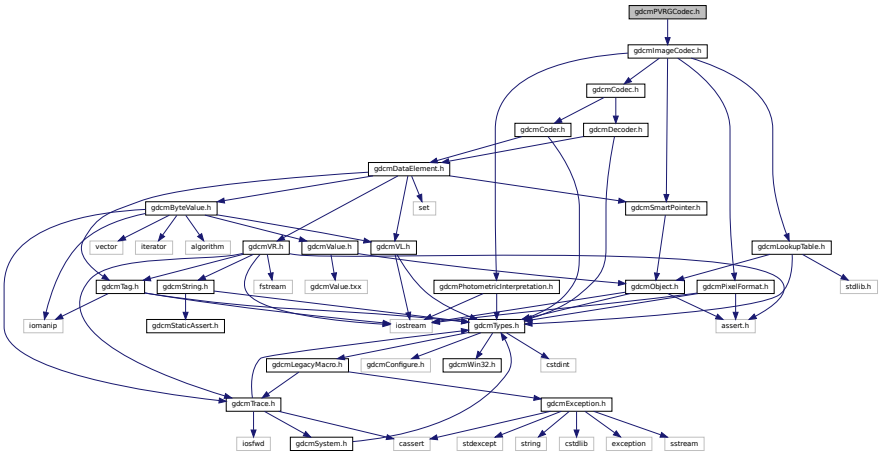
Go to the documentation of this file.

```
1 /*=========================================================================
2
3 Program:  GDCM (Grassroots DICOM). A DICOM library
4
5 Copyright (c) 2006-2011 Mathieu Malaterre
6 All rights reserved.
7 See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.
8
9 This software is distributed WITHOUT ANY WARRANTY; without even
10 the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
11 PURPOSE.  See the above copyright notice for more information.
12
13 =========================================================================*/
14 #ifndef GDCMPRINTER_H
15 #define GDCMPRINTER_H
16
17 // TODO Class to implement printing
18 // Since DICOM does printing ?
19 // Also I would like to encapsulate the IsCharacterPrintable thing
20 // (to avoid printing \0 and other weird characters)
21 // \todo I still need to implement skipping of group (shadow)
22 // need to implement longer field to read
23
24 /*
25 * Output:
26 * For ASCII:
27 * Typically will look like:
28 * [ORIGINAL\PRIMARY\OTHER]
29 * If a non printable character is found:  RED and INVERSE is used:
30 * [                .]
31 *
32 * when the VR is not found (file or dict), we check if we can print the output:
33 * on success ASCII mode is used, on failure the output is printed a series of bytes
34 *
35 * Special case when the data element is empty:
36 * INVERSE « (no value)
37 *
38 * retired public element are printed in red and underline
39 * unknown private element are printed in RED followed by 'UNKNOWN'
40 *
41 * Correct VR is printed in green just after the found VR
42 *
43 * length of data element is printed in bytes, followed by the VM, a green VM is appended
44 * if this is not compatible
45 */
46 #include "gdcmFile.h"
47 #include "gdcmDataElement.h"
48
49 namespace gdcm
50 {
51
52 class DataSet;
53 class DictEntry;
54 class Dicts;
58 // It's a sink there is no output
59 class GDCM_EXPORT Printer
60 {
61 public:
62   Printer();
63   ~Printer();
64
66   void SetFile(File const &f) { F = &f; }
67
69   void SetColor(bool c);
70
71   typedef enum {
72     VERBOSE_STYLE = 0, // GDCM Legacy VERBOSE one
73     CONDENSED_STYLE, //
74     // Ok I am missing voc here ...better naming would be nice
75     XML, //
76     CXX
77   } PrintStyles;
78
80   void SetStyle(PrintStyles ps) {
81     PrintStyle = ps;
82   }
```

```
84   PrintStyles GetPrintStyle()const {
85     return PrintStyle;
86     }
87
89   void Print(std::ostream& os);
90
92   void PrintDataSet(const DataSet &ds, std::ostream& os, const std::string &s = "");
93
94 protected:
95   VR PrintDataElement(std::ostringstream & os, const Dicts &dicts, const DataSet & ds, const DataElement &de,
      std::ostream &out, std::string const & indent );
96 void PrintSQ(const SequenceOfItems *sqi, std::ostream & os, std::string const & indent);
97
98   PrintStyles PrintStyle;
99   const File *F;
100   VL MaxPrintLength;
101 };
102
103 } // end namespace gdcm
104
105 #endif //GDCMPRINTER_H
```

11.387 gdcmPVRGCodec.h File Reference

`#include "gdcmImageCodec.h"`
Include dependency graph for gdcmPVRGCodec.h:



Classes

- class gdcm::PVRGCodec
  PVRGCodec.

Namespaces

- namespace gdcm

11.388 gdcmPVRGCodec.h

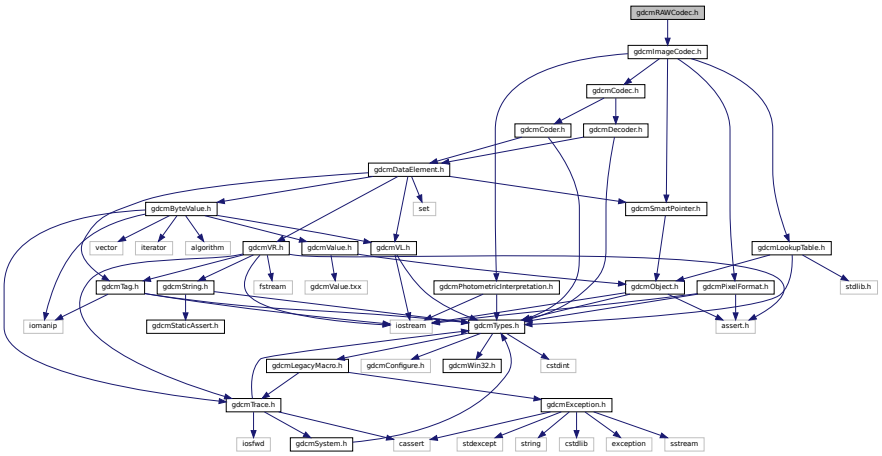
Go to the documentation of this file.

```
1 /*=========================================================================
2
3 Program:  GDCM (Grassroots DICOM). A DICOM library
4
5 Copyright (c) 2006-2011 Mathieu Malaterre
6 All rights reserved.
7 See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.
8
9 This software is distributed WITHOUT ANY WARRANTY; without even
10 the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
11 PURPOSE.  See the above copyright notice for more information.
12
13 =========================================================================*/
14 #ifndef GDCMPVRGCODEC_H
15 #define GDCMPVRGCODEC_H
16
17 #include "gdcmImageCodec.h"
18
19 namespace gdcm
20 {
21
35 class PVRGCodec :  public ImageCodec
36 {
37 public:
38   PVRGCodec();
39   ~PVRGCodec() override;
40   bool CanDecode(TransferSyntax const &ts) const override;
41   bool CanCode(TransferSyntax const &ts) const override;
42
43   bool Decode(DataElement const &is, DataElement &os) override;
44   bool Code(DataElement const &in, DataElement &out) override;
45   void SetLossyFlag( bool l );
46
47   ImageCodec * Clone() const override;
48 private:
49 };
50
51 } // end namespace gdcm
52
53 #endif //GDCMPVRGCODEC_H
```

11.389 gdcmRAWCodec.h File Reference

`#include "gdcmImageCodec.h"`

Include dependency graph for gdcmRAWCodec.h:



Classes

- class gdcm::RAWCodec

  RAWCodec class.

Namespaces

- namespace gdcm

11.390 gdcmRAWCodec.h

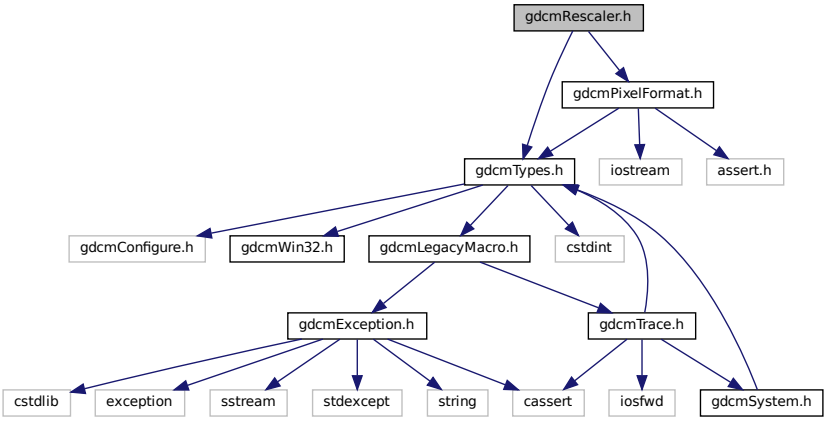
Go to the documentation of this file.

```
1 /*=========================================================================
2
3 Program:  GDCM (Grassroots DICOM). A DICOM library
4
5 Copyright (c) 2006-2011 Mathieu Malaterre
6 All rights reserved.
7 See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.
8
9 This software is distributed WITHOUT ANY WARRANTY; without even
10 the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
11 PURPOSE.  See the above copyright notice for more information.
12
13 =========================================================================*/
14 #ifndef GDCMRAWCODEC_H
15 #define GDCMRAWCODEC_H
16
17 #include "gdcmImageCodec.h"
18
19 namespace gdcm
20 {
21
22 class RAWInternals;
26 class GDCM_EXPORT RAWCodec :  public ImageCodec
27 {
28 public:
29   RAWCodec();
30   ~RAWCodec() override;
31   bool CanCode(TransferSyntax const &ts) const override;
32   bool CanDecode(TransferSyntax const &ts) const override;
33   bool Decode(DataElement const &is, DataElement &os) override;
34   bool Code(DataElement const &in, DataElement &out) override;
35
36   bool GetHeaderInfo(std::istream &is, TransferSyntax &ts) override;
37   ImageCodec * Clone() const override;
38
41   bool DecodeBytes(const char* inBytes, size_t inBufferLength,
42     char* outBytes, size_t inOutBufferLength);
43
44 protected:
45   bool DecodeByStreams(std::istream &is, std::ostream &os) override;
46
47 private:
48   RAWInternals *Internals;
49 };
50
51 } // end namespace gdcm
52
53 #endif // GDCMRAWCODEC_H
```

11.391 gdcmRescaler.h File Reference

```
#include "gdcmTypes.h"
#include "gdcmPixelFormat.h"
```

Include dependency graph for gdcmRescaler.h:



Classes

- class gdcm::Rescaler

  Rescale class.

Namespaces

- namespace gdcm

11.392 gdcmRescaler.h

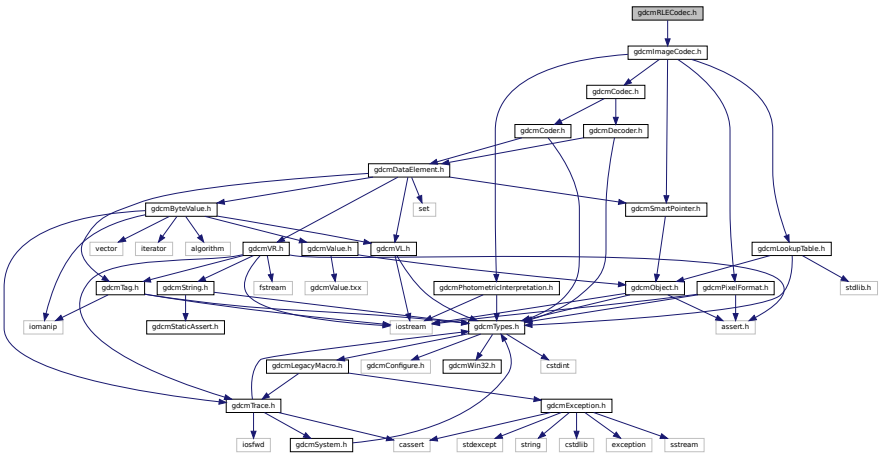
Go to the documentation of this file.

```
1 /*=========================================================================
2
3 Program:  GDCM (Grassroots DICOM). A DICOM library
4
5 Copyright (c) 2006-2011 Mathieu Malaterre
6 All rights reserved.
7 See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.
8
9 This software is distributed WITHOUT ANY WARRANTY; without even
10 the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
11 PURPOSE.  See the above copyright notice for more information.
12
13 =========================================================================*/
14 #ifndef GDCMRESCALER_H
15 #define GDCMRESCALER_H
16
17 #include "gdcmTypes.h"
```

```
18 #include "gdcmPixelFormat.h"
19
20 namespace gdcm
21 {
22
68 class GDCM_EXPORT Rescaler
69 {
70 public:
71   Rescaler():Intercept(0),Slope(1),PF(PixelFormat::UNKNOWN),TargetScalarType(PixelFormat::UNKNOWN),
       ScalarRangeMin(0), ScalarRangeMax(0), UseTargetPixelType(false) {}
72   ~Rescaler() = default;
73
75   bool Rescale(char *out, const char *in, size_t n);
76
78   bool InverseRescale(char *out, const char *in, size_t n);
79
81   void SetIntercept(double i) { Intercept = i; }
82   double GetIntercept()const { return Intercept; }
83
85   void SetSlope(double s) { Slope = s; }
86   double GetSlope()const { return Slope; }
87
92   void SetTargetPixelType( PixelFormat const & targetst );
93
95   void SetUseTargetPixelType(bool b);
96
98   void SetPixelFormat(PixelFormat const & pf) { PF = pf; }
99
102    PixelFormat::ScalarType ComputeInterceptSlopePixelType();
103
106    void SetMinMaxForPixelType(double min, double max);
107
110    PixelFormat ComputePixelTypeFromMinMax();
111
112 protected:
113   template <typename TIn>
114     void RescaleFunctionIntoBestFit(char *out, const TIn *in, size_t n);
115   template <typename TIn>
116     void InverseRescaleFunctionIntoBestFit(char *out, const TIn *in, size_t n);
117
118 private:
119   double Intercept; // 0028,1052
120   double Slope;     // 0028,1053
121   PixelFormat PF;
122   PixelFormat::ScalarType TargetScalarType;
123   double ScalarRangeMin;
124   double ScalarRangeMax;
125   bool UseTargetPixelType;
126 };
127
128 } // end namespace gdcm
129
130 #endif //GDCMRESCALER_H
```

11.393 gdcmRLECodec.h File Reference

`#include "gdcmImageCodec.h"`
Include dependency graph for gdcmRLECodec.h:



Classes

- class gdcm::RLECodec

  Class to do RLE.

Namespaces

- namespace gdcm

11.394 gdcmRLECodec.h

Go to the documentation of this file.

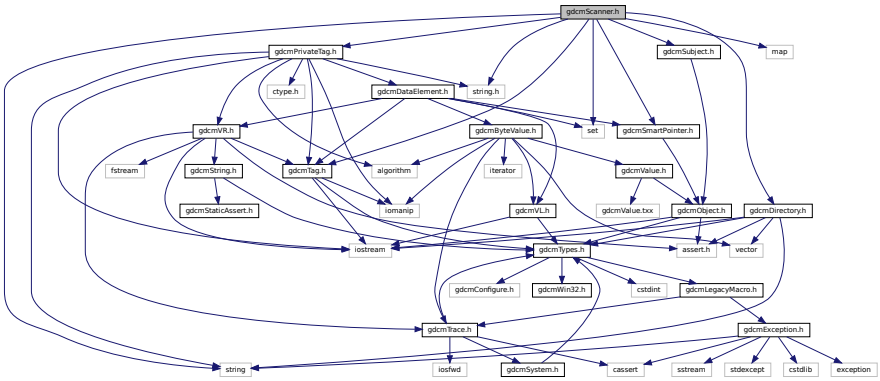
```
1 /*=========================================================================
2
3 Program:  GDCM (Grassroots DICOM). A DICOM library
4
5 Copyright (c) 2006-2011 Mathieu Malaterre
6 All rights reserved.
7 See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.
8
9 This software is distributed WITHOUT ANY WARRANTY; without even
10 the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
11 PURPOSE.  See the above copyright notice for more information.
12
13 =========================================================================*/
14 #ifndef GDCMRLECODEC_H
15 #define GDCMRLECODEC_H
16
17 #include "gdcmImageCodec.h"
18
19 namespace gdcm
20 {
```

```
21
22 class Fragment;
23 class RLEInternals;
36 class GDCM_EXPORT RLECodec :  public ImageCodec
37 {
38 friend class ImageRegionReader;
39 public:
40   RLECodec();
41   ~RLECodec() override;
42   bool CanCode(TransferSyntax const &ts) const override;
43   bool CanDecode(TransferSyntax const &ts) const override;
44   bool Decode(DataElement const &is, DataElement &os) override;
45   unsigned long GetBufferLength()const { return BufferLength; }
46   void SetBufferLength(unsigned long l) { BufferLength = l; }
47
48   bool Code(DataElement const &in, DataElement &out) override;
49   bool GetHeaderInfo(std::istream &is, TransferSyntax &ts) override;
50   ImageCodec * Clone() const override;
51
52 protected:
53   bool DecodeExtent(
54     char *buffer,
55     unsigned int XMin, unsigned int XMax,
56     unsigned int YMin, unsigned int YMax,
57     unsigned int ZMin, unsigned int ZMax,
58     std::istream & is
59   );
60
61   bool DecodeByStreams(std::istream &is, std::ostream &os) override;
62 public:
63
64   void SetLength(unsigned long l)
65     {
66     Length = l;
67     }
68
69 protected:
70   bool StartEncode( std::ostream & ) override;
71   bool IsRowEncoder() override;
72   bool IsFrameEncoder() override;
73   bool AppendRowEncode( std::ostream & out, const char * data, size_t datalen ) override;
74   bool AppendFrameEncode( std::ostream & out, const char * data, size_t datalen ) override;
75   bool StopEncode( std::ostream & ) override;
76
77 private:
78   bool DecodeByStreamsCommon(std::istream &is, std::ostream &os);
79   RLEInternals *Internals;
80   unsigned long Length;
81   unsigned long BufferLength;
82   size_t DecodeFragment(Fragment const & frag, char *buffer, size_t llen);
83 };
84
85 } // end namespace gdcm
86
87 #endif //GDCMRLECODEC_H
```

11.395 gdcmScanner.h File Reference

```
#include "gdcmDirectory.h"
#include "gdcmSubject.h"
#include "gdcmTag.h"
#include "gdcmPrivateTag.h"
#include "gdcmSmartPointer.h"
#include <map>
#include <set>
#include <string>
#include <string.h>
```

Include dependency graph for gdcmScanner.h:



Classes

- struct gdcm::Scanner::ltstr
- class gdcm::Scanner

  Scanner.

Namespaces

- namespace gdcm

Functions

- std::ostream & gdcm::operator<< (std::ostream &os, const Scanner &s)

11.396 gdcmScanner.h

Go to the documentation of this file.

```
1 /*=========================================================================
2
3 Program:  GDCM (Grassroots DICOM). A DICOM library
4
5 Copyright (c) 2006-2011 Mathieu Malaterre
6 All rights reserved.
7 See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.
8
9 This software is distributed WITHOUT ANY WARRANTY; without even
10 the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
11 PURPOSE.  See the above copyright notice for more information.
12
13 =========================================================================*/
14 #ifndef GDCMSCANNER_H
15 #define GDCMSCANNER_H
16
17 #include "gdcmDirectory.h"
18 #include "gdcmSubject.h"
19 #include "gdcmTag.h"
```

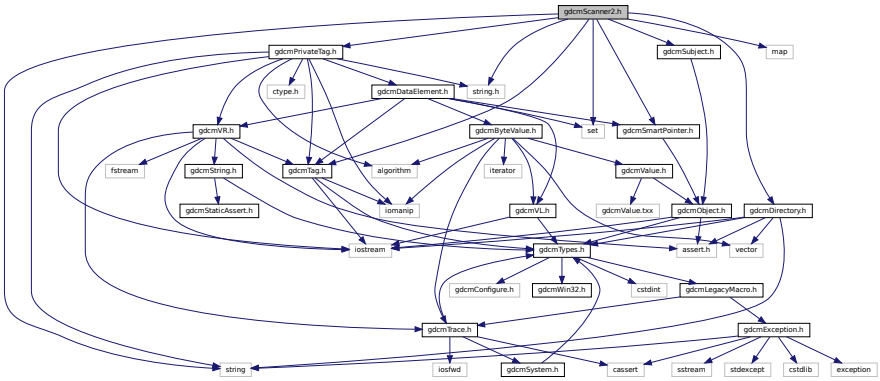
```
20 #include "gdcmPrivateTag.h"
21 #include "gdcmSmartPointer.h"
22
23 #include <map>
24 #include <set>
25 #include <string>
26
27 #include <string.h> // strcmp
28
29 namespace gdcm
30 {
31 class StringFilter;
32
55 class GDCM_EXPORT Scanner :  public Subject
56 {
57   friend std::ostream& operator«(std::ostream &_os, const Scanner &s);
58 public:
59   Scanner():Values(),Filenames(),Mappings() {}
60   ~Scanner() override;
61
68   typedef std::map<Tag, const char*> TagToValue;
69   //typedef std::map<Tag, ConstCharWrapper> TagToValue; //StringMap;
70   //typedef TagToStringMap TagToValue;
71   typedef TagToValue::value_type TagToValueValueType;
72
74   void AddTag( Tag const & t );
75   void ClearTags();
76
77   // Work in progress do not use:
78   void AddPrivateTag( PrivateTag const & t );
79
81   void AddSkipTag( Tag const & t );
82   void ClearSkipTags();
83
85   bool Scan( Directory::FilenamesType const & filenames );
86
87   Directory::FilenamesType const &GetFilenames()const { return Filenames; }
88
90   void Print( std::ostream & os ) const override;
91
92   void PrintTable( std::ostream & os ) const;
93
97   bool IsKey( const char * filename ) const;
98
101    Directory::FilenamesType GetKeys() const;
102
103   // struct to store all the values found:
104   typedef std::set< std::string > ValuesType;
105
107   ValuesType const & GetValues()const { return Values; }
108
110   ValuesType GetValues(Tag const &t) const;
111
115   Directory::FilenamesType GetOrderedValues(Tag const &t) const;
116
117   /* ltstr is CRITICAL, otherwise pointers value are used to do the key comparison */
118   struct ltstr
119     {
120     bool operator()(const char* s1, const char* s2)const
121 {
122       assert( s1 && s2 );
123       return strcmp(s1, s2) < 0;
124       }
125     };
126   typedef std::map<const char *,TagToValue, ltstr> MappingType;
127   typedef MappingType::const_iterator ConstIterator;
128   ConstIterator Begin()const { return Mappings.begin(); }
129   ConstIterator End()const { return Mappings.end(); }
130
132   MappingType const & GetMappings()const { return Mappings; }
133
135   TagToValue const & GetMapping(const char *filename) const;
136
139   const char *GetFilenameFromTagToValue(Tag const &t, const char *valueref) const;
140
143   Directory::FilenamesType GetAllFilenamesFromTagToValue(Tag const &t, const char *valueref) const;
144
146   // by a call to GetMapping()
147   TagToValue const & GetMappingFromTagToValue(Tag const &t, const char *value) const;
148
154   const char* GetValue(const char *filename, Tag const &t) const;
```

```
155
157   static SmartPointer<Scanner> New() { return new Scanner; }
158
159 protected:
160   void ProcessPublicTag(StringFilter &sf, const char *filename);
161 private:
162   // struct to store all uniq tags in ascending order:
163   typedef std::set< Tag > TagsType;
164   typedef std::set< PrivateTag > PrivateTagsType;
165   std::set< Tag > Tags;
166   std::set< PrivateTag > PrivateTags;
167   std::set< Tag > SkipTags;
168   ValuesType Values;
169   Directory::FilenamesType Filenames;
170
171   // Main struct that will hold all mapping:
172   MappingType Mappings;
173
174   double Progress;
175 };
176 //-----------------------------------------------------------------------------
177 inline std::ostream& operator<<(std::ostream &os, const Scanner &s)
178 {
179   s.Print( os );
180   return os;
181 }
182
183 #if defined(SWIGPYTHON) || defined(SWIGCSHARP) || defined(SWIGJAVA) || defined(SWIGPHP)
184 /*
185 * HACK: I need this temp class to be able to manipulate a std::map from python,
186 * swig does not support wrapping of simple class like std::map...
187 */
188 class SWIGTagToValue
189 {
190 public:
191   SWIGTagToValue(Scanner::TagToValue const &t2v):Internal(t2v),it(t2v.begin()) {}
192   const Scanner::TagToValueValueType& GetCurrent()const { return *it; }
193   const Tag& GetCurrentTag()const { return it->first; }
194   const char *GetCurrentValue()const { return it->second; }
195   void Start() { it = Internal.begin(); }
196   bool IsAtEnd()const { return it == Internal.end(); }
197   void Next() { ++it; }
198 private:
199   const Scanner::TagToValue& Internal;
200   Scanner::TagToValue::const_iterator it;
201 };
202 #endif /* SWIG */
203
209 } // end namespace gdcm
210
211 #endif //GDCMSCANNER_H
```

11.397 gdcmScanner2.h File Reference

```
#include "gdcmDirectory.h"
#include "gdcmSubject.h"
#include "gdcmTag.h"
#include "gdcmPrivateTag.h"
#include "gdcmSmartPointer.h"
#include <map>
#include <set>
#include <string>
#include <string.h>
```

Include dependency graph for gdcmScanner2.h:



Classes

- struct gdcm::Scanner2::ltstr
- class gdcm::Scanner2

  Scanner2.

Namespaces

- namespace gdcm

Functions

- std::ostream & gdcm::operator<< (std::ostream &os, const Scanner2 &s)

11.398 gdcmScanner2.h

Go to the documentation of this file.

```
1 /*=========================================================================
2
3 Program:  GDCM (Grassroots DICOM). A DICOM library
4
5 Copyright (c) 2006-2011 Mathieu Malaterre
6 All rights reserved.
7 See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.
8
9 This software is distributed WITHOUT ANY WARRANTY; without even
10 the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
11 PURPOSE.  See the above copyright notice for more information.
12
13 =========================================================================*/
14 #ifndef GDCMSCANNER2_H
15 #define GDCMSCANNER2_H
16
17 #include "gdcmDirectory.h"
18 #include "gdcmSubject.h"
19 #include "gdcmTag.h"
```

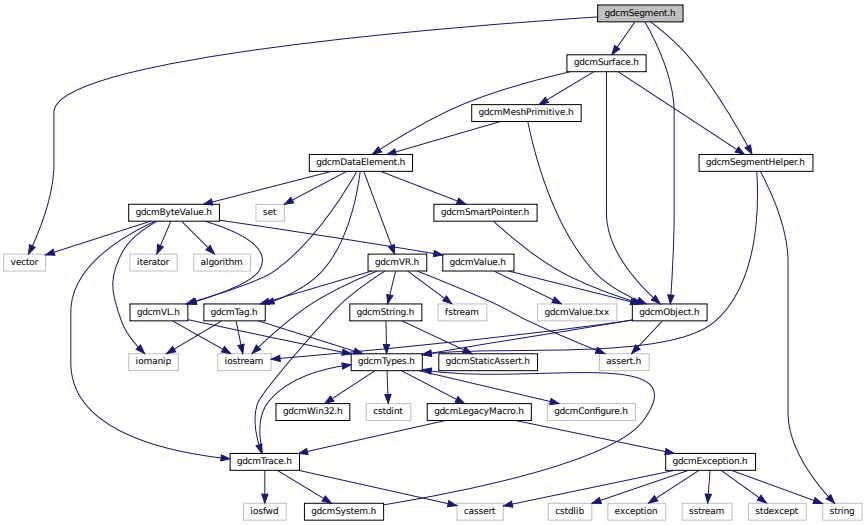
```
20 #include "gdcmPrivateTag.h"
21 #include "gdcmSmartPointer.h"
22
23 #include <map>
24 #include <set>
25 #include <string>
26
27 #include <string.h> // strcmp
28
29 namespace gdcm
30 {
31 class StringFilter;
32
55 class GDCM_EXPORT Scanner2 :  public Subject
56 {
57   friend std::ostream& operator<<(std::ostream &_os, const Scanner2 &s);
58 public:
59   Scanner2():Values(),Filenames(),PublicMappings(),PrivateMappings() {}
60   ~Scanner2() override;
61
68   typedef std::map<Tag, const char*> PublicTagToValue;
69   typedef PublicTagToValue::value_type PublicTagToValueValueType;
70
71   typedef std::map<PrivateTag, const char*> PrivateTagToValue;
72   typedef PrivateTagToValue::value_type PrivateTagToValueValueType;
73
75   bool AddPublicTag( Tag const & t );
76   void ClearPublicTags();
77
78   // Work in progress do not use:
79   bool AddPrivateTag( PrivateTag const & pt );
80   void ClearPrivateTags();
81
83   bool AddSkipTag( Tag const & t );
84   void ClearSkipTags();
85
87   bool Scan( Directory::FilenamesType const & filenames );
88
90   Directory::FilenamesType const &GetFilenames()const { return Filenames; }
91
93   void Print( std::ostream & os ) const override;
94
96   void PrintTable( std::ostream & os, bool header = false ) const;
97
101   bool IsKey( const char * filename ) const;
102
105   Directory::FilenamesType GetKeys() const;
106
107   // struct to store all the values found:
108   typedef std::set< std::string > ValuesType;
109
111   ValuesType const & GetValues()const { return Values; }
112
114   ValuesType GetPublicValues(Tag const &t) const;
115
117   ValuesType GetPrivateValues(PrivateTag const &pt) const;
118
122   Directory::FilenamesType GetPublicOrderedValues(Tag const &t) const;
123
124   Directory::FilenamesType GetPrivateOrderedValues(PrivateTag const &pt) const;
125
126   /* ltstr is CRITICAL, otherwise pointers value are used to do the key comparison */
127   struct ltstr
128     {
129     bool operator()(const char* s1, const char* s2)const
130 {
131       assert( s1 && s2 );
132       return strcmp(s1, s2) < 0;
133       }
134     };
135   typedef std::map<const char *,PublicTagToValue, ltstr> PublicMappingType;
136   typedef PublicMappingType::const_iterator PublicConstIterator;
137   PublicConstIterator Begin()const { return PublicMappings.begin(); }
138   PublicConstIterator End()const { return PublicMappings.end(); }
139
140   typedef std::map<const char *,PrivateTagToValue, ltstr> PrivateMappingType;
141   typedef PrivateMappingType::const_iterator PrivateConstIterator;
142   PrivateConstIterator PrivateBegin()const { return PrivateMappings.begin(); }
143   PrivateConstIterator PrivateEnd()const { return PrivateMappings.end(); }
144
146   PublicMappingType const & GetPublicMappings()const { return PublicMappings; }
```

```
147   PrivateMappingType const & GetPrivateMappings()const { return PrivateMappings; }
148
150   PublicTagToValue const & GetPublicMapping(const char *filename) const;
151   PrivateTagToValue const & GetPrivateMapping(const char *filename) const;
152
155   const char *GetFilenameFromPublicTagToValue(Tag const &t, const char *valueref) const;
156   const char *GetFilenameFromPrivateTagToValue(PrivateTag const &pt, const char *valueref) const;
157
160   Directory::FilenamesType GetAllFilenamesFromPublicTagToValue(Tag const &t, const char *valueref) const;
161   Directory::FilenamesType GetAllFilenamesFromPrivateTagToValue(PrivateTag const &pt, const char *valueref)
      const;
162
164   // by a call to GetMapping()
165   PublicTagToValue const & GetMappingFromPublicTagToValue(Tag const &t, const char *value) const;
166   PrivateTagToValue const & GetMappingFromPrivateTagToValue(PrivateTag const &pt, const char *value) const;
167
173   const char* GetPublicValue(const char *filename, Tag const &t) const;
174   const char* GetPrivateValue(const char *filename, PrivateTag const &t) const;
175
177   static SmartPointer<Scanner2> New() { return new Scanner2; }
178
179 protected:
180   void ProcessPublicTag(StringFilter &sf, const char *filename);
181   void ProcessPrivateTag(StringFilter &sf, const char *filename);
182 private:
183   // struct to store all uniq tags in ascending order:
184   typedef std::set< Tag > PublicTagsType;
185   typedef std::set< PrivateTag > PrivateTagsType;
186   std::set< Tag > PublicTags; // Public and Private Creator
187   std::set< PrivateTag > PrivateTags; // Only Private (no Private Creator)
188   std::set< Tag > SkipTags;
189   ValuesType Values;
190   Directory::FilenamesType Filenames;
191
192   // Main struct that will hold all public mapping:
193   PublicMappingType PublicMappings;
194   // Main struct that will hold all private mapping:
195   PrivateMappingType PrivateMappings;
196
197   double Progress;
198 };
199 //-----------------------------------------------------------------------------
200 inline std::ostream& operator<<(std::ostream &os, const Scanner2 &s)
201 {
202   s.Print( os );
203   return os;
204 }
205
206 } // end namespace gdcm
207
208 #endif //GDCMSCANNER2_H
```

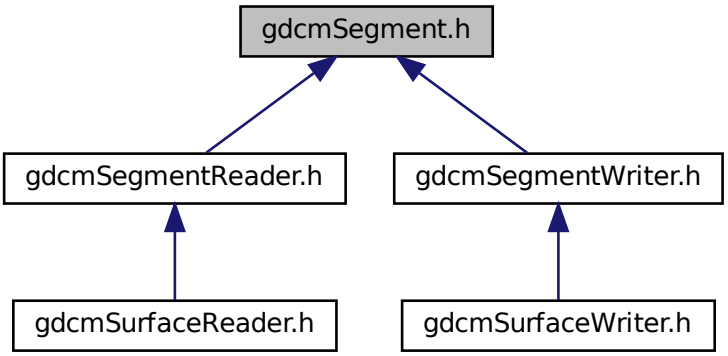
11.399 gdcmSegment.h File Reference

```
#include <vector>
#include <gdcmObject.h>
#include <gdcmSurface.h>
#include "gdcmSegmentHelper.h"
```

Include dependency graph for gdcmSegment.h:



This graph shows which files directly or indirectly include this file:



Classes

- class gdcm::Segment

  This class defines a segment.

Namespaces

- namespace gdcm

11.400 gdcmSegment.h

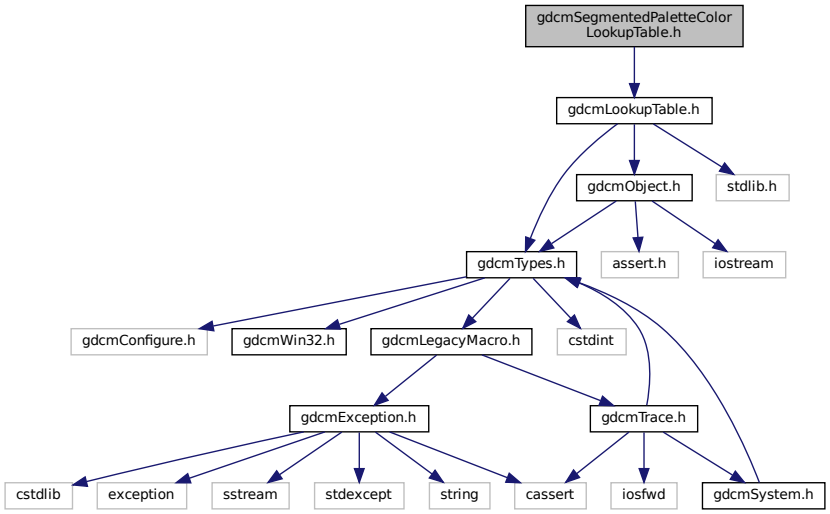
Go to the documentation of this file.

```
1 /*=========================================================================
2
3 Program:  GDCM (Grassroots DICOM). A DICOM library
4
5 Copyright (c) 2006-2011 Mathieu Malaterre
6 All rights reserved.
7 See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.
8
9 This software is distributed WITHOUT ANY WARRANTY; without even
10 the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
11 PURPOSE.  See the above copyright notice for more information.
12
13 =========================================================================*/
14 #ifndef GDCMSEGMENT_H
15 #define GDCMSEGMENT_H
16
17 #include <vector>
18
19 #include <gdcmObject.h>
20 #include <gdcmSurface.h>
21 #include "gdcmSegmentHelper.h"
22
23 namespace gdcm
24 {
25
33 class GDCM_EXPORT Segment :  public Object
34 {
35 public:
36
37   typedef std::vector< SmartPointer< Surface > > SurfaceVector;
38   typedef std::vector< SegmentHelper::BasicCodedEntry > BasicCodedEntryVector;
39
40   typedef enum {
41     AUTOMATIC = 0,
42     SEMIAUTOMATIC,
43     MANUAL,
44     ALGOType_END
45   } ALGOType;
46
47   static const char * GetALGOTypeString(ALGOType type);
48   static ALGOType GetALGOType(const char * type);
49
50
51   Segment();
52
53   ~Segment() override;
54
55   //**        Segment getters/setters     **//
56   unsigned short GetSegmentNumber() const;
57   void SetSegmentNumber(const unsigned short num);
58
59   const char * GetSegmentLabel() const;
60   void SetSegmentLabel(const char * label);
61
62   const char * GetSegmentDescription() const;
63   void SetSegmentDescription(const char * description);
64
65   SegmentHelper::BasicCodedEntry const & GetAnatomicRegion() const;
66   SegmentHelper::BasicCodedEntry & GetAnatomicRegion();
67   void SetAnatomicRegion(SegmentHelper::BasicCodedEntry const & BSE);
68
69   BasicCodedEntryVector const & GetAnatomicRegionModifiers() const;
70   BasicCodedEntryVector & GetAnatomicRegionModifiers();
71   void SetAnatomicRegionModifiers(BasicCodedEntryVector const & BSEV);
72
73   SegmentHelper::BasicCodedEntry const & GetPropertyCategory() const;
74   SegmentHelper::BasicCodedEntry & GetPropertyCategory();
75   void SetPropertyCategory(SegmentHelper::BasicCodedEntry const & BSE);
76
77   SegmentHelper::BasicCodedEntry const & GetPropertyType() const;
78   SegmentHelper::BasicCodedEntry & GetPropertyType();
79   void SetPropertyType(SegmentHelper::BasicCodedEntry const & BSE);
80
81   BasicCodedEntryVector const & GetPropertyTypeModifiers() const;
82   BasicCodedEntryVector & GetPropertyTypeModifiers();
83   void SetPropertyTypeModifiers(BasicCodedEntryVector const & BSEV);
```

```
84
85   ALGOType GetSegmentAlgorithmType() const;
86   void SetSegmentAlgorithmType(ALGOType type);
87   void SetSegmentAlgorithmType(const char * typeStr);
88
89   const char * GetSegmentAlgorithmName() const;
90   void SetSegmentAlgorithmName(const char * name);
91
92   //**        Surface getters/setters     **//
93   unsigned long GetSurfaceCount();
94   void SetSurfaceCount(const unsigned long nb);
95
96   SurfaceVector const & GetSurfaces() const;
97   SurfaceVector & GetSurfaces();
98
99   SmartPointer< Surface > GetSurface(const unsigned int idx = 0) const;
100
101   void AddSurface(SmartPointer< Surface > surface);
102
103 protected :
104   //**        Segment members     **//
105   //0062 0004 US 1 Segment Number
106   unsigned short  SegmentNumber;
107   //0062 0005 LO 1 Segment Label
108   std::string     SegmentLabel;
109   //0062 0006 ST 1 Segment Description
110   std::string     SegmentDescription;
111
112   // General Anatomic Region
113   SegmentHelper::BasicCodedEntry AnatomicRegion;
114   // General Anatomic Region Modifier
115   BasicCodedEntryVector AnatomicRegionModifiers;
116   // Property Category Code
117   SegmentHelper::BasicCodedEntry PropertyCategory;
118   // Property Type Code
119   SegmentHelper::BasicCodedEntry PropertyType;
120   // Property Type Modifier Code
121   BasicCodedEntryVector PropertyTypeModifiers;
122
123   //0062 0008 CS 1 Segment Algorithm Type
124   ALGOType        SegmentAlgorithmType;
125   //0062 0009 LO 1 Segment Algorithm Name
126   std::string     SegmentAlgorithmName;
127
128   //**        Surface members     **//
129   //0066 002a UL 1 Surface Count
130   unsigned long   SurfaceCount;
131
132   SurfaceVector   Surfaces;
133
134 private :
135   void ComputeSurfaceCount();
136 };
137
138 }
139
140 #endif // GDCMSEGMENT_H
```

11.401 gdcmSegmentedPaletteColorLookupTable.h File Reference

`#include "gdcmLookupTable.h"`
Include dependency graph for gdcmSegmentedPaletteColorLookupTable.h:



Classes

- class gdcm::SegmentedPaletteColorLookupTable

  SegmentedPaletteColorLookupTable class.

Namespaces

- namespace gdcm

11.402 gdcmSegmentedPaletteColorLookupTable.h

Go to the documentation of this file.

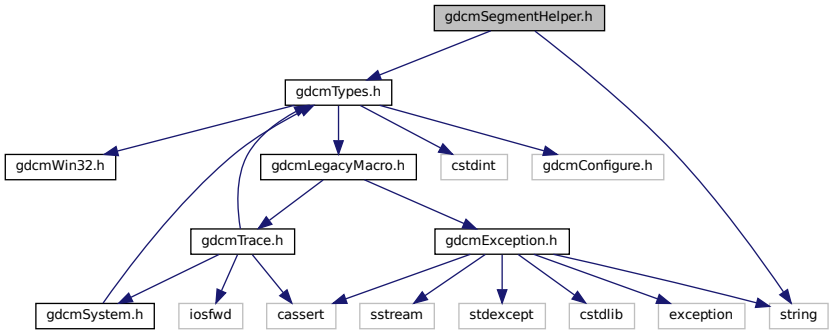
```
1 /*=========================================================================
2
3 Program:  GDCM (Grassroots DICOM). A DICOM library
4
5 Copyright (c) 2006-2011 Mathieu Malaterre
6 All rights reserved.
7 See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.
8
9 This software is distributed WITHOUT ANY WARRANTY; without even
10 the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
11 PURPOSE.  See the above copyright notice for more information.
12
13 =========================================================================*/
14
```

```
15 #ifndef GDCMSEGMENTEDPALETTECOLORLOOKUPTABLE_H
16 #define GDCMSEGMENTEDPALETTECOLORLOOKUPTABLE_H
17
18 #include "gdcmLookupTable.h"
19
20 namespace gdcm
21 {
22
26 class GDCM_EXPORT SegmentedPaletteColorLookupTable :  public LookupTable
27 {
28 public:
29   SegmentedPaletteColorLookupTable();
30   ~SegmentedPaletteColorLookupTable() override;
31   void Print(std::ostream &)const override {}
32
34   void SetLUT(LookupTableType type, const unsigned char *array,
35     unsigned int length) override;
36
37 };
38
39 } // end namespace gdcm
40
41 #endif //GDCMSEGMENTEDPALETTECOLORLOOKUPTABLE_H
```

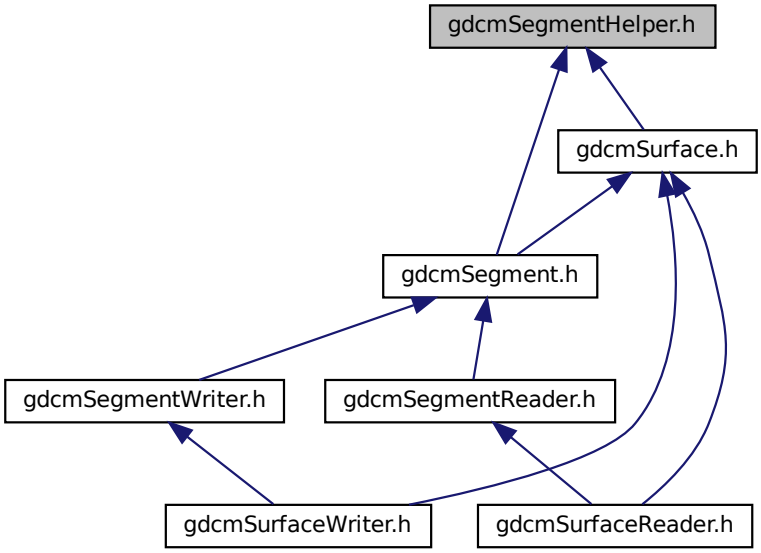
11.403 gdcmSegmentHelper.h File Reference

```
#include "gdcmTypes.h"
#include <string>
```

Include dependency graph for gdcmSegmentHelper.h:



This graph shows which files directly or indirectly include this file:



Classes

- struct gdcm::SegmentHelper::BasicCodedEntry

  This structure defines a basic coded entry with all of its attributes.

Namespaces

- namespace gdcm
- namespace gdcm::SegmentHelper

11.404 gdcmSegmentHelper.h

Go to the documentation of this file.

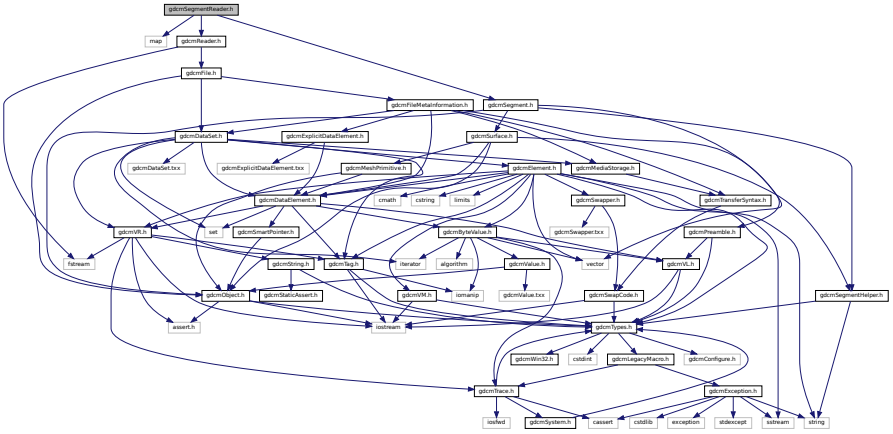
```
1 /*=========================================================================
2
3 Program:  GDCM (Grassroots DICOM). A DICOM library
4
5 Copyright (c) 2006-2011 Mathieu Malaterre
6 All rights reserved.
7 See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.
8
9 This software is distributed WITHOUT ANY WARRANTY; without even
10 the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
11 PURPOSE.  See the above copyright notice for more information.
12
13 =========================================================================*/
```

```
14 #ifndef GDCMSEGMENTHELPER_H
15 #define GDCMSEGMENTHELPER_H
16
17 #include "gdcmTypes.h"
18
19 #include <string>
20
21 namespace gdcm
22 {
23
24 namespace SegmentHelper
25 {
26
32 struct GDCM_EXPORT BasicCodedEntry
33 {
37   BasicCodedEntry():
38     CV(""),
39     CSD(""),
40     CSV(""),
41     CM("")
42   {}
43
47   BasicCodedEntry(const char * a_CV,
48                   const char * a_CSD,
49                   const char * a_CM):
50     CV(a_CV),
51     CSD(a_CSD),
52     CSV(""),
53     CM(a_CM)
54   {}
55
59   BasicCodedEntry(const char * a_CV,
60                   const char * a_CSD,
61                   const char * a_CSV,
62                   const char * a_CM):
63     CV(a_CV),
64     CSD(a_CSD),
65     CSV(a_CSV),
66     CM(a_CM)
67   {}
68
74   bool IsEmpty(const bool checkOptionalAttributes = false) const;
75
76
77   //**      Members     **//
78   // 0008 0100 1   Code Value
79   std::string CV;
80   // 0008 0102 1   Coding Scheme Designator
81   std::string CSD;
82   // 0008 0103 1C  Coding Scheme Version
83   std::string CSV;
84   // 0008 0104 1   Code Meaning
85   std::string CM;
86 };
87
88 } // end of SegmentHelper namespace
89
90 } // end of gdcm namespace
91
92 #endif // GDCMSEGMENTHELPER_H
```

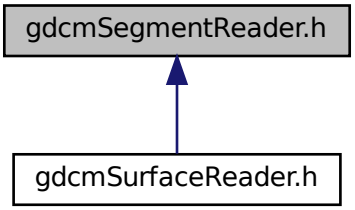
11.405 gdcmSegmentReader.h File Reference

```
#include <map>
#include <gdcmReader.h>
#include <gdcmSegment.h>
```

Include dependency graph for gdcmSegmentReader.h:



This graph shows which files directly or indirectly include this file:



Classes

- class gdcm::SegmentReader

  This class defines a segment reader.

Namespaces

- namespace gdcm

11.406 gdcmSegmentReader.h

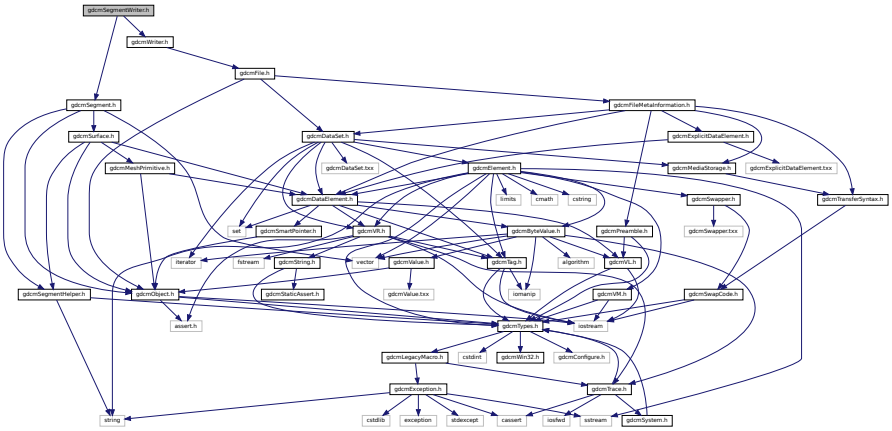
Go to the documentation of this file.

```
1 /*=========================================================================
2
3 Program:  GDCM (Grassroots DICOM). A DICOM library
4
5 Copyright (c) 2006-2011 Mathieu Malaterre
6 All rights reserved.
7 See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.
8
9 This software is distributed WITHOUT ANY WARRANTY; without even
10 the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
11 PURPOSE.  See the above copyright notice for more information.
12
13 =========================================================================*/
14 #ifndef GDCMSEGMENTREADER_H
15 #define GDCMSEGMENTREADER_H
16
17 #include <map>
18
19 #include <gdcmReader.h>
20 #include <gdcmSegment.h>
21
22 namespace gdcm
23 {
24
31 class GDCM_EXPORT SegmentReader :  public Reader
32 {
33 public:
34   typedef std::vector< SmartPointer< Segment > > SegmentVector;
35
36   SegmentReader();
37
38   ~SegmentReader() override;
39
41   bool Read() override; // Set to protected ?
42
43   //**        Segment getters/setters     **//
44   const SegmentVector GetSegments() const;
45   SegmentVector GetSegments();
46
47 //  unsigned int GetNumberOfSegments();
48
49 protected:
50
51   typedef std::map< unsigned long, SmartPointer< Segment > > SegmentMap;
52
53   bool ReadSegments();
54
55   bool ReadSegment(const Item & segmentItem, const unsigned int idx);
56
57
58   SegmentMap Segments;  // The key value is item number (in segment sequence)
59                         // or the surface number (for a surface segmentation).
60
61 };
62
63 }
64
65 #endif // GDCMSEGMENTREADER_H
```

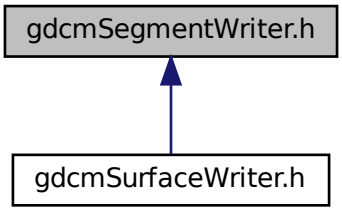
11.407 gdcmSegmentWriter.h File Reference

```
#include <gdcmWriter.h>
#include <gdcmSegment.h>
```

Include dependency graph for gdcmSegmentWriter.h:



This graph shows which files directly or indirectly include this file:



Classes

- class gdcm::SegmentWriter

  This class defines a segment writer.

Namespaces

- namespace gdcm

11.408 gdcmSegmentWriter.h

Go to the documentation of this file.

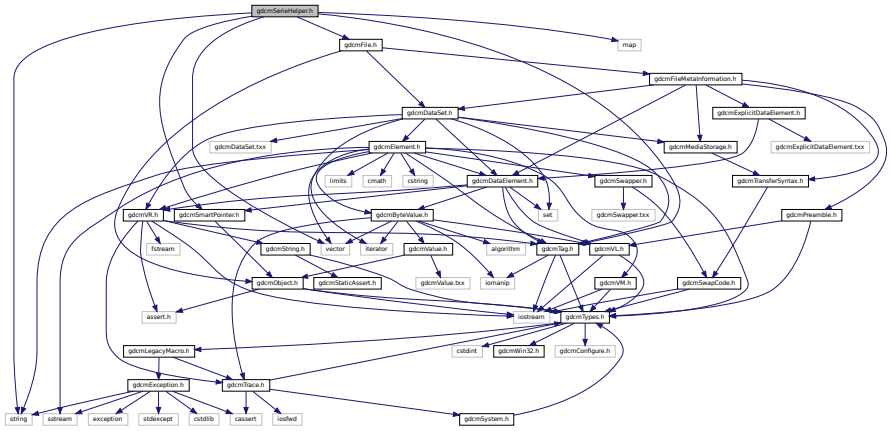
```
1 /*=========================================================================
2
3 Program:  GDCM (Grassroots DICOM). A DICOM library
4
5 Copyright (c) 2006-2011 Mathieu Malaterre
6 All rights reserved.
7 See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.
8
9 This software is distributed WITHOUT ANY WARRANTY; without even
10 the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
11 PURPOSE.  See the above copyright notice for more information.
12
13 =========================================================================*/
14 #ifndef GDCMSEGMENTWRITER_H
15 #define GDCMSEGMENTWRITER_H
16
17 #include <gdcmWriter.h>
18 #include <gdcmSegment.h>
19
20 namespace gdcm
21 {
22
29 class GDCM_EXPORT SegmentWriter :  public Writer
30 {
31 public:
32   typedef std::vector< SmartPointer< Segment > > SegmentVector;
33
34   SegmentWriter();
35
36   ~SegmentWriter() override;
37
39   bool Write() override; // Set to protected ?
40
41   //**        Segment getters/setters     **//
42   unsigned int GetNumberOfSegments() const;
43   void SetNumberOfSegments(const unsigned int size);
44
45   const SegmentVector & GetSegments() const;
46   SegmentVector & GetSegments();
47   SmartPointer< Segment > GetSegment(const unsigned int idx = 0) const;
48
49   void AddSegment(SmartPointer< Segment > segment);
50
51   void SetSegments(SegmentVector & segments);
52
53 protected:
54
55   bool PrepareWrite();
56
57
58   SegmentVector Segments;
59 };
60
61 }
62
63 #endif // GDCMSEGMENTWRITER_H
```

11.409 gdcmSerieHelper.h File Reference

```
#include "gdcmTag.h"
#include "gdcmSmartPointer.h"
#include "gdcmFile.h"
#include <vector>
#include <string>
```

```
#include <map>
```

Include dependency graph for gdcmSerieHelper.h:



Classes

- class gdcm::FileWithName

  FileWithName.
- class gdcm::SerieHelper

  SerieHelper DO NOT USE this class, it is only a temporary solution for ITK migration from GDCM 1.x to GDCM 2.x It will disappear soon, you've been warned.

Namespaces

- namespace gdcm

Typedefs

- typedef bool(∗ gdcm::BOOL_FUNCTION_PFILE_PFILE_POINTER) (File ∗, File ∗)
- typedef std::vector< SmartPointer< FileWithName > > gdcm::FileList

Enumerations

- enum gdcm::CompOperators {
  gdcm::GDCM_EQUAL = 0 ,
  gdcm::GDCM_DIFFERENT ,
  gdcm::GDCM_GREATER ,
  gdcm::GDCM_GREATEROREQUAL ,
  gdcm::GDCM_LESS ,
  gdcm::GDCM_LESSOREQUAL }
- enum gdcm::LodModeType {
  gdcm::LD_ALL = 0x00000000 ,
  gdcm::LD_NOSEQ = 0x00000001 ,
  gdcm::LD_NOSHADOW = 0x00000002 ,
  gdcm::LD_NOSHADOWSEQ = 0x00000004 }

11.410 gdcmSerieHelper.h

Go to the documentation of this file.

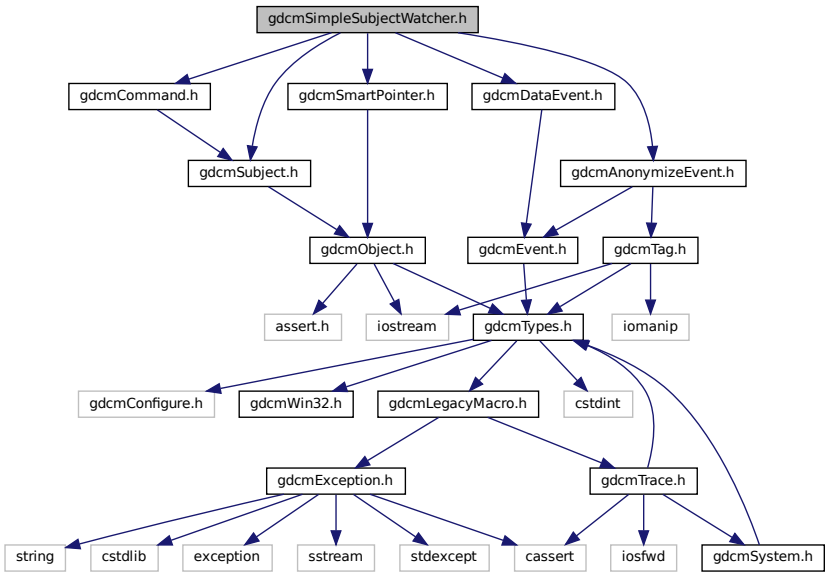
```
1 /*=========================================================================
2
3 Program:  GDCM (Grassroots DICOM). A DICOM library
4
5 Copyright (c) 2006-2011 Mathieu Malaterre
6 All rights reserved.
7 See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.
8
9 This software is distributed WITHOUT ANY WARRANTY; without even
10 the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
11 PURPOSE.  See the above copyright notice for more information.
12
13 =========================================================================*/
14 #ifndef GDCMSERIEHELPER_H
15 #define GDCMSERIEHELPER_H
16
17 #include "gdcmTag.h"
18 #include "gdcmSmartPointer.h"
19 #include "gdcmFile.h"
20 #include <vector>
21 #include <string>
22 #include <map>
23
24 namespace gdcm
25 {
26
27 enum CompOperators {
28    GDCM_EQUAL = 0,
29    GDCM_DIFFERENT,
30    GDCM_GREATER,
31    GDCM_GREATEROREQUAL,
32    GDCM_LESS,
33    GDCM_LESSOREQUAL
34 };
35 enum LodModeType
36 {
37    LD_ALL         = 0x00000000,
38    LD_NOSEQ       = 0x00000001,
39    LD_NOSHADOW    = 0x00000002,
40    LD_NOSHADOWSEQ = 0x00000004
41 };
42
43
50 class GDCM_EXPORT FileWithName :  public File
51 {
52 public:
53   FileWithName(File &f):File(f),filename(){}
54   std::string filename;
55 };
56
57 typedef std::vector< SmartPointer<FileWithName> > FileList;
58 typedef bool (*BOOL_FUNCTION_PFILE_PFILE_POINTER)(File *, File *);
59 class Scanner;
60
68 class GDCM_EXPORT SerieHelper
69 {
70 public:
71   SerieHelper();
72   ~SerieHelper();
73
74   void Clear();
75   void SetLoadMode (int ) {}
76   void SetDirectory(std::string const &dir, bool recursive=false);
77
78   void AddRestriction(const std::string & tag);
79   void SetUseSeriesDetails( bool useSeriesDetails );
80   void CreateDefaultUniqueSeriesIdentifier();
81   FileList *GetFirstSingleSerieUIDFileSet();
82   FileList *GetNextSingleSerieUIDFileSet();
83   std::string CreateUniqueSeriesIdentifier( File * inFile );
84   void OrderFileList(FileList *fileSet);
85   void AddRestriction(uint16_t group, uint16_t elem, std::string const &value, int op);
86
87 protected:
88   bool UserOrdering(FileList *fileSet);
89   void AddFileName(std::string const &filename);
```

```
90   bool AddFile(FileWithName &header);
91   void AddRestriction(const Tag& tag);
92   bool ImagePositionPatientOrdering(FileList *fileSet);
93   bool ImageNumberOrdering( FileList *fileList );
94   bool FileNameOrdering( FileList *fileList );
95
96   using Rule = struct RuleStructure{
97     uint16_t group;
98     uint16_t elem;
99     std::string value;
100     int op;
101   };
102   typedef std::vector<Rule> SerieRestrictions;
103
104   typedef std::map<std::string, FileList *> SingleSerieUIDFileSetmap;
105   SingleSerieUIDFileSetmap SingleSerieUIDFileSetHT;
106   SingleSerieUIDFileSetmap::iterator ItFileSetHt;
107
108 private:
109   SerieRestrictions Restrictions;
110   SerieRestrictions Refine;
111
112   bool UseSeriesDetails;
113   bool DirectOrder;
114
115   BOOL_FUNCTION_PFILE_PFILE_POINTER UserLessThanFunction;
116 };
117
118 // backward compat
119 } // end namespace gdcm
120
121
122 #endif //GDCMSERIEHELPER_H
```

11.411 gdcmSimpleSubjectWatcher.h File Reference

```
#include "gdcmSubject.h"
#include "gdcmCommand.h"
#include "gdcmSmartPointer.h"
#include "gdcmAnonymizeEvent.h"
#include "gdcmDataEvent.h"
```

Include dependency graph for gdcmSimpleSubjectWatcher.h:



Classes

- class gdcm::SimpleSubjectWatcher

  SimpleSubjectWatcher.

Namespaces

- namespace gdcm

11.412 gdcmSimpleSubjectWatcher.h

Go to the documentation of this file.

```
1 /*=========================================================================
2
3 Program:  GDCM (Grassroots DICOM). A DICOM library
4
5 Copyright (c) 2006-2011 Mathieu Malaterre
6 All rights reserved.
7 See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.
8
9 This software is distributed WITHOUT ANY WARRANTY; without even
10 the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
11 PURPOSE.  See the above copyright notice for more information.
12
13 =========================================================================*/
14 #ifndef GDCMSIMPLESUBJECTWATCHER_H
15 #define GDCMSIMPLESUBJECTWATCHER_H
16
17 #include "gdcmSubject.h"
```

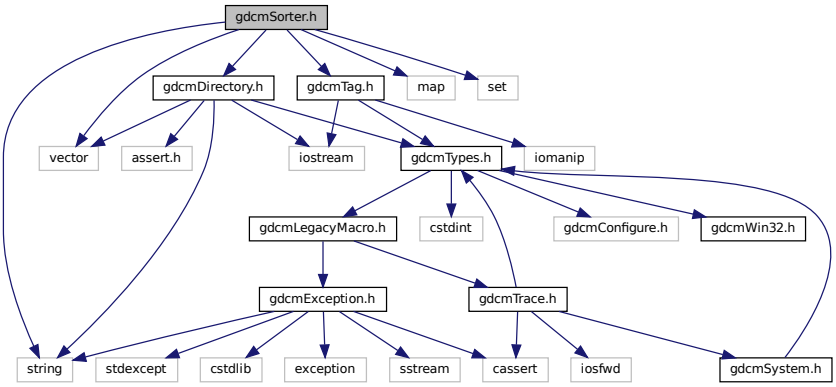
```
18 #include "gdcmCommand.h"
19 #include "gdcmSmartPointer.h"
20 #include "gdcmAnonymizeEvent.h"
21 #include "gdcmDataEvent.h"
22
23 namespace gdcm
24 {
25 //-----------------------------------------------------------------------------
26 class Event;
31 class GDCM_EXPORT SimpleSubjectWatcher
32 {
33 public:
34   SimpleSubjectWatcher(Subject * s, const char *comment = "");
35   virtual ~SimpleSubjectWatcher();
36   SimpleSubjectWatcher(const SimpleSubjectWatcher&) = delete;
37   void operator=(const SimpleSubjectWatcher&) = delete;
38
39 protected:
40   virtual void StartFilter();
41   virtual void EndFilter();
42   virtual void ShowProgress(Subject *caller, const Event &evt);
43   virtual void ShowFileName(Subject *caller, const Event &evt);
44   virtual void ShowIteration();
45   virtual void ShowAnonymization(Subject *caller, const Event &evt);
46   virtual void ShowDataSet(Subject *caller, const Event &evt);
47   virtual void ShowData(Subject *caller, const Event &evt);
48   virtual void ShowAbort();
49
50 protected:
51   // Custom API used for internal Testing do not use !
52   void TestAbortOn();
53   void TestAbortOff();
54
55 private:
56   SmartPointer<Subject> m_Subject;
57   std::string m_Comment;
58
59   typedef SimpleMemberCommand<SimpleSubjectWatcher> SimpleCommandType;
60   typedef MemberCommand<SimpleSubjectWatcher> CommandType;
61
62   SmartPointer<SimpleCommandType> m_StartFilterCommand;
63   SmartPointer<SimpleCommandType> m_EndFilterCommand;
64   SmartPointer<CommandType> m_ProgressFilterCommand;
65   SmartPointer<CommandType> m_FileNameFilterCommand;
66   SmartPointer<SimpleCommandType> m_IterationFilterCommand;
67   SmartPointer<SimpleCommandType> m_AbortFilterCommand;
68   SmartPointer<CommandType> m_AnonymizeFilterCommand;
69   SmartPointer<CommandType> m_DataFilterCommand;
70   SmartPointer<CommandType> m_DataSetFilterCommand;
71
72   unsigned long m_StartTag;
73   unsigned long m_EndTag;
74   unsigned long m_ProgressTag;
75   unsigned long m_FileNameTag;
76   unsigned long m_IterationTag;
77   unsigned long m_AbortTag;
78   unsigned long m_AnonymizeTag;
79   unsigned long m_DataTag;
80   unsigned long m_DataSetTag;
81
82   bool m_TestAbort;
83
84 };
85 } // end namespace gdcm
86 //-----------------------------------------------------------------------------
87 #endif //GDCMSIMPLESUBJECTWATCHER_H
```

11.413 gdcmSorter.h File Reference

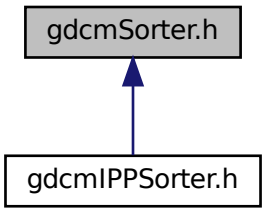
```
#include "gdcmDirectory.h"
#include "gdcmTag.h"
#include <vector>
#include <string>
```

```
#include <map>
#include <set>
```

Include dependency graph for gdcmSorter.h:



This graph shows which files directly or indirectly include this file:



Classes

- class gdcm::Sorter

  Sorter.

Namespaces

- namespace gdcm

Functions

- std::ostream & gdcm::operator<< (std::ostream &os, const Sorter &s)

11.414 gdcmSorter.h

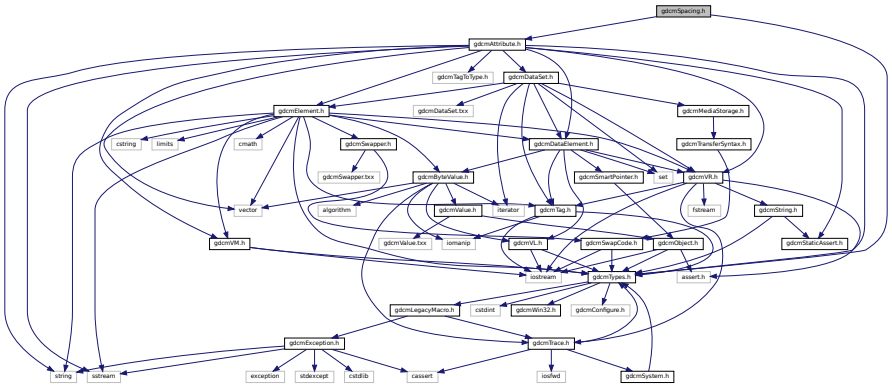
Go to the documentation of this file.

```
1 /*=========================================================================
2
3 Program:  GDCM (Grassroots DICOM). A DICOM library
4
5 Copyright (c) 2006-2011 Mathieu Malaterre
6 All rights reserved.
7 See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.
8
9 This software is distributed WITHOUT ANY WARRANTY; without even
10 the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
11 PURPOSE.  See the above copyright notice for more information.
12
13 =========================================================================*/
14 #ifndef GDCMSORTER_H
15 #define GDCMSORTER_H
16
17 #include "gdcmDirectory.h"
18 #include "gdcmTag.h"
19
20 #include <vector>
21 #include <string>
22 #include <map>
23 #include <set>
24
25 namespace gdcm
26 {
27 class DataSet;
28
39 class GDCM_EXPORT Sorter
40 {
41   friend std::ostream& operator<<(std::ostream &_os, const Sorter &s);
42 public:
43   Sorter();
44   virtual ~Sorter();
45
47   virtual bool Sort(std::vector<std::string> const & filenames);
48
51   const std::vector<std::string> &GetFilenames()const { return Filenames; }
52
54   void Print(std::ostream &os) const;
55
57   bool AddSelect( Tag const &tag, const char *value );
58
62   void SetTagsToRead( std::set<Tag> const & tags );
63
65   typedef bool (*SortFunction)(DataSet const &, DataSet const &);
66   void SetSortFunction( SortFunction f );
67
68   virtual bool StableSort(std::vector<std::string> const & filenames);
69
70 protected:
71   std::vector<std::string> Filenames;
72   typedef std::map<Tag,std::string> SelectionMap;
73   std::map<Tag,std::string> Selection;
74   SortFunction SortFunc;
75   std::set<Tag> TagsToRead;
76 };
77 //-----------------------------------------------------------------------------
78 inline std::ostream& operator<<(std::ostream &os, const Sorter &s)
79 {
80   s.Print( os );
81   return os;
82 }
83
84
85 } // end namespace gdcm
86
87 #endif //GDCMSORTER_H
```

11.415 gdcmSpacing.h File Reference

```
#include "gdcmTypes.h"
#include "gdcmAttribute.h"
```

Include dependency graph for gdcmSpacing.h:



Classes

- class gdcm::Spacing

  Class for Spacing.

Namespaces

- namespace gdcm

11.416 gdcmSpacing.h

Go to the documentation of this file.

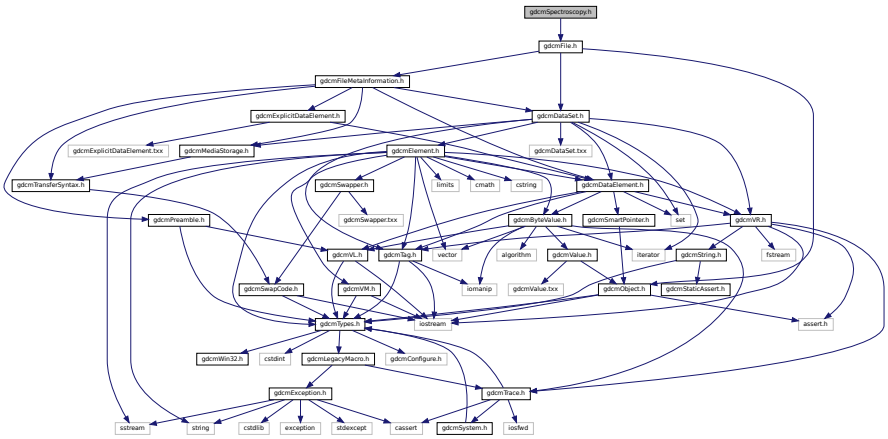
```
1 /*=========================================================================
2
3 Program:  GDCM (Grassroots DICOM). A DICOM library
4
5 Copyright (c) 2006-2011 Mathieu Malaterre
6 All rights reserved.
7 See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.
8
9 This software is distributed WITHOUT ANY WARRANTY; without even
10 the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
11 PURPOSE.  See the above copyright notice for more information.
12
13 =========================================================================*/
14 #ifndef GDCMSPACING_H
15 #define GDCMSPACING_H
16
17 #include "gdcmTypes.h"
18 #include "gdcmAttribute.h"
19
20 namespace gdcm
21 {
85 class GDCM_EXPORT Spacing
```

```
86 {
87 public :
88   Spacing();
89   ~Spacing();
90
91   // Here are the list of spacing we support:
92   // (0018,0088) DS [1.500000]                                    # 8,1 Spacing Between Slices
93   // (0018,1164) DS [0.5\0.5 ]                                    # 8,2 Imager Pixel Spacing
94   // (0018,2010) DS [0.664062\0.664062 ]                          # 18,2 Nominal Scanned Pixel Spacing
95   // (0018,7022) DS [0.125\0.125 ]                                # 12,2 Detector Element Spacing
96   // (0028,0030) DS [0.25\0.25 ]                                  # 10,2 Pixel Spacing
97   // > (0028,0a02) CS [FIDUCIAL]                                    # 8,1 Pixel Spacing Calibration Type
98   // > (0028,0a04) LO [Used fiducial ]                              # 14,1 Pixel Spacing Calibration
     Description
99   // (0028,0034) IS [4\3 ]                                        # 4,2 Pixel Aspect Ratio
100   // (3002,0011) DS [0.8\0.8 ]                                    # 8,2 Image Plane Pixel Spacing
101
102   // Here is the list of Spacing we do not support:
103   // <entry group="0018" element="7041" vr="LT" vm="1" name="Grid Spacing Material"/>
104   // <entry group="0018" element="9030" vr="FD" vm="1" name="Tag Spacing First Dimension"/>
105   // <entry group="0018" element="9218" vr="FD" vm="1" name="Tag Spacing Second Dimension"/>
106   // <entry group="0018" element="9322" vr="FD" vm="2" name="Reconstruction Pixel Spacing"/>
107   // <entry group="0018" element="9404" vr="FL" vm="2" name="Object Pixel Spacing in Center of Beam"/>
108   // <entry group="0040" element="08d8" vr="SQ" vm="1" name="Pixel Spacing Sequence"/>
109   // <entry group="0070" element="0101" vr="DS" vm="2" name="Presentation Pixel Spacing"/>
110   // <entry group="2010" element="0376" vr="DS" vm="2" name="Printer Pixel Spacing"/>
111   // <entry group="300a" element="00e9" vr="DS" vm="2" name="Compensator Pixel Spacing"/>
112
113   typedef enum {
114     DETECTOR = 0, // (0018,1164) Imager Pixel Spacing
115     MAGNIFIED,    // (0018,1114) (IHE Mammo)
116     CALIBRATED,   // (0028,0030) Pixel Spacing -> (0028,0a04) Pixel Spacing Calibration Description
117     UNKNOWN
118   } SpacingType;
119
120   static Attribute<0x28,0x34> ComputePixelAspectRatioFromPixelSpacing(const Attribute<0x28,0x30>&
     pixelspacing);
121 };
122 } // end namespace gdcm
123 //-----------------------------------------------------------------------------
124 #endif //GDCMSPACING_H
```

11.417 gdcmSpectroscopy.h File Reference

`#include "gdcmFile.h"`

Include dependency graph for gdcmSpectroscopy.h:



Classes

- class gdcm::Spectroscopy

  Spectroscopy class.

Namespaces

- namespace gdcm

11.418 gdcmSpectroscopy.h

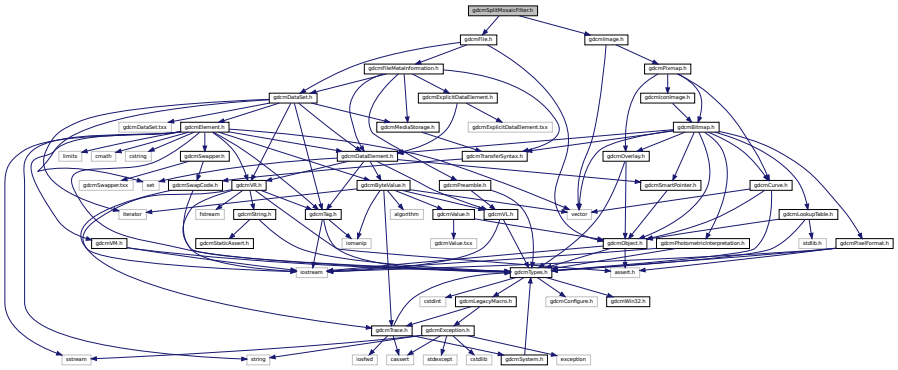
Go to the documentation of this file.

```
1 /*=========================================================================
2
3 Program:  GDCM (Grassroots DICOM). A DICOM library
4
5 Copyright (c) 2006-2011 Mathieu Malaterre
6 All rights reserved.
7 See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.
8
9 This software is distributed WITHOUT ANY WARRANTY; without even
10 the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
11 PURPOSE.  See the above copyright notice for more information.
12
13 =========================================================================*/
14 #ifndef GDCMSPECTROSCOPY_H
15 #define GDCMSPECTROSCOPY_H
16
17 #include "gdcmFile.h"
18
19 namespace gdcm
20 {
24 class GDCM_EXPORT Spectroscopy
25 {
26 public:
27   Spectroscopy() = default;
28
29 private:
30 };
31
32 } // end namespace gdcm
33
34 #endif //GDCMSPECTROSCOPY_H
```

11.419 gdcmSplitMosaicFilter.h File Reference

```
#include "gdcmFile.h"
#include "gdcmImage.h"
```

Include dependency graph for gdcmSplitMosaicFilter.h:



Classes

- class gdcm::SplitMosaicFilter

  SplitMosaicFilter class.

Namespaces

- namespace gdcm

11.420 gdcmSplitMosaicFilter.h

Go to the documentation of this file.

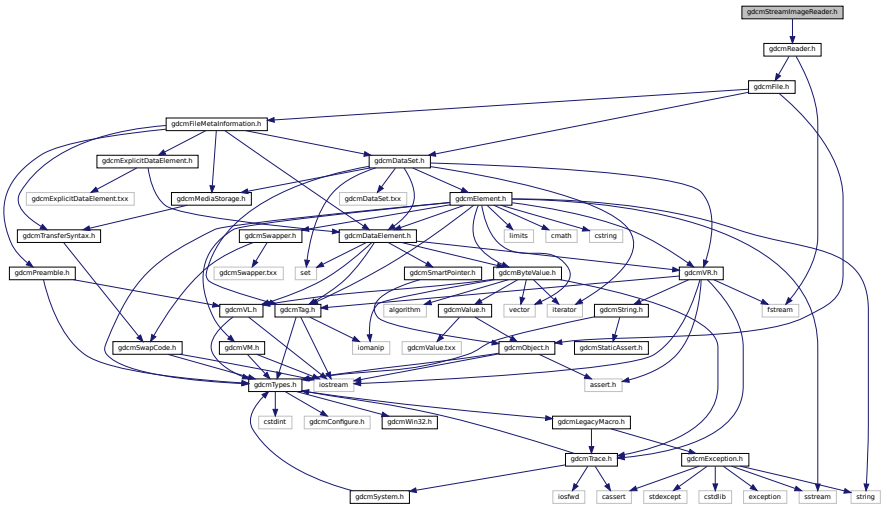
```
1 /*=========================================================================
2
3 Program:  GDCM (Grassroots DICOM). A DICOM library
4
5 Copyright (c) 2006-2011 Mathieu Malaterre
6 All rights reserved.
7 See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.
8
9 This software is distributed WITHOUT ANY WARRANTY; without even
10 the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
11 PURPOSE.  See the above copyright notice for more information.
12
13 =========================================================================*/
14 #ifndef GDCMSPLITMOSAICFILTER_H
15 #define GDCMSPLITMOSAICFILTER_H
16
17 #include "gdcmFile.h"
18 #include "gdcmImage.h"
19
20 namespace gdcm
21 {
22
23 /*
24 * Everything done in this code is for the sole purpose of writing interoperable
25 * software under Sect.  1201 (f) Reverse Engineering exception of the DMCA.
26 * If you believe anything in this code violates any law or any of your rights,
27 * please contact us (gdcm-developers@lists.sourceforge.net) so that we can
28 * find a solution.
29 */
43 class GDCM_EXPORT SplitMosaicFilter
```

```
44 {
45 public:
46   SplitMosaicFilter();
47   ~SplitMosaicFilter();
48
50   bool Split();
51
54   bool ComputeMOSAICDimensions(unsigned int dims[3]);
55
57   bool ComputeMOSAICSliceNormal( double dims[3], bool & inverted );
58
60   bool ComputeMOSAICSlicePosition( double pos[3], bool inverted );
61
62   void SetImage(const Image& image);
63   const Image &GetImage()const { return *I; }
64   Image &GetImage() { return *I; }
65
66   void SetFile(const File& f) { F = f; }
67   File &GetFile() { return *F; }
68   const File &GetFile()const { return *F; }
69
71   static bool GetAcquisitionSize(unsigned int size[2], DataSet const & ds);
72
74   static unsigned int GetNumberOfImagesInMosaic( File const & file );
75
76 protected:
77
78 private:
79   SmartPointer<File> F;
80   SmartPointer<Image> I;
81 };
82
83 } // end namespace gdcm
84
85 #endif //GDCMSPLITMOSAICFILTER_H
```

11.421 gdcmStreamImageReader.h File Reference

`#include "gdcmReader.h"`

Include dependency graph for gdcmStreamImageReader.h:



Classes

- class gdcm::StreamImageReader

StreamImageReader.

Namespaces

- namespace gdcm

11.422 gdcmStreamImageReader.h

Go to the documentation of this file.

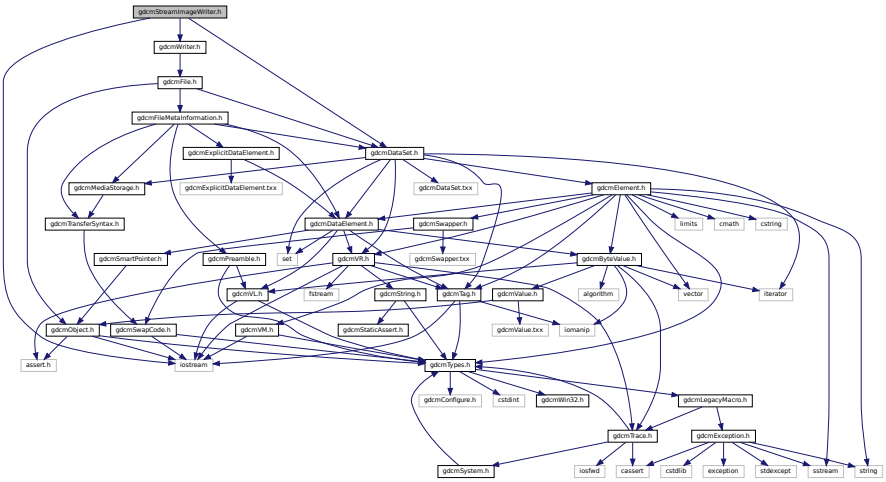
```
1 /*=========================================================================
2 *
3 *  Copyright NumFOCUS
4 *
5 *  Licensed under the Apache License, Version 2.0 (the "License");
6 *  you may not use this file except in compliance with the License.
7 *  You may obtain a copy of the License at
8 *
9 *         http://www.apache.org/licenses/LICENSE-2.0.txt
10 *
11 *  Unless required by applicable law or agreed to in writing, software
12 *  distributed under the License is distributed on an "AS IS" BASIS,
13 *  WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
14 *  See the License for the specific language governing permissions and
15 *  limitations under the License.
16 *
17 *=========================================================================*/
18 #ifndef GDCMSTREAMIMAGEREADER_H
19 #define GDCMSTREAMIMAGEREADER_H
20
21 #include "gdcmReader.h"
22
23 namespace gdcm
24 {
25
26 class MediaStorage;
38 class GDCM_EXPORT StreamImageReader
39 {
40
41 public:
42   StreamImageReader();
43   virtual ~StreamImageReader();
44
48   void SetFileName(const char* inFileName);
49   void SetStream(std::istream& inStream);
50
51   std::vector<unsigned int> GetDimensionsValueForResolution( unsigned int  );
52
60   void DefinePixelExtent(uint16_t inXMin, uint16_t inXMax,
61     uint16_t inYMin, uint16_t inYMax, uint16_t inZMin = 0, uint16_t inZMax = 1);
62
67   uint32_t DefineProperBufferLength() const;
68
76   bool Read(char* inReadBuffer, const std::size_t& inBufferLength);
77
83   bool CanReadImage() const;
84
88   virtual bool ReadImageInformation();
89
93   File const & GetFile() const;
94
95 protected:
96 private:
97   //contains a reader for being able to ReadUpToTag
98   //however, we don't want the user to be able to call Read
99   //either directly or via a parent class call, so we hide the reader in here.
100   Reader mReader;
101
102   std::streamoff mFileOffset; //the file offset for getting header information
103 #if 0
104   std::streamoff mFileOffset1;
105 #endif
```

```
106   DataSet mHeaderInformation; //all the non-pixel information
107
108   //for thread safety, these should not be stored here, but should be used
109   //for every read subregion operation.
110   uint16_t mXMin, mYMin, mXMax, mYMax, mZMin, mZMax;
111
116   bool ReadImageSubregionRAW(char* inReadBuffer, const std::size_t& inBufferLength);
117
120   bool ReadImageSubregionJpegLS(char* inReadBuffer, const std::size_t& inBufferLength);
121 };
122
123 } // end namespace gdcm
124
125 #endif //GDCMSTREAMIMAGEREADER_H
126
```

11.423 gdcmStreamImageWriter.h File Reference

```
#include "gdcmWriter.h"
#include <iostream>
#include "gdcmDataSet.h"
```

Include dependency graph for gdcmStreamImageWriter.h:



Classes

- class gdcm::StreamImageWriter

  StreamImageReader.

Namespaces

- namespace gdcm

11.424 gdcmStreamImageWriter.h

Go to the documentation of this file.

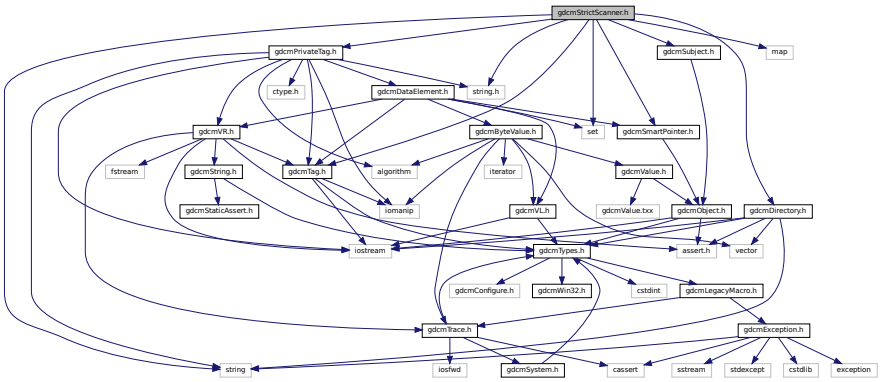
```
1 /*=========================================================================
2 *
3 *  Copyright NumFOCUS
4 *
5 *  Licensed under the Apache License, Version 2.0 (the "License");
6 *  you may not use this file except in compliance with the License.
7 *  You may obtain a copy of the License at
8 *
9 *         http://www.apache.org/licenses/LICENSE-2.0.txt
10 *
11 *  Unless required by applicable law or agreed to in writing, software
12 *  distributed under the License is distributed on an "AS IS" BASIS,
13 *  WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
14 *  See the License for the specific language governing permissions and
15 *  limitations under the License.
16 *
17 *=========================================================================*/
18
19 #ifndef GDCMSTREAMIMAGEWRITER_H
20 #define GDCMSTREAMIMAGEWRITER_H
21
22 #include "gdcmWriter.h"
23 #include <iostream>
24 #include "gdcmDataSet.h"
25
26 namespace gdcm
27 {
28
29 class MediaStorage;
30 class RAWCodec;
42 class GDCM_EXPORT StreamImageWriter
43 {
44
45 public:
46   StreamImageWriter();
47   virtual ~StreamImageWriter();
48
49
53   void SetFileName(const char* inFileName);
54   void SetStream(std::ostream& inStream);
55
64   void DefinePixelExtent(uint16_t inXMin, uint16_t inXMax,
65     uint16_t inYMin, uint16_t inYMax, uint16_t inZMin = 0, uint16_t inZMax = 1);
66
67
73   uint32_t DefineProperBufferLength();
74
82   bool Write(void* inWriteBuffer, const std::size_t& inBufferLength);
83
87   virtual bool WriteImageInformation();
88
92   bool CanWriteFile() const;
93
94
97   void SetFile(const File& inFile);
98
99 protected:
100
101   //contains the PrepareWrite function, which will get the given dataset ready
102   //for writing to disk by manufacturing the header information.
103   //note that if there is a pixel element in the given dataset, that will be removed
104   //during the copy, so that the imagewriter can write everything else out
105   Writer mWriter;
106
107   //is the offset necessary if we always append?
108   //std::streamoff mFileOffset; //the fileoffset for getting header information
109   SmartPointer<File> mspFile; //all the non-pixel information
110
111   //for thread safety, these should not be stored here, but should be used
112   //for every read subregion operation.
113   uint16_t mXMin, mYMin, mXMax, mYMax, mZMin, mZMax;
114
119   //virtual bool ReadImageSubregionRAW(std::ostream& os);
120   virtual bool WriteImageSubregionRAW(char* inWriteBuffer, const std::size_t& inBufferLength);
121
131   int WriteRawHeader(RAWCodec* inCodec, std::ostream* inStream);
```

```
132
137   int mElementOffsets;
138   int mElementOffsets1;
139
140 };
141
142
143 } // end namespace gdcm
144
145 #endif //GDCMSTREAMIMAGEWRITER_H
```

11.425 gdcmStrictScanner.h File Reference

```
#include "gdcmDirectory.h"
#include "gdcmSubject.h"
#include "gdcmTag.h"
#include "gdcmPrivateTag.h"
#include "gdcmSmartPointer.h"
#include <map>
#include <set>
#include <string>
#include <string.h>
```

Include dependency graph for gdcmStrictScanner.h:



Classes

- struct gdcm::StrictScanner::ltstr
- class gdcm::StrictScanner

  StrictScanner.

Namespaces

- namespace gdcm

Functions

- std::ostream & gdcm::operator<< (std::ostream &os, const StrictScanner &s)

11.426 gdcmStrictScanner.h

Go to the documentation of this file.

```
1 /*=========================================================================
2
3 Program:  GDCM (Grassroots DICOM). A DICOM library
4
5 Copyright (c) 2006-2011 Mathieu Malaterre
6 All rights reserved.
7 See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.
8
9 This software is distributed WITHOUT ANY WARRANTY; without even
10 the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
11 PURPOSE.  See the above copyright notice for more information.
12
13 =========================================================================*/
14 #ifndef GDCMSTRICTSCANNER_H
15 #define GDCMSTRICTSCANNER_H
16
17 #include "gdcmDirectory.h"
18 #include "gdcmSubject.h"
19 #include "gdcmTag.h"
20 #include "gdcmPrivateTag.h"
21 #include "gdcmSmartPointer.h"
22
23 #include <map>
24 #include <set>
25 #include <string>
26
27 #include <string.h> // strcmp
28
29 namespace gdcm
30 {
31 class StringFilter;
32
55 class GDCM_EXPORT StrictScanner :  public Subject
56 {
57   friend std::ostream& operator<<(std::ostream &_os, const StrictScanner &s);
58 public:
59   StrictScanner():Values(),Filenames(),Mappings() {}
60   ~StrictScanner() override;
61
68   typedef std::map<Tag, const char*> TagToValue;
69   //typedef std::map<Tag, ConstCharWrapper> TagToValue; //StringMap;
70   //typedef TagToStringMap TagToValue;
71   typedef TagToValue::value_type TagToValueValueType;
72
74   void AddTag( Tag const & t );
75   void ClearTags();
76
77   // Work in progress do not use:
78   void AddPrivateTag( PrivateTag const & t );
79
81   void AddSkipTag( Tag const & t );
82   void ClearSkipTags();
83
85   bool Scan( Directory::FilenamesType const & filenames );
86
87   Directory::FilenamesType const &GetFilenames()const { return Filenames; }
88
90   void Print( std::ostream & os ) const override;
91
92   void PrintTable( std::ostream & os ) const;
93
97   bool IsKey( const char * filename ) const;
98
101   Directory::FilenamesType GetKeys() const;
102
103   // struct to store all the values found:
104   typedef std::set< std::string > ValuesType;
```

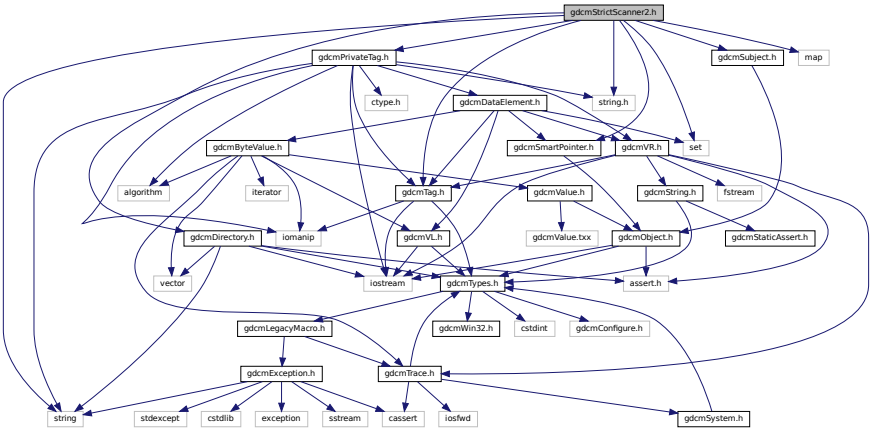
```
105
107   ValuesType const & GetValues()const { return Values; }
108
110   ValuesType GetValues(Tag const &t) const;
111
115   Directory::FilenamesType GetOrderedValues(Tag const &t) const;
116
117   /* ltstr is CRITICAL, otherwise pointers value are used to do the key comparison */
118   struct ltstr
119     {
120     bool operator()(const char* s1, const char* s2)const
121 {
122       assert( s1 && s2 );
123       return strcmp(s1, s2) < 0;
124       }
125     };
126   typedef std::map<const char *,TagToValue, ltstr> MappingType;
127   typedef MappingType::const_iterator ConstIterator;
128   ConstIterator Begin()const { return Mappings.begin(); }
129   ConstIterator End()const { return Mappings.end(); }
130
132   MappingType const & GetMappings()const { return Mappings; }
133
135   TagToValue const & GetMapping(const char *filename) const;
136
139   const char *GetFilenameFromTagToValue(Tag const &t, const char *valueref) const;
140
143   Directory::FilenamesType GetAllFilenamesFromTagToValue(Tag const &t, const char *valueref) const;
144
146   // by a call to GetMapping()
147   TagToValue const & GetMappingFromTagToValue(Tag const &t, const char *value) const;
148
154   const char* GetValue(const char *filename, Tag const &t) const;
155
157   static SmartPointer<StrictScanner> New() { return new StrictScanner; }
158
159 protected:
160   void ProcessPublicTag(StringFilter &sf, const char *filename);
161 private:
162   // struct to store all uniq tags in ascending order:
163   typedef std::set< Tag > TagsType;
164   typedef std::set< PrivateTag > PrivateTagsType;
165   std::set< Tag > Tags;
166   std::set< PrivateTag > PrivateTags;
167   std::set< Tag > SkipTags;
168   ValuesType Values;
169   Directory::FilenamesType Filenames;
170
171   // Main struct that will hold all mapping:
172   MappingType Mappings;
173
174   double Progress;
175 };
176 //-----------------------------------------------------------------------------
177 inline std::ostream& operator«(std::ostream &os, const StrictScanner &s)
178 {
179   s.Print( os );
180   return os;
181 }
182
183 } // end namespace gdcm
184
185 #endif //GDCMSTRICTSCANNER_H
```

11.427 gdcmStrictScanner2.h File Reference

```
#include "gdcmDirectory.h"
#include "gdcmPrivateTag.h"
#include "gdcmSmartPointer.h"
#include "gdcmSubject.h"
#include "gdcmTag.h"
#include <map>
```

```
#include <set>
#include <string>
#include <string.h>
```

Include dependency graph for gdcmStrictScanner2.h:



Classes

- struct gdcm::StrictScanner2::ltstr
- class gdcm::StrictScanner2

  StrictScanner2.

Namespaces

- namespace gdcm

Functions

- std::ostream & gdcm::operator<< (std::ostream &os, const StrictScanner2 &s)

11.428 gdcmStrictScanner2.h

Go to the documentation of this file.

```
1 /*=========================================================================
2
3 Program:  GDCM (Grassroots DICOM). A DICOM library
4
5 Copyright (c) 2006-2011 Mathieu Malaterre
6 All rights reserved.
7 See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.
8
9 This software is distributed WITHOUT ANY WARRANTY; without even
10 the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
11 PURPOSE.  See the above copyright notice for more information.
12
```

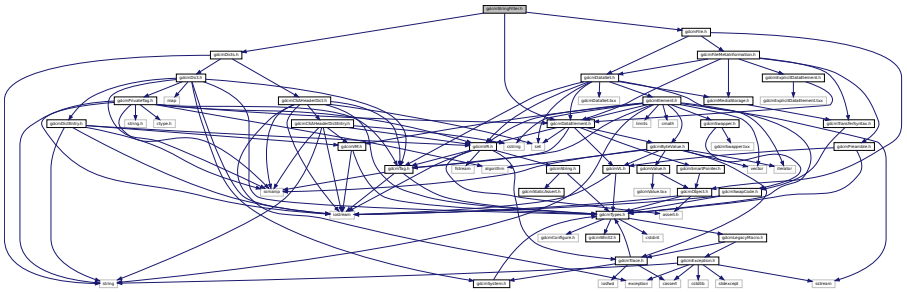
```
13 =========================================================================*/
14 #ifndef GDCMSTRICTSCANNER2_H
15 #define GDCMSTRICTSCANNER2_H
16
17 #include "gdcmDirectory.h"
18 #include "gdcmPrivateTag.h"
19 #include "gdcmSmartPointer.h"
20 #include "gdcmSubject.h"
21 #include "gdcmTag.h"
22
23 #include <map>
24 #include <set>
25 #include <string>
26
27 #include <string.h>  // strcmp
28
29 namespace gdcm {
30 class StringFilter;
31
52 class GDCM_EXPORT StrictScanner2 :  public Subject {
53   friend std::ostream &operator<<(std::ostream &_os, const StrictScanner2 &s);
54
55  public:
56   StrictScanner2() :  Values(), Filenames(), PublicMappings(), PrivateMappings() {}
57   ~StrictScanner2() override;
58
65   typedef std::map<Tag, const char *> PublicTagToValue;
66   typedef PublicTagToValue::value_type PublicTagToValueValueType;
67
68   typedef std::map<PrivateTag, const char *> PrivateTagToValue;
69   typedef PrivateTagToValue::value_type PrivateTagToValueValueType;
70
72   bool AddPublicTag(Tag const &t);
73   void ClearPublicTags();
74
75   // Work in progress do not use:
76   bool AddPrivateTag(PrivateTag const &pt);
77   void ClearPrivateTags();
78
80   bool AddSkipTag(Tag const &t);
81   void ClearSkipTags();
82
84   bool Scan(Directory::FilenamesType const &filenames);
85
87   Directory::FilenamesType const &GetFilenames()const { return Filenames; }
88
90   void Print(std::ostream &os) const override;
91
93   void PrintTable(std::ostream &os, bool header = false) const;
94
98   bool IsKey(const char *filename) const;
99
102   Directory::FilenamesType GetKeys() const;
103
104   // struct to store all the values found:
105   typedef std::set<std::string> ValuesType;
106
108   ValuesType const &GetValues()const { return Values; }
109
111   ValuesType GetPublicValues(Tag const &t) const;
112
115   ValuesType GetPrivateValues(PrivateTag const &pt) const;
116
120   Directory::FilenamesType GetPublicOrderedValues(Tag const &t) const;
121
122   Directory::FilenamesType GetPrivateOrderedValues(PrivateTag const &pt) const;
123
124   /* ltstr is CRITICAL, otherwise pointers value are used to do the key
125  * comparison */
126   struct ltstr {
127     bool operator()(const char *s1, const char *s2)const {
128       assert(s1 && s2);
129       return strcmp(s1, s2) < 0;
130     }
131   };
132   typedef std::map<const char *, PublicTagToValue, ltstr> PublicMappingType;
133   typedef PublicMappingType::const_iterator PublicConstIterator;
134   PublicConstIterator Begin()const { return PublicMappings.begin(); }
135   PublicConstIterator End()const { return PublicMappings.end(); }
136
137   typedef std::map<const char *, PrivateTagToValue, ltstr> PrivateMappingType;
```

```
138   typedef PrivateMappingType::const_iterator PrivateConstIterator;
139   PrivateConstIterator PrivateBegin()const { return PrivateMappings.begin(); }
140   PrivateConstIterator PrivateEnd()const { return PrivateMappings.end(); }
141
144   PublicMappingType const &GetPublicMappings()const { return PublicMappings; }
145   PrivateMappingType const &GetPrivateMappings()const {
146     return PrivateMappings;
147   }
148
150   PublicTagToValue const &GetPublicMapping(const char *filename) const;
151   PrivateTagToValue const &GetPrivateMapping(const char *filename) const;
152
155   const char *GetFilenameFromPublicTagToValue(Tag const &t,
156                                               const char *valueref) const;
157   const char *GetFilenameFromPrivateTagToValue(PrivateTag const &pt,
158                                                const char *valueref) const;
159
162   Directory::FilenamesType GetAllFilenamesFromPublicTagToValue(
163       Tag const &t, const char *valueref) const;
164   Directory::FilenamesType GetAllFilenamesFromPrivateTagToValue(
165       PrivateTag const &pt, const char *valueref) const;
166
169   // by a call to GetMapping()
170   PublicTagToValue const &GetMappingFromPublicTagToValue(
171       Tag const &t, const char *value) const;
172   PrivateTagToValue const &GetMappingFromPrivateTagToValue(
173       PrivateTag const &pt, const char *value) const;
174
180   const char *GetPublicValue(const char *filename, Tag const &t) const;
181   const char *GetPrivateValue(const char *filename, PrivateTag const &t) const;
182
184   static SmartPointer<StrictScanner2> New() { return new StrictScanner2; }
185
186  protected:
187   void ProcessPublicTag(StringFilter &sf, const char *filename);
188   void ProcessPrivateTag(StringFilter &sf, const char *filename);
189
190  private:
191   // struct to store all uniq tags in ascending order:
192   typedef std::set<Tag> PublicTagsType;
193   typedef std::set<PrivateTag> PrivateTagsType;
194   std::set<Tag> PublicTags;          // Public and Private Creator
195   std::set<PrivateTag> PrivateTags;  // Only Private (no Private Creator)
196   std::set<Tag> SkipTags;
197   ValuesType Values;
198   Directory::FilenamesType Filenames;
199
200   // Main struct that will hold all public mapping:
201   PublicMappingType PublicMappings;
202   // Main struct that will hold all private mapping:
203   PrivateMappingType PrivateMappings;
204
205   double Progress;
206 };
207 //-----------------------------------------------------------------------------
208 inline std::ostream &operator<<(std::ostream &os, const StrictScanner2 &s) {
209   s.Print(os);
210   return os;
211 }
212
213 }  // end namespace gdcm
214
215 #endif  // GDCMSTRICTSCANNER2_H
```

11.429 gdcmStringFilter.h File Reference

```
#include "gdcmDataElement.h"
#include "gdcmDicts.h"
#include "gdcmFile.h"
```

Include dependency graph for gdcmStringFilter.h:



Classes

- class gdcm::StringFilter

  StringFilter.

Namespaces

- namespace gdcm

11.430 gdcmStringFilter.h

Go to the documentation of this file.

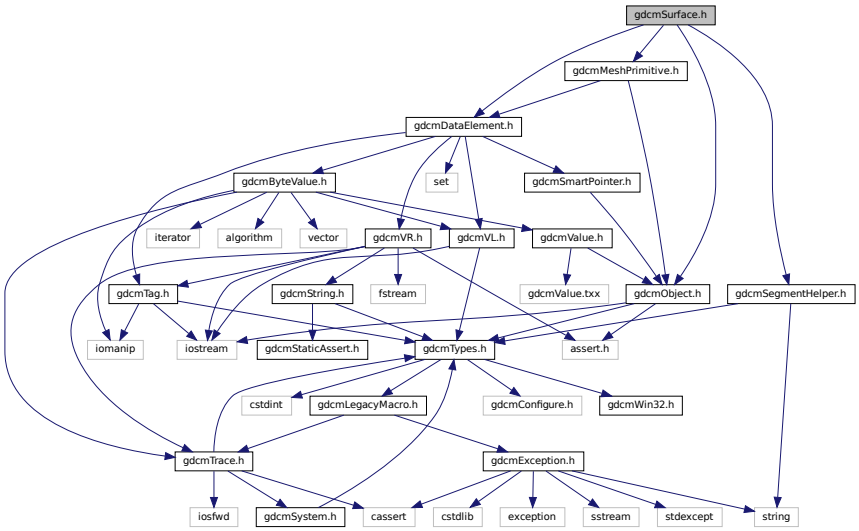
```
1 /*=========================================================================
2
3 Program:  GDCM (Grassroots DICOM). A DICOM library
4
5 Copyright (c) 2006-2011 Mathieu Malaterre
6 All rights reserved.
7 See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.
8
9 This software is distributed WITHOUT ANY WARRANTY; without even
10 the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
11 PURPOSE.  See the above copyright notice for more information.
12
13 =========================================================================*/
14 #ifndef GDCMSTRINGFILTER_H
15 #define GDCMSTRINGFILTER_H
16
17 #include "gdcmDataElement.h"
18 #include "gdcmDicts.h"
19 #include "gdcmFile.h"
20
21 namespace gdcm
22 {
23
29 class GDCM_EXPORT StringFilter
30 {
31 public:
32   StringFilter();
33   ~StringFilter();
34
36   void UseDictAlways(bool) {}
37
39   void SetDicts(const Dicts &dicts);
40
44   std::string ToString(const DataElement& de) const;
```

```
45
47   std::string ToString(const Tag& t) const;
48
49   std::string ToString(const PrivateTag& t) const;
50
55   std::pair<std::string, std::string> ToStringPair(const DataElement& de) const;
57   std::pair<std::string, std::string> ToStringPair(const Tag& t) const;
58
60   std::string FromString(const Tag&t, const char * value, size_t len);
61
63   void SetFile(const File& f) { F = f; }
64   File &GetFile() { return *F; }
65   const File &GetFile()const { return *F; }
66
70   bool ExecuteQuery(std::string const &query, std::string & value) const;
71
72 protected:
73   std::pair<std::string, std::string> ToStringPair(const Tag& t, DataSet const &ds) const;
74   bool ExecuteQuery(std::string const &query, DataSet const &ds, std::string & value) const;
75
76 private:
77   std::pair<std::string, std::string> ToStringPairInternal(const DataElement& de, DataSet const &ds) const;
78   SmartPointer<File> F;
79 };
80
81 } // end namespace gdcm
82
83 #endif //GDCMSTRINGFILTER_H
```

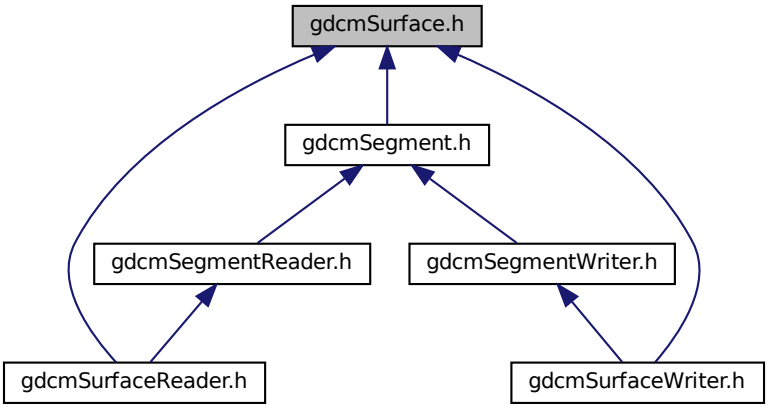
11.431 gdcmSurface.h File Reference

```
#include <gdcmObject.h>
#include <gdcmDataElement.h>
#include <gdcmMeshPrimitive.h>
#include "gdcmSegmentHelper.h"
```

Include dependency graph for gdcmSurface.h:



This graph shows which files directly or indirectly include this file:



Classes

- class gdcm::Surface

  This class defines a SURFACE IE.

Namespaces

- namespace gdcm

11.432 gdcmSurface.h

Go to the documentation of this file.

```
1 /*=========================================================================
2
3 Program:  GDCM (Grassroots DICOM). A DICOM library
4
5 Copyright (c) 2006-2011 Mathieu Malaterre
6 All rights reserved.
7 See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.
8
9 This software is distributed WITHOUT ANY WARRANTY; without even
10 the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
11 PURPOSE.  See the above copyright notice for more information.
12
13 =========================================================================*/
14 #ifndef GDCMSURFACE_H
15 #define GDCMSURFACE_H
16
17 #include <gdcmObject.h>
18 #include <gdcmDataElement.h>
19 #include <gdcmMeshPrimitive.h>
20 #include "gdcmSegmentHelper.h"  // for BasicCodedEntry
21
22 namespace gdcm
```

```
23 {
24
31 class GDCM_EXPORT Surface :  public Object
32 {
33 public:
34
35   typedef enum {
36     NO = 0,
37     YES,
38     UNKNOWN,
39     STATES_END
40   } STATES;
41
42   static const char * GetSTATESString(STATES state);
43   static STATES GetSTATES(const char * state);
44
50   typedef enum {
51     SURFACE = 0,
52     WIREFRAME,
53     POINTS,
54     VIEWType_END
55   } VIEWType;
56
57   static const char * GetVIEWTypeString(VIEWType type);
58   static VIEWType GetVIEWType(const char * type);
59
60   Surface();
61
62   ~Surface() override;
63
64   //**    Common getters/setters      **//
65   unsigned long GetSurfaceNumber() const;
66   void SetSurfaceNumber(const unsigned long nb);
67
68   const char * GetSurfaceComments() const;
69   void SetSurfaceComments(const char * comment);
70
71   bool GetSurfaceProcessing() const;
72   void SetSurfaceProcessing(bool b);
73
74   float GetSurfaceProcessingRatio() const;
75   void SetSurfaceProcessingRatio(const float ratio);
76
77   const char * GetSurfaceProcessingDescription() const;
78   void SetSurfaceProcessingDescription(const char * description);
79
80   SegmentHelper::BasicCodedEntry const & GetProcessingAlgorithm() const;
81   SegmentHelper::BasicCodedEntry & GetProcessingAlgorithm();
82   void SetProcessingAlgorithm(SegmentHelper::BasicCodedEntry const & BSE);
83
84   unsigned short GetRecommendedDisplayGrayscaleValue() const;
85   void SetRecommendedDisplayGrayscaleValue(const unsigned short vl);
86
87   const unsigned short * GetRecommendedDisplayCIELabValue() const;
88   unsigned short GetRecommendedDisplayCIELabValue(const unsigned int idx) const;
89   void SetRecommendedDisplayCIELabValue(const unsigned short vl[3]);
90   void SetRecommendedDisplayCIELabValue(const unsigned short vl, const unsigned int idx = 0);
91   void SetRecommendedDisplayCIELabValue(const std::vector< unsigned short > & vl);
92
93   float GetRecommendedPresentationOpacity() const;
94   void SetRecommendedPresentationOpacity(const float opacity);
95
96   VIEWType GetRecommendedPresentationType() const;
97   void SetRecommendedPresentationType(VIEWType type);
98
99   STATES GetFiniteVolume() const;
100   void SetFiniteVolume(STATES state);
101
102   STATES GetManifold() const;
103   void SetManifold(STATES state);
104
105   SegmentHelper::BasicCodedEntry const & GetAlgorithmFamily() const;
106   SegmentHelper::BasicCodedEntry & GetAlgorithmFamily();
107   void SetAlgorithmFamily(SegmentHelper::BasicCodedEntry const & BSE);
108
109   const char * GetAlgorithmVersion() const;
110   void SetAlgorithmVersion(const char * str);
111
112   const char * GetAlgorithmName() const;
113   void SetAlgorithmName(const char * str);
114
```

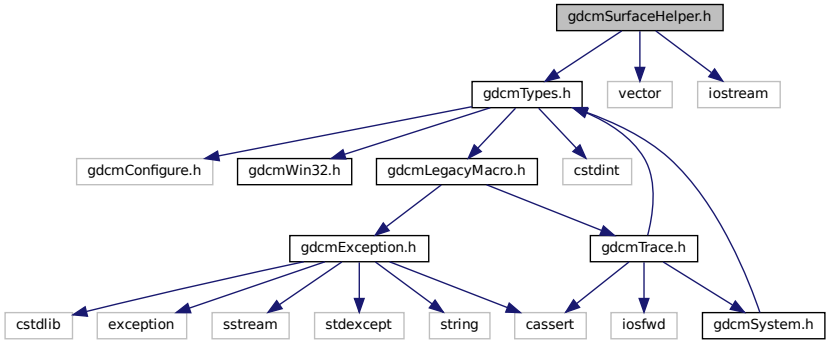
```
115   //**    Points getters/setters      **//
116   unsigned long GetNumberOfSurfacePoints() const;
117   void SetNumberOfSurfacePoints(const unsigned long nb);
118
119   const DataElement & GetPointCoordinatesData() const;
120   DataElement & GetPointCoordinatesData();
121
122   void SetPointCoordinatesData(DataElement const & de);
123
127   const float * GetPointPositionAccuracy() const;
128   void SetPointPositionAccuracy(const float * accuracies);
129
130   float GetMeanPointDistance() const;
131   void SetMeanPointDistance(float average);
132
133   float GetMaximumPointDistance() const;
134   void SetMaximumPointDistance(float maximum);
135
139   const float * GetPointsBoundingBoxCoordinates() const;
140   void SetPointsBoundingBoxCoordinates(const float * coordinates);
141
145   const float * GetAxisOfRotation() const;
146   void SetAxisOfRotation(const float * axis);
147
151   const float * GetCenterOfRotation() const;
152   void SetCenterOfRotation(const float * center);
153
154   //**    Vectors getters/setters      **//
155   unsigned long GetNumberOfVectors() const;
156   void SetNumberOfVectors(const unsigned long nb);
157
158   unsigned short GetVectorDimensionality() const;
159   void SetVectorDimensionality(const unsigned short dim);
160
161   const float * GetVectorAccuracy() const;
162   void SetVectorAccuracy(const float * accuracy);
163
164   const DataElement & GetVectorCoordinateData() const;
165   DataElement & GetVectorCoordinateData();
166
167   void SetVectorCoordinateData(DataElement const & de);
168
169   //**    Primitive getters/setters      **//
170   MeshPrimitive const & GetMeshPrimitive() const;
171   MeshPrimitive & GetMeshPrimitive();
172
173   void SetMeshPrimitive(MeshPrimitive & mp);
174
175 private:
176
177   //**        Common members      **//
178
179   //0066 0003 UL 1 Surface Number
180   unsigned long SurfaceNumber;
181   //0066 0004 LT 1 Surface Comments
182   std::string   SurfaceComments;
183
184   //0066 0009 CS 1 Surface Processing
185   bool          SurfaceProcessing;
186   //0066 000a FL 1 Surface Processing Ratio
187   float         SurfaceProcessingRatio;
188   //0066 000b LO 1 Surface Processing Description
189   std::string   SurfaceProcessingDescription;
190   // Processing Algorithm Code
191   SegmentHelper::BasicCodedEntry ProcessingAlgorithm;
192
193   //0062 000c US 1 Recommended Display Grayscale Value
194   unsigned short  RecommendedDisplayGrayscaleValue;
195   //0062 000d US 3 Recommended Display CIELab Value
196   unsigned short  RecommendedDisplayCIELabValue[3];
197
198   // 0066 000c FL 1 Recommended Presentation Opacity
199   float           RecommendedPresentationOpacity;
200   // 0066 000d CS 1 Recommended Presentation Type
201   VIEWType        RecommendedPresentationType;
202
203   //0066 000e CS 1 Finite Volume
204   STATES        FiniteVolume;
205   //0066 0010 CS 1 Manifold
206   STATES        Manifold;
207
```

```
208   // Algorithm Family Code
209   SegmentHelper::BasicCodedEntry AlgorithmFamily;
210
211   //0066 0031 LO 1 Algorithm Version
212   std::string   AlgorithmVersion;
213   //0066 0032 LT 1 Algorithm Parameters
214   //0066 0036 LO 1 Algorithm Name
215   std::string   AlgorithmName;
216
217
218   //**        Point members      **//
219
220   //0066 0015 UL 1 Number of Surface Points
221   unsigned long NumberOfSurfacePoints;
222   //0066 0016 OF 1 Point Coordinates Data
223   DataElement   PointCoordinatesData;
224   //0066 0017 FL 3 Point Position Accuracy
225   float *       PointPositionAccuracy;
226   //0066 0018 FL 1 Mean Point Distance
227   float         MeanPointDistance;
228   //0066 0019 FL 1 Maximum Point Distance
229   float         MaximumPointDistance;
230   //0066 001a FL 6 Points Bounding Box Coordinates
231   float *       PointsBoundingBoxCoordinates;
232   //0066 001b FL 3 Axis of Rotation
233   float *       AxisOfRotation;
234   //0066 001c FL 3 Center of Rotation
235   float *       CenterOfRotation;
236
237
238   //**        Normal members      **//
239
240   //0066 001e UL 1 Number of Vectors
241   unsigned long NumberOfVectors;
242   //0066 001f US 1 Vector Dimensionality
243   unsigned short VectorDimensionality;
244   //0066 0020 FL 1-n Vector Accuracy
245   float *       VectorAccuracy;
246   //0066 0021 OF 1 Vector Coordinate Data
247   DataElement   VectorCoordinateData;
248
249
250   //**        Primitive members      **//
251   SmartPointer< MeshPrimitive > Primitive;
252 };
253
254 }
255
256 #endif // GDCMSURFACE_H
```

11.433 gdcmSurfaceHelper.h File Reference

```
#include "gdcmTypes.h"
#include <vector>
#include <iostream>
```

Include dependency graph for gdcmSurfaceHelper.h:



Classes

- class gdcm::SurfaceHelper

  SurfaceHelper.

Namespaces

- namespace gdcm

11.434 gdcmSurfaceHelper.h

Go to the documentation of this file.

```
1 /*=========================================================================
2
3 Program:  GDCM (Grassroots DICOM). A DICOM library
4
5 Copyright (c) 2006-2017 Mathieu Malaterre
6 All rights reserved.
7 See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.
8
9 This software is distributed WITHOUT ANY WARRANTY; without even
10 the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
11 PURPOSE.  See the above copyright notice for more information.
12
13 =========================================================================*/
14 #ifndef GDCMSURFACEHELPER_H
15 #define GDCMSURFACEHELPER_H
16
17 #include "gdcmTypes.h"  // for GDCM_EXPORT
18
19 #include <vector>
20 #include <iostream>
21
22 namespace gdcm
23 {
24
29 class GDCM_EXPORT SurfaceHelper
30 {
31 public:
32
```

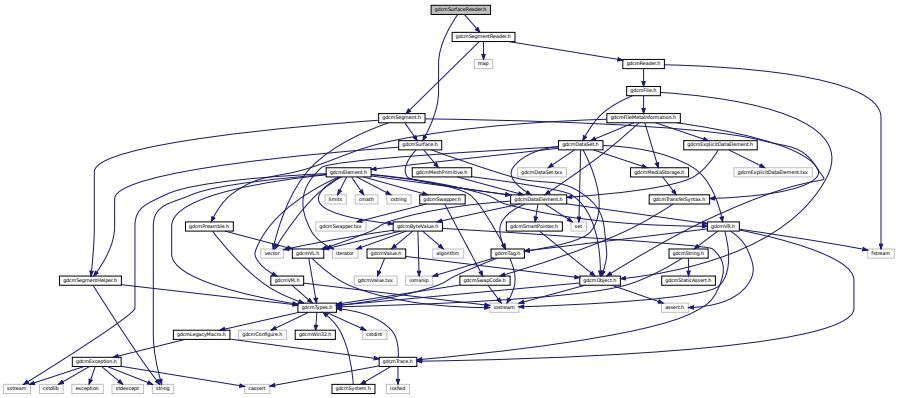
```
33   typedef std::vector< unsigned short > ColorArray;
34
46   template <typename T, typename U>
47   static unsigned short RGBToRecommendedDisplayGrayscale(const std::vector<T> & RGB,
48                                                          const U rangeMax = 255);
60   template <typename T, typename U>
61   static ColorArray RGBToRecommendedDisplayCIELab(const std::vector<T> & RGB,
62                                                   const U rangeMax = 255);
74   template <typename T, typename U>
75   static std::vector<T> RecommendedDisplayCIELabToRGB(const ColorArray & CIELab,
76                                                       const U rangeMax = 255);
87   template <typename U>
88   static std::vector<float> RecommendedDisplayCIELabToRGB(const ColorArray & CIELab,
89                                                       const U rangeMax = 255);
90
91 private:
92
93   static std::vector< float > RGBToXYZ(const std::vector<float> & RGB);
94
95   static std::vector< float > XYZToRGB(const std::vector<float> & XYZ);
96
97   static std::vector< float > XYZToCIELab(const std::vector<float> & XYZ);
98
99   static std::vector< float > CIELabToXYZ(const std::vector<float> & CIELab);
100 };
101
102 template <typename T, typename U>
103 unsigned short SurfaceHelper::RGBToRecommendedDisplayGrayscale(const std::vector<T> & RGB,
104                                                                const U rangeMax/* = 255*/)
105 {
106   assert(RGB.size() > 2);
107
108   unsigned short Grayscale = 0;
109
110   const float inverseRangeMax = 1.0f / (float) rangeMax;
111
112   // 0xFFFF "=" 255 "=" white
113   Grayscale = (unsigned short) ((0.2989 * RGB[0] + 0.5870 * RGB[1] + 0.1140 * RGB[2])
114                                 * inverseRangeMax // Convert to range 0-1
115                                 * 0xFFFF);        // Convert to range 0x0000-0xFFFF
116
117   return Grayscale;
118 }
119
120 template <typename T, typename U>
121 SurfaceHelper::ColorArray SurfaceHelper::RGBToRecommendedDisplayCIELab(const std::vector<T> & RGB,
122                                                                        const U rangeMax/* = 255*/)
123 {
124   assert(RGB.size() > 2);
125
126   ColorArray CIELab(3);
127   std::vector<float> tmp(3);
128
129   // Convert to range 0-1
130   const float inverseRangeMax = 1.0f / (float) rangeMax;
131   tmp[0] = (float) (RGB[0] * inverseRangeMax);
132   tmp[1] = (float) (RGB[1] * inverseRangeMax);
133   tmp[2] = (float) (RGB[2] * inverseRangeMax);
134
135   tmp = SurfaceHelper::XYZToCIELab( SurfaceHelper::RGBToXYZ( tmp ) );
136
137   // Convert to range 0x0000-0xFFFF
138   // 0xFFFF "=" 127, 0x8080 "=" 0, 0x0000 "=" -128
139   CIELab[0] = (unsigned short) (          0xFFFF           * (tmp[0]*0.01f));
140   if(tmp[1] >= -128 && tmp[1] <= 0)
141   {
142       CIELab[1] = (unsigned short)(((float)(0x8080)/128.0f)*tmp[1] + ((float)0x8080));
143   }
144   else if(tmp[1] <= 127 && tmp[1] > 0)
145   {
146       CIELab[1] = (unsigned short)(((float)(0xFFFF - 0x8080)/127.0f)*tmp[1] + (float)(0x8080));
147   }
148   if(tmp[2] >= -128 && tmp[2] <= 0)
149   {
150       CIELab[2] = (unsigned short)(((float)0x8080/128.0f)*tmp[2] + ((float)0x8080));
151   }
152   else if(tmp[2] <= 127 && tmp[2] > 0)
153   {
154       CIELab[2] = (unsigned short)(((float)(0xFFFF - 0x8080)/127.0f)*tmp[2] + (float)(0x8080));
155   }
156
```

```
157   return CIELab;
158 }
159
160 template <typename T, typename U>
161 std::vector<T> SurfaceHelper::RecommendedDisplayCIELabToRGB(const ColorArray & CIELab,
162                                                             const U rangeMax/* = 255*/)
163 {
164   assert(CIELab.size() > 2);
165
166   std::vector<T> RGB(3);
167   std::vector<float> tmp(3);
168
169   // Convert to range 0-1
170
171   tmp[0] = 100.0f*CIELab[0] /(float)(0xFFFF);
172   if(CIELab[1] <= 0x8080)
173   {
174       tmp[1] = (float)(((CIELab[1] - 0x8080) * 128.0f)/(float)0x8080);
175   }
176   else
177   {
178       tmp[1] = (float)((CIELab[1]-0x8080)*127.0f / (float)(0xFFFF - 0x8080));
179   }
180   if(CIELab[2] <= 0x8080)
181   {
182       tmp[2] = (float)(((CIELab[2] - 0x8080) * 128.0f)/(float)0x8080);
183   }
184   else
185   {
186       tmp[2] = (float)((CIELab[2]-0x8080)*127.0f / (float)(0XFFFF - 0x8080));
187   }
188
189   tmp = SurfaceHelper::XYZToRGB( SurfaceHelper::CIELabToXYZ( tmp ) );
190
191   // Convert to range 0-rangeMax
192   RGB[0] = (T) (tmp[0] * rangeMax);
193   RGB[1] = (T) (tmp[1] * rangeMax);
194   RGB[2] = (T) (tmp[2] * rangeMax);
195
196   return RGB;
197 }
198
199 template <typename U>
200 std::vector<float> SurfaceHelper::RecommendedDisplayCIELabToRGB(const ColorArray & CIELab,
201                                                                 const U rangeMax/* = 255*/)
202 {
203   return RecommendedDisplayCIELabToRGB<float>(CIELab, rangeMax);
204 }
205
206 } // end namespace gdcm
207
208 #endif // GDCMSURFACEHELPER_H
```

11.435 gdcmSurfaceReader.h File Reference

```
#include <gdcmSegmentReader.h>
#include <gdcmSurface.h>
```

Include dependency graph for gdcmSurfaceReader.h:



Classes

- class gdcm::SurfaceReader

  This class defines a SURFACE IE reader.

Namespaces

- namespace gdcm

11.436 gdcmSurfaceReader.h

Go to the documentation of this file.

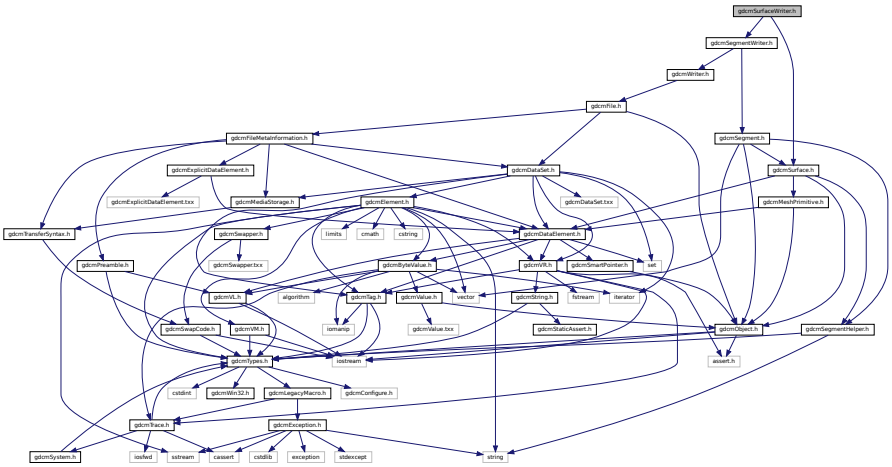
```
1 /*=========================================================================
2
3 Program:  GDCM (Grassroots DICOM). A DICOM library
4
5 Copyright (c) 2006-2011 Mathieu Malaterre
6 All rights reserved.
7 See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.
8
9 This software is distributed WITHOUT ANY WARRANTY; without even
10 the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
11 PURPOSE.  See the above copyright notice for more information.
12
13 =========================================================================*/
14 #ifndef GDCMSURFACEREADER_H
15 #define GDCMSURFACEREADER_H
16
17 #include <gdcmSegmentReader.h>
18 #include <gdcmSurface.h>
19
20 namespace gdcm
21 {
22
29 class GDCM_EXPORT SurfaceReader :  public SegmentReader
30 {
31 public:
32     SurfaceReader();
33
34     ~SurfaceReader() override;
35
```

```
37     bool Read() override;
38
39     unsigned long GetNumberOfSurfaces() const;
40
41   protected:
42
43     bool ReadSurfaces();
44
45     bool ReadSurface(const Item & surfaceItem, const unsigned long idx);
46
47     bool ReadPointMacro(SmartPointer< Surface > surface, const DataSet & surfaceDS);
48 };
49
50 }
51
52 #endif // GDCMSURFACEREADER_H
```

11.437 gdcmSurfaceWriter.h File Reference

```
#include <gdcmSegmentWriter.h>
#include <gdcmSurface.h>
```

Include dependency graph for gdcmSurfaceWriter.h:



Classes

- class gdcm::SurfaceWriter

  This class defines a SURFACE IE writer.

Namespaces

- namespace gdcm

11.438 gdcmSurfaceWriter.h

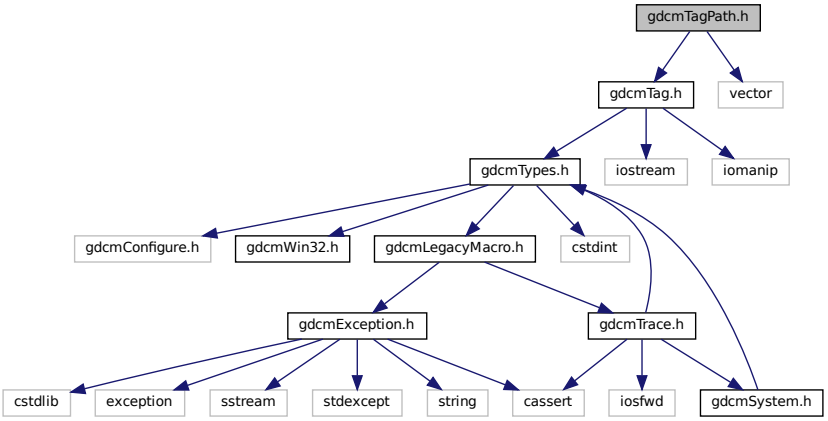
Go to the documentation of this file.

```
1 /*=========================================================================
2
3 Program:  GDCM (Grassroots DICOM). A DICOM library
4
5 Copyright (c) 2006-2011 Mathieu Malaterre
6 All rights reserved.
7 See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.
8
9 This software is distributed WITHOUT ANY WARRANTY; without even
10 the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
11 PURPOSE.  See the above copyright notice for more information.
12
13 =========================================================================*/
14 #ifndef GDCMSURFACEWRITER_H
15 #define GDCMSURFACEWRITER_H
16
17 #include <gdcmSegmentWriter.h>
18 #include <gdcmSurface.h>
19
20 namespace gdcm
21 {
22
29 class GDCM_EXPORT SurfaceWriter :  public SegmentWriter
30 {
31 public:
32     SurfaceWriter();
33
34     ~SurfaceWriter() override;
35
36 //    const Surface & GetSurface() const { return *SurfaceData; }
37 //    Surface & GetSurface() { return *SurfaceData; }
38 //    void SetSurface(Surface const & segment);
39
41     bool Write() override; // Execute()
42
43     unsigned long GetNumberOfSurfaces();
44     void SetNumberOfSurfaces(const unsigned long nb);
45
46 protected:
47
48     bool PrepareWrite();
49
50     void ComputeNumberOfSurfaces();
51
52     bool PrepareWritePointMacro(SmartPointer< Surface > surface,
53                                 DataSet & surfaceDS,
54                                 const TransferSyntax & ts);
55
56     //0066 0001 UL 1 Number of Surfaces
57     unsigned long NumberOfSurfaces;
58 };
59
60 }
61
62 #endif // GDCMSURFACEWRITER_H
```

11.439 gdcmTagPath.h File Reference

```
#include "gdcmTag.h"
#include <vector>
```

Include dependency graph for gdcmTagPath.h:



Classes

- class gdcm::TagPath

  class to handle a path of tag.

Namespaces

- namespace gdcm

11.440 gdcmTagPath.h

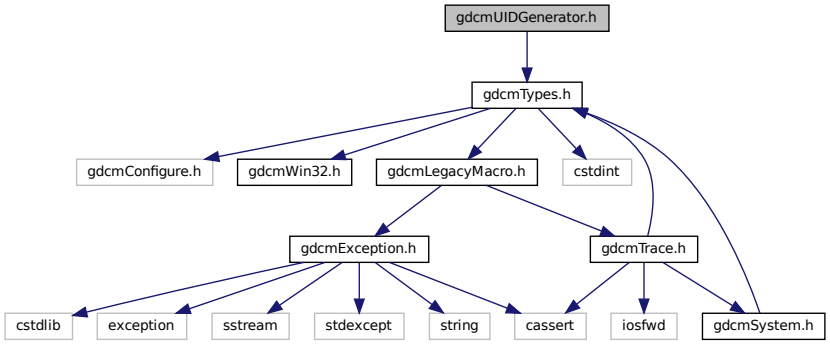
Go to the documentation of this file.

```
1 /*=========================================================================
2
3 Program:  GDCM (Grassroots DICOM). A DICOM library
4
5 Copyright (c) 2006-2011 Mathieu Malaterre
6 All rights reserved.
7 See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.
8
9 This software is distributed WITHOUT ANY WARRANTY; without even
10 the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
11 PURPOSE.  See the above copyright notice for more information.
12
13 =========================================================================*/
14 #ifndef GDCMTAGPATH_H
15 #define GDCMTAGPATH_H
16
17 #include "gdcmTag.h"
18
19 #include <vector>
20
21 namespace gdcm
22 {
23
30 class GDCM_EXPORT TagPath
```

```
31 {
32 public:
33   TagPath();
34   ~TagPath();
35   void Print(std::ostream &) const;
36
41   bool ConstructFromString(const char *path);
42
44   static bool IsValid(const char *path);
45
47   bool ConstructFromTagList(Tag const *l, unsigned int n);
48
49   bool Push(Tag const & t);
50   bool Push(unsigned int itemnum);
51
52 private:
53   std::vector<Tag> Path;
54 };
55
56 } // end namespace gdcm
57
58 #endif //GDCMTAGPATH_H
```

11.441 gdcmUIDGenerator.h File Reference

`#include "gdcmTypes.h"`
Include dependency graph for gdcmUIDGenerator.h:



Classes

- class gdcm::UIDGenerator

  Class for generating unique UID.

Namespaces

- namespace gdcm

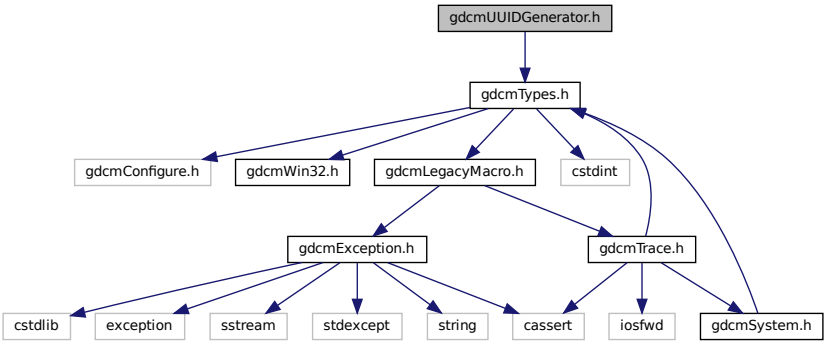
11.442 gdcmUIDGenerator.h

Go to the documentation of this file.

```
1 /*=========================================================================
2
3 Program:  GDCM (Grassroots DICOM). A DICOM library
4
5 Copyright (c) 2006-2011 Mathieu Malaterre
6 All rights reserved.
7 See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.
8
9 This software is distributed WITHOUT ANY WARRANTY; without even
10 the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
11 PURPOSE.  See the above copyright notice for more information.
12
13 =========================================================================*/
14 #ifndef GDCMUIDGENERATOR_H
15 #define GDCMUIDGENERATOR_H
16
17 #include "gdcmTypes.h"
18
19 namespace gdcm
20 {
21
30 class GDCM_EXPORT UIDGenerator
31 {
32 public:
34   UIDGenerator():Unique() {}
35
36   // Function to override the GDCM root with a user one:
37   // WARNING: This need to be a valid root, otherwise call will fail
38   // Implementation note.  According to DICOM standard PS 3.5, Section 9 :
39   // Unique Identifiers (UIDs), we have:
40   /*
41 ...
42 The <org root> portion of the UID uniquely identifies an organization, (i.e., manufacturer, research
43 organization, NEMA, etc.), and is composed of a number of numeric components as defined by ISO 8824.
44 The <suffix> portion of the UID is also composed of a number of numeric components, and shall be
45 unique within the scope of the <org root>.  This implies that the organization identified in the <org root>
    is
46 responsible for guaranteeing <suffix> uniqueness by providing registration policies.  These policies shall
47 guarantee <suffix> uniqueness for all UID's created by that organization.  Unlike the <org root>, which may
48 be common for UID's in an organization, the <suffix> shall take different unique values between different
49 UID's that identify different objects.
50 ...
51 */
57   static void SetRoot(const char * root);
58   static const char *GetRoot();
59
66   const char* Generate();
67
70   static bool IsValid(const char *uid);
71
73   static const char *GetGDCMUID(); // who would want that in the public API ??
74
75 protected:
76   static bool GenerateUUID(unsigned char *uuid_data);
77
78 private:
79   static const char GDCM_UID[];
80   static std::string Root;
81   static std::string EncodedHardwareAddress;
82   std::string Unique; // Buffer
83 };
84
85
86 } // end namespace gdcm
87
88 #endif //GDCMUIDGENERATOR_H
```

11.443 gdcmUUIDGenerator.h File Reference

`#include "gdcmTypes.h"`
Include dependency graph for gdcmUUIDGenerator.h:



Classes

- class gdcm::UUIDGenerator

  Class for generating unique UUID.

Namespaces

- namespace gdcm

11.444 gdcmUUIDGenerator.h

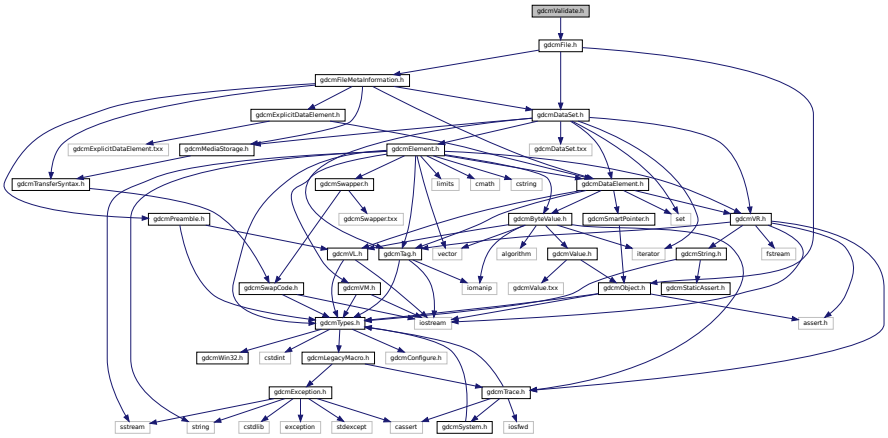
Go to the documentation of this file.

```
1 /*=========================================================================
2
3 Program:  GDCM (Grassroots DICOM). A DICOM library
4
5 Copyright (c) 2006-2011 Mathieu Malaterre
6 All rights reserved.
7 See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.
8
9 This software is distributed WITHOUT ANY WARRANTY; without even
10 the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
11 PURPOSE.  See the above copyright notice for more information.
12
13 =========================================================================*/
14 #ifndef GDCMUUIDGENERATOR_H
15 #define GDCMUUIDGENERATOR_H
16
17 #include "gdcmTypes.h"
18
19 namespace gdcm
20 {
21
26 class GDCM_EXPORT UUIDGenerator
```

```
27 {
28 public:
31   const char* Generate();
32
34   static bool IsValid(const char *uid);
35
36 private:
37   std::string Unique; // Buffer
38 };
39
40 } // end namespace gdcm
41
42 #endif //GDCMUUIDGENERATOR_H
```

11.445 gdcmValidate.h File Reference

`#include "gdcmFile.h"`
Include dependency graph for gdcmValidate.h:



Classes

- class gdcm::Validate

  Validate class.

Namespaces

- namespace gdcm

11.446 gdcmValidate.h

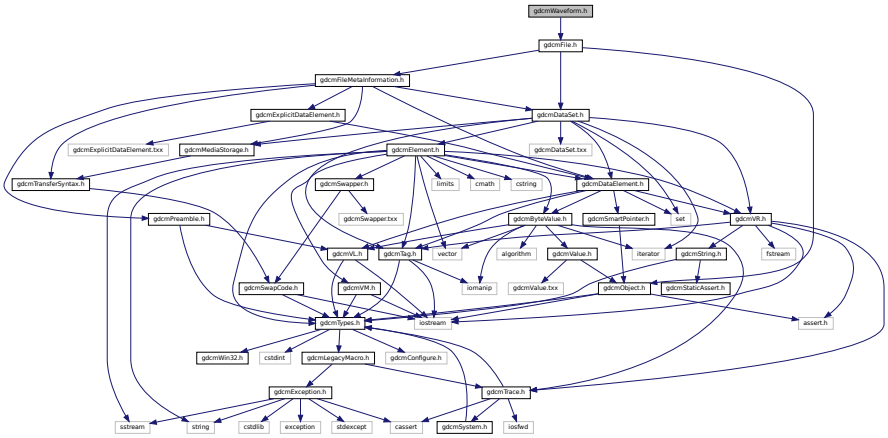
Go to the documentation of this file.

```
1 /*=========================================================================
2
3 Program:  GDCM (Grassroots DICOM). A DICOM library
4
5 Copyright (c) 2006-2011 Mathieu Malaterre
6 All rights reserved.
7 See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.
8
9 This software is distributed WITHOUT ANY WARRANTY; without even
10 the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
11 PURPOSE.  See the above copyright notice for more information.
12
13 =========================================================================*/
14 #ifndef GDCMVALIDATE_H
15 #define GDCMVALIDATE_H
16
17 #include "gdcmFile.h"
18
19 namespace gdcm
20 {
21
25 class GDCM_EXPORT Validate
26 {
27 public:
28   Validate();
29   ~Validate();
30
31   void SetFile(File const &f) { F = &f; }
32   const File& GetValidatedFile() { return V; }
33
34   void Validation();
35
36 protected:
37   const File *F;
38   File V; // Validated file
39 };
40
41 } // end namespace gdcm
42
43 #endif //GDCMVALIDATE_H
```

11.447 gdcmWaveform.h File Reference

`#include "gdcmFile.h"`

Include dependency graph for gdcmWaveform.h:



Classes

- class gdcm::Waveform

  Waveform class.

Namespaces

- namespace gdcm

11.448 gdcmWaveform.h

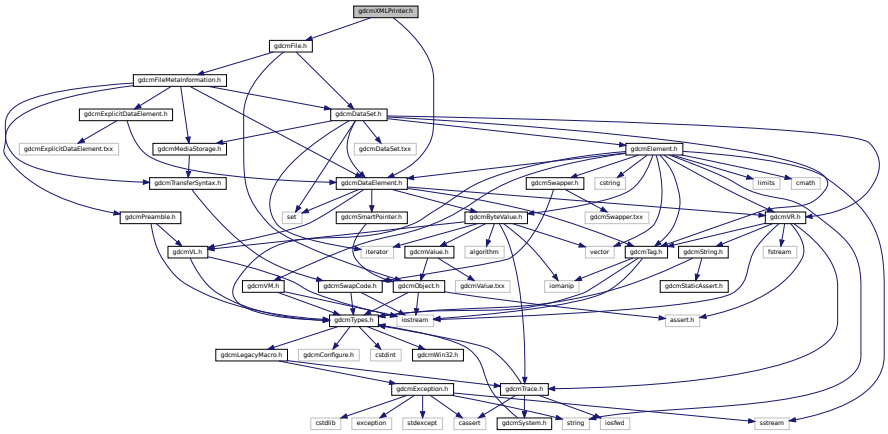
Go to the documentation of this file.

```
1 /*=========================================================================
2
3 Program:  GDCM (Grassroots DICOM). A DICOM library
4
5 Copyright (c) 2006-2011 Mathieu Malaterre
6 All rights reserved.
7 See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.
8
9 This software is distributed WITHOUT ANY WARRANTY; without even
10 the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
11 PURPOSE.  See the above copyright notice for more information.
12
13 =========================================================================*/
14 #ifndef GDCMWAVEFORM_H
15 #define GDCMWAVEFORM_H
16
17 #include "gdcmFile.h"
18
19 namespace gdcm
20 {
24 class GDCM_EXPORT Waveform
25 {
26 public:
27   Waveform() = default;
28
29 private:
30 };
31
32 } // end namespace gdcm
33
34 #endif //GDCMWAVEFORM_H
```

11.449 gdcmXMLPrinter.h File Reference

```
#include "gdcmFile.h"
#include "gdcmDataElement.h"
```

Include dependency graph for gdcmXMLPrinter.h:



Classes

- class gdcm::XMLPrinter

Namespaces

- namespace gdcm

11.450 gdcmXMLPrinter.h

Go to the documentation of this file.

```
1 /*=========================================================================
2
3 Program:  GDCM (Grassroots DICOM). A DICOM library
4
5 Copyright (c) 2006-2011 Mathieu Malaterre
6 All rights reserved.
7 See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.
8
9 This software is distributed WITHOUT ANY WARRANTY; without even
10 the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
11 PURPOSE.  See the above copyright notice for more information.
12
13 =========================================================================*/
14 #ifndef GDCMXMLPRINTER_H
15 #define GDCMXMLPRINTER_H
16
17 /*
18
19 The Normative version of the XML Schema for the Native DICOM Model follows:
20
21
22
23 start = element NativeDicomModel { DicomDataSet }
24
25 # A DICOM Data Set is as defined in PS3.5.  It does not appear
26 # as an XML Element, since it does not appear in the binary encoded
27 # DICOM objects.  It exists here merely as a documentation aid.
28
```

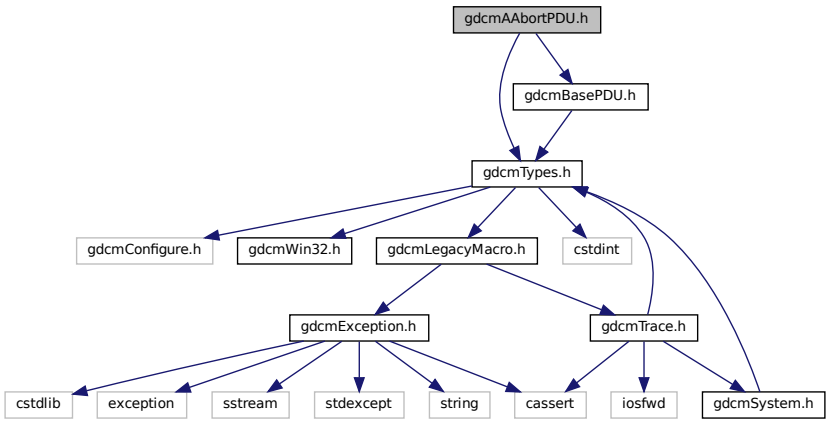
```
29 DicomDataSet = DicomAttribute*
30 DicomAttribute = element DicomAttribute {
31 Tag, VR, Keyword?, PrivateCreator?,
32 ( BulkData | Value+ | Item+ | PersonName+ )?
33 }
34
35 BulkData = element BulkData{ UUID }
36 Value = element Value { Number, xsd:string }
37 Item = element Item { Number, DicomDataSet }
38 PersonName = element PersonName {
39 Number,
40 element SingleByte { NameComponents }?,
41 element Ideographic { NameComponents }?,
42 element Phonetic
43 { NameComponents }?
44 }
45
46 NameComponents =
47 element FamilyName {xsd:string}?,
48 element GivenName {xsd:string}?,
49 element MiddleName {xsd:string}?,
50 element NamePrefix {xsd:string}?,
51 element NameSuffix {xsd:string}?
52
53 # keyword is the attribute tag from PS3.6
54 # (derived from the DICOM Attribute's name)
55 Keyword = attribute keyword { xsd:token }
56 # canonical XML definition of Hex, with lowercase letters disallowed
57 Tag = attribute tag { xsd:string{ minLength="8" maxLength="8" pattern="[0-9A-F]{8}" } }
58 VR = attribute vr { "AE" | "AS" | "AT"| "CS" | "DA" | "DS" | "DT" | "FL" | "FD"
59 | "IS" | "LO" | "LT" | "OB" | "OF" | "OW" | "PN" | "SH" | "SL"
60 | "SQ" | "SS" | "ST" | "TM" | "UI" | "UL" | "UN" | "US" | "UT" }
61 PrivateCreator = attribute privateCreator{ xsd:string }
62 UUID = attribute uuid { xsd:string }
63 Number = attribute number { xsd:positiveInteger }
64
65
66 */
67
68 #include "gdcmFile.h"
69 #include "gdcmDataElement.h"
70
71 namespace gdcm
72 {
73
74 class DataSet;
75 class DictEntry;
76 class Dicts;
77
78 class GDCM_EXPORT XMLPrinter
79 {
80 public:
81   XMLPrinter();
82   virtual ~XMLPrinter();
83
84   // Set file
85   void SetFile(File const &f) { F = &f; }
86
87
88
89   typedef enum {
90
91     OnlyUUID = 0 ,
92     LOADBULKDATA = 1
93
94   } PrintStyles;
95
96   // Set PrintStyle value
97   void SetStyle(PrintStyles ps)
98   {
99     PrintStyle = ps;
100   }
101
102   // Get PrintStyle value
103   PrintStyles GetPrintStyle()const
104 {
105     return PrintStyle;
106   }
107
108   // Print
109   void Print(std::ostream& os);
```

```
110
111   // Print an individual dataset
112   void PrintDataSet(const DataSet &ds, const TransferSyntax & ts, std::ostream& os);
113
114   //void PrintUID(std::ostream &os);
115
119   virtual void HandleBulkData(const char *uuid, const TransferSyntax &ts,
120     const char *bulkdata, size_t bulklen);
121
122 protected:
123
124   VR PrintDataElement(std::ostream &os, const Dicts &dicts, const DataSet & ds, const DataElement &de, const
      TransferSyntax & ts);
125
126   void PrintSQ(const SequenceOfItems *sqi, const TransferSyntax & ts, std::ostream &os);
127
128   PrintStyles PrintStyle;
129
130   const File *F;
131
132 };
133
134 } // end namespace gdcm
135
136 #endif //GDCMXMLPRINTER_H
```

11.451 gdcmAAbortPDU.h File Reference

```
#include "gdcmTypes.h"
#include "gdcmBasePDU.h"
```

Include dependency graph for gdcmAAbortPDU.h:



Classes

- class gdcm::network::AAbortPDU

  AAbortPDU.

Namespaces

- namespace gdcm
- namespace gdcm::network

11.452 gdcmAAbortPDU.h

Go to the documentation of this file.

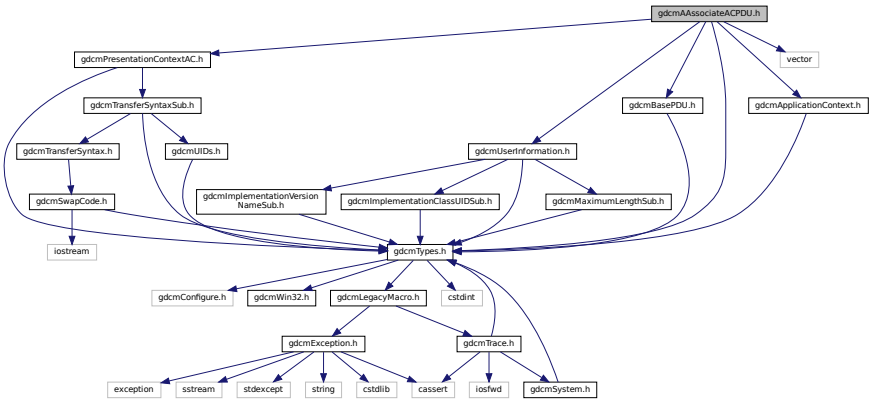
```
1 /*=========================================================================
2
3 Program:  GDCM (Grassroots DICOM). A DICOM library
4
5 Copyright (c) 2006-2011 Mathieu Malaterre
6 All rights reserved.
7 See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.
8
9 This software is distributed WITHOUT ANY WARRANTY; without even
10 the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
11 PURPOSE.  See the above copyright notice for more information.
12
13 =========================================================================*/
14 #ifndef GDCMAABORTPDU_H
15 #define GDCMAABORTPDU_H
16
17 #include "gdcmTypes.h"
18 #include "gdcmBasePDU.h"
19
20 namespace gdcm
21 {
22
23 namespace network
24 {
25
30 class GDCM_EXPORT AAbortPDU :  public BasePDU
31 {
32 public:
33   AAbortPDU();
34   std::istream &Read(std::istream &is) override;
35   const std::ostream &Write(std::ostream &os) const override;
36
38   size_t Size() const override;
39   void Print(std::ostream &os) const override;
40
41   bool IsLastFragment()const override { return true; }
42
43   void SetSource(const uint8_t s);
44   void SetReason(const uint8_t r);
45
46 private:
47   static const uint8_t ItemType; // PDUType ?
48   static const uint8_t Reserved2;
49   uint32_t ItemLength; // PDU Length
50   static const uint8_t Reserved7;
51   static const uint8_t Reserved8;
52   uint8_t Source;
53   uint8_t Reason; // diag
54 };
55
56 } // end namespace network
57
58 } // end namespace gdcm
59
60 #endif //GDCMAABORTPDU_H
```

11.453 gdcmAAssociateACPDU.h File Reference

```
#include "gdcmTypes.h"
#include "gdcmApplicationContext.h"
#include "gdcmPresentationContextAC.h"
#include "gdcmUserInformation.h"
#include "gdcmBasePDU.h"
```

`#include <vector>`

Include dependency graph for gdcmAAssociateACPDU.h:



Classes

- class gdcm::network::AAssociateACPDU

  AAssociateACPDU.

Namespaces

- namespace gdcm
- namespace gdcm::network

11.454 gdcmAAssociateACPDU.h

Go to the documentation of this file.

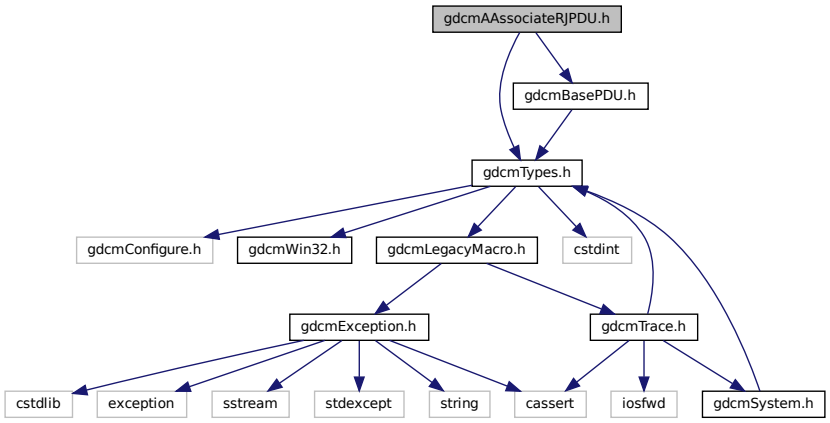
```
1 /*=========================================================================
2
3 Program:  GDCM (Grassroots DICOM). A DICOM library
4
5 Copyright (c) 2006-2011 Mathieu Malaterre
6 All rights reserved.
7 See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.
8
9 This software is distributed WITHOUT ANY WARRANTY; without even
10 the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
11 PURPOSE.  See the above copyright notice for more information.
12
13 =========================================================================*/
14 #ifndef GDCMAASSOCIATEACPDU_H
15 #define GDCMAASSOCIATEACPDU_H
16
17 #include "gdcmTypes.h"
18 #include "gdcmApplicationContext.h"
19 #include "gdcmPresentationContextAC.h"
20 #include "gdcmUserInformation.h"
21 #include "gdcmBasePDU.h"
22
23 #include <vector>
24
```

```
25 namespace gdcm
26 {
27
28 namespace network
29 {
30 class AAssociateRQPDU;
31
37 class AAssociateACPDU :  public BasePDU
38 {
39 public:
40   AAssociateACPDU();
41   std::istream &Read(std::istream &is) override;
42   const std::ostream &Write(std::ostream &os) const override;
43
44   void AddPresentationContextAC( PresentationContextAC const &pcac );
45
46   typedef std::vector<PresentationContextAC>::size_type SizeType;
47   const PresentationContextAC &GetPresentationContextAC( SizeType i ) {
48     assert( !PresContextAC.empty() && i < PresContextAC.size() );
49     return PresContextAC[i];
50   }
51   SizeType GetNumberOfPresentationContextAC()const {
52     return PresContextAC.size();
53   }
54   const UserInformation &GetUserInformation()const { return UserInfo; }
55
56   SizeType Size() const override;
57
58   void Print(std::ostream &os) const override;
59   bool IsLastFragment()const override { return true; }
60
61   void InitFromRQ( AAssociateRQPDU const & rqpdu );
62 protected:
63   friend class AAssociateRQPDU;
64   void SetCalledAETitle(const char calledaetitle[16]);
65   void SetCallingAETitle(const char callingaetitle[16]);
66
67 private:
68   void InitSimple( AAssociateRQPDU const & rqpdu );
69
70 private:
71   static const uint8_t ItemType; // PDUType ?
72   static const uint8_t Reserved2;
73   uint32_t PDULength; // len of
74   static const uint16_t ProtocolVersion;
75   static const uint16_t Reserved9_10;
76
77   // This reserved field shall be sent with a value identical to the value
78   // received in the same field of the A-ASSOCIATE-RQ PDU, but its value
79   // shall not be tested when received.
80   char Reserved11_26[16];
81   // This reserved field shall be sent with a value identical to the value
82   // received in the same field of the A-ASSOCIATE-RQ PDU, but its value
83   // shall not be tested when received.
84   char Reserved27_42[16];
85   // This reserved field shall be sent with a value identical to the value
86   // received in the same field of the A-ASSOCIATE-RQ PDU, but its value
87   // shall not be tested when received.
88   char Reserved43_74[32];
89   /*
90 75-xxx Variable items This variable field shall contain the following items:  one Application
91 Context Item, one or more Presentation Context Item(s) and one User
92 Information Item.  For a complete description of these items see Sections
93 7.1.1.2, 7.1.1.14, and 7.1.1.6.
94 */
95   ApplicationContext AppContext;
96   std::vector<PresentationContextAC>  PresContextAC;
97   UserInformation UserInfo;
98 };
99
100 } // end namespace network
101
102 } // end namespace gdcm
103
104 #endif //GDCMAASSOCIATEACPDU_H
```

11.455 gdcmAAssociateRJPDU.h File Reference

```
#include "gdcmTypes.h"
#include "gdcmBasePDU.h"
```

Include dependency graph for gdcmAAssociateRJPDU.h:



Classes

- class gdcm::network::AAssociateRJPDU

  AAssociateRJPDU.

Namespaces

- namespace gdcm
- namespace gdcm::network

11.456 gdcmAAssociateRJPDU.h

Go to the documentation of this file.

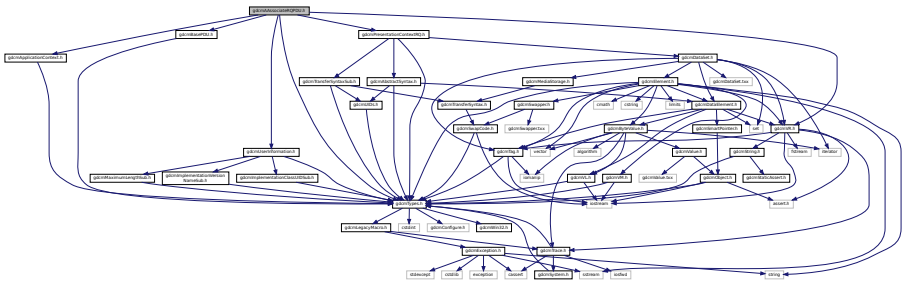
```
1 /*=========================================================================
2
3 Program:  GDCM (Grassroots DICOM). A DICOM library
4
5 Copyright (c) 2006-2011 Mathieu Malaterre
6 All rights reserved.
7 See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.
8
9 This software is distributed WITHOUT ANY WARRANTY; without even
10 the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
11 PURPOSE.  See the above copyright notice for more information.
12
13 =========================================================================*/
14 #ifndef GDCMAASSOCIATERJPDU_H
15 #define GDCMAASSOCIATERJPDU_H
```

```
16
17 #include "gdcmTypes.h"
18 #include "gdcmBasePDU.h"
19
20 namespace gdcm
21 {
22
23 namespace network
24 {
25
31 class AAssociateRJPDU :  public BasePDU
32 {
33 public:
34   AAssociateRJPDU();
35   std::istream &Read(std::istream &is) override;
36   const std::ostream &Write(std::ostream &os) const override;
37   void Print(std::ostream &os) const override;
38   size_t Size() const override;
39   bool IsLastFragment()const override { return true; }
40 private:
41   static const uint8_t ItemType; // PDUType ?
42   static const uint8_t Reserved2;
43   uint32_t ItemLength; // PDU Length ?
44   static const uint8_t Reserved8;
45   uint8_t Result;
46   uint8_t Source;
47   uint8_t Reason; // diag ?
48 };
49
50 } // end namespace network
51
52 } // end namespace gdcm
53
54 #endif //GDCMAASSOCIATERJPDU_H
```

11.457 gdcmAAssociateRQPDU.h File Reference

```
#include "gdcmTypes.h"
#include "gdcmVR.h"
#include "gdcmApplicationContext.h"
#include "gdcmPresentationContextRQ.h"
#include "gdcmUserInformation.h"
#include "gdcmBasePDU.h"
```

Include dependency graph for gdcmAAssociateRQPDU.h:



Classes

- class gdcm::network::AAssociateRQPDU

  AAssociateRQPDU.

Namespaces

- namespace gdcm
- namespace gdcm::network

11.458 gdcmAAssociateRQPDU.h

Go to the documentation of this file.

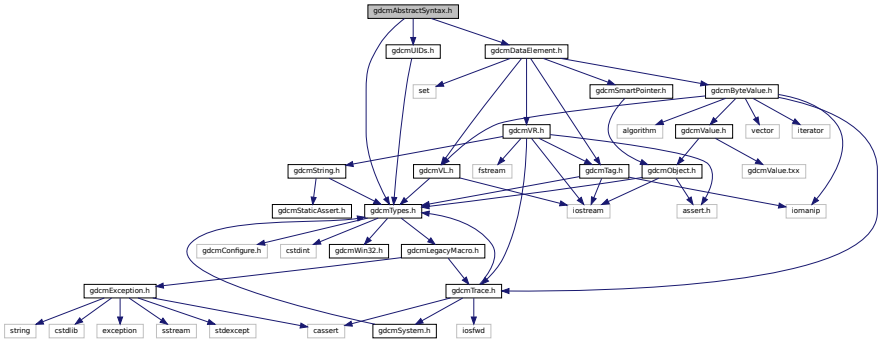
```
1 /*=========================================================================
2
3 Program:  GDCM (Grassroots DICOM). A DICOM library
4
5 Copyright (c) 2006-2011 Mathieu Malaterre
6 All rights reserved.
7 See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.
8
9 This software is distributed WITHOUT ANY WARRANTY; without even
10 the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
11 PURPOSE.  See the above copyright notice for more information.
12
13 =========================================================================*/
14 #ifndef GDCMAASSOCIATERQPDU_H
15 #define GDCMAASSOCIATERQPDU_H
16
17 #include "gdcmTypes.h"
18 #include "gdcmVR.h" // AEComp
19 #include "gdcmApplicationContext.h"
20 #include "gdcmPresentationContextRQ.h"
21 #include "gdcmUserInformation.h"
22 #include "gdcmBasePDU.h"
23
24 namespace gdcm
25 {
26
27 namespace network
28 {
29
30 class AAssociateACPDU;
35 class AAssociateRQPDU :  public BasePDU
36 {
37 public:
38   AAssociateRQPDU();
39   std::istream &Read(std::istream &is) override;
40   const std::ostream &Write(std::ostream &os) const override;
41   size_t Size() const override;
42   void AddPresentationContext( PresentationContextRQ const &pc );
43
45   void SetCalledAETitle(const char calledaetitle[16]);
46   std::string GetCalledAETitle()const { return std::string(CalledAETitle,16); }
47
49   void SetCallingAETitle(const char callingaetitle[16]);
50   std::string GetCallingAETitle()const { return std::string(CallingAETitle,16); }
51
53   static bool IsAETitleValid(const char title[16]);
54
57   //void InitFromRQ( AAssociateACPDU & acpdu );
58
59   void Print(std::ostream &os) const override;
60
61   AAssociateRQPDU(const AAssociateRQPDU &pdu):BasePDU(pdu)
62     {
63     assert( 0 );
64     }
65   //this function fails to compile on windows.
66 //  AAssociateRQPDU &operator=(const AAssociateRQPDU &_val)
67 //    {
68 //    assert( 0 );
69 //    }
70
71   typedef std::vector<PresentationContextRQ>::size_type SizeType;
72   SizeType GetNumberOfPresentationContext()const {
73     return PresContext.size();
74   }
```

```
75   PresentationContextRQ const &GetPresentationContext(SizeType i)const {
76     assert( !PresContext.empty() && i < PresContext.size() );
77     return PresContext[i];
78   }
79   typedef std::vector<PresentationContextRQ> PresentationContextArrayType;
80   PresentationContextArrayType const &GetPresentationContexts() { return PresContext; }
81
82   const PresentationContextRQ *GetPresentationContextByID(uint8_t i) const;
83   const PresentationContextRQ *GetPresentationContextByAbstractSyntax(AbstractSyntax const & absyn ) const;
84   bool IsLastFragment()const override { return true; }
85
86   const UserInformation & GetUserInformation()const { return UserInfo; }
87   void SetUserInformation( UserInformation const & ui );
88
89 protected:
90   friend class AAssociateACPDU;
91   std::string GetReserved43_74() const;
92
93 private:
94   // 1 PDU-type 01H
95   static const uint8_t ItemType; // PDUType ?
96   // 2 Reserved This reserved field shall be sent with a value 00H but not tested to this value when
      received.
97   static const uint8_t Reserved2;
98   /* 3-6 PDU-length This PDU-length shall be the number of bytes from the first byte of the
99 following field to the last byte of the variable field.  It shall be encoded as
100 an unsigned binary number
101 */
102   uint32_t ItemLength; // PDU Length
103   /*
104 7-8 Protocol-version This two byte field shall use one bit to identify each version of the
105 DICOM UL protocol supported by the calling end-system.  This is
106 Version 1 and shall be identified with bit 0 set.  A receiver of this PDU
107 implementing only this version of the DICOM UL protocol shall only test
108 that bit 0 is set.
109 */
110   static const uint16_t ProtocolVersion;
111   /*
112 9-10 Reserved This reserved field shall be sent with a value 0000H but not tested to
113 this value when received.
114 */
115   static const uint16_t Reserved9_10;
116   /*
117 11-26 Called-AE-title Destination DICOM Application Name.  It shall be encoded as 16
118 characters as defined by the ISO 646:1990-Basic G0 Set with leading
119 and trailing spaces (20H) being non-significant.  The value made of 16
120 spaces (20H) meaning "no Application Name specified" shall not be
121 used.  For a complete description of the use of this field, see Section
122 7.1.1.4.
123 */
124   char CalledAETitle[16];
125   /*
126 27-42 Calling-AE-title Source DICOM Application Name.  It shall be encoded as 16
127 characters as defined by the ISO 646:1990-Basic G0 Set with leading
128 and trailing spaces (20H) being non-significant.  The value made of 16
129 spaces (20H) meaning "no Application Name specified" shall not be
130 used.  For a complete description of the use of this field, see Section
131 7.1.1.3.
132 */
133   char CallingAETitle[16];
134   /*
135 43-74 Reserved This reserved field shall be sent with a value 00H for all bytes but not
136 tested to this value when received
137 */
138   char Reserved43_74[32]; // { 0 }
139   /*
140 75-xxx Variable items This variable field shall contain the following items:  one Application
141 Context Item, one or more Presentation Context Items and one User
142 Information Item.  For a complete description of the use of these items
143 see Sections 7.1.1.2, 7.1.1.13, and 7.1.1.6.
144 */
145   ApplicationContext AppContext;
146   std::vector<PresentationContextRQ> PresContext;
147   UserInformation UserInfo;
148 };
149
150 } // end namespace network
151 } // end namespace gdcm
152
153 #endif //GDCMAASSOCIATERQPDU_H
```

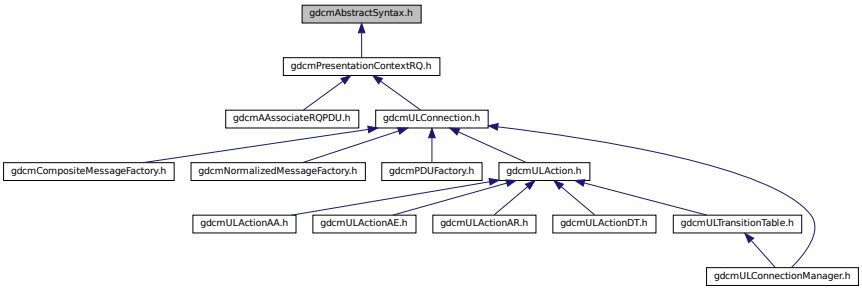
11.459 gdcmAbstractSyntax.h File Reference

```
#include "gdcmTypes.h"
#include "gdcmUIDs.h"
#include "gdcmDataElement.h"
```

Include dependency graph for gdcmAbstractSyntax.h:



This graph shows which files directly or indirectly include this file:



Classes

- class gdcm::network::AbstractSyntax

  AbstractSyntax.

Namespaces

- namespace gdcm
- namespace gdcm::network

11.460 gdcmAbstractSyntax.h

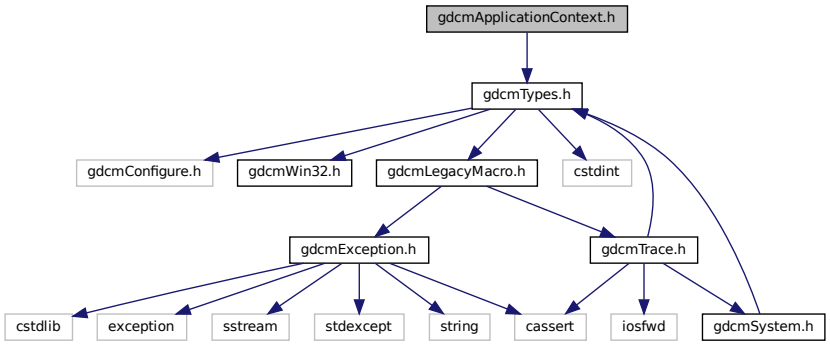
Go to the documentation of this file.

```
1 /*=========================================================================
2
3 Program:  GDCM (Grassroots DICOM). A DICOM library
4
5 Copyright (c) 2006-2011 Mathieu Malaterre
6 All rights reserved.
7 See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.
8
9 This software is distributed WITHOUT ANY WARRANTY; without even
10 the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
11 PURPOSE.  See the above copyright notice for more information.
12
13 =========================================================================*/
14 #ifndef GDCMABSTRACTSYNTAX_H
15 #define GDCMABSTRACTSYNTAX_H
16
17 #include "gdcmTypes.h"
18 #include "gdcmUIDs.h"
19 #include "gdcmDataElement.h"
20
21 namespace gdcm
22 {
23
24 namespace network
25 {
26
32 class AbstractSyntax
33 {
34 public:
35   AbstractSyntax();
36   std::istream &Read(std::istream &is);
37   const std::ostream &Write(std::ostream &os) const;
38
39   void SetName( const char *name ) { UpdateName( name ); }
40   const char *GetName()const { return Name.c_str(); }
41
42   // accept a UIDs::TSType also...
43   void SetNameFromUID( UIDs::TSName tsname );
44   //now that the PresentationContext messes around with UIDs and returns a string
45   //use that string as well.
46   //void SetNameFromUIDString( const std::string& inUIDName );
47
48   size_t Size() const;
49
50   void Print(std::ostream &os) const;
51
52   bool operator==(const AbstractSyntax & as)const
53 {
54     return Name == as.Name;
55     }
56
57   DataElement GetAsDataElement() const;
58
59 private:
60   void UpdateName( const char *name );
61   static const uint8_t ItemType;
62   static const uint8_t Reserved2;
63   uint16_t ItemLength; // len of
64   std::string /*AbstractSyntax*/ Name; // UID
65 };
66
67 } // end namespace network
68 } // end namespace gdcm
69
70 #endif //GDCMABSTRACTSYNTAX_H
```

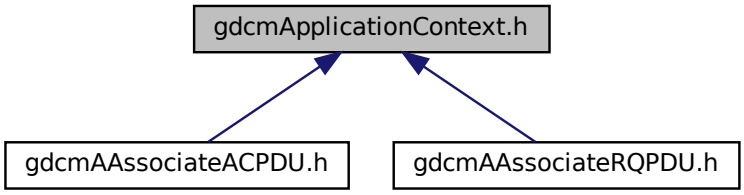
11.461 gdcmApplicationContext.h File Reference

```
#include "gdcmTypes.h"
```

Include dependency graph for gdcmApplicationContext.h:



This graph shows which files directly or indirectly include this file:



Classes

- class gdcm::network::ApplicationContext

  ApplicationContext.

Namespaces

- namespace gdcm
- namespace gdcm::network

11.462 gdcmApplicationContext.h

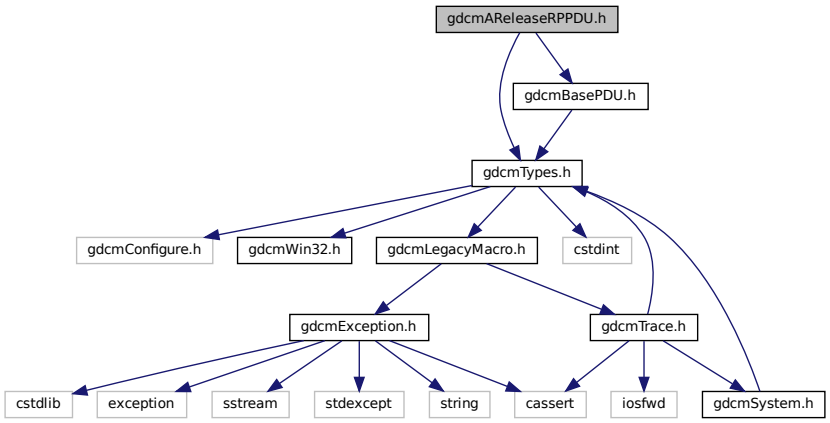
Go to the documentation of this file.

```
1 /*=========================================================================
2
3 Program:  GDCM (Grassroots DICOM). A DICOM library
4
5 Copyright (c) 2006-2011 Mathieu Malaterre
6 All rights reserved.
7 See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.
8
9 This software is distributed WITHOUT ANY WARRANTY; without even
10 the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
11 PURPOSE.  See the above copyright notice for more information.
12
13 =========================================================================*/
14 #ifndef GDCMAPPLICATIONCONTEXT_H
15 #define GDCMAPPLICATIONCONTEXT_H
16
17 #include "gdcmTypes.h"
18
19 namespace gdcm
20 {
21
22 namespace network
23 {
24
32 class ApplicationContext
33 {
34 public:
35   ApplicationContext();
36   std::istream &Read(std::istream &is);
37   const std::ostream &Write(std::ostream &os) const;
38
39   void SetName( const char *name ) { UpdateName( name ); }
40   const char *GetName()const { return Name.c_str(); }
41   size_t Size() const;
42
43   //static const uint8_t GetItemType() { return ItemType; }
44   void Print(std::ostream &os) const;
45
46 private:
47   void UpdateName( const char *name );
48   static const uint8_t ItemType;
49   static const uint8_t Reserved2;
50   uint16_t ItemLength; // len of application context name
51   std::string /*ApplicationContext*/ Name; // UID
52 };
53
54 } // end namespace network
55
56 } // end namespace gdcm
57
58 #endif //GDCMAPPLICATIONCONTEXT_H
```

11.463 gdcmAReleaseRPPDU.h File Reference

```
#include "gdcmTypes.h"
#include "gdcmBasePDU.h"
```

Include dependency graph for gdcmAReleaseRPPDU.h:



Classes

- class gdcm::network::AReleaseRPPDU

  AReleaseRPPDU.

Namespaces

- namespace gdcm
- namespace gdcm::network

11.464 gdcmAReleaseRPPDU.h

Go to the documentation of this file.

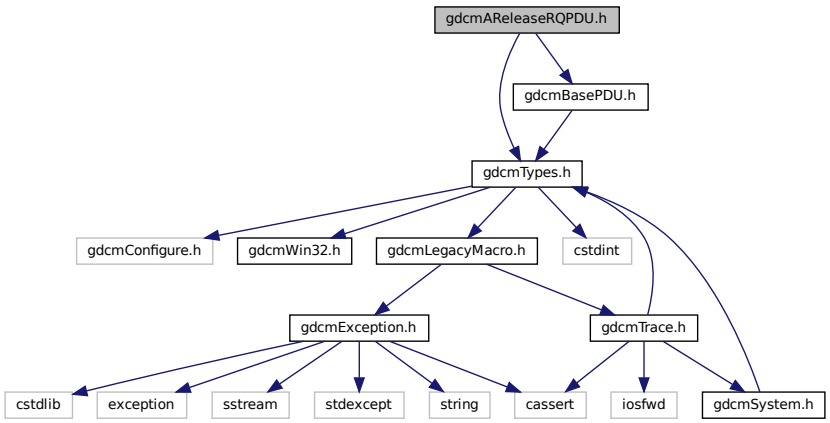
```
1 /*=========================================================================
2
3 Program:  GDCM (Grassroots DICOM). A DICOM library
4
5 Copyright (c) 2006-2011 Mathieu Malaterre
6 All rights reserved.
7 See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.
8
9 This software is distributed WITHOUT ANY WARRANTY; without even
10 the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
11 PURPOSE.  See the above copyright notice for more information.
12
13 =========================================================================*/
14 #ifndef GDCMARELEASERPPDU_H
15 #define GDCMARELEASERPPDU_H
16
17 #include "gdcmTypes.h"
18 #include "gdcmBasePDU.h"
19
20 namespace gdcm
21 {
22
23 namespace network
```

```
24 {
25
31 class AReleaseRPPDU :  public BasePDU
32 {
33 public:
34   AReleaseRPPDU();
35   std::istream &Read(std::istream &is) override;
36   const std::ostream &Write(std::ostream &os) const override;
37   size_t Size() const override;
38   void Print(std::ostream &os) const override;
39   bool IsLastFragment()const override { return true; }
40 private:
41   static const uint8_t ItemType; // PDUType ?
42   static const uint8_t Reserved2;
43   uint32_t ItemLength; // PDU Length
44   static const uint32_t Reserved7_10;
45 };
46
47 } // end namespace network
48
49 } // end namespace gdcm
50
51 #endif //GDCMARELEASERPPDU_H
```

11.465 gdcmAReleaseRQPDU.h File Reference

```
#include "gdcmTypes.h"
#include "gdcmBasePDU.h"
```

Include dependency graph for gdcmAReleaseRQPDU.h:



Classes

- class gdcm::network::AReleaseRQPDU
  AReleaseRQPDU.

Namespaces

- namespace gdcm
- namespace gdcm::network

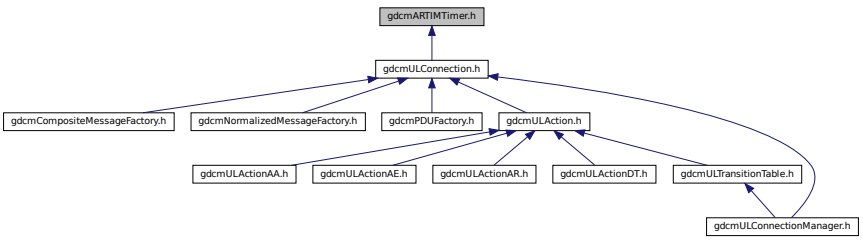
11.466 gdcmAReleaseRQPDU.h

Go to the documentation of this file.

```
1 /*=========================================================================
2
3 Program:  GDCM (Grassroots DICOM). A DICOM library
4
5 Copyright (c) 2006-2011 Mathieu Malaterre
6 All rights reserved.
7 See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.
8
9 This software is distributed WITHOUT ANY WARRANTY; without even
10 the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
11 PURPOSE.  See the above copyright notice for more information.
12
13 =========================================================================*/
14 #ifndef GDCMARELEASERQPDU_H
15 #define GDCMARELEASERQPDU_H
16
17 #include "gdcmTypes.h"
18 #include "gdcmBasePDU.h"
19
20 namespace gdcm
21 {
22
23 namespace network
24 {
25
31 class AReleaseRQPDU :  public BasePDU
32 {
33 public:
34   AReleaseRQPDU();
35   std::istream &Read(std::istream &is) override;
36   const std::ostream &Write(std::ostream &os) const override;
37   size_t Size() const override;
38   void Print(std::ostream &os) const override;
39   bool IsLastFragment()const override { return true; }
40 private:
41   static const uint8_t ItemType; // PDUType ?
42   static const uint8_t Reserved2;
43   uint32_t ItemLength; // PDU Length
44   static const uint32_t Reserved7_10;
45 };
46
47 } // end namespace network
48
49 } // end namespace gdcm
50
51 #endif //GDCMARELEASERQPDU_H
```

11.467 gdcmARTIMTimer.h File Reference

This graph shows which files directly or indirectly include this file:



Classes

- class gdcm::network::ARTIMTimer

  ARTIMTimer.

Namespaces

- namespace gdcm
- namespace gdcm::network

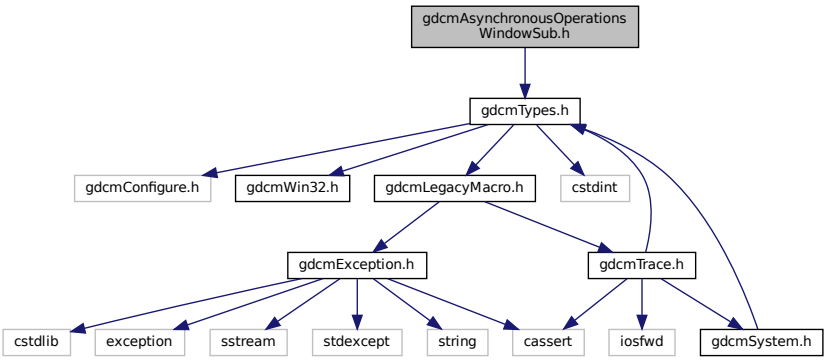
11.468 gdcmARTIMTimer.h

Go to the documentation of this file.

```
1 /*=========================================================================
2 *
3 *  Copyright NumFOCUS
4 *
5 *  Licensed under the Apache License, Version 2.0 (the "License");
6 *  you may not use this file except in compliance with the License.
7 *  You may obtain a copy of the License at
8 *
9 *         http://www.apache.org/licenses/LICENSE-2.0.txt
10 *
11 *  Unless required by applicable law or agreed to in writing, software
12 *  distributed under the License is distributed on an "AS IS" BASIS,
13 *  WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
14 *  See the License for the specific language governing permissions and
15 *  limitations under the License.
16 *
17 *=========================================================================*/
18 #ifndef GDCMARTIMTIMER_H
19 #define GDCMARTIMTIMER_H
20
21 namespace gdcm {
22   namespace network{
38 class ARTIMTimer
39 {
40     private:
41       double mStartTime; //ms timing should be good enough, but there are also
42       //high-resolution timing options.    Those return doubles.    For now,
43       //go with integer timing solutions based on milliseconds (DWORD on windows),
44       //but leave as doubles to ease transitions to other timing methods.
45
46       double mTimeOut;
47       //once GetCurrentTime() -mStartTime > mTimeout, GetHasExpired returns true.
48
49       double GetCurrentTime() const;//a platform-specific implementation of getting the
50       //current time.
51
52     public:
53       ARTIMTimer(); //initiates the start and timeout at -1;
54       void Start(); //'start' the timer by getting the current wall time
55       void Stop();//'stop' the timer by resetting the 'start' to -1;
56       void SetTimeout(double inTimeout);
57       double GetTimeout() const;
58
59       double GetElapsedTime() const;
60
61       bool GetHasExpired() const;
62
63     };
64   }
65 }
66
67 #endif //GDCMARTIMTIMER_H
```

11.469 gdcmAsynchronousOperationsWindowSub.h File Reference

`#include "gdcmTypes.h"`
Include dependency graph for gdcmAsynchronousOperationsWindowSub.h:



Classes

- class gdcm::network::AsynchronousOperationsWindowSub
  AsynchronousOperationsWindowSub.

Namespaces

- namespace gdcm
- namespace gdcm::network

11.470 gdcmAsynchronousOperationsWindowSub.h

Go to the documentation of this file.

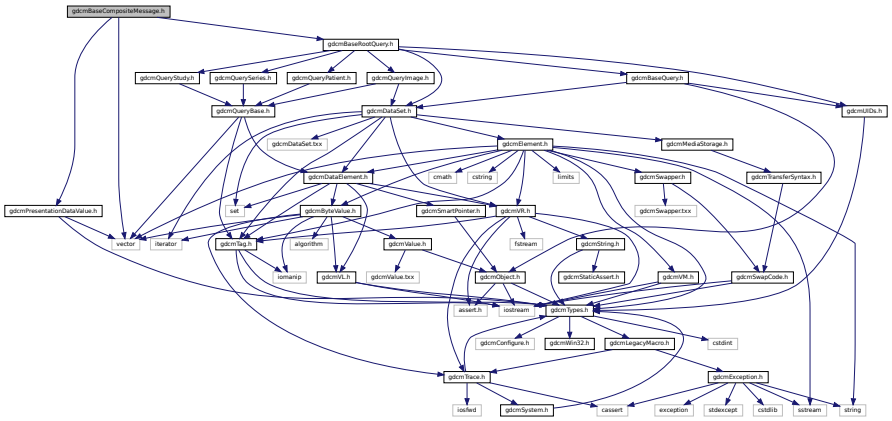
```
1 /*=========================================================================
2
3 Program:  GDCM (Grassroots DICOM). A DICOM library
4
5 Copyright (c) 2006-2011 Mathieu Malaterre
6 All rights reserved.
7 See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.
8
9 This software is distributed WITHOUT ANY WARRANTY; without even
10 the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
11 PURPOSE.  See the above copyright notice for more information.
12
13 =========================================================================*/
14 #ifndef GDCMASYNCHRONOUSOPERATIONSWINDOWSUB_H
15 #define GDCMASYNCHRONOUSOPERATIONSWINDOWSUB_H
16
17 #include "gdcmTypes.h"
18
19 namespace gdcm
20 {
```

```
21
22 namespace network
23 {
24
32 class AsynchronousOperationsWindowSub
33 {
34 public:
35   AsynchronousOperationsWindowSub();
36   std::istream &Read(std::istream &is);
37   const std::ostream &Write(std::ostream &os) const;
38
39   size_t Size() const;
40   void Print(std::ostream &os) const;
41
42 private:
43   static const uint8_t ItemType;
44   static const uint8_t Reserved2;
45   uint16_t ItemLength;
46   uint16_t MaximumNumberOperationsInvoked;
47   uint16_t MaximumNumberOperationsPerformed;
48 };
49
50 } // end namespace network
51
52 } // end namespace gdcm
53
54 #endif // GDCMASYNCHRONOUSOPERATIONSWINDOWSUB_H
```

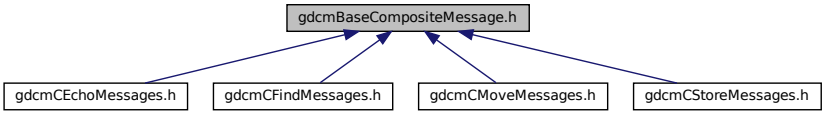
11.471 gdcmBaseCompositeMessage.h File Reference

```
#include "gdcmPresentationDataValue.h"
#include "gdcmBaseRootQuery.h"
#include <vector>
```

Include dependency graph for gdcmBaseCompositeMessage.h:



This graph shows which files directly or indirectly include this file:



Classes

- class gdcm::network::BaseCompositeMessage

  BaseCompositeMessage.

Namespaces

- namespace gdcm
- namespace gdcm::network

11.472 gdcmBaseCompositeMessage.h

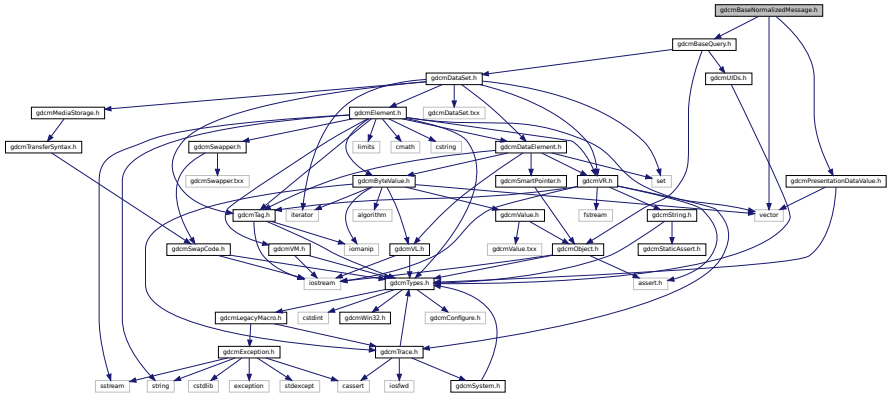
Go to the documentation of this file.

```
1 /*=========================================================================
2 *
3 *  Copyright NumFOCUS
4 *
5 *  Licensed under the Apache License, Version 2.0 (the "License");
6 *  you may not use this file except in compliance with the License.
7 *  You may obtain a copy of the License at
8 *
9 *         http://www.apache.org/licenses/LICENSE-2.0.txt
10 *
11 *  Unless required by applicable law or agreed to in writing, software
12 *  distributed under the License is distributed on an "AS IS" BASIS,
13 *  WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
14 *  See the License for the specific language governing permissions and
15 *  limitations under the License.
16 *
17 *=========================================================================*/
18 #ifndef GDCMBASECOMPOSITEMESSSAGE_H
19 #define GDCMBASECOMPOSITEMESSSAGE_H
20
21 #include "gdcmPresentationDataValue.h"
22 #include "gdcmBaseRootQuery.h"
23
24 #include <vector>
25
26 namespace gdcm
27 {
28   namespace network
29     {
30 class ULConnection;
52 class BaseCompositeMessage
53 {
54     public:
55       virtual ~BaseCompositeMessage() = default;
56       //construct the appropriate pdv and dataset for this message
57       //for instance, setting tag 0x0,0x100 to the appropriate value
58       //the pdv, as described in Annex E of 3.8-2009, is the first byte
59       //of the message (the MessageHeader), and then the subsequent dataset
60       //that describes the operation.
61       virtual std::vector<PresentationDataValue> ConstructPDV(const ULConnection &inConnection,
62         const BaseRootQuery * inRootQuery) = 0;
63     };
64   }
65 }
66 #endif //BASECOMPOSITEMESSSAGE_H
```

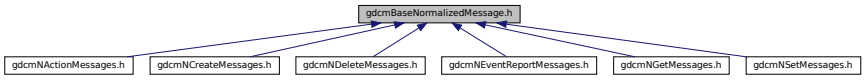
11.473 gdcmBaseNormalizedMessage.h File Reference

```
#include "gdcmPresentationDataValue.h"
#include "gdcmBaseQuery.h"
#include <vector>
```

Include dependency graph for gdcmBaseNormalizedMessage.h:



This graph shows which files directly or indirectly include this file:



Classes

- class gdcm::network::BaseNormalizedMessage

  BaseNormalizedMessage.

Namespaces

- namespace gdcm
- namespace gdcm::network

11.474 gdcmBaseNormalizedMessage.h

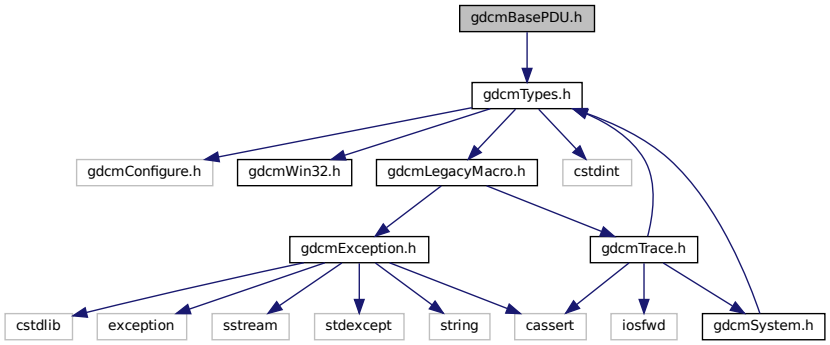
Go to the documentation of this file.

```
1 /*=========================================================================
2
3 Program:  GDCM (Grassroots DICOM). A DICOM library
4
5 Copyright (c) 2006-2014 Mathieu Malaterre
6 All rights reserved.
7 See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.
8
9 This software is distributed WITHOUT ANY WARRANTY; without even
10 the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
11 PURPOSE.  See the above copyright notice for more information.
12
13 =========================================================================*/
14 #ifndef GDCMBASENORMALIZEDMESSSAGE_H
15 #define GDCMBASENORMALIZEDMESSSAGE_H
16
17 #include "gdcmPresentationDataValue.h"
18 #include "gdcmBaseQuery.h"
19
20 #include <vector>
21
22 namespace gdcm
23 {
24   namespace network
25     {
26 class ULConnection;
49 class BaseNormalizedMessage
50 {
51     public:
52       virtual ~BaseNormalizedMessage() = default;
53       //construct the appropriate pdv and dataset for this message
54       //for instance, setting tag 0x0,0x100 to the appropriate value
55       //the pdv, as described in Annex E of 3.8-2009, is the first byte
56       //of the message (the MessageHeader), and then the subsequent dataset
57       //that describes the operation.
58       virtual std::vector<PresentationDataValue> ConstructPDV(  const ULConnection &inConnection,
59                                 const BaseQuery * inQuery) = 0;
60     };
61   }
62 }
63 #endif //GDCMBASENORMALIZEDMESSSAGE_H
```

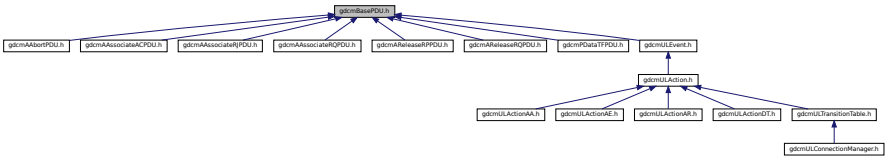
11.475 gdcmBasePDU.h File Reference

`#include "gdcmTypes.h"`

Include dependency graph for gdcmBasePDU.h:



This graph shows which files directly or indirectly include this file:



Classes

- class gdcm::network::BasePDU

  BasePDU.

Namespaces

- namespace gdcm
- namespace gdcm::network

11.476 gdcmBasePDU.h

Go to the documentation of this file.

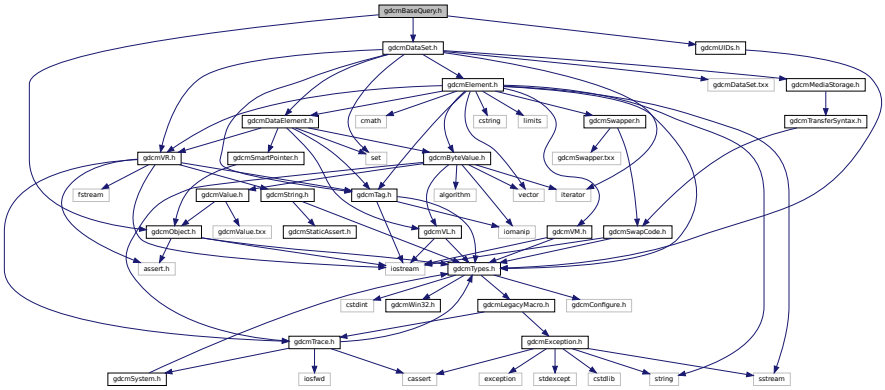
```
1 /*=========================================================================
2 *
3 *  Copyright NumFOCUS
4 *
5 *  Licensed under the Apache License, Version 2.0 (the "License");
6 *  you may not use this file except in compliance with the License.
7 *  You may obtain a copy of the License at
8 *
9 *         http://www.apache.org/licenses/LICENSE-2.0.txt
10 *
11 *  Unless required by applicable law or agreed to in writing, software
12 *  distributed under the License is distributed on an "AS IS" BASIS,
13 *  WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
14 *  See the License for the specific language governing permissions and
15 *  limitations under the License.
16 *
17 *=========================================================================*/
18 #ifndef GDCMBASEPDU_H
19 #define GDCMBASEPDU_H
20
21 #include "gdcmTypes.h"
22
23 namespace gdcm
24 {
25 namespace network
26 {
27
50 class BasePDU
51 {
52 public:
53   virtual ~BasePDU() = default;
54
55   virtual std::istream &Read(std::istream &is) = 0;
56   virtual const std::ostream &Write(std::ostream &os) const = 0;
57
58   virtual size_t Size() const = 0;
59   virtual void Print(std::ostream &os) const = 0;
60
```

```
61   virtual bool IsLastFragment() const = 0;
62 };
63
64 } // end namespace network
65 } // end namespace gdcm
66
67 #endif // GDCMBASEPDU_H
```

11.477 gdcmBaseQuery.h File Reference

```
#include "gdcmDataSet.h"
#include "gdcmUIDs.h"
#include "gdcmObject.h"
```

Include dependency graph for gdcmBaseQuery.h:



This graph shows which files directly or indirectly include this file:

Classes

- class gdcm::BaseQuery

  BaseQuery.

Namespaces

- namespace gdcm

Enumerations

- enum gdcm::ENQueryType {
  gdcm::eCreateMMPS = 0 ,
  gdcm::eSetMMPS }

11.478 gdcmBaseQuery.h

Go to the documentation of this file.

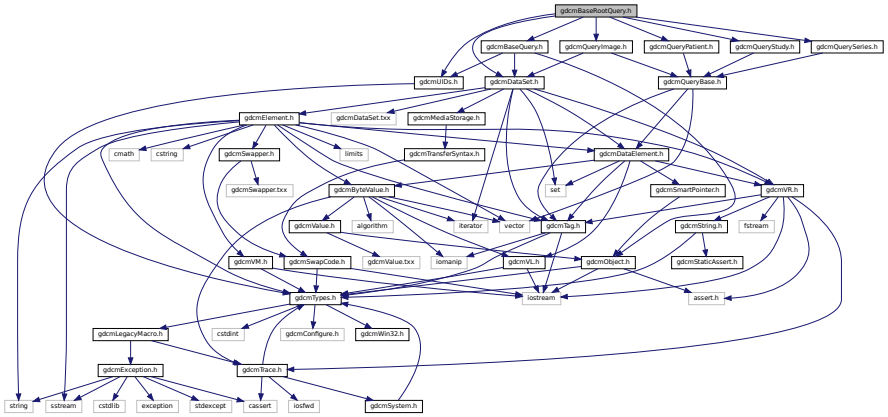
```
1 /*=========================================================================
2 *
3 *  Copyright NumFOCUS
4 *
5 *  Licensed under the Apache License, Version 2.0 (the "License");
6 *  you may not use this file except in compliance with the License.
7 *  You may obtain a copy of the License at
8 *
9 *         http://www.apache.org/licenses/LICENSE-2.0.txt
10 *
11 *  Unless required by applicable law or agreed to in writing, software
12 *  distributed under the License is distributed on an "AS IS" BASIS,
13 *  WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
14 *  See the License for the specific language governing permissions and
15 *  limitations under the License.
16 *
17 *=========================================================================*/
18 #ifndef GDCMBASEQUERY_H
19 #define GDCMBASEQUERY_H
20
21 #include "gdcmDataSet.h"
22 #include "gdcmUIDs.h"
23 #include "gdcmObject.h"
24
25 namespace gdcm
26 {
27   class QueryFactory;
28   class DictEntry;
29
30   enum ENQueryType
31     {
32     eCreateMMPS = 0,
33     eSetMMPS
34     };
41 class GDCM_EXPORT BaseQuery :  public Object
42 {
43   //these four classes contain the required, unique, and optional tags from the standard.
44   //used both to list the tags as well as to validate a dataset, if ever we were to do so.
45 protected:
46
47   DataSet mDataSet;
48   friend class QueryFactory;
49   BaseQuery();
50
51   std::string mSopInstanceUID;
52
53   void SetSearchParameter(const Tag& inTag, const DictEntry& inDictEntry, const std::string& inValue);
54
55   bool ValidDataSet( const DataSet & dataSetToValid, const DataSet & dataSetReference  ) const ;
56 public:
57   ~BaseQuery() override;
58
59   void SetSearchParameter(const Tag& inTag, const std::string& inValue);
60   void SetSearchParameter(const std::string& inKeyword, const std::string& inValue);
61
62   const std::ostream &WriteHelpFile(std::ostream &os);
63
64   //this function allows writing of the query to disk for storing for future use
65   //virtual in case it needs to be overridden
66   //returns false if the operation failed
67   bool WriteQuery(const std::string& inFileName);
68
70   DataSet const & GetQueryDataSet() const;
71   DataSet & GetQueryDataSet();
```

```
72   void AddQueryDataSet(const DataSet & ds);
73
74   virtual bool ValidateQuery( bool inStrict = true ) const = 0;
75
76   virtual UIDs::TSName GetAbstractSyntaxUID() const = 0;
77   std::string GetSOPInstanceUID()const { return mSopInstanceUID ; }
78   void SetSOPInstanceUID( const std::string & iSopInstanceUID ) { mSopInstanceUID = iSopInstanceUID ; }
79
80
81
82   void Print(std::ostream &os) const override;
83 };
84
85 } // end namespace gdcm
86
87 #endif //GDCMBASEROOTQUERY_H
```

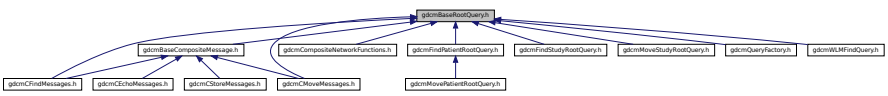
11.479 gdcmBaseRootQuery.h File Reference

```
#include "gdcmDataSet.h"
#include "gdcmUIDs.h"
#include "gdcmBaseQuery.h"
#include "gdcmQueryPatient.h"
#include "gdcmQueryStudy.h"
#include "gdcmQuerySeries.h"
#include "gdcmQueryImage.h"
```

Include dependency graph for gdcmBaseRootQuery.h:



This graph shows which files directly or indirectly include this file:



Classes

- class gdcm::BaseRootQuery

  BaseRootQuery.

Namespaces

- namespace gdcm

Enumerations

- enum gdcm::EQueryLevel {
  gdcm::ePatient = 0 ,
  gdcm::eStudy = 1 ,
  gdcm::eSeries = 2 ,
  gdcm::eImage = 3 }
- enum gdcm::EQueryType {
  gdcm::eFind = 0 ,
  gdcm::eMove ,
  gdcm::eWLMFind }

11.480 gdcmBaseRootQuery.h

Go to the documentation of this file.

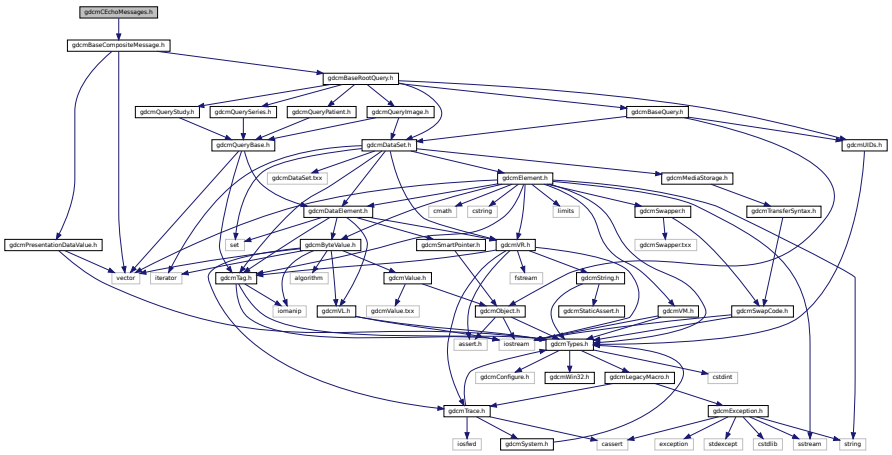
```
1 /*=========================================================================
2 *
3 *  Copyright NumFOCUS
4 *
5 *  Licensed under the Apache License, Version 2.0 (the "License");
6 *  you may not use this file except in compliance with the License.
7 *  You may obtain a copy of the License at
8 *
9 *         http://www.apache.org/licenses/LICENSE-2.0.txt
10 *
11 *  Unless required by applicable law or agreed to in writing, software
12 *  distributed under the License is distributed on an "AS IS" BASIS,
13 *  WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
14 *  See the License for the specific language governing permissions and
15 *  limitations under the License.
16 *
17 *=========================================================================*/
18 #ifndef GDCMBASEROOTQUERY_H
19 #define GDCMBASEROOTQUERY_H
20
21 #include "gdcmDataSet.h"
22 #include "gdcmUIDs.h"
23 #include "gdcmBaseQuery.h"
24 #include "gdcmQueryPatient.h"
25 #include "gdcmQueryStudy.h"
26 #include "gdcmQuerySeries.h"
27 #include "gdcmQueryImage.h"
28
29 namespace gdcm
30 {
31   class QueryFactory;
32   class DictEntry;
33
34   enum EQueryLevel
35     {
36     // -1 is reserved do not use
37     ePatient = 0,
38     eStudy   = 1,
39     eSeries  = 2,
40     eImage   = 3
41     };
42   enum EQueryType
43     {
44     eFind= 0,
45     eMove,
46     eWLMFind
47     };
```

```
48
66 class GDCM_EXPORT BaseRootQuery :  public BaseQuery
67 {
68   //these four classes contain the required, unique, and optional tags from the standard.
69   //used both to list the tags as well as to validate a dataset, if ever we were to do so.
70 protected:
71   QueryPatient mPatient;
72   QueryStudy mStudy;
73   QuerySeries mSeries;
74   QueryImage mImage;
75
76   friend class QueryFactory;
77   BaseRootQuery();
78
79   ERootType mRootType; //set in construction, and it's something else in the study root type
80   std::string mHelpDescription; //used when generating the help output
81
82 public:
83   ~BaseRootQuery() override;
84
88   virtual std::vector<Tag> GetTagListByLevel(const EQueryLevel& inQueryLevel) = 0;
89
93   virtual void InitializeDataSet(const EQueryLevel& inQueryLevel) = 0;
94
108   bool ValidateQuery( bool inStrict = true ) const override = 0;
109
110   static const char *GetQueryLevelString( EQueryLevel ql );
111   static int GetQueryLevelFromString( const char * str );
112
113   static QueryBase * Construct(ERootType inRootType, EQueryLevel qlevel);
114   EQueryLevel GetQueryLevelFromQueryRoot( ERootType roottype );
115 };
116
117 } // end namespace gdcm
118
119 #endif //GDCMBASEROOTQUERY_H
```

11.481 gdcmCEchoMessages.h File Reference

`#include "gdcmBaseCompositeMessage.h"`

Include dependency graph for gdcmCEchoMessages.h:



Classes

- class gdcm::network::CEchoRQ

CEchoRQ.

- class gdcm::network::CEchoRSP

  CEchoRSP this file defines the messages for the cecho action.

Namespaces

- namespace gdcm
- namespace gdcm::network

11.482 gdcmCEchoMessages.h

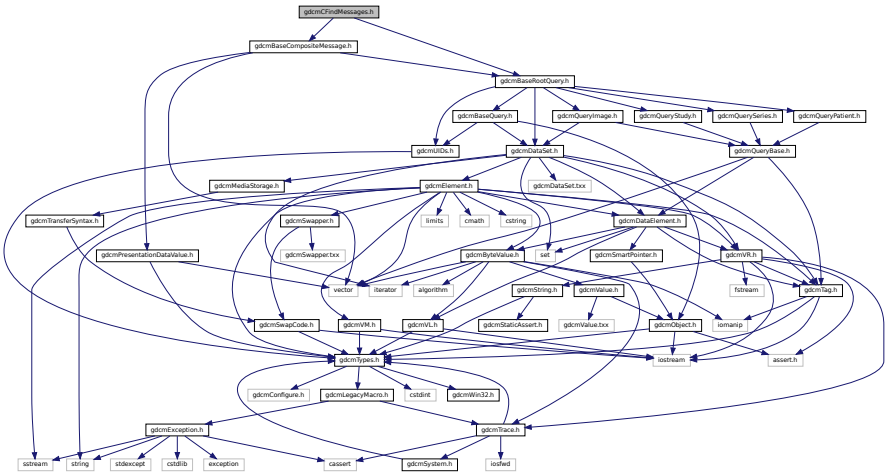
Go to the documentation of this file.

```
1 /*=========================================================================
2 *
3 *  Copyright NumFOCUS
4 *
5 *  Licensed under the Apache License, Version 2.0 (the "License");
6 *  you may not use this file except in compliance with the License.
7 *  You may obtain a copy of the License at
8 *
9 *         http://www.apache.org/licenses/LICENSE-2.0.txt
10 *
11 *  Unless required by applicable law or agreed to in writing, software
12 *  distributed under the License is distributed on an "AS IS" BASIS,
13 *  WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
14 *  See the License for the specific language governing permissions and
15 *  limitations under the License.
16 *
17 *=========================================================================*/
18 #ifndef GDCMCECHOMESSAGES_H
19 #define GDCMCECHOMESSAGES_H
20
21 #include "gdcmBaseCompositeMessage.h"
22
23 namespace gdcm{
24   namespace network{
25
26 class ULConnection;
27
32 class CEchoRQ :  public BaseCompositeMessage {
33     public:
34       std::vector<PresentationDataValue> ConstructPDV(const ULConnection &inConnection,
35         const BaseRootQuery* inRootQuery) override;
36     };
37
42     class CEchoRSP :  public BaseCompositeMessage {
43     public:
44       std::vector<PresentationDataValue> ConstructPDVByDataSet(const DataSet* inDataSet);
45     };
46   }
47 }
48 #endif // GDCMCECHOMESSAGES_H
```

11.483 gdcmCFindMessages.h File Reference

```
#include "gdcmBaseCompositeMessage.h"
#include "gdcmBaseRootQuery.h"
```

Include dependency graph for gdcmCFindMessages.h:



Classes

- class gdcm::network::CFindCancelRQ

  CFindCancelRQ this file defines the messages for the cfind action.
- class gdcm::network::CFindRQ

  CFindRQ.
- class gdcm::network::CFindRSP

  CFindRSP this file defines the messages for the cfind action.

Namespaces

- namespace gdcm
- namespace gdcm::network

11.484 gdcmCFindMessages.h

Go to the documentation of this file.

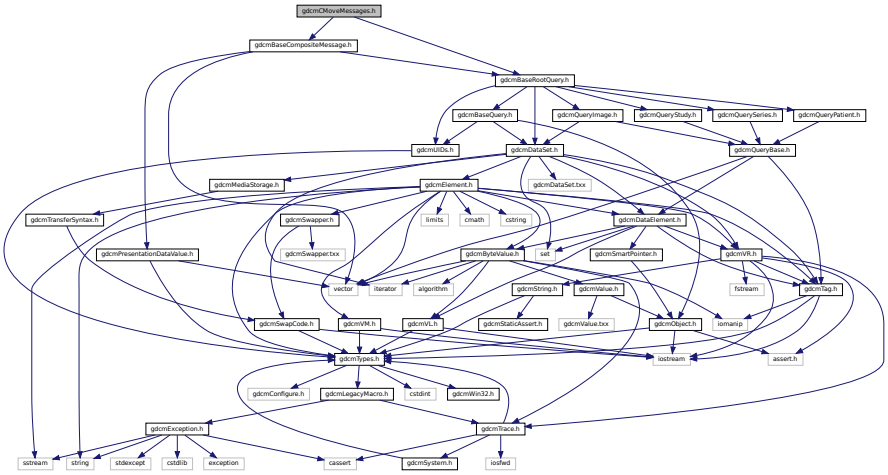
```
1 /*=========================================================================
2 *
3 *  Copyright NumFOCUS
4 *
5 *  Licensed under the Apache License, Version 2.0 (the "License");
6 *  you may not use this file except in compliance with the License.
7 *  You may obtain a copy of the License at
8 *
9 *         http://www.apache.org/licenses/LICENSE-2.0.txt
10 *
11 *  Unless required by applicable law or agreed to in writing, software
12 *  distributed under the License is distributed on an "AS IS" BASIS,
13 *  WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
14 *  See the License for the specific language governing permissions and
15 *  limitations under the License.
16 *
```

```
17 *=========================================================================*/
18 #ifndef GDCMCFINDMESSAGES_H
19 #define GDCMCFINDMESSAGES_H
20
21 #include "gdcmBaseCompositeMessage.h"
22 #include "gdcmBaseRootQuery.h"
23
24 namespace gdcm
25 {
26 namespace network
27 {
28
33 class CFindRQ :  public BaseCompositeMessage
34 {
35 public:
36   std::vector<PresentationDataValue> ConstructPDV(const ULConnection &inConnection,
37     const BaseRootQuery* inRootQuery) override;
38 };
39
44 class CFindRSP :  public BaseCompositeMessage {
45 public:
46   std::vector<PresentationDataValue> ConstructPDVByDataSet(const DataSet* inDataSet);
47 };
48
53   class CFindCancelRQ :  public BaseCompositeMessage {
54   public:
55     std::vector<PresentationDataValue> ConstructPDVByDataSet(const DataSet* inDataSet);
56   };
57 }
58 }
59 #endif
```

11.485 gdcmCMoveMessages.h File Reference

```
#include "gdcmBaseCompositeMessage.h"
#include "gdcmBaseRootQuery.h"
```

Include dependency graph for gdcmCMoveMessages.h:



Classes

- class gdcm::network::CMoveCancelRq
- class gdcm::network::CMoveRQ

CMoveRQ.

- class gdcm::network::CMoveRSP

  CMoveRSP this file defines the messages for the cmove action.

Namespaces

- namespace gdcm
- namespace gdcm::network

11.486 gdcmCMoveMessages.h

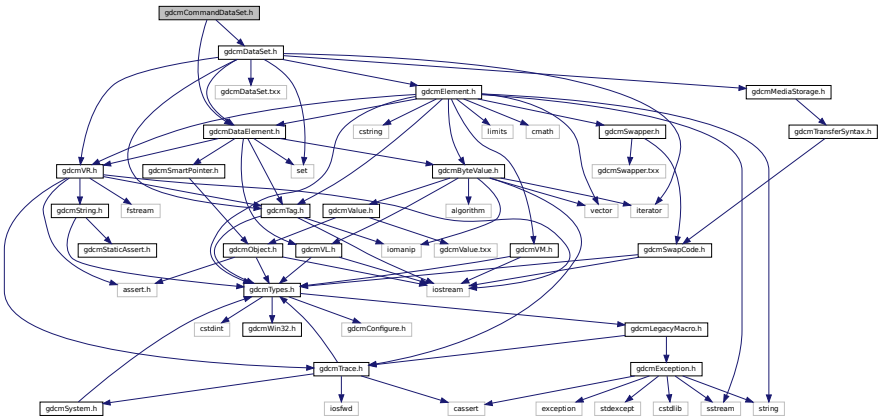
Go to the documentation of this file.

```
1 /*=========================================================================
2 *
3 *  Copyright NumFOCUS
4 *
5 *  Licensed under the Apache License, Version 2.0 (the "License");
6 *  you may not use this file except in compliance with the License.
7 *  You may obtain a copy of the License at
8 *
9 *         http://www.apache.org/licenses/LICENSE-2.0.txt
10 *
11 *  Unless required by applicable law or agreed to in writing, software
12 *  distributed under the License is distributed on an "AS IS" BASIS,
13 *  WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
14 *  See the License for the specific language governing permissions and
15 *  limitations under the License.
16 *
17 *=========================================================================*/
18 #ifndef GDCMCMOVEMESSAGES_H
19 #define GDCMCMOVEMESSAGES_H
20
21 #include "gdcmBaseCompositeMessage.h"
22 #include "gdcmBaseRootQuery.h"
23
24 namespace gdcm{
25   namespace network{
26   class ULConnection;
31 class CMoveRQ :  public BaseCompositeMessage {
32       //this class will fulfill the inheritance,
33       //but additional information is needed by cmovd
34       //namely, the root type or the calling AE-TITLE
35       std::vector<PresentationDataValue> ConstructPDVByDataSet(const DataSet* inDataSet);
36     public:
37       std::vector<PresentationDataValue> ConstructPDV(
38         const ULConnection &inConnection,
39         const BaseRootQuery* inRootQuery) override;
40     };
41
46 class CMoveRSP :  public BaseCompositeMessage {
47     public:
48       std::vector<PresentationDataValue> ConstructPDVByDataSet(const DataSet* inDataSet);
49     };
50
51     class CMoveCancelRq :  public BaseCompositeMessage {
52     public:
53       std::vector<PresentationDataValue> ConstructPDVByDataSet(const DataSet* inDataSet);
54     };
55   }
56 }
57 #endif
```

11.487 gdcmCommandDataSet.h File Reference

```
#include "gdcmDataSet.h"
#include "gdcmDataElement.h"
```

Include dependency graph for gdcmCommandDataSet.h:



Classes

- class gdcm::CommandDataSet

  Class to represent a Command DataSet.

Namespaces

- namespace gdcm

Functions

- std::ostream & gdcm::operator<< (std::ostream &os, const CommandDataSet &val)

11.488 gdcmCommandDataSet.h

Go to the documentation of this file.

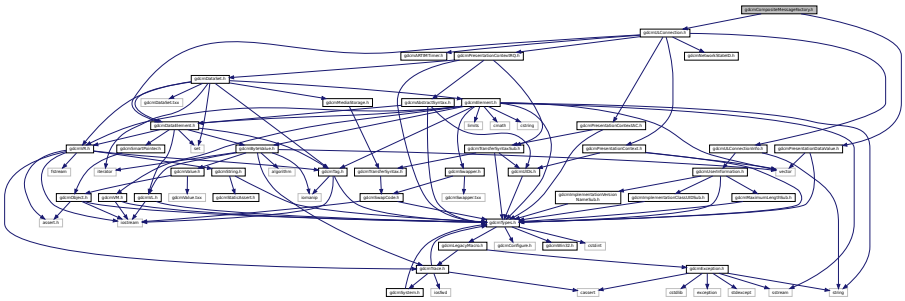
```
1 /*=========================================================================
2
3 Program:  GDCM (Grassroots DICOM). A DICOM library
4
5 Copyright (c) 2006-2011 Mathieu Malaterre
6 All rights reserved.
7 See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.
8
9 This software is distributed WITHOUT ANY WARRANTY; without even
10 the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
11 PURPOSE.  See the above copyright notice for more information.
12
```

```
13 =========================================================================*/
14 #ifndef GDCMCOMMANDDATASET_H
15 #define GDCMCOMMANDDATASET_H
16
17 #include "gdcmDataSet.h"
18 #include "gdcmDataElement.h"
19
20 namespace gdcm
21 {
27 class GDCM_EXPORT CommandDataSet :  public DataSet
28 {
29 public:
30   CommandDataSet() = default;
31  ~CommandDataSet() = default;
32
33   friend std::ostream &operator«(std::ostream &_os, const CommandDataSet &_val);
34
35   // FIXME: no virtual function means:  duplicate code...
36   void Insert(const DataElement& de) {
37     if( de.GetTag().GetGroup() == 0x0000 )
38       {
39       InsertDataElement( de );
40       }
41     else
42       {
43       gdcmErrorMacro( "Cannot add element with group != 0x0000 in the command dataset :  " « de );
44       }
45   }
46   void Replace(const DataElement& de) {
47     Remove(de.GetTag());
48     Insert(de);
49   }
50
52   std::istream &Read(std::istream &is);
53
55   std::ostream &Write(std::ostream &os) const;
56
57 protected:
58 };
59 //-----------------------------------------------------------------------------
60 inline std::ostream& operator«(std::ostream &os, const CommandDataSet &val)
61 {
62   val.Print( os );
63   return os;
64 }
65
66 } // end namespace gdcm
67
68 #endif //GDCMFILEMETAINFORMATION_H
```

11.489 gdcmCompositeMessageFactory.h File Reference

```
#include "gdcmPresentationDataValue.h"
#include "gdcmULConnection.h"
```

Include dependency graph for gdcmCompositeMessageFactory.h:



Classes

- class gdcm::network::CompositeMessageFactory

  CompositeMessageFactory.

Namespaces

- namespace gdcm
- namespace gdcm::network

11.490 gdcmCompositeMessageFactory.h

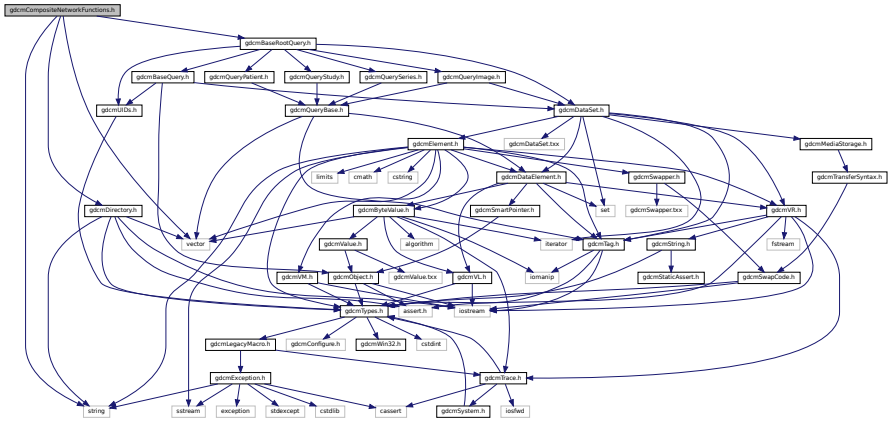
Go to the documentation of this file.

```
1 /*=========================================================================
2 *
3 *  Copyright NumFOCUS
4 *
5 *  Licensed under the Apache License, Version 2.0 (the "License");
6 *  you may not use this file except in compliance with the License.
7 *  You may obtain a copy of the License at
8 *
9 *         http://www.apache.org/licenses/LICENSE-2.0.txt
10 *
11 *  Unless required by applicable law or agreed to in writing, software
12 *  distributed under the License is distributed on an "AS IS" BASIS,
13 *  WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
14 *  See the License for the specific language governing permissions and
15 *  limitations under the License.
16 *
17 *=========================================================================*/
18 #ifndef GDCMCOMPOSITEMESSAGEFACTORY_H
19 #define GDCMCOMPOSITEMESSAGEFACTORY_H
20
21 #include "gdcmPresentationDataValue.h"
22 #include "gdcmULConnection.h"
23
24 namespace gdcm {
25   class BaseRootQuery;
26   class File;
27   namespace network {
28     class BasePDU;
37 class CompositeMessageFactory
38 {
39     public:
40       //the echo request only needs a properly constructed PDV.
41       //find, move, etc, may need something more robust, but since those are
42       //easily placed into the appropriate pdatapdu in the pdufactory,
43       //this approach without a base class (but done internally) is useful.
44       static std::vector<PresentationDataValue> ConstructCEchoRQ(const ULConnection& inConnection);
45
46       static std::vector<PresentationDataValue> ConstructCStoreRQ(const ULConnection& inConnection,const File
     &file, bool writeDataSet = true );
47       static std::vector<PresentationDataValue> ConstructCStoreRSP(const DataSet *inDataSet, const BasePDU*
     inPC);
48
49       static  std::vector<PresentationDataValue> ConstructCFindRQ(const ULConnection& inConnection, const
     BaseRootQuery* inRootQuery);
50
51       static  std::vector<PresentationDataValue> ConstructCMoveRQ(const ULConnection& inConnection, const
     BaseRootQuery* inRootQuery);
52
53
54     };
55   }
56 }
57
58 #endif // GDCMCOMPOSITEMESSAGEFACTORY_H
```

11.491 gdcmCompositeNetworkFunctions.h File Reference

```
#include "gdcmDirectory.h"
#include "gdcmBaseRootQuery.h"
#include <vector>
#include <string>
```

Include dependency graph for gdcmCompositeNetworkFunctions.h:



Classes

- class gdcm::CompositeNetworkFunctions

  Composite Network Functions.

Namespaces

- namespace gdcm

11.492 gdcmCompositeNetworkFunctions.h

Go to the documentation of this file.

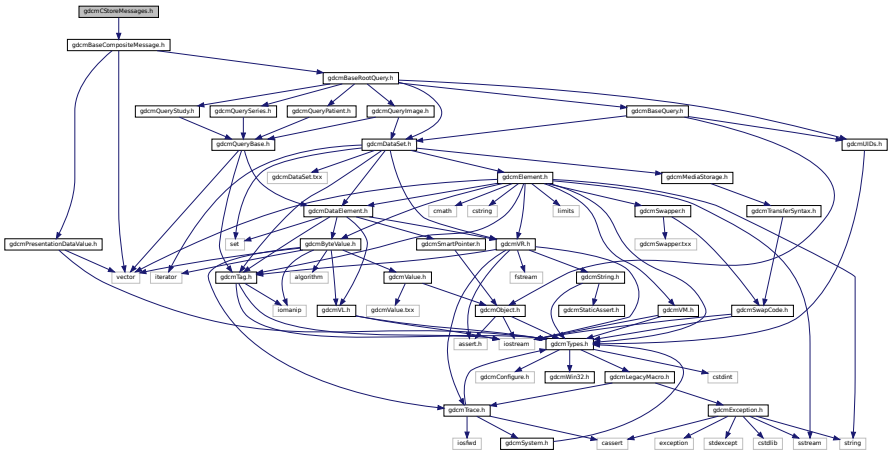

```
1 /*=========================================================================
2 *
3 *  Copyright NumFOCUS
4 *
5 *  Licensed under the Apache License, Version 2.0 (the "License");
6 *  you may not use this file except in compliance with the License.
7 *  You may obtain a copy of the License at
8 *
9 *         http://www.apache.org/licenses/LICENSE-2.0.txt
10 *
11 *  Unless required by applicable law or agreed to in writing, software
12 *  distributed under the License is distributed on an "AS IS" BASIS,
13 *  WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
14 *  See the License for the specific language governing permissions and
15 *  limitations under the License.
16 *
17 *=========================================================================*/
```

```
18 #ifndef GDCMCOMPOSITENETWORKFUNCTIONS_H
19 #define GDCMCOMPOSITENETWORKFUNCTIONS_H
20
21 #include "gdcmDirectory.h"
22 #include "gdcmBaseRootQuery.h" // EQueryLevel / EQueryType
23
24 #include <vector>
25 #include <string>
26
27 namespace gdcm
28 {
48 class GDCM_EXPORT CompositeNetworkFunctions
49 {
50 public:
57   static bool CEcho( const char *remote, uint16_t portno, const char *aetitle = nullptr,
58     const char *call = nullptr );
59
60   typedef std::pair<Tag, std::string> KeyValuePairType;
61   typedef std::vector< KeyValuePairType  > KeyValuePairArrayType;
62
69   static BaseRootQuery* ConstructQuery(ERootType inRootType, EQueryLevel inQueryLevel,
70     const DataSet& queryds, EQueryType queryType = eFind );
71
73   static BaseRootQuery* ConstructQuery(ERootType inRootType, EQueryLevel inQueryLevel,
74     const KeyValuePairArrayType& keys, EQueryType queryType = eFind );
75
91   static bool CMove( const char *remote, uint16_t portno, const BaseRootQuery* query,
92     uint16_t portscp, const char *aetitle = nullptr,
93     const char *call = nullptr, const char *outputdir = nullptr);
94
104   static bool CFind( const char *remote, uint16_t portno,
105     const BaseRootQuery* query,
106     std::vector<DataSet> &retDataSets,
107     const char *aetitle = nullptr,
108     const char *call = nullptr );
109
117   static bool CStore( const char *remote, uint16_t portno,
118     const Directory::FilenamesType & filenames,
119     const char *aetitle = nullptr, const char *call = nullptr);
120 };
121
122 } // end namespace gdcm
123
124 #endif // GDCMCOMPOSITENETWORKFUNCTIONS_H
```

11.493 gdcmCStoreMessages.h File Reference

`#include "gdcmBaseCompositeMessage.h"`

Include dependency graph for gdcmCStoreMessages.h:



Classes

- class gdcm::network::CStoreRQ

  CStoreRQ.
- class gdcm::network::CStoreRSP

  CStoreRSP this file defines the messages for the cecho action.

Namespaces

- namespace gdcm
- namespace gdcm::network

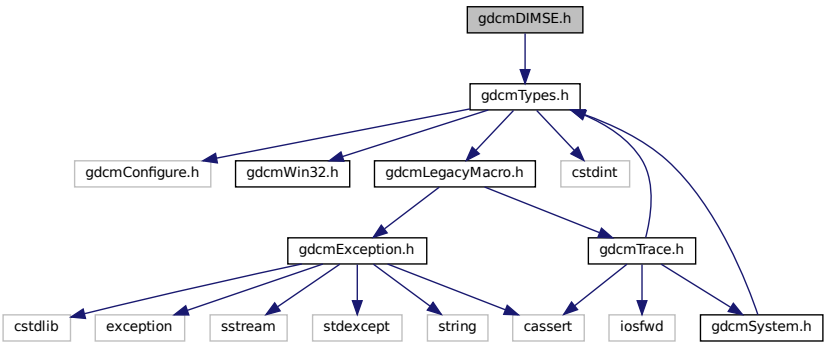
11.494 gdcmCStoreMessages.h

Go to the documentation of this file.

```
1 /*=========================================================================
2 *
3 *  Copyright NumFOCUS
4 *
5 *  Licensed under the Apache License, Version 2.0 (the "License");
6 *  you may not use this file except in compliance with the License.
7 *  You may obtain a copy of the License at
8 *
9 *         http://www.apache.org/licenses/LICENSE-2.0.txt
10 *
11 *  Unless required by applicable law or agreed to in writing, software
12 *  distributed under the License is distributed on an "AS IS" BASIS,
13 *  WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
14 *  See the License for the specific language governing permissions and
15 *  limitations under the License.
16 *
17 *=========================================================================*/
18 #ifndef GDCMCSTOREMESSAGES_H
19 #define GDCMCSTOREMESSAGES_H
20
21 #include "gdcmBaseCompositeMessage.h"
22
23 namespace gdcm{
24 class File;
25   namespace network{
26     class BasePDU;
31 class CStoreRQ :  public BaseCompositeMessage {
32       std::vector<PresentationDataValue> ConstructPDV(const ULConnection &inConnection, const BaseRootQuery*
      inRootQuery) override;//to fulfill the virtual contract
33     public:
34       std::vector<PresentationDataValue> ConstructPDV(const ULConnection &inConnection,
35         const File& file,  bool writeDataSet = true );
36     };
37
42     class CStoreRSP :  public BaseCompositeMessage {
43       std::vector<PresentationDataValue> ConstructPDV(const ULConnection &inConnection, const BaseRootQuery*
      inRootQuery) override;//to fulfill the virtual contract
44     public:
45       std::vector<PresentationDataValue> ConstructPDV(const DataSet* inDataSet, const BasePDU* inPC);
46     };
47   }
48 }
49 #endif // GDCMCSTOREMESSAGES_H
```

11.495 gdcmDIMSE.h File Reference

`#include "gdcmTypes.h"`
Include dependency graph for gdcmDIMSE.h:



Classes

- class gdcm::network::CEchoRQ

  CEchoRQ.
- class gdcm::network::CEchoRSP

  CEchoRSP this file defines the messages for the cecho action.
- class gdcm::network::CFind
- class gdcm::network::DIMSE

  DIMSE.

Namespaces

- namespace gdcm
- namespace gdcm::network

11.496 gdcmDIMSE.h

Go to the documentation of this file.

```
1 /*=========================================================================
2
3 Program:  GDCM (Grassroots DICOM). A DICOM library
4
5 Copyright (c) 2006-2011 Mathieu Malaterre
6 All rights reserved.
7 See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.
8
9 This software is distributed WITHOUT ANY WARRANTY; without even
10 the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
11 PURPOSE.  See the above copyright notice for more information.
12
```

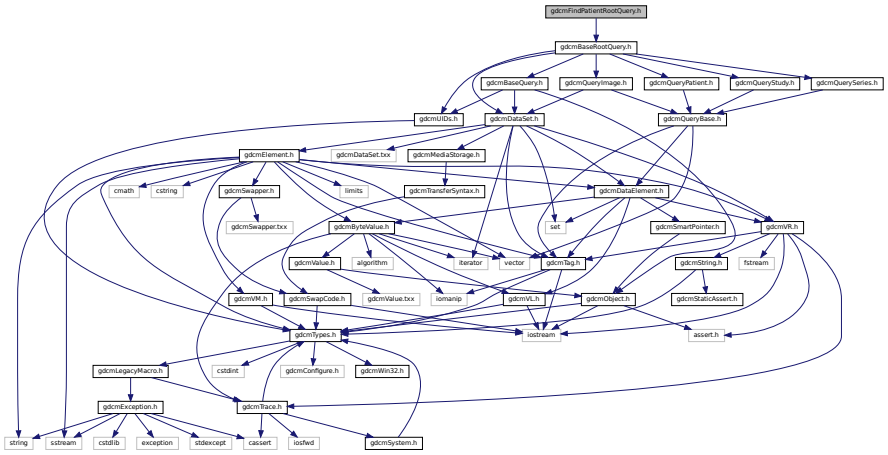
```
13 ============================================================================*/
14 #ifndef GDCMDIMSE_H
15 #define GDCMDIMSE_H
16
17 #include "gdcmTypes.h"
18
19 namespace gdcm
20 {
21
22 namespace network
23 {
24
33 class DIMSE {
34 public:
35   typedef enum {
36 C_STORE_RQ         = 0x0001,
37 C_STORE_RSP        = 0x8001,
38 C_GET_RQ           = 0x0010,
39 C_GET_RSP          = 0x8010,
40 C_FIND_RQ          = 0x0020,
41 C_FIND_RSP         = 0x8020,
42 C_MOVE_RQ          = 0x0021,
43 C_MOVE_RSP         = 0x8021,
44 C_ECHO_RQ          = 0x0030,
45 C_ECHO_RSP         = 0x8030,
46 N_EVENT_REPORT_RQ  = 0x0100,
47 N_EVENT_REPORT_RSP = 0x8100,
48 N_GET_RQ          = 0x0110,
49 N_GET_RSP         = 0x8110,
50 N_SET_RQ          = 0x0120,
51 N_SET_RSP         = 0x8120,
52 N_ACTION_RQ       = 0x0130,
53 N_ACTION_RSP      = 0x8130,
54 N_CREATE_RQ       = 0x0140,
55 N_CREATE_RSP      = 0x8140,
56 N_DELETE_RQ       = 0x0150,
57 N_DELETE_RSP      = 0x8150,
58 C_CANCEL_RQ       = 0x0FFF
59   } CommandTypes;
60 };
61
62 /*
63 9.1.5.1 C-ECHO parameters
64 Table 9.1-5
65 C-ECHO PARAMETERS
66 */
67 class CEchoRQ
68 {
69 public:
70   uint16_t          MessageID;                          /* M */
71   UIComp            AffectedSOPClassUID;                /* M */
72 };
73
74 class CEchoRSP
75 {
76 public:
77 /*
78 Message ID M U
79 Message ID Being Responded To M
80 Affected SOP Class UID M U(=)
81 Status M
82 */
83 };
84
89 class CFind
90 {
91 /*
92 Failure Refused:  Out of Resources A700 (0000,0902)
93 Identifier does not match SOP Class A900 (0000,0901)
94 (0000,0902)
95 Unable to process Cxxx (0000,0901)
96 (0000,0902)
97 Cancel Matching terminated due to Cancel
98 request
99 FE00 None
100 Success Matching is complete - No final Identifier
101 is supplied.
102 0000 None
103 Pending Matches are continuing - Current Match
104 is supplied and any Optional Keys were
105 supported in the same manner as
```

```
106 Required Keys.
107 FF00 Identifier
108 Matches are continuing - Warning that
109 one or more Optional Keys were not
110 supported for existence and/or matching
111 for this Identifier.
112 FF01 Identifier
113 */
114 };
115
116
117 } // end namespace network
118
119 } // end namespace gdcm
120
121 #endif //GDCMDIMSE_H
```

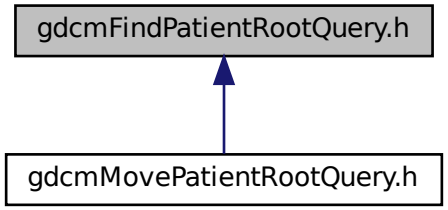
11.497 gdcmFindPatientRootQuery.h File Reference

`#include "gdcmBaseRootQuery.h"`

Include dependency graph for gdcmFindPatientRootQuery.h:



This graph shows which files directly or indirectly include this file:



Classes

- class gdcm::FindPatientRootQuery

  PatientRootQuery.

Namespaces

- namespace gdcm

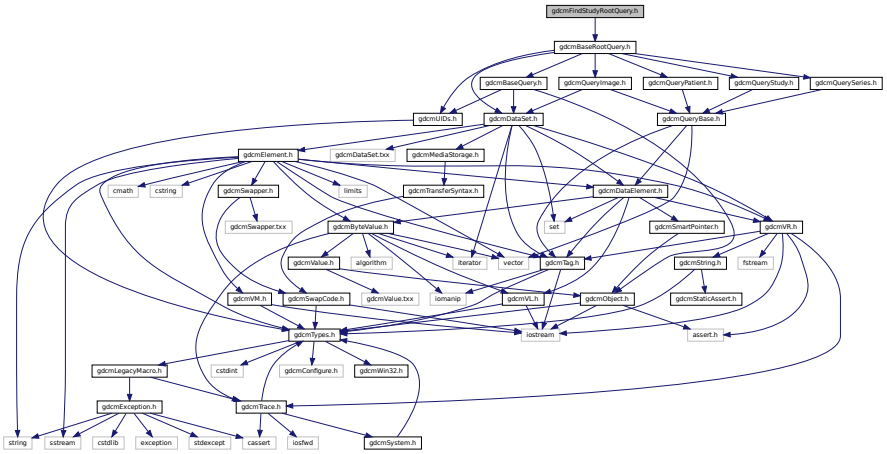
11.498 gdcmFindPatientRootQuery.h

Go to the documentation of this file.

```
1 /*=========================================================================
2
3 Program:  GDCM (Grassroots DICOM). A DICOM library
4
5 Copyright (c) 2006-2011 Mathieu Malaterre
6 All rights reserved.
7 See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.
8
9 This software is distributed WITHOUT ANY WARRANTY; without even
10 the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
11 PURPOSE.  See the above copyright notice for more information.
12
13 =========================================================================*/
14 #ifndef GDCMFINDPATIENTROOTQUERY_H
15 #define GDCMFINDPATIENTROOTQUERY_H
16
17 #include "gdcmBaseRootQuery.h"
18
19 namespace gdcm
20 {
25 class GDCM_EXPORT FindPatientRootQuery :  public BaseRootQuery
26 {
27   friend class QueryFactory;
28 public:
29   FindPatientRootQuery();
30
31   void InitializeDataSet(const EQueryLevel& inQueryLevel) override;
32
33   std::vector<Tag> GetTagListByLevel(const EQueryLevel& inQueryLevel) override;
34   bool ValidateQuery(bool inStrict = true) const override;
35
36   UIDs::TSName GetAbstractSyntaxUID() const override;
37 };
38
39 } // end namespace gdcm
40
41 #endif // GDCMFINDPATIENTROOTQUERY_H
```

11.499 gdcmFindStudyRootQuery.h File Reference

`#include "gdcmBaseRootQuery.h"`
Include dependency graph for gdcmFindStudyRootQuery.h:



Classes

- class gdcm::FindStudyRootQuery

  FindStudyRootQuery.

Namespaces

- namespace gdcm

11.500 gdcmFindStudyRootQuery.h

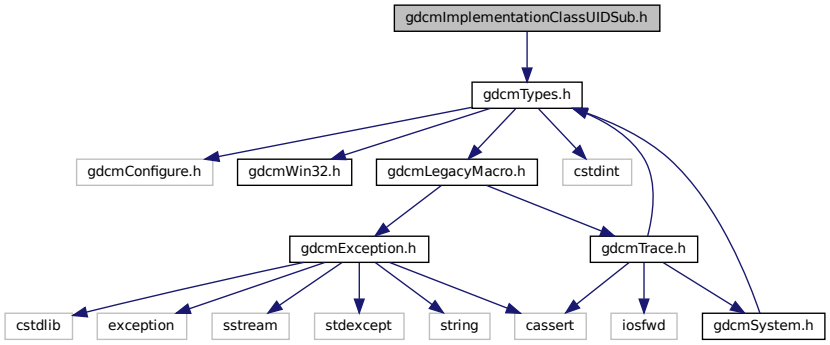
Go to the documentation of this file.

```
1 /*=========================================================================
2
3 Program:  GDCM (Grassroots DICOM). A DICOM library
4
5 Copyright (c) 2006-2011 Mathieu Malaterre
6 All rights reserved.
7 See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.
8
9 This software is distributed WITHOUT ANY WARRANTY; without even
10 the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
11 PURPOSE.  See the above copyright notice for more information.
12
13 =========================================================================*/
14 #ifndef GDCMFINDSTUDYROOTQUERY_H
15 #define GDCMFINDSTUDYROOTQUERY_H
16
17 #include "gdcmBaseRootQuery.h"
18
19 namespace gdcm
20 {
```

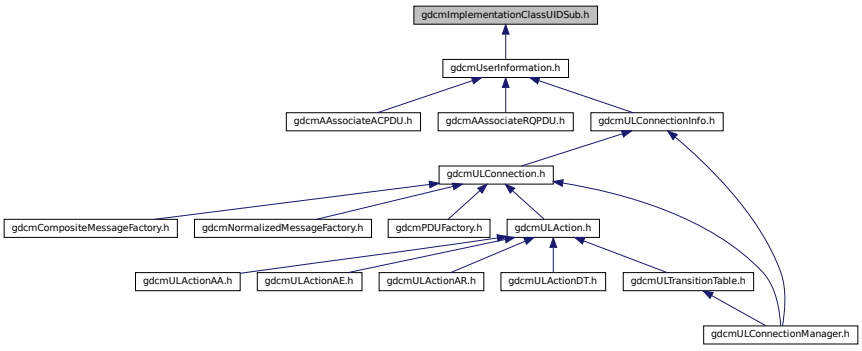
```
25 class GDCM_EXPORT FindStudyRootQuery :  public BaseRootQuery
26 {
27   friend class QueryFactory;
28 public:
29   FindStudyRootQuery();
30
31   void InitializeDataSet(const EQueryLevel& inQueryLevel) override;
32
33   std::vector<Tag> GetTagListByLevel(const EQueryLevel& inQueryLevel) override;
34
38   bool ValidateQuery(bool inStrict = true) const override;
39
40   UIDs::TSName GetAbstractSyntaxUID() const override;
41 };
42
43 } // end namespace gdcm
44
45 #endif // GDCMFINDSTUDYROOTQUERY_H
```

11.501 gdcmImplementationClassUIDSub.h File Reference

`#include "gdcmTypes.h"`
Include dependency graph for gdcmImplementationClassUIDSub.h:



This graph shows which files directly or indirectly include this file:



Classes

- class gdcm::network::ImplementationClassUIDSub

  ImplementationClassUIDSub.

Namespaces

- namespace gdcm
- namespace gdcm::network

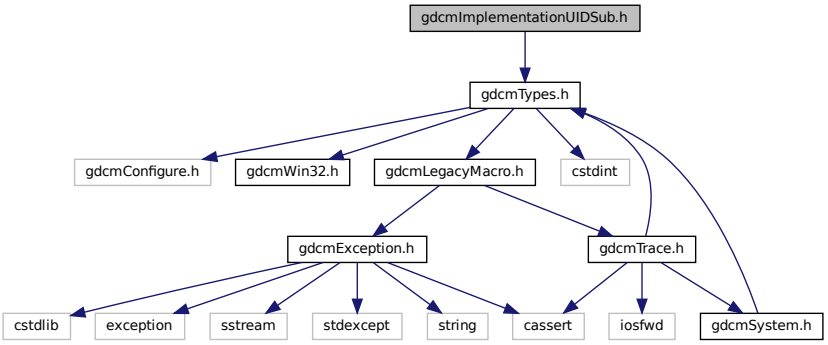
11.502 gdcmImplementationClassUIDSub.h

Go to the documentation of this file.

```
1 /*=========================================================================
2
3 Program:  GDCM (Grassroots DICOM). A DICOM library
4
5 Copyright (c) 2006-2011 Mathieu Malaterre
6 All rights reserved.
7 See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.
8
9 This software is distributed WITHOUT ANY WARRANTY; without even
10 the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
11 PURPOSE.  See the above copyright notice for more information.
12
13 =========================================================================*/
14 #ifndef GDCMIMPLEMENTATIONCLASSUIDSUB_H
15 #define GDCMIMPLEMENTATIONCLASSUIDSUB_H
16
17 #include "gdcmTypes.h"
18
19 namespace gdcm
20 {
21
22 namespace network
23 {
24
31 class ImplementationClassUIDSub
32 {
33 public:
34   ImplementationClassUIDSub();
35   std::istream &Read(std::istream &is);
36   const std::ostream &Write(std::ostream &os) const;
37
38   size_t Size() const;
39
40   void Print(std::ostream &os) const;
41
42 private:
43   static const uint8_t ItemType;
44   static const uint8_t Reserved2;
45   uint16_t ItemLength;
46   std::string ImplementationClassUID;
47 };
48
49 } // end namespace network
50
51 } // end namespace gdcm
52
53 #endif //GDCMMAXIMUMLENGTHSUB_H
```

11.503 gdcmImplementationUIDSub.h File Reference

`#include "gdcmTypes.h"`
Include dependency graph for gdcmImplementationUIDSub.h:



Classes

- class gdcm::network::ImplementationUIDSub
  ImplementationUIDSub.

Namespaces

- namespace gdcm
- namespace gdcm::network

11.504 gdcmImplementationUIDSub.h

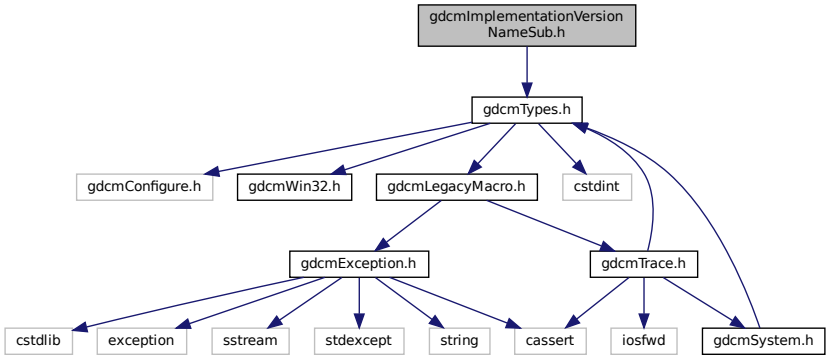
Go to the documentation of this file.

```
1 /*=========================================================================
2
3 Program:  GDCM (Grassroots DICOM). A DICOM library
4
5 Copyright (c) 2006-2011 Mathieu Malaterre
6 All rights reserved.
7 See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.
8
9 This software is distributed WITHOUT ANY WARRANTY; without even
10 the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
11 PURPOSE.  See the above copyright notice for more information.
12
13 =========================================================================*/
14 #ifndef GDCMIMPLEMENTATIONUIDSUB_H
15 #define GDCMIMPLEMENTATIONUIDSUB_H
16
17 #include "gdcmTypes.h"
18
19 namespace gdcm
20 {
21
```

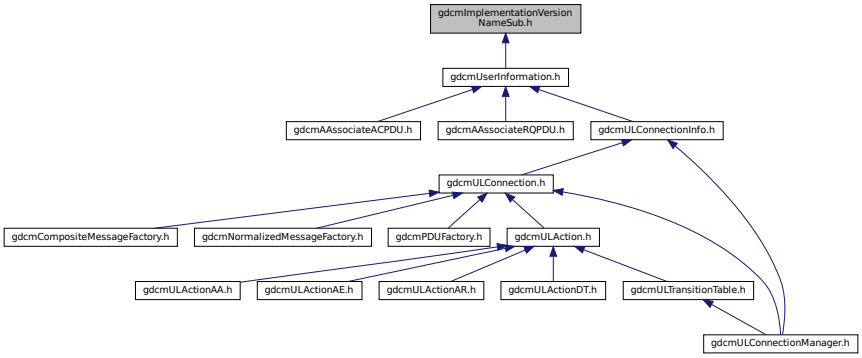
```
22 namespace network
23 {
24
30 class GDCM_EXPORT ImplementationUIDSub
31 {
32 public:
33   ImplementationUIDSub();
34   const std::ostream &Write(std::ostream &os) const;
35 private:
36   static const uint8_t ItemType;
37   static const uint8_t Reserved2;
38   uint16_t ItemLength;
39   std::string ImplementationClassUID;
40 };
41
42 } // end namespace network
43
44 } // end namespace gdcm
45
46 #endif //GDCMMAXIMUMLENGTHSUB_H
```

11.505 gdcmImplementationVersionNameSub.h File Reference

`#include "gdcmTypes.h"`
Include dependency graph for gdcmImplementationVersionNameSub.h:



This graph shows which files directly or indirectly include this file:



Classes

- class gdcm::network::ImplementationVersionNameSub

  ImplementationVersionNameSub.

Namespaces

- namespace gdcm
- namespace gdcm::network

11.506 gdcmImplementationVersionNameSub.h

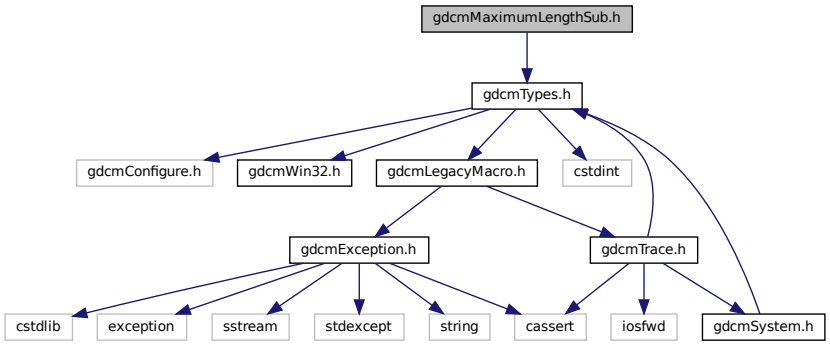
Go to the documentation of this file.

```
1 /*=========================================================================
2
3 Program:  GDCM (Grassroots DICOM). A DICOM library
4
5 Copyright (c) 2006-2011 Mathieu Malaterre
6 All rights reserved.
7 See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.
8
9 This software is distributed WITHOUT ANY WARRANTY; without even
10 the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
11 PURPOSE.  See the above copyright notice for more information.
12
13 =========================================================================*/
14 #ifndef GDCMIMPLEMENTATIONVERSIONNAMESUB_H
15 #define GDCMIMPLEMENTATIONVERSIONNAMESUB_H
16
17 #include "gdcmTypes.h"
18
19 namespace gdcm
20 {
21
22 namespace network
23 {
24
30 class ImplementationVersionNameSub
31 {
32 public:
33   ImplementationVersionNameSub();
34   std::istream &Read(std::istream &is);
35   const std::ostream &Write(std::ostream &os) const;
36
37   size_t Size() const;
38   void Print(std::ostream &os) const;
39
40 private:
41   static const uint8_t ItemType;
42   static const uint8_t Reserved2;
43   uint16_t ItemLength;
44   std::string ImplementationVersionName;
45 };
46
47 } // end namespace network
48
49 } // end namespace gdcm
50
51 #endif //GDCMMAXIMUMLENGTHSUB_H
```

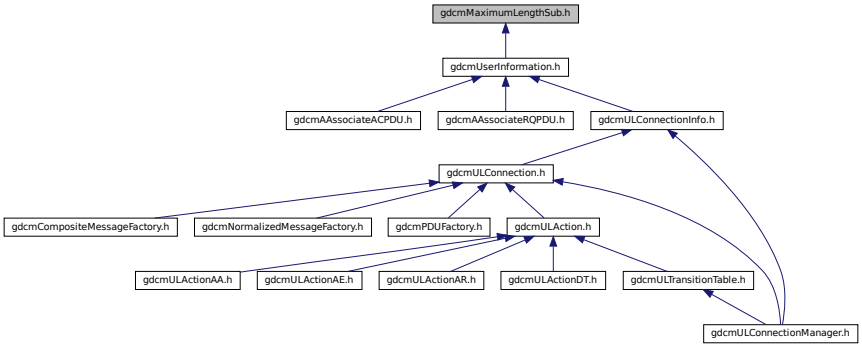
11.507 gdcmMaximumLengthSub.h File Reference

```
#include "gdcmTypes.h"
```

Include dependency graph for gdcmMaximumLengthSub.h:



This graph shows which files directly or indirectly include this file:



Classes

- class gdcm::network::MaximumLengthSub

  MaximumLengthSub.

Namespaces

- namespace gdcm
- namespace gdcm::network

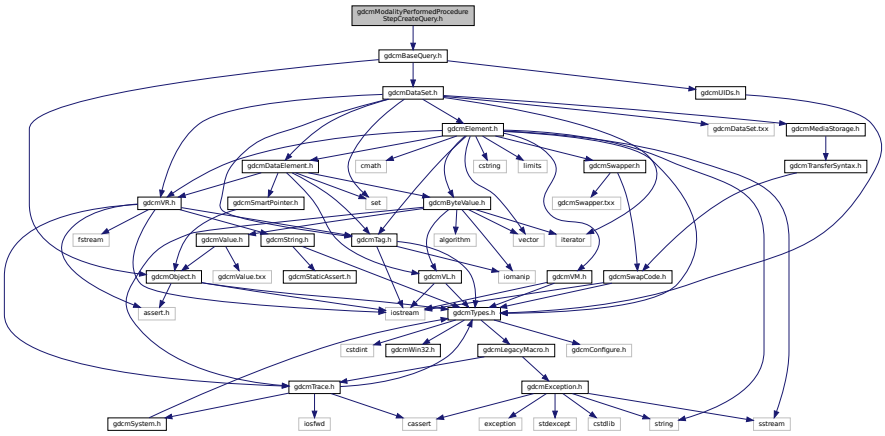
11.508 gdcmMaximumLengthSub.h

Go to the documentation of this file.

```
1 /*=========================================================================
2
3 Program:  GDCM (Grassroots DICOM). A DICOM library
4
5 Copyright (c) 2006-2011 Mathieu Malaterre
6 All rights reserved.
7 See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.
8
9 This software is distributed WITHOUT ANY WARRANTY; without even
10 the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
11 PURPOSE.  See the above copyright notice for more information.
12
13 =========================================================================*/
14 #ifndef GDCMMAXIMUMLENGTHSUB_H
15 #define GDCMMAXIMUMLENGTHSUB_H
16
17 #include "gdcmTypes.h"
18
19 namespace gdcm
20 {
21
22 namespace network
23 {
24
36 class MaximumLengthSub
37 {
38 public:
39   MaximumLengthSub();
40   std::istream &Read(std::istream &is);
41   const std::ostream &Write(std::ostream &os) const;
42
43   size_t Size() const;
44
45   uint32_t GetMaximumLength()const { return MaximumLength; }
46   void SetMaximumLength(uint32_t maximumlength);
47
48   void Print(std::ostream &os) const;
49
50 private:
51   static const uint8_t ItemType;
52   static const uint8_t Reserved2;
53   uint16_t ItemLength;
54   uint32_t MaximumLength;
55 };
56
57 } // end namespace network
58
59 } // end namespace gdcm
60
61 #endif //GDCMMAXIMUMLENGTHSUB_H
```

11.509 gdcmModalityPerformedProcedureStepCreateQuery.h File Reference

`#include "gdcmBaseQuery.h"`
Include dependency graph for gdcmModalityPerformedProcedureStepCreateQuery.h:



Classes

- class gdcm::ModalityPerformedProcedureStepCreateQuery

  ModalityPerformedProcedureStepCreateQuery.

Namespaces

- namespace gdcm

11.510 gdcmModalityPerformedProcedureStepCreateQuery.h

Go to the documentation of this file.

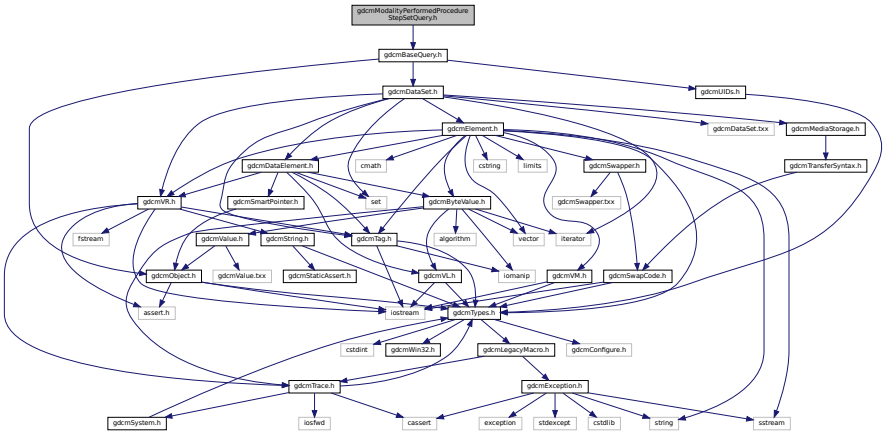
```
1 /*=========================================================================
2
3 Program:  GDCM (Grassroots DICOM). A DICOM library
4
5 Copyright (c) 2006-2014 Mathieu Malaterre
6 All rights reserved.
7 See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.
8
9 This software is distributed WITHOUT ANY WARRANTY; without even
10 the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
11 PURPOSE.  See the above copyright notice for more information.
12
13 =========================================================================*/
14 #ifndef GDCMMODALITYPERFORMEDPROCEDURESTEPCREATEQUERY_H
15 #define GDCMMODALITYPERFORMEDPROCEDURESTEPCREATEQUERY_H
16
17 #include "gdcmBaseQuery.h"
18
```

```
19 namespace gdcm
20 {
25 class GDCM_EXPORT ModalityPerformedProcedureStepCreateQuery :  public BaseQuery{
26   friend class QueryFactory;
27 public:
28   ModalityPerformedProcedureStepCreateQuery( const std::string & iSopInstanceUID );
29
30   gdcm::DataSet GetRequiredDataSet() const;
31   bool ValidateQuery(bool inStrict = true) const override;
32   UIDs::TSName GetAbstractSyntaxUID() const override;
33 };
34
35 } // end namespace gdcm
36
37 #endif // GDCMMODALITYPERFORMEDPROCEDURESTEPCREATEQUERY_H
```

11.511 gdcmModalityPerformedProcedureStepSetQuery.h File Reference

`#include "gdcmBaseQuery.h"`

Include dependency graph for gdcmModalityPerformedProcedureStepSetQuery.h:



Classes

- class gdcm::ModalityPerformedProcedureStepSetQuery

  ModalityPerformedProcedureStepSetQuery.

Namespaces

- namespace gdcm

11.512 gdcmModalityPerformedProcedureStepSetQuery.h

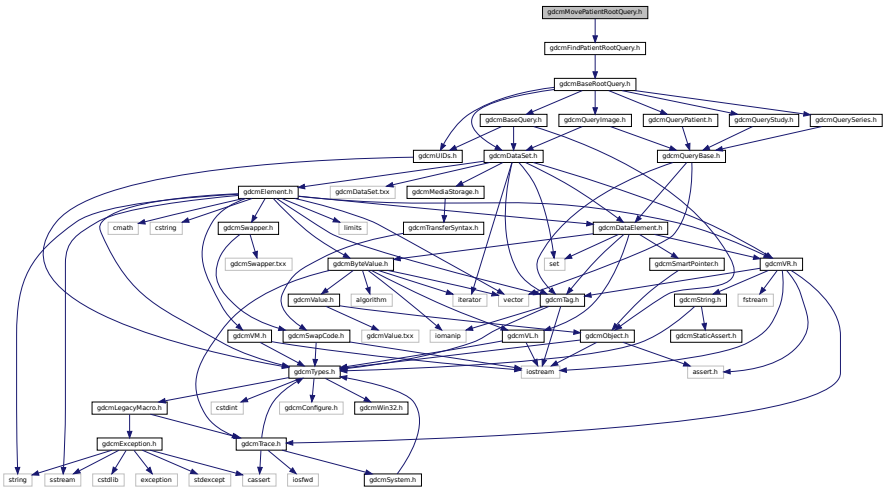
Go to the documentation of this file.

```
1 /*=========================================================================
2
3 Program:  GDCM (Grassroots DICOM). A DICOM library
4
5 Copyright (c) 2006-2011 Mathieu Malaterre
6 All rights reserved.
7 See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.
8
9 This software is distributed WITHOUT ANY WARRANTY; without even
10 the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
11 PURPOSE.  See the above copyright notice for more information.
12
13 =========================================================================*/
14 #ifndef GDCMMODALITYPERFORMEDPROCEDURESTEPSETQUERY_H
15 #define GDCMMODALITYPERFORMEDPROCEDURESTEPSETQUERY_H
16
17 #include "gdcmBaseQuery.h"
18
19 namespace gdcm
20 {
25 class GDCM_EXPORT ModalityPerformedProcedureStepSetQuery :  public BaseQuery{
26   friend class QueryFactory;
27 public:
28   ModalityPerformedProcedureStepSetQuery( const std::string & iSopInstanceUID );
29
30   gdcm::DataSet GetRequiredDataSet() const;
31   bool ValidateQuery(bool inStrict = true) const override;
32   UIDs::TSName GetAbstractSyntaxUID() const override;
33 };
34
35 } // end namespace gdcm
36
37 #endif // GDCMMODALITYPERFORMEDPROCEDURESTEPSETQUERY_H
```

11.513 gdcmMovePatientRootQuery.h File Reference

```
#include "gdcmFindPatientRootQuery.h"
```

Include dependency graph for gdcmMovePatientRootQuery.h:



Classes

- class gdcm::MovePatientRootQuery

  MovePatientRootQuery.

Namespaces

- namespace gdcm

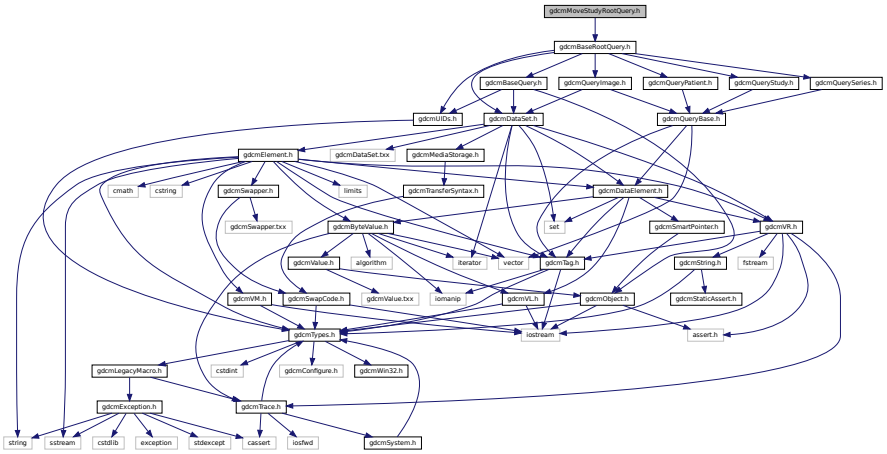
11.514 gdcmMovePatientRootQuery.h

Go to the documentation of this file.

```
1 /*=========================================================================
2
3 Program:  GDCM (Grassroots DICOM). A DICOM library
4
5 Copyright (c) 2006-2011 Mathieu Malaterre
6 All rights reserved.
7 See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.
8
9 This software is distributed WITHOUT ANY WARRANTY; without even
10 the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
11 PURPOSE.  See the above copyright notice for more information.
12
13 =========================================================================*/
14 #ifndef GDCMMOVEPATIENTROOTQUERY_H
15 #define GDCMMOVEPATIENTROOTQUERY_H
16
17 #include "gdcmFindPatientRootQuery.h"
18
19 namespace gdcm
20 {
25 class GDCM_EXPORT MovePatientRootQuery :  public BaseRootQuery
26 {
27   friend class QueryFactory;
28 public:
29   MovePatientRootQuery();
30
31   void InitializeDataSet(const EQueryLevel& inQueryLevel) override;
32
33   std::vector<Tag> GetTagListByLevel(const EQueryLevel& inQueryLevel) override;
34
35   bool ValidateQuery(bool inStrict = true) const override;
36
37   UIDs::TSName GetAbstractSyntaxUID() const override;
38 };
39
40 } // end namespace gdcm
41
42 #endif // GDCMMOVEPATIENTROOTQUERY_H
```

11.515 gdcmMoveStudyRootQuery.h File Reference

`#include "gdcmBaseRootQuery.h"`
Include dependency graph for gdcmMoveStudyRootQuery.h:



Classes

- class gdcm::MoveStudyRootQuery

  MoveStudyRootQuery.

Namespaces

- namespace gdcm

11.516 gdcmMoveStudyRootQuery.h

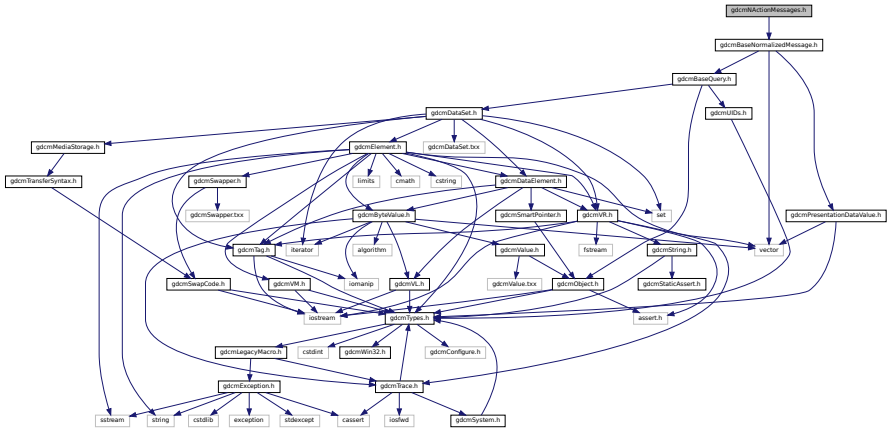
Go to the documentation of this file.

```
1 /*=========================================================================
2
3 Program:  GDCM (Grassroots DICOM). A DICOM library
4
5 Copyright (c) 2006-2011 Mathieu Malaterre
6 All rights reserved.
7 See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.
8
9 This software is distributed WITHOUT ANY WARRANTY; without even
10 the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
11 PURPOSE.  See the above copyright notice for more information.
12
13 =========================================================================*/
14 #ifndef GDCMMOVESTUDYROOTQUERY_H
15 #define GDCMMOVESTUDYROOTQUERY_H
16
17 #include "gdcmBaseRootQuery.h"
18
19 namespace gdcm
20 {
```

```
25 class GDCM_EXPORT MoveStudyRootQuery :  public BaseRootQuery
26 {
27   friend class QueryFactory;
28 public:
29   MoveStudyRootQuery();
30
31   void InitializeDataSet(const EQueryLevel& inQueryLevel) override;
32
33   std::vector<Tag> GetTagListByLevel(const EQueryLevel& inQueryLevel) override;
34
35   bool ValidateQuery(bool inStrict = true) const override;
36
37   UIDs::TSName GetAbstractSyntaxUID() const override;
38 };
39
40 } // end namespace gdcm
41
42 #endif // GDCMMOVESTUDYROOTQUERY_H
```

11.517 gdcmNActionMessages.h File Reference

`#include "gdcmBaseNormalizedMessage.h"`
Include dependency graph for gdcmNActionMessages.h:



Classes

- class gdcm::network::NActionRQ

  NActionRQ.
- class gdcm::network::NActionRSP

  NActionRSP this file defines the messages for the NAction action.

Namespaces

- namespace gdcm
- namespace gdcm::network

11.518 gdcmNActionMessages.h

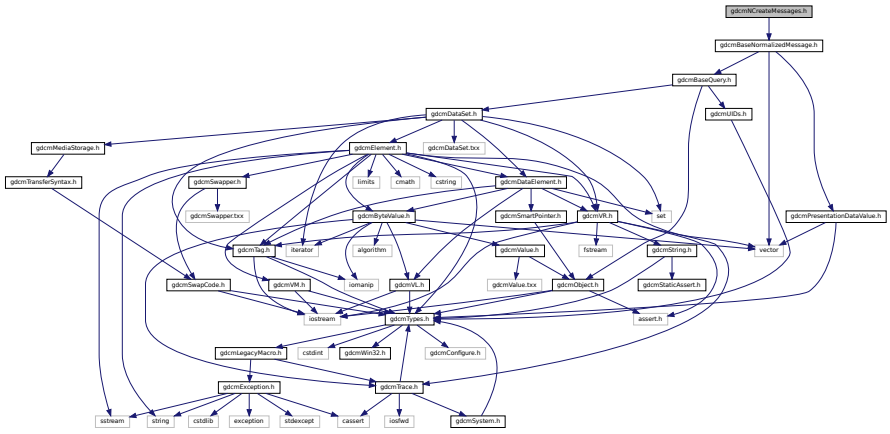
Go to the documentation of this file.

```
1 /*=========================================================================
2
3 Program:  GDCM (Grassroots DICOM). A DICOM library
4
5 Copyright (c) 2006-2014 Mathieu Malaterre
6 All rights reserved.
7 See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.
8
9 This software is distributed WITHOUT ANY WARRANTY; without even
10 the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
11 PURPOSE.  See the above copyright notice for more information.
12
13 =========================================================================*/
14 #ifndef GDCMCNACTIONMESSAGES_H
15 #define GDCMCNACTIONMESSAGES_H
16
17 #include "gdcmBaseNormalizedMessage.h"
18
19 namespace gdcm{
20   namespace network{
21
22 class ULConnection;
23
28 class NActionRQ :  public BaseNormalizedMessage {
29     public:
30       std::vector<PresentationDataValue> ConstructPDV(const ULConnection &inConnection,
31         const BaseQuery* inQuery) override;
32     };
33
38     class NActionRSP :  public BaseNormalizedMessage {
39     public:
40       std::vector<PresentationDataValue> ConstructPDVByDataSet(const DataSet* inDataSet);
41     };
42   }
43 }
44 #endif // GDCMCNACTIONMESSAGES_H
```

11.519 gdcmNCreateMessages.h File Reference

`#include "gdcmBaseNormalizedMessage.h"`

Include dependency graph for gdcmNCreateMessages.h:



Classes

- class gdcm::network::NCreateRQ

  NCreateRQ.
- class gdcm::network::NCreateRSP

  NCreateRSP this file defines the messages for the ncreate action.

Namespaces

- namespace gdcm
- namespace gdcm::network

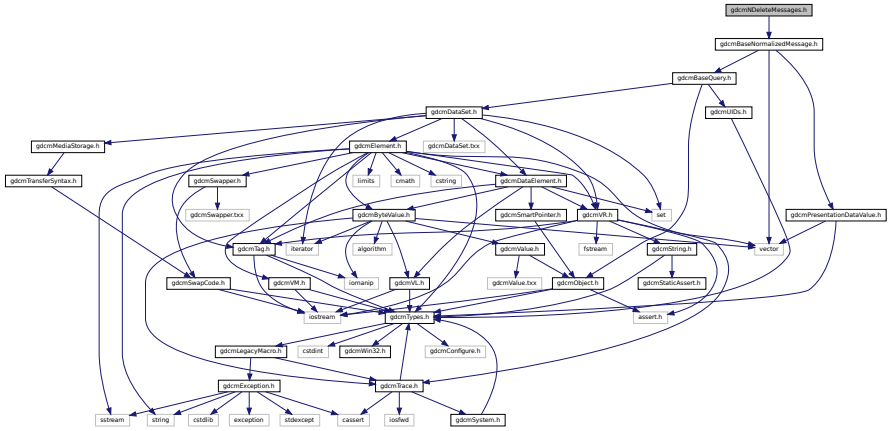
11.520 gdcmNCreateMessages.h

Go to the documentation of this file.

```
1 /*=========================================================================
2
3 Program:  GDCM (Grassroots DICOM). A DICOM library
4
5 Copyright (c) 2006-2014 Mathieu Malaterre
6 All rights reserved.
7 See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.
8
9 This software is distributed WITHOUT ANY WARRANTY; without even
10 the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
11 PURPOSE.  See the above copyright notice for more information.
12
13 =========================================================================*/
14 #ifndef GDCMCNCREATEMESSAGES_H
15 #define GDCMCNCREATEMESSAGES_H
16
17 #include "gdcmBaseNormalizedMessage.h"
18
19 namespace gdcm{
20   namespace network{
21
22 class ULConnection;
23
28 class NCreateRQ :  public BaseNormalizedMessage {
29     public:
30       std::vector<PresentationDataValue> ConstructPDV(const ULConnection &inConnection,
31         const BaseQuery* inQuery) override;
32     };
33
38     class NCreateRSP :  public BaseNormalizedMessage {
39     public:
40       std::vector<PresentationDataValue> ConstructPDVByDataSet(const DataSet* inDataSet);
41     };
42   }
43 }
44 #endif // GDCMCNCREATEMESSAGES_H
```

11.521 gdcmNDeleteMessages.h File Reference

`#include "gdcmBaseNormalizedMessage.h"`
Include dependency graph for gdcmNDeleteMessages.h:



Classes

- class gdcm::network::NDeleteRQ

  NDeleteRQ.
- class gdcm::network::NDeleteRSP

  NDeleteRSP this file defines the messages for the ndelete action.

Namespaces

- namespace gdcm
- namespace gdcm::network

11.522 gdcmNDeleteMessages.h

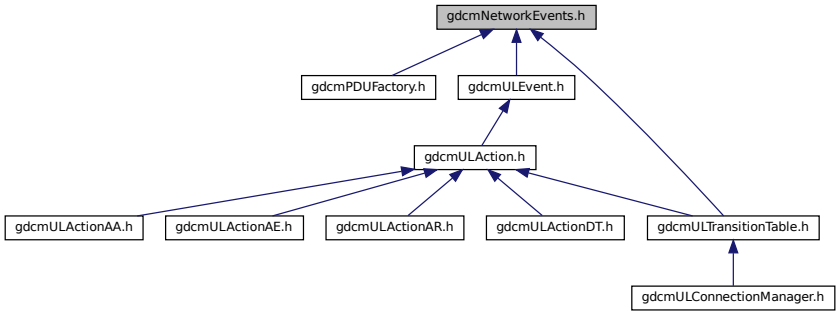
Go to the documentation of this file.

```
1 /*=========================================================================
2
3 Program:  GDCM (Grassroots DICOM). A DICOM library
4
5 Copyright (c) 2006-2014 Mathieu Malaterre
6 All rights reserved.
7 See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.
8
9 This software is distributed WITHOUT ANY WARRANTY; without even
10 the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
11 PURPOSE.  See the above copyright notice for more information.
12
13 =========================================================================*/
14 #ifndef GDCMCNDELETEMESSAGES_H
15 #define GDCMCNDELETEMESSAGES_H
16
```

```
17 #include "gdcmBaseNormalizedMessage.h"
18
19 namespace gdcm{
20   namespace network{
21
22 class ULConnection;
23
28 class NDeleteRQ :  public BaseNormalizedMessage {
29     public:
30       std::vector<PresentationDataValue> ConstructPDV(const ULConnection &inConnection,
31         const BaseQuery* inQuery) override;
32     };
33
38     class NDeleteRSP :  public BaseNormalizedMessage {
39     public:
40       std::vector<PresentationDataValue> ConstructPDVByDataSet(const DataSet* inDataSet);
41     };
42   }
43 }
44 #endif // GDCMCNDELETEMESSAGES_H
```

11.523 gdcmNetworkEvents.h File Reference

This graph shows which files directly or indirectly include this file:



Namespaces

- namespace gdcm
- namespace gdcm::network

Enumerations

- enum gdcm::network::EEventID {
  gdcm::network::eAASSOCIATERequestLocalUser = 0 ,
  gdcm::network::eTransportConnConfirmLocal ,
  gdcm::network::eASSOCIATE_ACPDUreceived ,
  gdcm::network::eASSOCIATE_RJPDUreceived ,
  gdcm::network::eTransportConnIndicLocal ,
  gdcm::network::eAASSOCIATE_RQPDUreceived ,
  gdcm::network::eAASSOCIATEresponseAccept ,

gdcm::network::eAASSOCIATEresponseReject ,
gdcm::network::ePDATArequest ,
gdcm::network::ePDATATFPDU ,
gdcm::network::eARELEASERequest ,
gdcm::network::eARELEASE_RQPDUReceivedOpen ,
gdcm::network::eARELEASE_RPPDUReceived ,
gdcm::network::eARELEASEResponse ,
gdcm::network::eAABORTRequest ,
gdcm::network::eAABORTPDUReceivedOpen ,
gdcm::network::eTransportConnectionClosed ,
gdcm::network::eARTIMTimerExpired ,
gdcm::network::eUnrecognizedPDUReceived ,
gdcm::network::eEventDoesNotExist }

Variables

- const int gdcm::network::cMaxEventID = eEventDoesNotExist

11.524 gdcmNetworkEvents.h

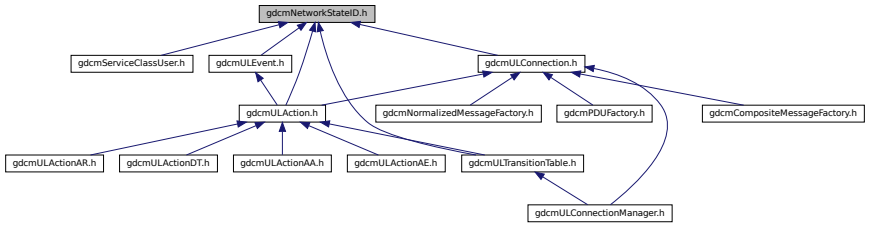
Go to the documentation of this file.

```
1 /*=========================================================================
2 *
3 *  Copyright NumFOCUS
4 *
5 *  Licensed under the Apache License, Version 2.0 (the "License");
6 *  you may not use this file except in compliance with the License.
7 *  You may obtain a copy of the License at
8 *
9 *         http://www.apache.org/licenses/LICENSE-2.0.txt
10 *
11 *  Unless required by applicable law or agreed to in writing, software
12 *  distributed under the License is distributed on an "AS IS" BASIS,
13 *  WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
14 *  See the License for the specific language governing permissions and
15 *  limitations under the License.
16 *
17 *=========================================================================*/
18 /*
19 The NetworkEvents enumeration defines the inputs into the state of the network connection.
20
21 These inputs can come either from user input or input from other things on the socket,
22 ie, responses from the peer or ARTIM timeouts.
23
24 Note that this enumeration is not 'power of two', like the states, because you can't have
25 multiple simultaneous events.    Multiple state outputs in transition tables, however, is possible.
26
27 */
28 #ifndef GDCMNETWORKEVENTS_H
29 #define GDCMNETWORKEVENTS_H
30
31 namespace gdcm {
32   namespace network{
33     typedef enum {
34       eAASSOCIATERequestLocalUser = 0,
35       eTransportConnConfirmLocal,
36       eASSOCIATE_ACPDUreceived,
37       eASSOCIATE_RJPDUreceived,
38       eTransportConnIndicLocal,
39       eAASSOCIATE_RQPDUreceived,
40       eAASSOCIATEresponseAccept,
41       eAASSOCIATEresponseReject,
42       ePDATArequest,
43       ePDATATFPDU,
44       eARELEASERequest,
```

```
45        eARELEASE_RQPDUReceivedOpen,
46        eARELEASE_RPPDUReceived,
47        eARELEASEResponse,
48        eAABORTRequest,
49        eAABORTPDUReceivedOpen,
50        eTransportConnectionClosed,
51        eARTIMTimerExpired,
52        eUnrecognizedPDUReceived,
53        eEventDoesNotExist
54      } EEventID;
55
56      const int cMaxEventID = eEventDoesNotExist;
57    }
58 }
59
60 #endif //NETWORKEVENTS_H
```

11.525 gdcmNetworkStateID.h File Reference

This graph shows which files directly or indirectly include this file:



Namespaces

- namespace gdcm
- namespace gdcm::network

Enumerations

- enum gdcm::network::EStateID {
  gdcm::network::eStaDoesNotExist = 0 ,
  gdcm::network::eSta1Idle = 1 ,
  gdcm::network::eSta2Open = 2 ,
  gdcm::network::eSta3WaitLocalAssoc = 4 ,
  gdcm::network::eSta4LocalAssocDone = 8 ,
  gdcm::network::eSta5WaitRemoteAssoc = 16 ,
  gdcm::network::eSta6TransferReady = 32 ,
  gdcm::network::eSta7WaitRelease = 64 ,
  gdcm::network::eSta8WaitLocalRelease = 128 ,
  gdcm::network::eSta9ReleaseCollisionRqLocal = 256 ,
  gdcm::network::eSta10ReleaseCollisionAc = 512 ,
  gdcm::network::eSta11ReleaseCollisionRq = 1024 ,
  gdcm::network::eSta12ReleaseCollisionAcLocal = 2048 ,
  gdcm::network::eSta13AwaitingClose = 4096 }

Functions

- int gdcm::network::GetStateIndex (EStateID inState)

Variables

- const int gdcm::network::cMaxStateID = 13

11.526 gdcmNetworkStateID.h

Go to the documentation of this file.

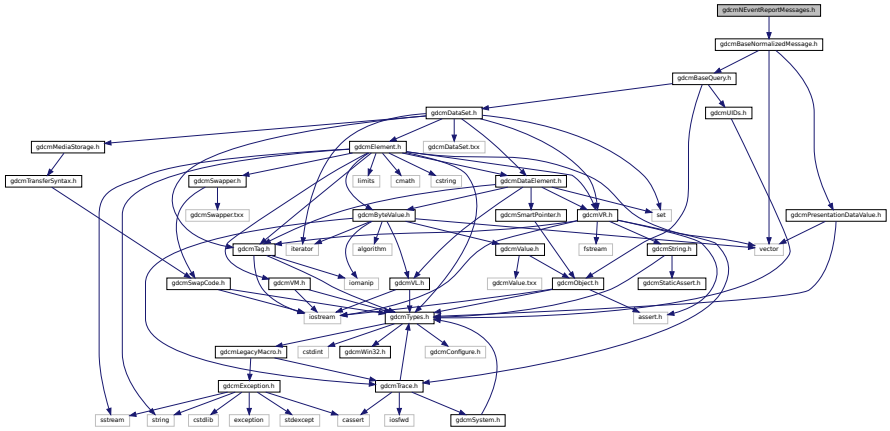
```
1 /*=========================================================================
2 *
3 *  Copyright NumFOCUS
4 *
5 *  Licensed under the Apache License, Version 2.0 (the "License");
6 *  you may not use this file except in compliance with the License.
7 *  You may obtain a copy of the License at
8 *
9 *         http://www.apache.org/licenses/LICENSE-2.0.txt
10 *
11 *  Unless required by applicable law or agreed to in writing, software
12 *  distributed under the License is distributed on an "AS IS" BASIS,
13 *  WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
14 *  See the License for the specific language governing permissions and
15 *  limitations under the License.
16 *
17 *=========================================================================*/
18 #ifndef GDCMNETWORKSTATEID_H
19 #define GDCMNETWORKSTATEID_H
20
21 namespace gdcm {
22   namespace network {
23
32     enum EStateID {
33       eStaDoesNotExist = 0,
34       eSta1Idle = 1,
35       eSta2Open = 2,
36       eSta3WaitLocalAssoc = 4,
37       eSta4LocalAssocDone = 8,
38       eSta5WaitRemoteAssoc = 16,
39       eSta6TransferReady = 32,
40       eSta7WaitRelease = 64,
41       eSta8WaitLocalRelease = 128,
42       eSta9ReleaseCollisionRqLocal = 256,
43       eSta10ReleaseCollisionAc = 512,
44       eSta11ReleaseCollisionRq = 1024,
45       eSta12ReleaseCollisionAcLocal = 2048,
46       eSta13AwaitingClose = 4096
47     };
48
49     const int cMaxStateID = 13;
50
51     //the transition table is built on state indices
52     //this function will produce the index from the power-of-two EStateID
53     inline int GetStateIndex(EStateID inState){
54       switch (inState){
55         case eStaDoesNotExist:
56         default:
57           return -1;
58         case eSta1Idle:
59           return 0;
60         case eSta2Open:
61           return 1;
62         case eSta3WaitLocalAssoc:
63           return 2;
64         case eSta4LocalAssocDone:
65           return 3;
66         case eSta5WaitRemoteAssoc:
67           return 4;
```

```
68          case eSta6TransferReady:
69            return 5;
70          case eSta7WaitRelease:
71            return 6;
72          case eSta8WaitLocalRelease:
73            return 7;
74          case eSta9ReleaseCollisionRqLocal:
75            return 8;
76          case eSta10ReleaseCollisionAc:
77            return 9;
78          case eSta11ReleaseCollisionRq:
79            return 10;
80          case eSta12ReleaseCollisionAcLocal:
81            return 11;
82          case eSta13AwaitingClose:
83            return 12;
84        }
85      }
86    }
87 }
88
89 #endif //GDCMNETWORKSTATEID_H
```

11.527 gdcmNEventReportMessages.h File Reference

`#include "gdcmBaseNormalizedMessage.h"`

Include dependency graph for gdcmNEventReportMessages.h:



Classes

- class gdcm::network::NEventReportRQ

  NEventReportRQ.
- class gdcm::network::NEventReportRSP

  NEventReportRSP this file defines the messages for the neventreport action.

Namespaces

- namespace gdcm
- namespace gdcm::network

11.528 gdcmNEventReportMessages.h

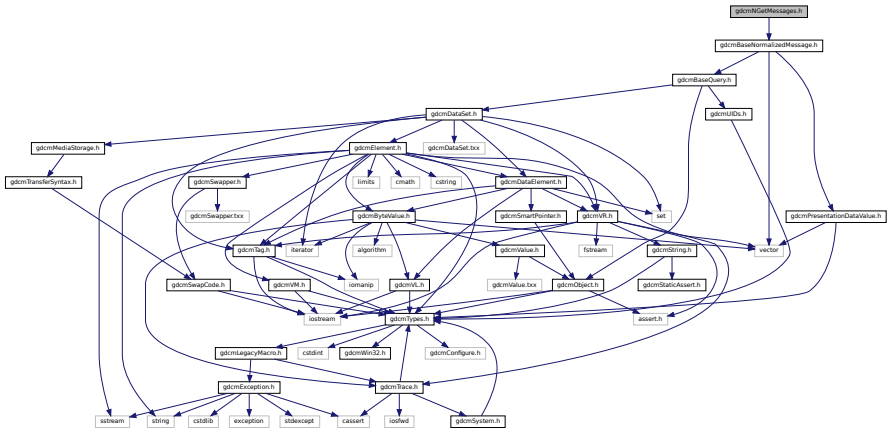
Go to the documentation of this file.

```
1 /*=========================================================================
2
3 Program:  GDCM (Grassroots DICOM). A DICOM library
4
5 Copyright (c) 2006-2014 Mathieu Malaterre
6 All rights reserved.
7 See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.
8
9 This software is distributed WITHOUT ANY WARRANTY; without even
10 the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
11 PURPOSE.  See the above copyright notice for more information.
12
13 =========================================================================*/
14 #ifndef GDCMCNEVENTREPORTMESSAGES_H
15 #define GDCMCNEVENTREPORTMESSAGES_H
16
17 #include "gdcmBaseNormalizedMessage.h"
18
19 namespace gdcm{
20   namespace network{
21
22 class ULConnection;
23
28 class NEventReportRQ :  public BaseNormalizedMessage {
29     public:
30       std::vector<PresentationDataValue> ConstructPDV(const ULConnection &inConnection,
31         const BaseQuery* inQuery) override;
32     };
33
38     class NEventReportRSP :  public BaseNormalizedMessage {
39     public:
40       std::vector<PresentationDataValue> ConstructPDVByDataSet(const DataSet* inDataSet);
41     };
42   }
43 }
44 #endif // GDCMCNEVENTREPORTMESSAGES_H
```

11.529 gdcmNGetMessages.h File Reference

```
#include "gdcmBaseNormalizedMessage.h"
```

Include dependency graph for gdcmNGetMessages.h:



Classes

- class gdcm::network::NGetRQ

  NGetRQ.
- class gdcm::network::NGetRSP

  NGetRSP this file defines the messages for the nget action.

Namespaces

- namespace gdcm
- namespace gdcm::network

11.530 gdcmNGetMessages.h

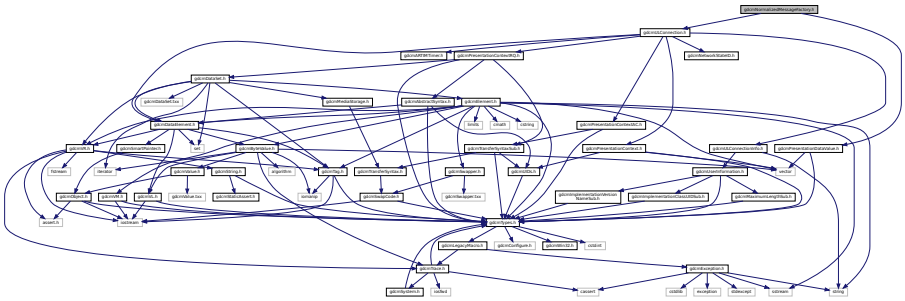
Go to the documentation of this file.

```
1 /*=========================================================================
2
3 Program:  GDCM (Grassroots DICOM). A DICOM library
4
5 Copyright (c) 2006-2014 Mathieu Malaterre
6 All rights reserved.
7 See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.
8
9 This software is distributed WITHOUT ANY WARRANTY; without even
10 the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
11 PURPOSE.  See the above copyright notice for more information.
12
13 =========================================================================*/
14 #ifndef GDCMCNGETMESSAGES_H
15 #define GDCMCNGETMESSAGES_H
16
17 #include "gdcmBaseNormalizedMessage.h"
18
19 namespace gdcm{
20   namespace network{
21
22 class ULConnection;
23
28 class NGetRQ :  public BaseNormalizedMessage {
29     public:
30       std::vector<PresentationDataValue> ConstructPDV(const ULConnection &inConnection,
31         const BaseQuery* inQuery) override;
32     };
33
38     class NGetRSP :  public BaseNormalizedMessage {
39     public:
40       std::vector<PresentationDataValue> ConstructPDVByDataSet(const DataSet* inDataSet);
41     };
42   }
43 }
44 #endif // GDCMCNGETMESSAGES_H
```

11.531 gdcmNormalizedMessageFactory.h File Reference

```
#include "gdcmPresentationDataValue.h"
#include "gdcmULConnection.h"
```

Include dependency graph for gdcmNormalizedMessageFactory.h:



Classes

- class gdcm::network::NormalizedMessageFactory

Namespaces

- namespace gdcm
- namespace gdcm::network

11.532 gdcmNormalizedMessageFactory.h

Go to the documentation of this file.

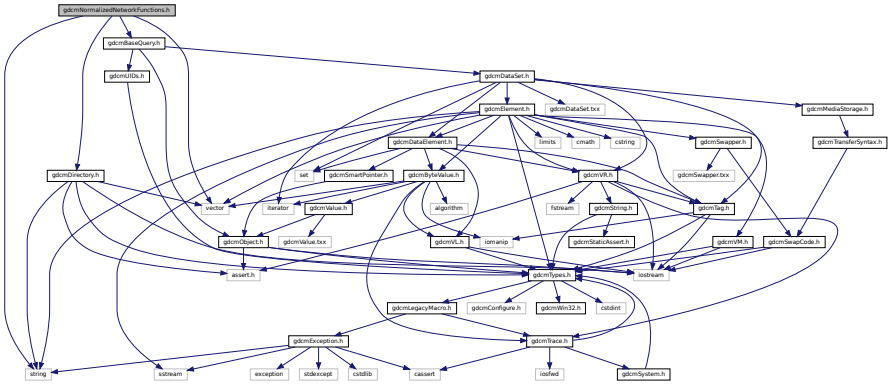
```
1 /*=========================================================================
2
3 Program:  GDCM (Grassroots DICOM). A DICOM library
4
5 Copyright (c) 2006-2014 Mathieu Malaterre
6 All rights reserved.
7 See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.
8
9 This software is distributed WITHOUT ANY WARRANTY; without even
10 the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
11 PURPOSE.  See the above copyright notice for more information.
12
13 =========================================================================*/
14 #ifndef GDCMNORMALIZEDMESSAGEFACTORY_H
15 #define GDCMNORMALIZEDMESSAGEFACTORY_H
16
17 #include "gdcmPresentationDataValue.h"
18 #include "gdcmULConnection.h"
19
20 namespace gdcm {
21   class BaseQuery;
22   class File;
23   namespace network {
24     class BasePDU;
25
26 class NormalizedMessageFactory
27 {
28     public:
29       static  std::vector<PresentationDataValue> ConstructNEventReport (const ULConnection& inConnection,
      const BaseQuery* inQuery);
30       static  std::vector<PresentationDataValue> ConstructNGet     (const ULConnection& inConnection, const
      BaseQuery* inQuery);
31       static  std::vector<PresentationDataValue> ConstructNSet     (const ULConnection& inConnection, const
      BaseQuery* inQuery);
```

```
32       static  std::vector<PresentationDataValue> ConstructNAction   (const ULConnection& inConnection, const
     BaseQuery* inQuery);
33       static  std::vector<PresentationDataValue> ConstructNCreate   (const ULConnection& inConnection, const
     BaseQuery* inQuery);
34       static  std::vector<PresentationDataValue> ConstructNDelete   (const ULConnection& inConnection, const
     BaseQuery* inQuery);
35
36
37     };
38   }
39 }
40
41 #endif // GDCMNORMALIZEDMESSAGEFACTORY_H
```

11.533 gdcmNormalizedNetworkFunctions.h File Reference

```
#include "gdcmDirectory.h"
#include "gdcmBaseQuery.h"
#include <vector>
#include <string>
```

Include dependency graph for gdcmNormalizedNetworkFunctions.h:



Classes

- class gdcm::NormalizedNetworkFunctions

  Normalized Network Functions.

Namespaces

- namespace gdcm

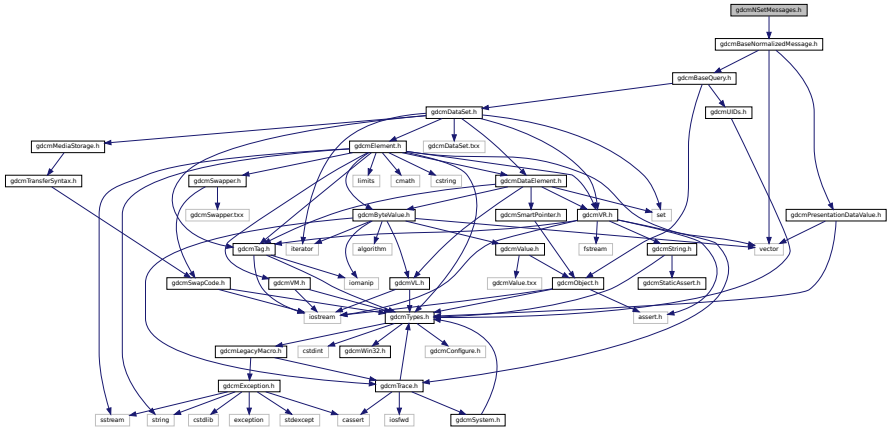
11.534 gdcmNormalizedNetworkFunctions.h

Go to the documentation of this file.

```
1 /*=========================================================================
2
3 Program:  GDCM (Grassroots DICOM). A DICOM library
4
5 Copyright (c) 2006-2014 Mathieu Malaterre
6 All rights reserved.
7 See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.
8
9 This software is distributed WITHOUT ANY WARRANTY; without even
10 the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
11 PURPOSE.  See the above copyright notice for more information.
12
13 =========================================================================*/
14 #ifndef GDCMNORMALIZEDNETWORKFUNCTIONS_H
15 #define GDCMNORMALIZEDNETWORKFUNCTIONS_H
16
17 #include "gdcmDirectory.h"
18 #include "gdcmBaseQuery.h" // EQueryLevel / EQueryType
19
20 #include <vector>
21 #include <string>
22
23 namespace gdcm
24 {
46 class GDCM_EXPORT NormalizedNetworkFunctions
47 {
48 public:
49   static BaseQuery* ConstructQuery(  const std::string & sopInstanceUID,
50                      const DataSet& queryds, ENQueryType queryType = eCreateMMPS );
51   static bool NEventReport( const char *remote, uint16_t portno,
52                 const BaseQuery* query, std::vector<DataSet> &retDataSets,
53                 const char *aetitle, const char *call );
54   static bool NGet( const char *remote, uint16_t portno,
55                       const BaseQuery* query, std::vector<DataSet> &retDataSets,
56                       const char *aetitle, const char *call );
57   static bool NSet( const char *remote, uint16_t portno,
58                       const BaseQuery* query, std::vector<DataSet> &retDataSets,
59                       const char *aetitle, const char *call );
60   static bool NAction( const char *remote, uint16_t portno,
61                       const BaseQuery* query, std::vector<DataSet> &retDataSets,
62                       const char *aetitle, const char *call );
63   static bool NCreate( const char *remote, uint16_t portno,
64                       BaseQuery* query, std::vector<DataSet> &retDataSets,
65                       const char *aetitle, const char *call );
66   static bool NDelete( const char *remote, uint16_t portno,
67                       const BaseQuery* query, std::vector<DataSet> &retDataSets,
68                       const char *aetitle, const char *call );
69 };
70
71 } // end namespace gdcm
72
73 #endif // GDCMCOMPOSITENETWORKFUNCTIONS_H
```

11.535 gdcmNSetMessages.h File Reference

`#include "gdcmBaseNormalizedMessage.h"`
Include dependency graph for gdcmNSetMessages.h:



Classes

- class gdcm::network::NSetRQ

  NSetRQ.
- class gdcm::network::NSetRSP

  NSetRSP this file defines the messages for the nset action.

Namespaces

- namespace gdcm
- namespace gdcm::network

11.536 gdcmNSetMessages.h

Go to the documentation of this file.

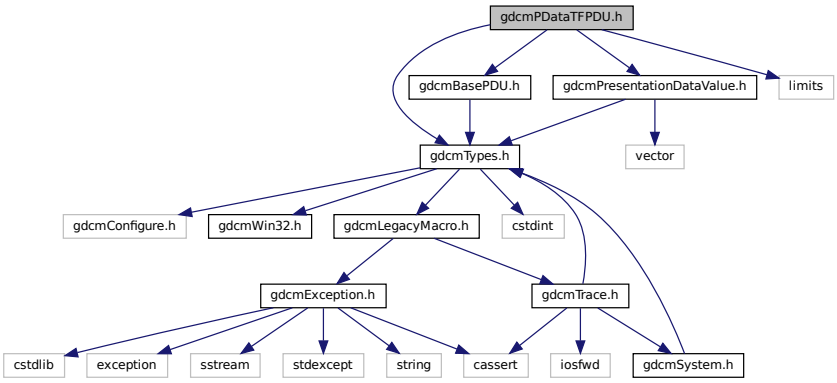
```
1 /*=========================================================================
2
3 Program:  GDCM (Grassroots DICOM). A DICOM library
4
5 Copyright (c) 2006-2014 Mathieu Malaterre
6 All rights reserved.
7 See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.
8
9 This software is distributed WITHOUT ANY WARRANTY; without even
10 the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
11 PURPOSE.  See the above copyright notice for more information.
12
13 =========================================================================*/
14 #ifndef GDCMCNSETMESSAGES_H
15 #define GDCMCNSETMESSAGES_H
16
```

```
17 #include "gdcmBaseNormalizedMessage.h"
18
19 namespace gdcm{
20   namespace network{
21
22 class ULConnection;
23
28 class NSetRQ :  public BaseNormalizedMessage {
29     public:
30       std::vector<PresentationDataValue> ConstructPDV(const ULConnection &inConnection,
31         const BaseQuery* inQuery) override;
32     };
33
38     class NSetRSP :  public BaseNormalizedMessage {
39     public:
40       std::vector<PresentationDataValue> ConstructPDVByDataSet(const DataSet* inDataSet);
41     };
42   }
43 }
44 #endif // GDCMCNSETMESSAGES_H
```

11.537 gdcmPDataTFPDU.h File Reference

```
#include "gdcmTypes.h"
#include "gdcmPresentationDataValue.h"
#include "gdcmBasePDU.h"
#include <limits>
```

Include dependency graph for gdcmPDataTFPDU.h:



Classes

- class gdcm::network::PDataTFPDU

  PDataTFPDU.

Namespaces

- namespace gdcm
- namespace gdcm::network

11.538 gdcmPDataTFPDU.h

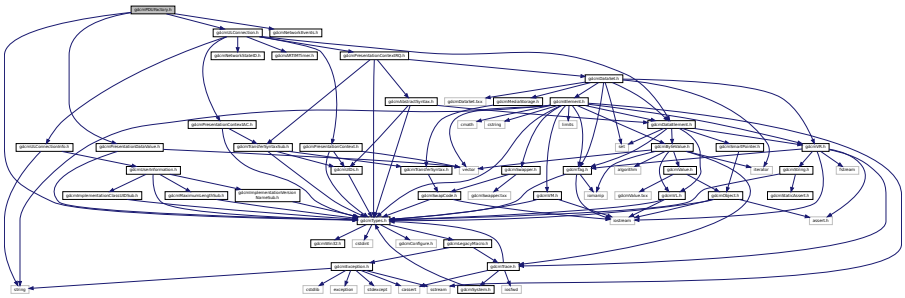
Go to the documentation of this file.

```
1 /*=========================================================================
2
3 Program:  GDCM (Grassroots DICOM). A DICOM library
4
5 Copyright (c) 2006-2011 Mathieu Malaterre
6 All rights reserved.
7 See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.
8
9 This software is distributed WITHOUT ANY WARRANTY; without even
10 the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
11 PURPOSE.  See the above copyright notice for more information.
12
13 =========================================================================*/
14 #ifndef GDCMPDATATFPDU_H
15 #define GDCMPDATATFPDU_H
16
17 #include "gdcmTypes.h"
18 #include "gdcmPresentationDataValue.h"
19 #include "gdcmBasePDU.h"
20 #include <limits>
21
22 namespace gdcm
23 {
24
25 namespace network
26 {
27
33 class GDCM_EXPORT PDataTFPDU :  public BasePDU
34 {
35 public:
36   PDataTFPDU();
37   std::istream &Read(std::istream &is) override;
38   const std::ostream &Write(std::ostream &os) const override;
39
41   size_t Size() const override;
42
43   void AddPresentationDataValue( PresentationDataValue const &pdv ) {
44     V.push_back( pdv );
45     assert(Size() < std::numeric_limits<uint32_t>::max());
46     ItemLength = (uint32_t)Size() - 6;
47     }
48
49   typedef std::vector<PresentationDataValue>::size_type SizeType;
50   PresentationDataValue const &GetPresentationDataValue(SizeType i)const {
51     assert( !V.empty() && i < V.size() );
52     return V[i];
53   }
54   SizeType GetNumberOfPresentationDataValues()const {
55     return V.size();
56     }
57
58   void Print(std::ostream &os) const override;
59   bool IsLastFragment() const override;
60
61 protected:
62   std::istream &ReadInto(std::istream &is, std::ostream &os);
63 private:
64   static const uint8_t ItemType; // PDUType ?
65   static const uint8_t Reserved2;
66   uint32_t ItemLength; // PDU Length ?
67   std::vector<PresentationDataValue> V;
68 };
69
70 } // end namespace network
71
72 } // end namespace gdcm
73
74 #endif //GDCMPDATATFPDU_H
```

11.539 gdcmPDUFactory.h File Reference

```
#include "gdcmTypes.h"
#include "gdcmNetworkEvents.h"
#include "gdcmULConnection.h"
#include "gdcmPresentationDataValue.h"
```

Include dependency graph for gdcmPDUFactory.h:



Classes

- class gdcm::network::PDUFactory

  PDUFactory basically, given an initial byte, construct the.

Namespaces

- namespace gdcm
- namespace gdcm::network

11.540 gdcmPDUFactory.h

Go to the documentation of this file.

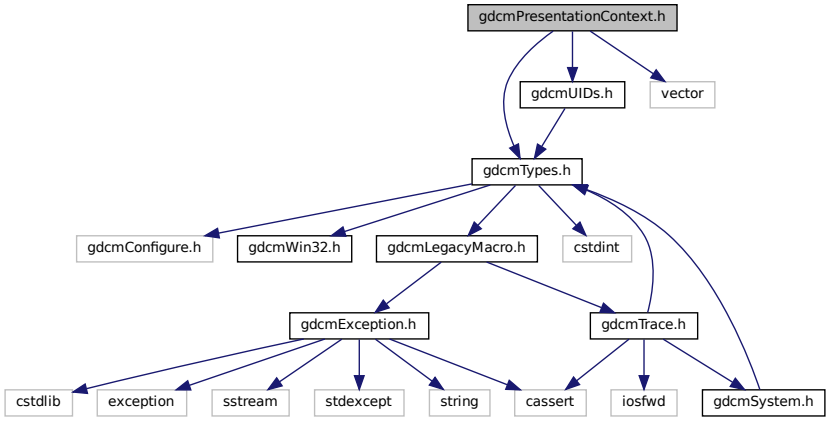
```
1 /*=========================================================================
2 *
3 *  Copyright NumFOCUS
4 *
5 *  Licensed under the Apache License, Version 2.0 (the "License");
6 *  you may not use this file except in compliance with the License.
7 *  You may obtain a copy of the License at
8 *
9 *         http://www.apache.org/licenses/LICENSE-2.0.txt
10 *
11 *  Unless required by applicable law or agreed to in writing, software
12 *  distributed under the License is distributed on an "AS IS" BASIS,
13 *  WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
14 *  See the License for the specific language governing permissions and
15 *  limitations under the License.
16 *
17 *=========================================================================*/
18 #ifndef GDCMPDUFACTORY_H
19 #define GDCMPDUFACTORY_H
20
21 #include "gdcmTypes.h"
22 #include "gdcmNetworkEvents.h"
```

```
23 #include "gdcmULConnection.h"
24 #include "gdcmPresentationDataValue.h"
25
26 namespace gdcm{
27   class BaseRootQuery;
28   class BaseQuery;
29   class File;
30   namespace network{
31     class BasePDU;
32
38   class PDUFactory {
39       public:
40       static BasePDU* ConstructPDU(uint8_t itemtype);//eventually needs to be smartpointer'd
41       static EEventID DetermineEventByPDU(const BasePDU* inPDU);
42       static BasePDU* ConstructReleasePDU();
43       static BasePDU* ConstructAbortPDU();
44
45       //these are the composite PDU construction methods for the PDataPDUs.
46       //basically, builds a pdatapdu, and then puts the appropriate information in
47       //for the appropriate composite service (c-echo, c-find, c-store, c-get, c-move)
48       //the connection is necessary to construct the stream of PDVs that will
49       //be then placed into the vector of PDUs
50       static std::vector<BasePDU*> CreateCEchoPDU(const ULConnection& inConnection);
51       static std::vector<BasePDU*> CreateCStoreRQPDU(const ULConnection& inConnection, const File &file, bool
     writeDataSet = true );
52       static std::vector<BasePDU*> CreateCStoreRSPPDU(const DataSet *inDataSet, const BasePDU* inPC);
53       static std::vector<BasePDU*> CreateCFindPDU(const ULConnection& inConnection, const BaseRootQuery*
     inRootQuery);
54       static std::vector<BasePDU*> CreateCMovePDU(const ULConnection& inConnection, const BaseRootQuery*
     inRootQuery);
55
56      static std::vector<BasePDU*> CreateNEventReportPDU (const ULConnection& inConnection, const BaseQuery
     *inQuery);
57      static std::vector<BasePDU*> CreateNGetPDU     (const ULConnection& inConnection, const BaseQuery
     *inQuery);
58      static std::vector<BasePDU*> CreateNSetPDU     (const ULConnection& inConnection, const BaseQuery
     *inQuery);
59      static std::vector<BasePDU*> CreateNActionPDU     (const ULConnection& inConnection, const BaseQuery
     *inQuery);
60      static std::vector<BasePDU*> CreateNCreatePDU     (const ULConnection& inConnection, const BaseQuery
     *inQuery);
61      static std::vector<BasePDU*> CreateNDeletePDU     (const ULConnection& inConnection, const BaseQuery
     *inQuery);
62
63
64       //given data pdus, produce the presentation data values stored within.
65       //all operations have these as the payload of the data sending operation
66       //however, echo does not have a dataset in the pdv.
67       static std::vector<PresentationDataValue> GetPDVs(const std::vector<BasePDU*> & inDataPDUs);
68     };
69   }
70 }
71 #endif //GDCMPDUFACTORY_H
```

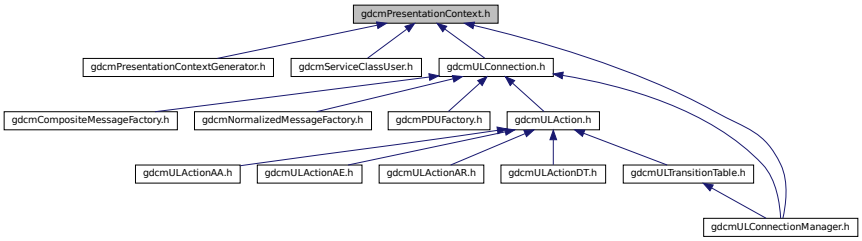
11.541 gdcmPresentationContext.h File Reference

```
#include "gdcmTypes.h"
#include "gdcmUIDs.h"
#include <vector>
```

Include dependency graph for gdcmPresentationContext.h:



This graph shows which files directly or indirectly include this file:



Classes

- class gdcm::PresentationContext

  PresentationContext.

Namespaces

- namespace gdcm

11.542 gdcmPresentationContext.h

Go to the documentation of this file.

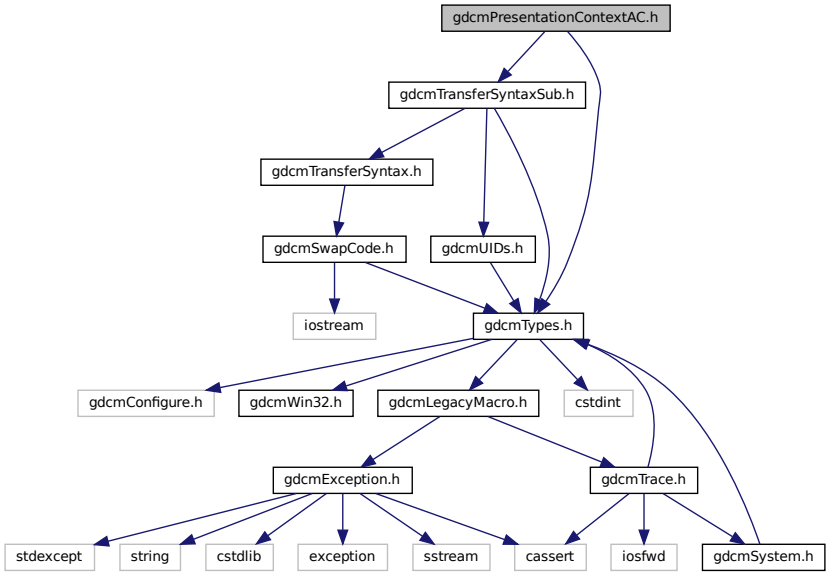
```
1 /*=========================================================================
2
```

```
3 Program:  GDCM (Grassroots DICOM). A DICOM library
4
5 Copyright (c) 2006-2011 Mathieu Malaterre
6 All rights reserved.
7 See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.
8
9 This software is distributed WITHOUT ANY WARRANTY; without even
10 the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
11 PURPOSE.  See the above copyright notice for more information.
12
13 =========================================================================*/
14 #ifndef GDCMPRESENTATIONCONTEXT_H
15 #define GDCMPRESENTATIONCONTEXT_H
16
17 #include "gdcmTypes.h"
18 #include "gdcmUIDs.h"
19
20 #include <vector>
21
22 namespace gdcm
23 {
24
29 class GDCM_EXPORT PresentationContext
30 {
31 public:
32   PresentationContext();
33
37   PresentationContext( UIDs::TSName asname,
38     UIDs::TSName tsname = UIDs::ImplicitVRLittleEndianDefaultTransferSyntaxforDICOM );
39
40   void SetAbstractSyntax( const char *absyn ) { AbstractSyntax = absyn; }
41   const char *GetAbstractSyntax()const { return AbstractSyntax.c_str(); }
42
43   void AddTransferSyntax( const char *tsstr );
44   typedef std::vector<std::string> TransferSyntaxArrayType;
45   typedef TransferSyntaxArrayType::size_type SizeType;
46   const char *GetTransferSyntax(SizeType i)const { return TransferSyntaxes[i].c_str(); }
47   SizeType GetNumberOfTransferSyntaxes()const { return TransferSyntaxes.size(); }
48
49   void SetPresentationContextID( uint8_t id );
50   uint8_t GetPresentationContextID() const;
51
52   void Print(std::ostream &os) const;
53
54   bool operator==(const PresentationContext & pc)const
55 {
56     assert( TransferSyntaxes.size() == 1 ); // TODO
57     assert( pc.TransferSyntaxes.size() == 1 );
58     return AbstractSyntax == pc.AbstractSyntax && TransferSyntaxes == pc.TransferSyntaxes;
59     }
60
61 protected :
62   std::string AbstractSyntax;
63   std::vector<std::string> TransferSyntaxes;
64   uint8_t /*PresentationContext*/ID;
65 };
66
67 } // end namespace gdcm
68
69 #endif //GDCMPRESENTATIONCONTEXT_H
```

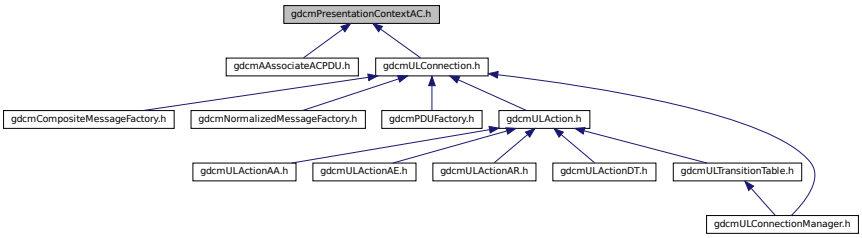
11.543 gdcmPresentationContextAC.h File Reference

```
#include "gdcmTypes.h"
#include "gdcmTransferSyntaxSub.h"
```

Include dependency graph for gdcmPresentationContextAC.h:



This graph shows which files directly or indirectly include this file:



Classes

- class gdcm::network::PresentationContextAC

  PresentationContextAC.

Namespaces

- namespace gdcm
- namespace gdcm::network

11.544 gdcmPresentationContextAC.h

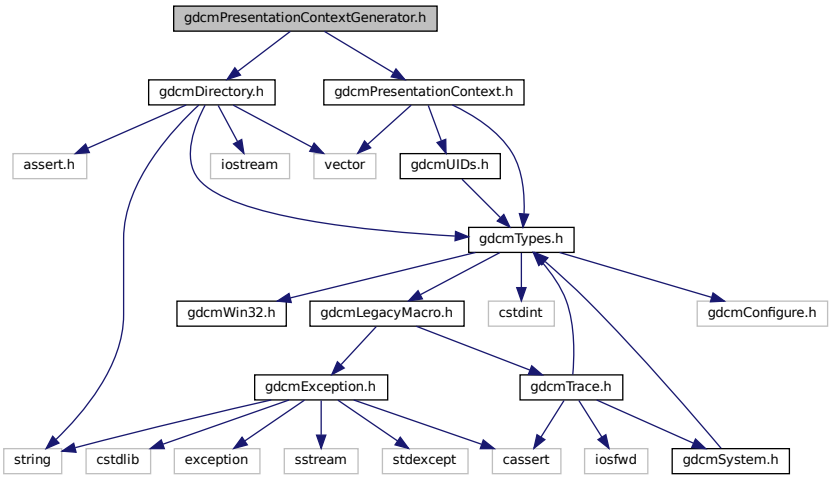
Go to the documentation of this file.

```
1 /*=========================================================================
2
3 Program:  GDCM (Grassroots DICOM). A DICOM library
4
5 Copyright (c) 2006-2011 Mathieu Malaterre
6 All rights reserved.
7 See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.
8
9 This software is distributed WITHOUT ANY WARRANTY; without even
10 the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
11 PURPOSE.  See the above copyright notice for more information.
12
13 =========================================================================*/
14 #ifndef GDCMPRESENTATIONCONTEXTAC_H
15 #define GDCMPRESENTATIONCONTEXTAC_H
16
17 #include "gdcmTypes.h"
18 #include "gdcmTransferSyntaxSub.h"
19
20 namespace gdcm
21 {
22
23 namespace network
24 {
25
32 class PresentationContextAC
33 {
34 public:
35   PresentationContextAC();
36   std::istream &Read(std::istream &is);
37   const std::ostream &Write(std::ostream &os) const;
38
39   size_t Size() const;
40
41   void SetTransferSyntax( TransferSyntaxSub const &ts );
42   void SetPresentationContextID( uint8_t id );
43
44   void Print(std::ostream &os) const;
45
46   uint8_t GetPresentationContextID()const
47 {
48     return ID;
49     }
50   TransferSyntaxSub const & GetTransferSyntax()const { return SubItems; }
51
52   void SetReason( uint8_t r ) { Result = r; }
53   uint8_t GetReason()const { return Result; }
54
55 private:
56   static const uint8_t ItemType;
57   static const uint8_t Reserved2;
58   uint16_t ItemLength; // len of last transfer syntax
59   uint8_t /*PresentationContext*/ID;
60   static const uint8_t Reserved6;
61   uint8_t /*Reason*/Result;
62   static const uint8_t Reserved8;
63   TransferSyntaxSub SubItems;
64 };
65
66 } // end namespace network
67
68 } // end namespace gdcm
69
70 #endif //GDCMPRESENTATIONCONTEXTAC_H
```

11.545 gdcmPresentationContextGenerator.h File Reference

```
#include "gdcmDirectory.h"
#include "gdcmPresentationContext.h"
```

Include dependency graph for gdcmPresentationContextGenerator.h:



Classes

- class gdcm::PresentationContextGenerator

  PresentationContextGenerator.

Namespaces

- namespace gdcm

11.546 gdcmPresentationContextGenerator.h

Go to the documentation of this file.

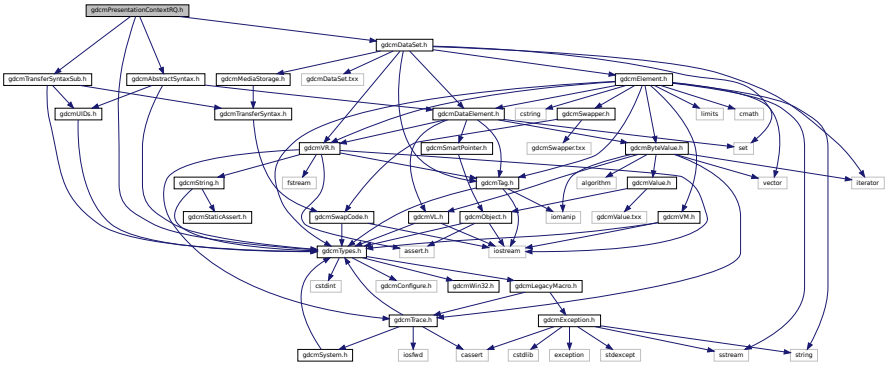
```
1 /*=========================================================================
2
3 Program:  GDCM (Grassroots DICOM). A DICOM library
4
5 Copyright (c) 2006-2011 Mathieu Malaterre
6 All rights reserved.
7 See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.
8
9 This software is distributed WITHOUT ANY WARRANTY; without even
10 the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
11 PURPOSE.  See the above copyright notice for more information.
12
13 =========================================================================*/
14 #ifndef GDCMPRESENTATIONCONTEXTGENERATOR_H
15 #define GDCMPRESENTATIONCONTEXTGENERATOR_H
16
17 #include "gdcmDirectory.h"
18 #include "gdcmPresentationContext.h"
19
20 namespace gdcm
21 {
22 class TransferSyntax;
```

```
23 class File;
24
53 class GDCM_EXPORT PresentationContextGenerator
54 {
55 public:
56   PresentationContextGenerator();
57
58   // Set MergeMode
59   // Default mode, each pair AbstractSyntax/TransferSyntax are only merged when
60   // exactly identical
61   void SetMergeModeToAbstractSyntax();
62
63   // Set MergeMode
64   // Merge is done on a per AbstractSyntax basis.  Any new TransferSyntax for a
65   // given AbstractSyntax is merge to the existing PresentationContext referring
66   // to that AbstractSyntax
67   void SetMergeModeToTransferSyntax();
68
70   bool GenerateFromUID(UIDs::TSName asname);
71
75   bool GenerateFromFilenames(const Directory::FilenamesType &files);
76
79   bool AddFromFile(const File &file);
80
81   typedef std::vector<PresentationContext> PresentationContextArrayType;
82   typedef PresentationContextArrayType::size_type SizeType;
83   PresentationContextArrayType const &GetPresentationContexts() { return PresContext; }
84
86   void SetDefaultTransferSyntax( const TransferSyntax &ts );
87 protected:
88   bool AddPresentationContext( const char *absyn, const char *ts );
89   const char *GetDefaultTransferSyntax() const;
90
91 private:
92   std::vector<PresentationContext> PresContext;
93   static std::string DefaultTransferSyntax;
94 };
95
96 } // end namespace gdcm
97
98 #endif //GDCMPRESENTATIONCONTEXTGENERATOR_H
```

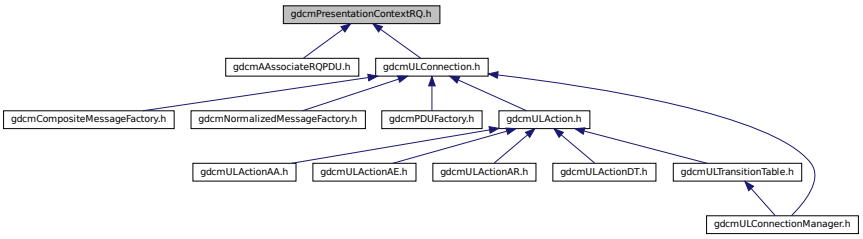
11.547 gdcmPresentationContextRQ.h File Reference

```
#include "gdcmTypes.h"
#include "gdcmAbstractSyntax.h"
#include "gdcmTransferSyntaxSub.h"
#include "gdcmDataSet.h"
```

Include dependency graph for gdcmPresentationContextRQ.h:



This graph shows which files directly or indirectly include this file:



Classes

- class gdcm::network::PresentationContextRQ

  PresentationContextRQ.

Namespaces

- namespace gdcm
- namespace gdcm::network

11.548 gdcmPresentationContextRQ.h

Go to the documentation of this file.

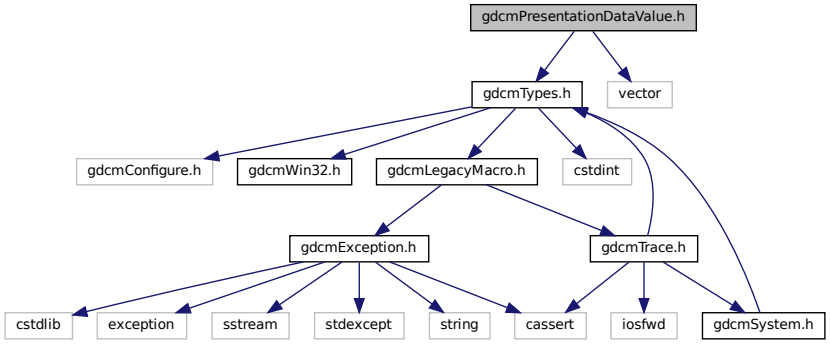
```
1 /*=========================================================================
2
3 Program:  GDCM (Grassroots DICOM). A DICOM library
4
5 Copyright (c) 2006-2011 Mathieu Malaterre
6 All rights reserved.
7 See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.
8
9 This software is distributed WITHOUT ANY WARRANTY; without even
10 the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
11 PURPOSE.  See the above copyright notice for more information.
12
13 =========================================================================*/
14 #ifndef GDCMPRESENTATIONCONTEXTRQ_H
15 #define GDCMPRESENTATIONCONTEXTRQ_H
16
17 #include "gdcmTypes.h"
18 #include "gdcmAbstractSyntax.h"
19 #include "gdcmTransferSyntaxSub.h"
20 #include "gdcmDataSet.h"
21
22 namespace gdcm
23 {
24 class PresentationContext;
25 namespace network
26 {
27
34 class GDCM_EXPORT PresentationContextRQ
35 {
36 public:
37   PresentationContextRQ();
38
42   PresentationContextRQ( UIDs::TSName asname, UIDs::TSName tsname =
```

```
43     UIDs::ImplicitVRLittleEndianDefaultTransferSyntaxforDICOM  );
44
45   std::istream &Read(std::istream &is);
46   const std::ostream &Write(std::ostream &os) const;
47   size_t Size() const;
48
49   void SetAbstractSyntax( AbstractSyntax const & absyn );
50   AbstractSyntax const &GetAbstractSyntax()const { return SubItems; }
51   AbstractSyntax &GetAbstractSyntax() { return SubItems; }
52
53   void AddTransferSyntax( TransferSyntaxSub const &ts );
54   typedef std::vector<TransferSyntaxSub>::size_type SizeType;
55   TransferSyntaxSub const & GetTransferSyntax(SizeType i)const { return TransferSyntaxes[i]; }
56   TransferSyntaxSub & GetTransferSyntax(SizeType i) { return TransferSyntaxes[i]; }
57   std::vector<TransferSyntaxSub> const & GetTransferSyntaxes()const {return TransferSyntaxes; }
58   SizeType GetNumberOfTransferSyntaxes()const { return TransferSyntaxes.size(); }
59
60   void SetPresentationContextID( uint8_t id );
61   uint8_t GetPresentationContextID() const;
62
63   void Print(std::ostream &os) const;
64
65   bool operator==(const PresentationContextRQ & pc)const
66 {
67     assert( TransferSyntaxes.size() == 1 ); // TODO
68     assert( pc.TransferSyntaxes.size() == 1 );
69     return SubItems == pc.SubItems && TransferSyntaxes == pc.TransferSyntaxes;
70     }
71
72   PresentationContextRQ(const PresentationContext & pc);
73
74 private:
75   static const uint8_t ItemType;
76   static const uint8_t Reserved2;
77   uint16_t ItemLength; // len of last transfer syntax
78   uint8_t /*PresentationContext*/ID;
79   static const uint8_t Reserved6;
80   static const uint8_t Reserved7;
81   static const uint8_t Reserved8;
82   /*
83 This variable field shall contain the following sub-items:  one Abstract
84 Syntax and one or more Transfer Syntax(es).  For a complete
85 description of the use and encoding of these sub-items see Sections
86 9.3.2.2.1 and 9.3.2.2.2.
87 */
88   AbstractSyntax SubItems;
89   std::vector<TransferSyntaxSub> TransferSyntaxes;
90 };
91
92 } // end namespace network
93
94 } // end namespace gdcm
95
96 #endif //GDCMPRESENTATIONCONTEXTRQ_H
```

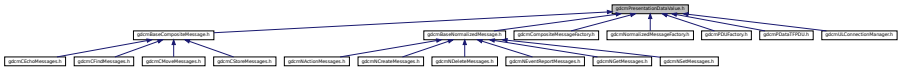
11.549 gdcmPresentationDataValue.h File Reference

```
#include "gdcmTypes.h"
#include <vector>
```

Include dependency graph for gdcmPresentationDataValue.h:



This graph shows which files directly or indirectly include this file:



Classes

- class gdcm::network::PresentationDataValue

  PresentationDataValue.

Namespaces

- namespace gdcm
- namespace gdcm::network

11.550 gdcmPresentationDataValue.h

Go to the documentation of this file.

```
1 /*=========================================================================
2
3 Program:  GDCM (Grassroots DICOM). A DICOM library
4
5 Copyright (c) 2006-2011 Mathieu Malaterre
6 All rights reserved.
7 See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.
8
9 This software is distributed WITHOUT ANY WARRANTY; without even
10 the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
11 PURPOSE.  See the above copyright notice for more information.
12
13 =========================================================================*/
14 #ifndef GDCMPRESENTATIONDATAVALUE_H
```

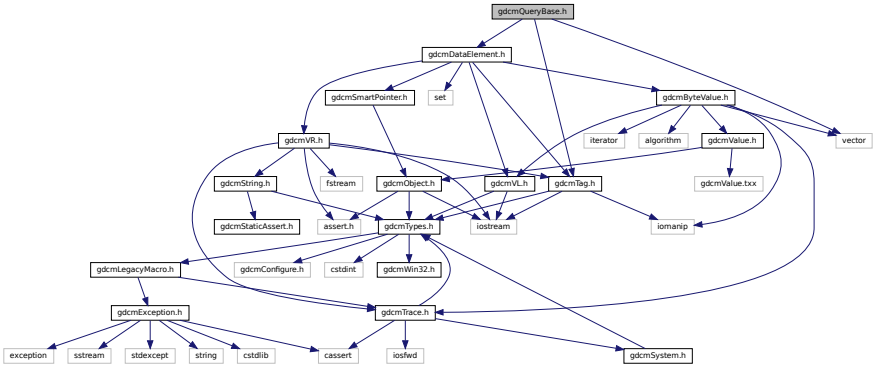
```
15 #define GDCMPRESENTATIONDATAVALUE_H
16
17 #include "gdcmTypes.h"
18
19 #include <vector>
20
21 namespace gdcm
22 {
23 class DataSet;
24 namespace network
25 {
26
32 class GDCM_EXPORT PresentationDataValue
33 {
34 public:
35   PresentationDataValue();
36   std::istream &Read(std::istream &is);
37   std::istream &ReadInto(std::istream &is, std::ostream &os);
38
39   const std::ostream &Write(std::ostream &os) const;
40
42   size_t Size() const;
43
46   void SetDataSet(const DataSet & ds);
47   void SetBlob(const std::string & partialblob);
48   const std::string &GetBlob() const;
49
50   uint8_t GetPresentationContextID()const { return PresentationContextID; }
51   void SetPresentationContextID(uint8_t id) {
52     assert( id );
53     PresentationContextID = id;
54   }
55   uint8_t GetMessageHeader()const {
56     assert( MessageHeader <= 0x3 );
57     return MessageHeader;
58   }
59   // E.2 MESSAGE CONTROL HEADER ENCODING
60   // Only the first two bits are considered
61   void SetMessageHeader(uint8_t messageheader) {
62     MessageHeader = messageheader;
63     assert( MessageHeader <= 0x3 );
64   }
65   //flip the least significant bit of the message header to 1
66   //if this is a command, else set it to 0.
67   void SetCommand(bool inCommand);
68   void SetLastFragment(bool inLast);//set to true if this is the last PDV of a set
69
70   bool GetIsCommand() const;
71   bool GetIsLastFragment() const;
72
73   void Print(std::ostream &os) const;
74
75   //NOTE that the PDVs have to be given in the order in which they were received!
76   //also note that a dataset may be across multiple PDVs
78   static DataSet ConcatenatePDVBlobs(const std::vector<PresentationDataValue>& inPDVs);
79
80   static DataSet ConcatenatePDVBlobsAsExplicit(const std::vector<PresentationDataValue>& inPDVs);
81
82 private:
83   uint32_t ItemLength;
84   uint8_t PresentationContextID;
85   uint8_t MessageHeader;
86   std::string Blob;
87 };
88 } // end namespace network
89
90 } // end namespace gdcm
91
92 #endif //GDCMPRESENTATIONDATAVALUE_H
```

11.551 gdcmQueryBase.h File Reference

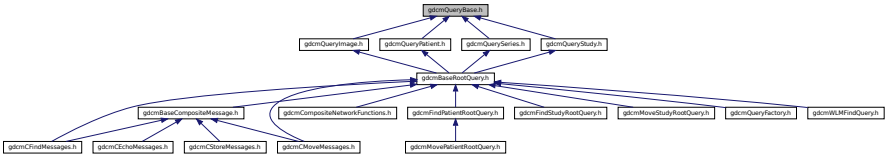
```
#include "gdcmTag.h"
#include "gdcmDataElement.h"
```

```
#include <vector>
```

Include dependency graph for gdcmQueryBase.h:



This graph shows which files directly or indirectly include this file:



Classes

- class gdcm::QueryBase

  QueryBase.

Namespaces

- namespace gdcm

Enumerations

- enum gdcm::ERootType {
  gdcm::ePatientRootType ,
  gdcm::eStudyRootType }

11.552 gdcmQueryBase.h

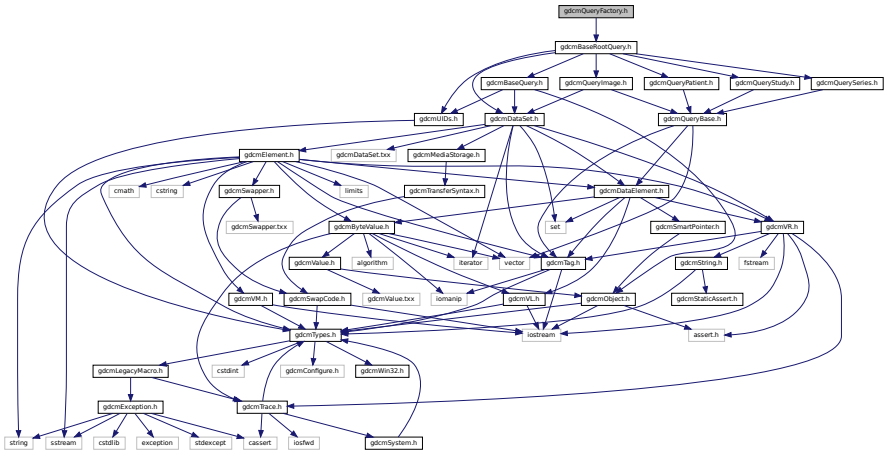
Go to the documentation of this file.

```
1 /*=========================================================================
2 *
3 *  Copyright NumFOCUS
4 *
5 *  Licensed under the Apache License, Version 2.0 (the "License");
6 *  you may not use this file except in compliance with the License.
7 *  You may obtain a copy of the License at
8 *
9 *         http://www.apache.org/licenses/LICENSE-2.0.txt
10 *
11 *  Unless required by applicable law or agreed to in writing, software
12 *  distributed under the License is distributed on an "AS IS" BASIS,
13 *  WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
14 *  See the License for the specific language governing permissions and
15 *  limitations under the License.
16 *
17 *=========================================================================*/
18 #ifndef GDCMQUERYBASE_H
19 #define GDCMQUERYBASE_H
20
21 #include "gdcmTag.h"
22 #include "gdcmDataElement.h"
23
24 #include <vector>
25
26 namespace gdcm
27 {
28   enum ERootType
29     {
30     ePatientRootType,
31     eStudyRootType
32     };
33
60 class GDCM_EXPORT QueryBase
61 {
62   public:
63     virtual ~QueryBase() = default;
64
65     virtual std::vector<Tag> GetRequiredTags(const ERootType& inRootType) const = 0;
66     virtual std::vector<Tag> GetUniqueTags(const ERootType& inRootType) const = 0;
67     virtual std::vector<Tag> GetOptionalTags(const ERootType& inRootType) const = 0;
68     // C.4.1.2.1 Baseline Behavior of SCU
69     // All C-FIND SCUs shall be capable of generating query requests which
70     // meet the requirements of the Hierarchical Search.
71     // The Identifier contained in a C-FIND request shall contain a single
72     // value in the Unique Key Attribute for each level above the
73     // Query/Retrieve level.  No Required or Optional Keys shall be
74     // specified which are associated with levels above the Query/Retrieve
75     // level.
77     virtual std::vector<Tag> GetHierachicalSearchTags(const ERootType& inRootType) const = 0;
78
81     std::vector<Tag> GetAllTags(const ERootType& inRootType) const;
82
85     std::vector<Tag> GetAllRequiredTags(const ERootType& inRootType) const;
86
87     virtual const char * GetName() const = 0;
88     virtual DataElement GetQueryLevel() const = 0;
89   };
90 }
91
92 #endif //GDCMQUERYBASE_H
```

11.553 gdcmQueryFactory.h File Reference

```
#include "gdcmBaseRootQuery.h"
```

Include dependency graph for gdcmQueryFactory.h:



Classes

- class gdcm::QueryFactory

  QueryFactory.h.

Namespaces

- namespace gdcm

Enumerations

- enum gdcm::ECharSet {
  gdcm::eLatin1 = 0 ,
  gdcm::eLatin2 ,
  gdcm::eLatin3 ,
  gdcm::eLatin4 ,
  gdcm::eCyrillic ,
  gdcm::eArabic ,
  gdcm::eGreek ,
  gdcm::eHebrew ,
  gdcm::eLatin5 ,
  gdcm::eJapanese ,
  gdcm::eThai ,
  gdcm::eJapaneseKanjiMultibyte ,
  gdcm::eJapaneseSupplementaryKanjiMultibyte ,
  gdcm::eKoreanHangulHanjaMultibyte ,
  gdcm::eUTF8 ,
  gdcm::eGB18030 }

11.554 gdcmQueryFactory.h

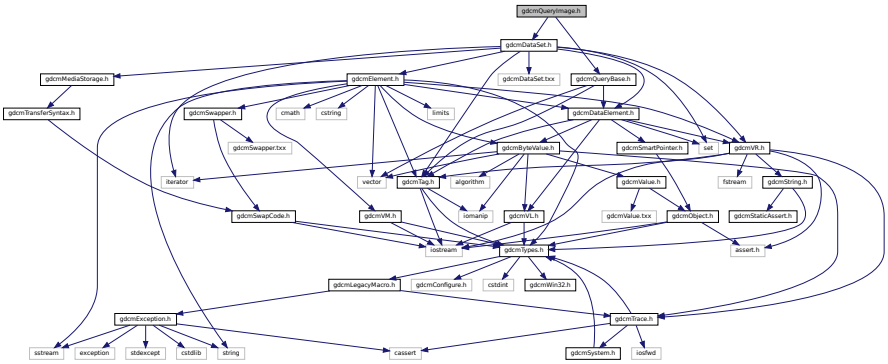
Go to the documentation of this file.

```
1 /*=========================================================================
2 *
3 *  Copyright NumFOCUS
4 *
5 *  Licensed under the Apache License, Version 2.0 (the "License");
6 *  you may not use this file except in compliance with the License.
7 *  You may obtain a copy of the License at
8 *
9 *         http://www.apache.org/licenses/LICENSE-2.0.txt
10 *
11 *  Unless required by applicable law or agreed to in writing, software
12 *  distributed under the License is distributed on an "AS IS" BASIS,
13 *  WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
14 *  See the License for the specific language governing permissions and
15 *  limitations under the License.
16 *
17 *=========================================================================*/
18 #ifndef GDCMQUERYFACTORY_H
19 #define GDCMQUERYFACTORY_H
20
21 #include "gdcmBaseRootQuery.h"
22
23 namespace gdcm{
27   enum ECharSet {
28     eLatin1 = 0,
29     eLatin2,
30     eLatin3,
31     eLatin4,
32     eCyrillic,
33     eArabic,
34     eGreek,
35     eHebrew,
36     eLatin5, // Latin Alphabet No.  5 (Turkish) Extended
37     eJapanese, // JIS X 0201 (Shift JIS) Extended
38     eThai, // TIS 620-2533 (Thai) Extended
39     eJapaneseKanjiMultibyte, // JIS X 0208 (Kanji) Extended
40     eJapaneseSupplementaryKanjiMultibyte, // JIS X 0212 (Kanji) Extended
41     eKoreanHangulHanjaMultibyte, // KS X 1001 (Hangul and Hanja) Extended
42     eUTF8,
43     eGB18030 // Chinese (Simplified) Extended
44   };
45
58 class GDCM_EXPORT QueryFactory
59 {
60 public:
61
62   static BaseQuery* ProduceQuery( const std::string & sopInstanceUID, ENQueryType inQueryType );
68   static BaseRootQuery* ProduceQuery(ERootType inRootType, EQueryType inQueryType,
69     EQueryLevel inQueryLevel);
70
78   static DataElement ProduceCharacterSetDataElement(
79     const std::vector<ECharSet>& inCharSetType);
80
83   static ECharSet GetCharacterFromCurrentLocale();
84
86   static void ListCharSets(std::ostream& os);
87 };
88
89 } // end namespace gdcm
90
91 #endif // GDCMQUERYFACTORY_H
```

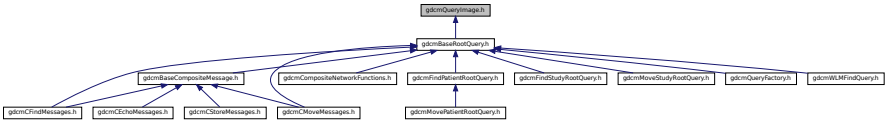
11.555 gdcmQueryImage.h File Reference

```
#include "gdcmQueryBase.h"
#include "gdcmDataSet.h"
```

Include dependency graph for gdcmQueryImage.h:



This graph shows which files directly or indirectly include this file:



Classes

- class gdcm::QueryImage

  QueryImage.

Namespaces

- namespace gdcm

11.556 gdcmQueryImage.h

Go to the documentation of this file.

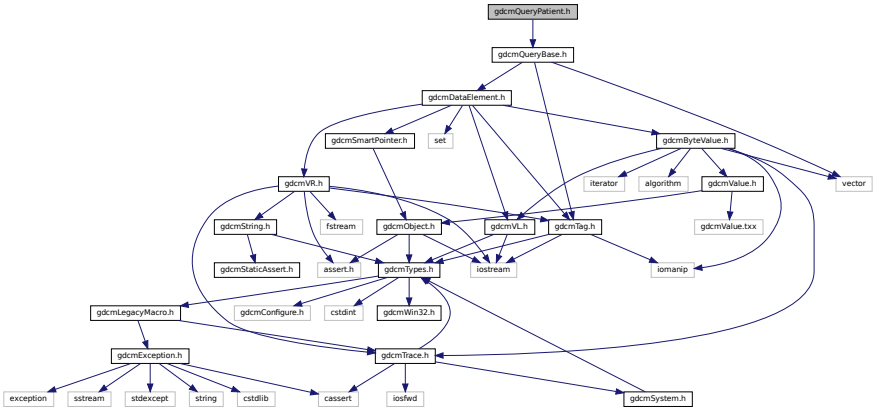
```
1 /*=========================================================================
2 *
3 *  Copyright NumFOCUS
4 *
5 *  Licensed under the Apache License, Version 2.0 (the "License");
6 *  you may not use this file except in compliance with the License.
7 *  You may obtain a copy of the License at
8 *
9 *         http://www.apache.org/licenses/LICENSE-2.0.txt
10 *
11 *  Unless required by applicable law or agreed to in writing, software
12 *  distributed under the License is distributed on an "AS IS" BASIS,
13 *  WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
14 *  See the License for the specific language governing permissions and
15 *  limitations under the License.
```

```
16 *
17 *=========================================================================*/
18 #ifndef GDCMQUERYIMAGE_H
19 #define GDCMQUERYIMAGE_H
20
21 #include "gdcmQueryBase.h"
22 #include "gdcmDataSet.h"
23
24 namespace gdcm
25 {
30 class GDCM_EXPORT QueryImage :  public QueryBase
31 {
32 public:
33   std::vector<Tag> GetRequiredTags(const ERootType& inRootType) const override;
34   std::vector<Tag> GetUniqueTags(const ERootType& inRootType) const override;
35   std::vector<Tag> GetOptionalTags(const ERootType& inRootType) const override;
36   std::vector<Tag> GetHierachicalSearchTags(const ERootType& inRootType) const override;
37
38   const char * GetName() const override;
39
40   DataElement GetQueryLevel() const override;
41 };
42
43 } // end namespace gdcm
44
45 #endif // GDCMQUERYIMAGE_H
```

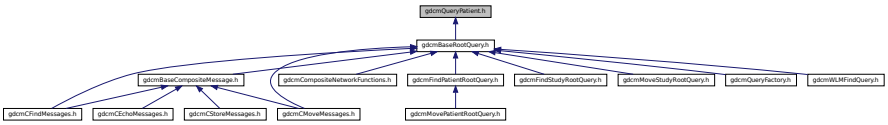
11.557 gdcmQueryPatient.h File Reference

`#include "gdcmQueryBase.h"`

Include dependency graph for gdcmQueryPatient.h:



This graph shows which files directly or indirectly include this file:



Classes

- class gdcm::QueryPatient

  QueryPatient.

Namespaces

- namespace gdcm

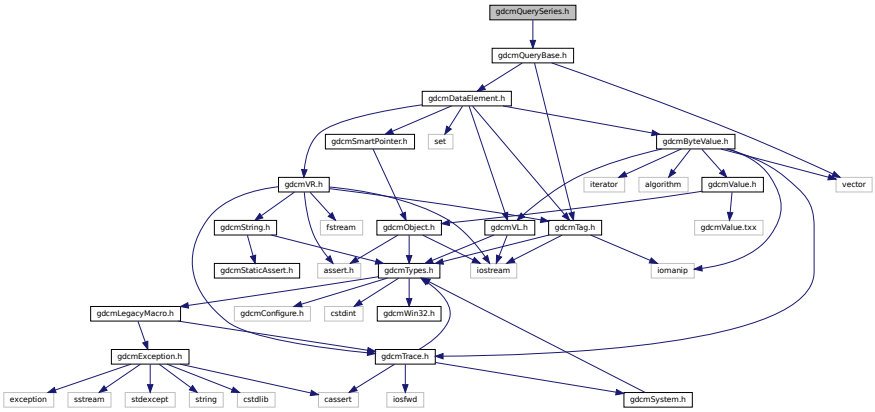
11.558 gdcmQueryPatient.h

Go to the documentation of this file.

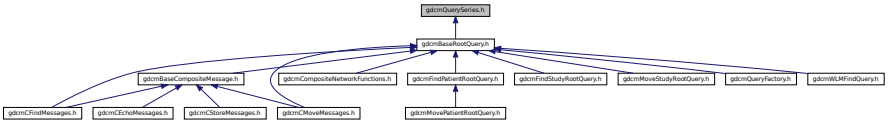
```
1 /*=========================================================================
2 *
3 *  Copyright NumFOCUS
4 *
5 *  Licensed under the Apache License, Version 2.0 (the "License");
6 *  you may not use this file except in compliance with the License.
7 *  You may obtain a copy of the License at
8 *
9 *         http://www.apache.org/licenses/LICENSE-2.0.txt
10 *
11 *  Unless required by applicable law or agreed to in writing, software
12 *  distributed under the License is distributed on an "AS IS" BASIS,
13 *  WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
14 *  See the License for the specific language governing permissions and
15 *  limitations under the License.
16 *
17 *=========================================================================*/
18 #ifndef GDCMQUERYPATIENT_H
19 #define GDCMQUERYPATIENT_H
20
21 #include "gdcmQueryBase.h"
22
23 namespace gdcm
24 {
29 class GDCM_EXPORT QueryPatient :  public QueryBase
30 {
31 public:
32   std::vector<Tag> GetRequiredTags(const ERootType& inRootType) const override;
33   std::vector<Tag> GetUniqueTags(const ERootType& inRootType) const override;
34   std::vector<Tag> GetOptionalTags(const ERootType& inRootType) const override;
35   std::vector<Tag> GetHierachicalSearchTags(const ERootType& inRootType) const override;
36
37   const char * GetName() const override;
38   DataElement GetQueryLevel() const override;
39 };
40
41 } // end namespace gdcm
42
43 #endif //GDCMQUERYPATIENT_H
```

11.559 gdcmQuerySeries.h File Reference

`#include "gdcmQueryBase.h"`
Include dependency graph for gdcmQuerySeries.h:



This graph shows which files directly or indirectly include this file:



Classes

- class gdcm::QuerySeries
  QuerySeries.

Namespaces

- namespace gdcm

11.560 gdcmQuerySeries.h

Go to the documentation of this file.

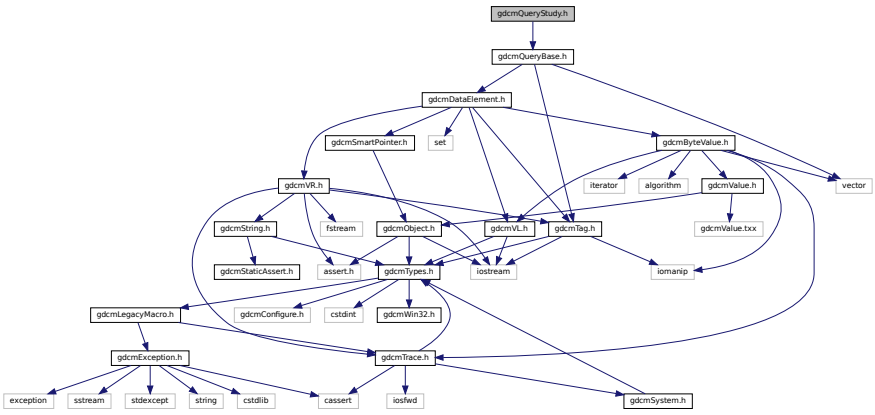
```
1 /*=========================================================================
2 *
3 *  Copyright NumFOCUS
4 *
5 *  Licensed under the Apache License, Version 2.0 (the "License");
6 *  you may not use this file except in compliance with the License.
```

```
 7 *  You may obtain a copy of the License at
 8 *
 9 *         http://www.apache.org/licenses/LICENSE-2.0.txt
10 *
11 *  Unless required by applicable law or agreed to in writing, software
12 *  distributed under the License is distributed on an "AS IS" BASIS,
13 *  WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
14 *  See the License for the specific language governing permissions and
15 *  limitations under the License.
16 *
17 *=========================================================================*/
18 #ifndef GDCMQUERYSERIES_H
19 #define GDCMQUERYSERIES_H
20
21 #include "gdcmQueryBase.h"
22
23 namespace gdcm
24 {
29 class GDCM_EXPORT QuerySeries :  public QueryBase
30 {
31 public:
32   std::vector<Tag> GetRequiredTags(const ERootType& inRootType) const override;
33   std::vector<Tag> GetUniqueTags(const ERootType& inRootType) const override;
34   std::vector<Tag> GetOptionalTags(const ERootType& inRootType) const override;
35   std::vector<Tag> GetHierachicalSearchTags(const ERootType& inRootType) const override;
36
37   const char * GetName() const override;
38   DataElement GetQueryLevel() const override;
39 };
40
41 } // end namespace gdcm
42
43 #endif //GDCMQUERYSERIES_H
```

11.561 gdcmQueryStudy.h File Reference

`#include "gdcmQueryBase.h"`

Include dependency graph for gdcmQueryStudy.h:



This graph shows which files directly or indirectly include this file:

Classes

- class gdcm::QueryStudy

  QueryStudy.h.

Namespaces

- namespace gdcm

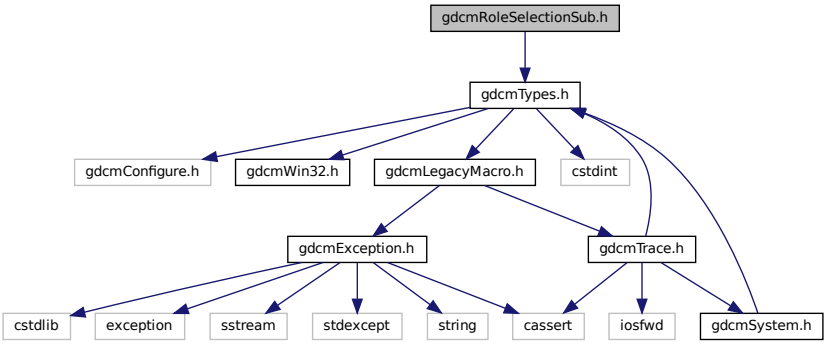
11.562 gdcmQueryStudy.h

Go to the documentation of this file.

```
1 /*=========================================================================
2 *
3 *  Copyright NumFOCUS
4 *
5 *  Licensed under the Apache License, Version 2.0 (the "License");
6 *  you may not use this file except in compliance with the License.
7 *  You may obtain a copy of the License at
8 *
9 *         http://www.apache.org/licenses/LICENSE-2.0.txt
10 *
11 *  Unless required by applicable law or agreed to in writing, software
12 *  distributed under the License is distributed on an "AS IS" BASIS,
13 *  WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
14 *  See the License for the specific language governing permissions and
15 *  limitations under the License.
16 *
17 *=========================================================================*/
18 #ifndef GDCMQUERYSTUDY_H
19 #define GDCMQUERYSTUDY_H
20
21 #include "gdcmQueryBase.h"
22
23 namespace gdcm
24 {
29 class GDCM_EXPORT QueryStudy :  public QueryBase
30 {
31 public:
32   std::vector<Tag> GetRequiredTags(const ERootType& inRootType) const override;
33   std::vector<Tag> GetUniqueTags(const ERootType& inRootType) const override;
34   std::vector<Tag> GetOptionalTags(const ERootType& inRootType) const override;
35   std::vector<Tag> GetHierachicalSearchTags(const ERootType& inRootType) const override;
36
37   const char *GetName() const override;
38   DataElement GetQueryLevel() const override;
39 };
40
41 } // end namespace gdcm
42
43 #endif //GDCMQUERYSTUDY_H
```

11.563 gdcmRoleSelectionSub.h File Reference

`#include "gdcmTypes.h"`
Include dependency graph for gdcmRoleSelectionSub.h:



Classes

- class gdcm::network::RoleSelectionSub
  RoleSelectionSub.

Namespaces

- namespace gdcm
- namespace gdcm::network

11.564 gdcmRoleSelectionSub.h

Go to the documentation of this file.

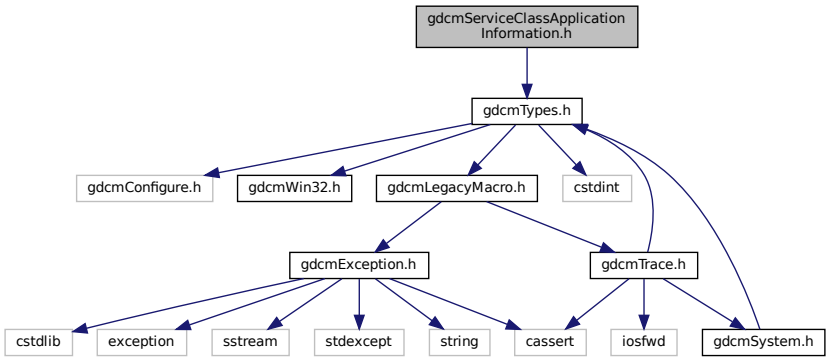
```
1 /*=========================================================================
2
3 Program:  GDCM (Grassroots DICOM). A DICOM library
4
5 Copyright (c) 2006-2011 Mathieu Malaterre
6 All rights reserved.
7 See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.
8
9 This software is distributed WITHOUT ANY WARRANTY; without even
10 the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
11 PURPOSE.  See the above copyright notice for more information.
12
13 =========================================================================*/
14 #ifndef GDCMROLESELECTIONSUB_H
15 #define GDCMROLESELECTIONSUB_H
16
17 #include "gdcmTypes.h"
18
19 namespace gdcm
20 {
21
```

```
22 namespace network
23 {
24
31 class RoleSelectionSub
32 {
33 public:
34   RoleSelectionSub();
35   std::istream &Read(std::istream &is);
36   const std::ostream &Write(std::ostream &os) const;
37
38   size_t Size() const;
39   void Print(std::ostream &os) const;
40
41   void SetTuple(const char *uid, uint8_t scurole, uint8_t scprole);
42
43 private:
44   static const uint8_t ItemType;
45   static const uint8_t Reserved2;
46   uint16_t ItemLength;
47   uint16_t UIDLength;
48   std::string /*SOP-class-uid*/ Name; // UID
49   uint8_t SCURole;
50   uint8_t SCPRole;
51 };
52
53 } // end namespace network
54
55 } // end namespace gdcm
56
57 #endif // GDCMROLESELECTIONSUB_H
```

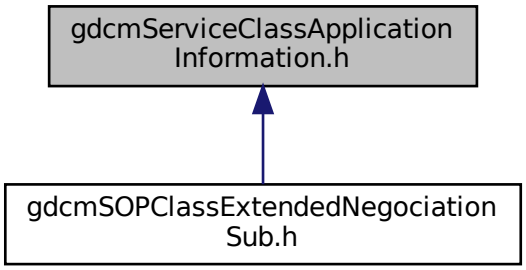
11.565 gdcmServiceClassApplicationInformation.h File Reference

`#include "gdcmTypes.h"`

Include dependency graph for gdcmServiceClassApplicationInformation.h:



This graph shows which files directly or indirectly include this file:



Classes

- class gdcm::network::ServiceClassApplicationInformation

Namespaces

- namespace gdcm
- namespace gdcm::network

11.566 gdcmServiceClassApplicationInformation.h

Go to the documentation of this file.

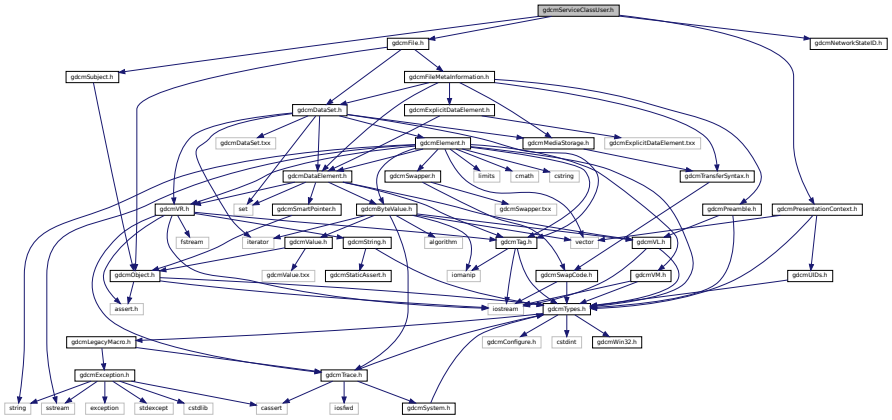
```
1 /*=========================================================================
2
3 Program:  GDCM (Grassroots DICOM). A DICOM library
4
5 Copyright (c) 2006-2011 Mathieu Malaterre
6 All rights reserved.
7 See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.
8
9 This software is distributed WITHOUT ANY WARRANTY; without even
10 the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
11 PURPOSE.  See the above copyright notice for more information.
12
13 =========================================================================*/
14 #ifndef GDCMSERVICECLASSAPPLICATIONINFORMATION_H
15 #define GDCMSERVICECLASSAPPLICATIONINFORMATION_H
16
17 #include "gdcmTypes.h"
18
19 namespace gdcm
20 {
21
22 namespace network
23 {
24
30 class ServiceClassApplicationInformation
31 {
32 public:
```

```
33   ServiceClassApplicationInformation();
34   std::istream &Read(std::istream &is);
35   const std::ostream &Write(std::ostream &os) const;
36
37   size_t Size() const;
38   void SetTuple(uint8_t levelofsupport, uint8_t levelofdigitalsig,
39     uint8_t elementcoercion);
40
41   void Print(std::ostream &os) const;
42 private:
43   uint8_t InternalArray[6];
44 };
45
46 } // end namespace network
47
48 } // end namespace gdcm
49
50 #endif //GDCMSERVICECLASSAPPLICATIONINFORMATION_H
```

11.567 gdcmServiceClassUser.h File Reference

```
#include "gdcmSubject.h"
#include "gdcmPresentationContext.h"
#include "gdcmFile.h"
#include "gdcmNetworkStateID.h"
```

Include dependency graph for gdcmServiceClassUser.h:



Classes

- class gdcm::ServiceClassUser

  ServiceClassUser.

Namespaces

- namespace gdcm
- namespace gdcm::network

11.568 gdcmServiceClassUser.h

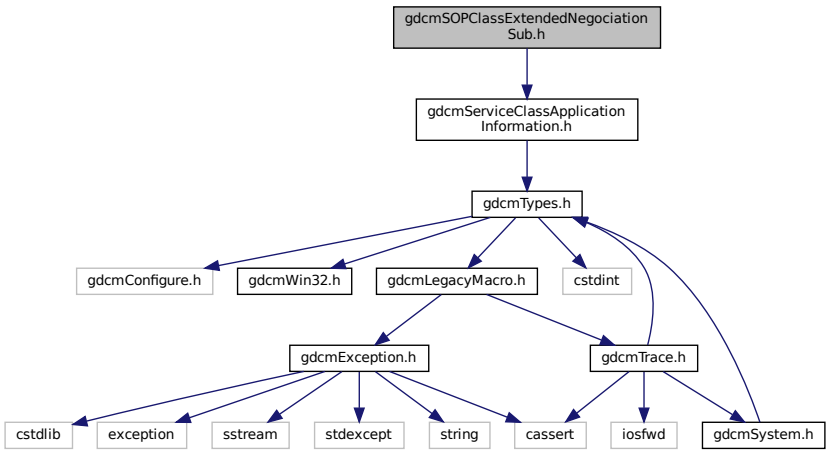
Go to the documentation of this file.

```
1 /*=========================================================================
2
3 Program:  GDCM (Grassroots DICOM). A DICOM library
4
5 Copyright (c) 2006-2011 Mathieu Malaterre
6 All rights reserved.
7 See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.
8
9 This software is distributed WITHOUT ANY WARRANTY; without even
10 the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
11 PURPOSE.  See the above copyright notice for more information.
12
13 =========================================================================*/
14 #ifndef GDCMSERVICECLASSUSER_H
15 #define GDCMSERVICECLASSUSER_H
16
17 #include "gdcmSubject.h"
18
19 #include "gdcmPresentationContext.h"
20 #include "gdcmFile.h"
21
22 #include "gdcmNetworkStateID.h" // EStateID
23
24 namespace gdcm
25 {
26 class ServiceClassUserInternals;
27 class BaseRootQuery;
28 namespace network{
29 class ULEvent;
30 class ULConnection;
31 class ULConnectionCallback;
32 }
36 class GDCM_EXPORT ServiceClassUser :  public Subject
37 {
38 public:
42   ServiceClassUser();
43   ~ServiceClassUser() override;
44   ServiceClassUser(const ServiceClassUser&) = delete;
45   void operator=(const ServiceClassUser &) = delete;
46
48   void SetHostname( const char *hostname );
49
51   void SetPort( uint16_t port );
52
54   void SetPortSCP( uint16_t portscp );
55
57   void SetAETitle(const char *aetitle);
58   const char *GetAETitle() const;
59
61   void SetCalledAETitle(const char *aetitle);
62   const char *GetCalledAETitle() const;
63
65   void SetTimeout(double t);
66   double GetTimeout() const;
67
71   bool InitializeConnection();
72
74   void SetPresentationContexts(std::vector<PresentationContext> const & pcs);
75
77   bool IsPresentationContextAccepted(const PresentationContext& pc) const;
78
80   bool StartAssociation();
81
83   bool StopAssociation();
84
86   bool SendEcho();
87
89   bool SendStore(const char *filename);
92   bool SendStore(File const &file);
94   bool SendStore(DataSet const &ds);
95
97   bool SendFind(const BaseRootQuery* query, std::vector<DataSet> &retDatasets);
98
100   bool SendMove(const BaseRootQuery* query, const char *outputdir);
102   bool SendMove(const BaseRootQuery* query, std::vector<DataSet> &retDatasets);
104   bool SendMove(const BaseRootQuery* query, std::vector<File> &retFile);
```

```
105
107   static SmartPointer<ServiceClassUser> New() { return new ServiceClassUser; }
108
109 private:
110   network::EStateID RunEventLoop(network::ULEvent& inEvent,
111     network::ULConnection* inWhichConnection,
112     network::ULConnectionCallback* inCallback, const bool& startWaiting);
113   network::EStateID RunMoveEventLoop(network::ULEvent& inEvent,
114     network::ULConnectionCallback* inCallback);
115
116 private:
117   ServiceClassUserInternals *Internals;
118 };
119
120 } // end namespace gdcm
121
122 #endif // GDCMSERVICECLASSUSER_H
```

11.569 gdcmSOPClassExtendedNegociationSub.h File Reference

`#include "gdcmServiceClassApplicationInformation.h"`
Include dependency graph for gdcmSOPClassExtendedNegociationSub.h:



Classes

- class gdcm::network::SOPClassExtendedNegociationSub
  SOPClassExtendedNegociationSub.

Namespaces

- namespace gdcm
- namespace gdcm::network

11.570 gdcmSOPClassExtendedNegociationSub.h

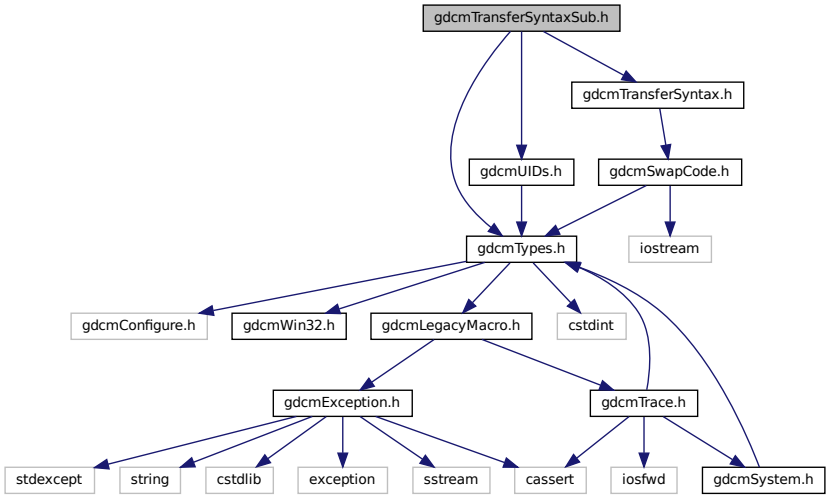
Go to the documentation of this file.

```
1 /*=========================================================================
2
3 Program:  GDCM (Grassroots DICOM). A DICOM library
4
5 Copyright (c) 2006-2011 Mathieu Malaterre
6 All rights reserved.
7 See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.
8
9 This software is distributed WITHOUT ANY WARRANTY; without even
10 the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
11 PURPOSE.  See the above copyright notice for more information.
12
13 =========================================================================*/
14 #ifndef GDCMSOPCLASSEXTENDEDNEGOCIATIONSUB_H
15 #define GDCMSOPCLASSEXTENDEDNEGOCIATIONSUB_H
16
17 #include "gdcmServiceClassApplicationInformation.h"
18
19 namespace gdcm
20 {
21 namespace network
22 {
23
31 class SOPClassExtendedNegociationSub
32 {
33 public:
34   SOPClassExtendedNegociationSub();
35   std::istream &Read(std::istream &is);
36   const std::ostream &Write(std::ostream &os) const;
37
38   size_t Size() const;
39   void Print(std::ostream &os) const;
40
41   void SetTuple(const char *uid, uint8_t levelofsupport = 3,
42     uint8_t levelofdigitalsig = 0,
43     uint8_t elementcoercion = 2);
44
45 private:
46   static const uint8_t ItemType;
47   static const uint8_t Reserved2;
48   uint16_t ItemLength;
49   uint16_t UIDLength;
50   std::string /*SOP-class-uid*/ Name; // UID
51   ServiceClassApplicationInformation SCAI;
52 };
53
54 } // end namespace network
55
56 } // end namespace gdcm
57
58 #endif // GDCMSOPCLASSEXTENDEDNEGOCIATIONSUB_H
```

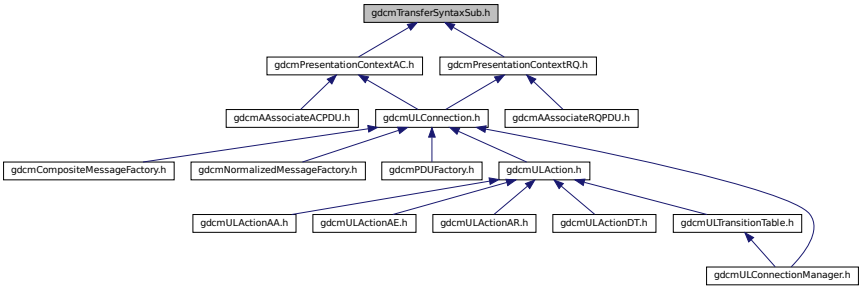
11.571 gdcmTransferSyntaxSub.h File Reference

```
#include "gdcmTypes.h"
#include "gdcmTransferSyntax.h"
#include "gdcmUIDs.h"
```

Include dependency graph for gdcmTransferSyntaxSub.h:



This graph shows which files directly or indirectly include this file:



Classes

- class gdcm::network::TransferSyntaxSub

  TransferSyntaxSub.

Namespaces

- namespace gdcm
- namespace gdcm::network

11.572 gdcmTransferSyntaxSub.h

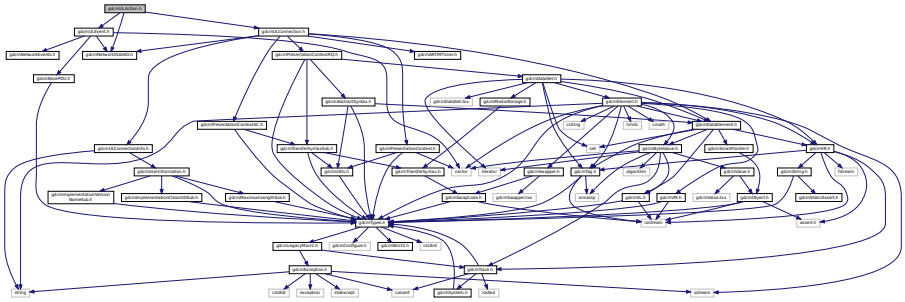
Go to the documentation of this file.

```
1 /*=========================================================================
2
3 Program:  GDCM (Grassroots DICOM). A DICOM library
4
5 Copyright (c) 2006-2011 Mathieu Malaterre
6 All rights reserved.
7 See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.
8
9 This software is distributed WITHOUT ANY WARRANTY; without even
10 the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
11 PURPOSE.  See the above copyright notice for more information.
12
13 =========================================================================*/
14 #ifndef GDCMTRANSFERSYNTAXSUB_H
15 #define GDCMTRANSFERSYNTAXSUB_H
16
17 #include "gdcmTypes.h"
18 #include "gdcmTransferSyntax.h"
19 #include "gdcmUIDs.h"
20
21 namespace gdcm
22 {
23
24 namespace network
25 {
26
37 class TransferSyntaxSub
38 {
39 public:
40   TransferSyntaxSub();
41   void SetName( const char *name );
42   const char *GetName()const { return Name.c_str(); }
43
44   // accept a UIDs::TSType also...
45   void SetNameFromUID( UIDs::TSName tsname );
46
47   std::istream &Read(std::istream &is);
48   const std::ostream &Write(std::ostream &os) const;
49   size_t Size() const;
50   void Print(std::ostream &os) const;
51
52   bool operator==(const TransferSyntaxSub & ts)const
53 {
54     return Name == ts.Name;
55     }
56
57 private:
58   void UpdateName( const char *name );
59   static const uint8_t ItemType;
60   static const uint8_t Reserved2;
61   uint16_t ItemLength; // len of
62   std::string /*TransferSyntaxSub*/ Name; // UID
63 };
64
65 } // end namespace network
66
67 } // end namespace gdcm
68
69 #endif //GDCMTRANSFERSYNTAXSUB_H
```

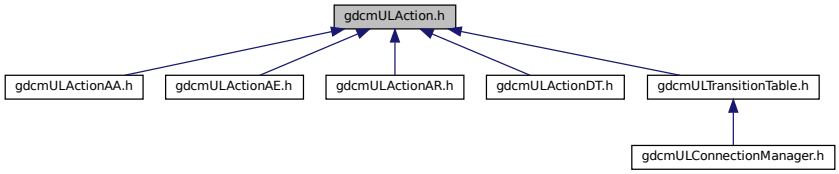
11.573 gdcmULAction.h File Reference

```
#include "gdcmNetworkStateID.h"
#include "gdcmULEvent.h"
```

`#include "gdcmULConnection.h"`
Include dependency graph for gdcmULAction.h:



This graph shows which files directly or indirectly include this file:



Classes

- class gdcm::network::ULAction
  ULAction.

Namespaces

- namespace gdcm
- namespace gdcm::network

11.574 gdcmULAction.h

Go to the documentation of this file.

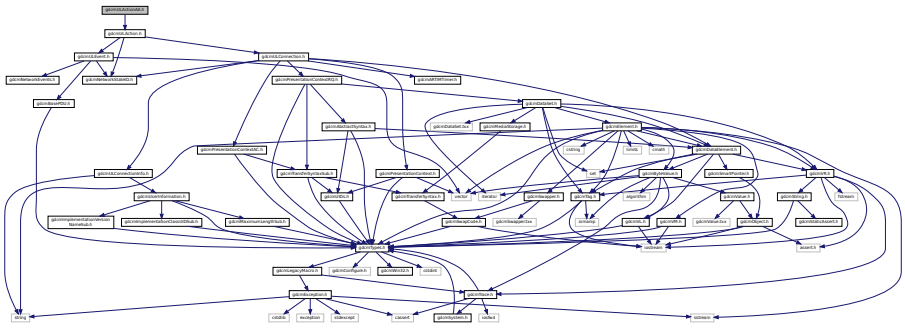
```
1 /*=========================================================================
2 *
3 *  Copyright NumFOCUS
4 *
5 *  Licensed under the Apache License, Version 2.0 (the "License");
6 *  you may not use this file except in compliance with the License.
7 *  You may obtain a copy of the License at
8 *
9 *         http://www.apache.org/licenses/LICENSE-2.0.txt
10 *
```

```
11 *  Unless required by applicable law or agreed to in writing, software
12 *  distributed under the License is distributed on an "AS IS" BASIS,
13 *  WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
14 *  See the License for the specific language governing permissions and
15 *  limitations under the License.
16 *
17 *=========================================================================*/
18 #ifndef GDCMULACTION_H
19 #define GDCMULACTION_H
20
21 #include "gdcmNetworkStateID.h"
22 #include "gdcmULEvent.h"
23 #include "gdcmULConnection.h"
24
25 namespace gdcm {
26 class Subject;
27   namespace network {
28
62 class ULAction {
63
64     protected:
65
66
67     public:
68       ULAction() = default;
69       //make sure destructors are virtual to avoid memory leaks
70       virtual ~ULAction() = default;
71       //cannot copy a ULAction
72       ULAction(const ULAction& inAction) = delete;
73       void operator=(const ULAction&) = delete;
74
75       virtual EStateID PerformAction(Subject *s, ULEvent& inEvent, ULConnection& inConnection,
76         bool& outWaitingForEvent, EEventID& outRaisedEvent) = 0;
77     };
78   }
79 }
80
81 #endif // GDCMULACTION_H
```

11.575 gdcmULActionAA.h File Reference

`#include "gdcmULAction.h"`

Include dependency graph for gdcmULActionAA.h:



Classes

- class gdcm::network::ULActionAA1
- class gdcm::network::ULActionAA2
- class gdcm::network::ULActionAA3

- class gdcm::network::ULActionAA4
- class gdcm::network::ULActionAA5
- class gdcm::network::ULActionAA6
- class gdcm::network::ULActionAA7
- class gdcm::network::ULActionAA8

Namespaces

- namespace gdcm
- namespace gdcm::network

11.576 gdcmULActionAA.h

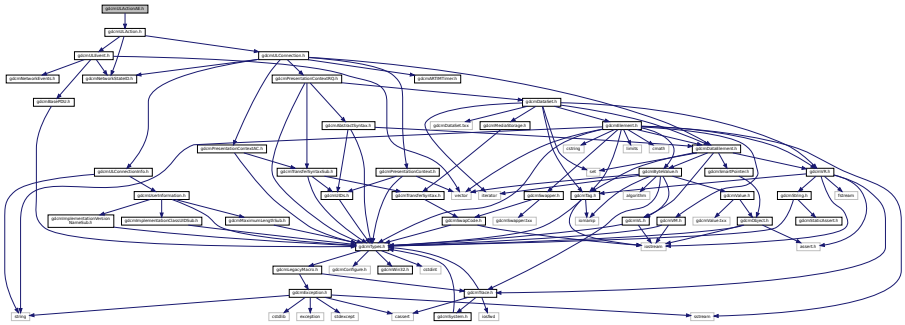
Go to the documentation of this file.

```
1 /*=========================================================================
2 *
3 *  Copyright NumFOCUS
4 *
5 *  Licensed under the Apache License, Version 2.0 (the "License");
6 *  you may not use this file except in compliance with the License.
7 *  You may obtain a copy of the License at
8 *
9 *         http://www.apache.org/licenses/LICENSE-2.0.txt
10 *
11 *  Unless required by applicable law or agreed to in writing, software
12 *  distributed under the License is distributed on an "AS IS" BASIS,
13 *  WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
14 *  See the License for the specific language governing permissions and
15 *  limitations under the License.
16 *
17 *=========================================================================*/
18 #ifndef GDCMULACTIONAA_H
19 #define GDCMULACTIONAA_H
20
21 #include "gdcmULAction.h"
22
31 namespace gdcm {
32   namespace network {
33
34     //Send A-ABORT PDU (service-user source) and start (or restart if already started) ARTIM timer
35     //Next State:  eSta13AwaitingClose
36     class ULActionAA1 :  public ULAction {
37     public:
38       EStateID PerformAction(Subject *s, ULEvent& inEvent, ULConnection& inConnection,
39         bool& outWaitingForEvent, EEventID& outRaisedEvent) override;
40     };
41
42     //Stop ARTIM timer if running.    Close transport connection.
43     //Next State:  eSta1Idle
44     class ULActionAA2 :  public ULAction {
45     public:
46       EStateID PerformAction(Subject *s, ULEvent& inEvent, ULConnection& inConnection,
47         bool& outWaitingForEvent, EEventID& outRaisedEvent) override;
48     };
49
50     //If (service-user initiated abort)
51     //- issue A-ABORT indication and close transport connection
52     //otherwise (service-provider initiated abort):
53     //- issue A-P-ABORT indication and close transport connection
54     //Next State:  eSta1Idle
55     class ULActionAA3 :  public ULAction {
56     public:
57       EStateID PerformAction(Subject *s, ULEvent& inEvent, ULConnection& inConnection,
58         bool& outWaitingForEvent, EEventID& outRaisedEvent) override;
59     };
60
61     //Issue A-P-ABORT indication primitive
62     //Next State:  eSta1Idle
```

```
63      class ULActionAA4 :  public ULAction {
64      public:
65        EStateID PerformAction(Subject *s, ULEvent& inEvent, ULConnection& inConnection,
66          bool& outWaitingForEvent, EEventID& outRaisedEvent) override;
67      };
68
69      //Stop ARTIM timer
70      //Next State:  eSta1Idle
71      class ULActionAA5 :  public ULAction {
72      public:
73        EStateID PerformAction(Subject *s, ULEvent& inEvent, ULConnection& inConnection,
74          bool& outWaitingForEvent, EEventID& outRaisedEvent) override;
75      };
76
77      //Ignore PDU
78      //Next State:  eSta13AwaitingClose
79      class ULActionAA6 :  public ULAction {
80      public:
81        EStateID PerformAction(Subject *s, ULEvent& inEvent, ULConnection& inConnection,
82          bool& outWaitingForEvent, EEventID& outRaisedEvent) override;
83      };
84
85      //Send A-ABORT PDU
86      //Next State:  eSta13AwaitingClose
87      class ULActionAA7 :  public ULAction {
88      public:
89        EStateID PerformAction(Subject *s, ULEvent& inEvent, ULConnection& inConnection,
90          bool& outWaitingForEvent, EEventID& outRaisedEvent) override;
91      };
92
93      //Send A-ABORT PDU (service-provider source), issue an A-P-ABORT indication, and start ARTIM timer
94      //Next State:  eSta13AwaitingClose
95      class ULActionAA8 :  public ULAction {
96      public:
97        EStateID PerformAction(Subject *s, ULEvent& inEvent, ULConnection& inConnection,
98          bool& outWaitingForEvent, EEventID& outRaisedEvent) override;
99      };
100   }
101 }
102
103 #endif // GDCMULACTIONAA_H
```

11.577 gdcmULActionAE.h File Reference

`#include "gdcmULAction.h"`
Include dependency graph for gdcmULActionAE.h:



Classes

- class gdcm::network::ULActionAE1

- class gdcm::network::ULActionAE2
- class gdcm::network::ULActionAE3
- class gdcm::network::ULActionAE4
- class gdcm::network::ULActionAE5
- class gdcm::network::ULActionAE6
- class gdcm::network::ULActionAE7
- class gdcm::network::ULActionAE8

Namespaces

- namespace gdcm
- namespace gdcm::network

11.578 gdcmULActionAE.h

Go to the documentation of this file.

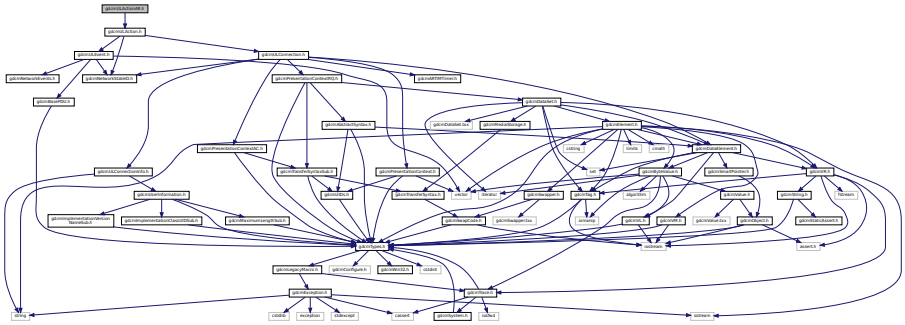
```
1 /*=========================================================================
2 *
3 *  Copyright NumFOCUS
4 *
5 *  Licensed under the Apache License, Version 2.0 (the "License");
6 *  you may not use this file except in compliance with the License.
7 *  You may obtain a copy of the License at
8 *
9 *         http://www.apache.org/licenses/LICENSE-2.0.txt
10 *
11 *  Unless required by applicable law or agreed to in writing, software
12 *  distributed under the License is distributed on an "AS IS" BASIS,
13 *  WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
14 *  See the License for the specific language governing permissions and
15 *  limitations under the License.
16 *
17 *=========================================================================*/
18 #ifndef GDCMULACTIONAE_H
19 #define GDCMULACTIONAE_H
20
21 #include "gdcmULAction.h"
22
23 namespace gdcm {
24   namespace network {
25
26     //Issue TRANSPORT CONNECT request primitive to local transport service.
27     class ULActionAE1 :  public ULAction {
28     public:
29       EStateID PerformAction(Subject *s, ULEvent& inEvent, ULConnection& inConnection,
30         bool& outWaitingForEvent, EEventID& outRaisedEvent) override;
31     };
32
33     //Send A-ASSOCIATE-RQ-PDU
34     //Next State:  eSta5WaitRemoteAssoc
35     class ULActionAE2 :  public ULAction {
36     public:
37       EStateID PerformAction(Subject *s, ULEvent& inEvent, ULConnection& inConnection,
38         bool& outWaitingForEvent, EEventID& outRaisedEvent) override;
39     };
40
41     //Issue A-ASSOCIATE confirmation (accept) primitive
42     //Next State:  eSta6TransferReady
43     class ULActionAE3 :  public ULAction {
44     public:
45       EStateID PerformAction(Subject *s, ULEvent& inEvent, ULConnection& inConnection,
46         bool& outWaitingForEvent, EEventID& outRaisedEvent) override;
47     };
48
49     //Issue A-ASSOCIATE confirmation (reject) primitive and close transport connection
50     //Next State:  eSta1Idle
51     class ULActionAE4 :  public ULAction {
52
```

```
61      public:
62        EStateID PerformAction(Subject *s, ULEvent& inEvent, ULConnection& inConnection,
63          bool& outWaitingForEvent, EEventID& outRaisedEvent) override;
64      };
65
66      //Issue Transport connection response primitive, start ARTIM timer
67      //Next State:  eSta2Open
68      class ULActionAE5 :  public ULAction {
69      public:
70        EStateID PerformAction(Subject *s, ULEvent& inEvent, ULConnection& inConnection,
71          bool& outWaitingForEvent, EEventID& outRaisedEvent) override;
72      };
73
74      //Stop ARTIM timer and if A-ASSOCIATE-RQ acceptable by service-provider:
75      //- issue A-ASSOCIATE indication primitive
76      //Next state:  eSta3WaitLocalAssoc
77      //otherwise:
78      //- issue A-ASSOCIATE-RJ-PDU and start ARTIM timer
79      //Next state:  eSta13AwaitingClose
80      class ULActionAE6 :  public ULAction {
81      public:
82        EStateID PerformAction(Subject *s, ULEvent& inEvent, ULConnection& inConnection,
83          bool& outWaitingForEvent, EEventID& outRaisedEvent) override;
84      };
85
86      //Send A-ASSOCIATE-AC PDU
87      //Next State:  eSta6TransferReady
88      class ULActionAE7 :  public ULAction {
89      public:
90        EStateID PerformAction(Subject *s, ULEvent& inEvent, ULConnection& inConnection,
91          bool& outWaitingForEvent, EEventID& outRaisedEvent) override;
92      };
93
94      //Send A-ASSOCIATE-RJ PDU and start ARTIM timer
95      //Next State:  eSta13AwaitingClose
96      class ULActionAE8 :  public ULAction {
97      public:
98        EStateID PerformAction(Subject *s, ULEvent& inEvent, ULConnection& inConnection,
99          bool& outWaitingForEvent, EEventID& outRaisedEvent) override;
100     };
101   }
102 }
103 #endif // GDCMULACTIONAE_H
```

11.579 gdcmULActionAR.h File Reference

`#include "gdcmULAction.h"`

Include dependency graph for gdcmULActionAR.h:



Classes

- class gdcm::network::ULActionAR1

- class gdcm::network::ULActionAR10
- class gdcm::network::ULActionAR2
- class gdcm::network::ULActionAR3
- class gdcm::network::ULActionAR4
- class gdcm::network::ULActionAR5
- class gdcm::network::ULActionAR6
- class gdcm::network::ULActionAR7
- class gdcm::network::ULActionAR8
- class gdcm::network::ULActionAR9

Namespaces

- namespace gdcm
- namespace gdcm::network

11.580 gdcmULActionAR.h

Go to the documentation of this file.

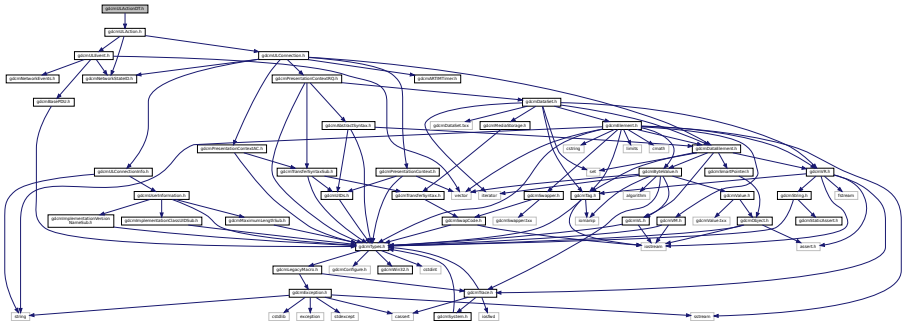
```
1 /*=========================================================================
2 *
3 *  Copyright NumFOCUS
4 *
5 *  Licensed under the Apache License, Version 2.0 (the "License");
6 *  you may not use this file except in compliance with the License.
7 *  You may obtain a copy of the License at
8 *
9 *         http://www.apache.org/licenses/LICENSE-2.0.txt
10 *
11 *  Unless required by applicable law or agreed to in writing, software
12 *  distributed under the License is distributed on an "AS IS" BASIS,
13 *  WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
14 *  See the License for the specific language governing permissions and
15 *  limitations under the License.
16 *
17 *=========================================================================*/
18 #ifndef GDCMULACTIONAR_H
19 #define GDCMULACTIONAR_H
20
21 #include "gdcmULAction.h"
22
31 namespace gdcm {
32   namespace network {
33
34     //Send A-RELEASE-RQ-PDU
35     //Next State:  eSta7WaitRelease
36     class ULActionAR1 :  public ULAction {
37     public:
38       EStateID PerformAction(Subject *s, ULEvent& inEvent, ULConnection& inConnection,
39         bool& outWaitingForEvent, EEventID& outRaisedEvent) override;
40     };
41
42     //Issue A-RELEASE indication primitive
43     //Next State:  eSta8WaitLocalRelease
44     class ULActionAR2 :  public ULAction {
45     public:
46       EStateID PerformAction(Subject *s, ULEvent& inEvent, ULConnection& inConnection,
47         bool& outWaitingForEvent, EEventID& outRaisedEvent) override;
48     };
49
50     //Issue A-RELEASE confirmation primitive, and close transport connection
51     //Next State:  eSta1Idle
52     class ULActionAR3 :  public ULAction {
53     public:
54       EStateID PerformAction(Subject *s, ULEvent& inEvent, ULConnection& inConnection,
55         bool& outWaitingForEvent, EEventID& outRaisedEvent) override;
```

```
56      };
57
58      //Issue A-RELEASE-RP PDU and start ARTIM timer
59      //Next State:  eSta13AwaitingClose
60      class ULActionAR4 :  public ULAction {
61      public:
62        EStateID PerformAction(Subject *s, ULEvent& inEvent, ULConnection& inConnection,
63          bool& outWaitingForEvent, EEventID& outRaisedEvent) override;
64      };
65
66      //Stop ARTIM timer
67      //Next State:  eSta1Idle
68      class ULActionAR5 :  public ULAction {
69      public:
70        EStateID PerformAction(Subject *s, ULEvent& inEvent, ULConnection& inConnection,
71          bool& outWaitingForEvent, EEventID& outRaisedEvent) override;
72      };
73
74      //Issue P-Data indication
75      //Next State:  eSta7WaitRelease
76      class ULActionAR6 :  public ULAction {
77      public:
78        EStateID PerformAction(Subject *s, ULEvent& inEvent, ULConnection& inConnection,
79          bool& outWaitingForEvent, EEventID& outRaisedEvent) override;
80      };
81
82      //Issue P-DATA-TF PDU
83      //Next State:  eSta8WaitLocalRelease
84      class ULActionAR7 :  public ULAction {
85      public:
86        EStateID PerformAction(Subject *s, ULEvent& inEvent, ULConnection& inConnection,
87          bool& outWaitingForEvent, EEventID& outRaisedEvent) override;
88      };
89
90      //Issue A-RELEASE indication (release collision):
91      //- If association-requestor, next state is eSta9ReleaseCollisionRqLocal
92      //- if not, next state is eSta10ReleaseCollisionAc
93      class ULActionAR8 :  public ULAction {
94      public:
95        EStateID PerformAction(Subject *s, ULEvent& inEvent, ULConnection& inConnection,
96          bool& outWaitingForEvent, EEventID& outRaisedEvent) override;
97      };
98
99      //Send A-RELEASE-RP PDU
100      //Next State:  eSta11ReleaseCollisionRq
101      class ULActionAR9 :  public ULAction {
102      public:
103        EStateID PerformAction(Subject *s, ULEvent& inEvent, ULConnection& inConnection,
104          bool& outWaitingForEvent, EEventID& outRaisedEvent) override;
105      };
106
107      //Issue A-RELEASE confirmation primitive
108      //Next State:  eSta12ReleaseCollisionAcLocal
109      class ULActionAR10 :  public ULAction {
110      public:
111        EStateID PerformAction(Subject *s, ULEvent& inEvent, ULConnection& inConnection,
112          bool& outWaitingForEvent, EEventID& outRaisedEvent) override;
113      };
114    }
115 }
116 #endif // GDCMULACTIONAR_H
```

11.581 gdcmULActionDT.h File Reference

```
#include "gdcmULAction.h"
```

Include dependency graph for gdcmULActionDT.h:



Classes

- class gdcm::network::ULActionDT1
- class gdcm::network::ULActionDT2

Namespaces

- namespace gdcm
- namespace gdcm::network

11.582 gdcmULActionDT.h

Go to the documentation of this file.

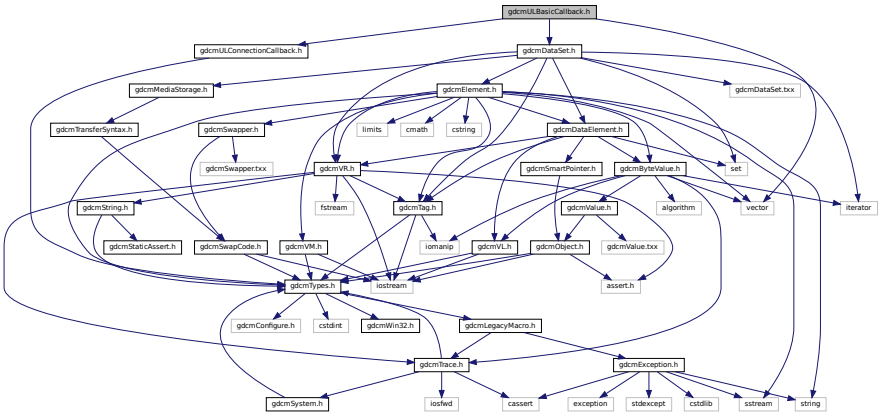
```
1 /*=========================================================================
2 *
3 *  Copyright NumFOCUS
4 *
5 *  Licensed under the Apache License, Version 2.0 (the "License");
6 *  you may not use this file except in compliance with the License.
7 *  You may obtain a copy of the License at
8 *
9 *         http://www.apache.org/licenses/LICENSE-2.0.txt
10 *
11 *  Unless required by applicable law or agreed to in writing, software
12 *  distributed under the License is distributed on an "AS IS" BASIS,
13 *  WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
14 *  See the License for the specific language governing permissions and
15 *  limitations under the License.
16 *
17 *=========================================================================*/
18 #ifndef GDCMULACTIONDT_H
19 #define GDCMULACTIONDT_H
20
21 #include "gdcmULAction.h"
22
31 namespace gdcm {
32   namespace network {
33
```

```
34     //Send P-DATA-TF PDU
35     //Next state:  eSta6TransferReady
36     class ULActionDT1 :  public ULAction {
37     public:
38       EStateID PerformAction(Subject *s, ULEvent& inEvent, ULConnection& inConnection,
39         bool& outWaitingForEvent, EEventID& outRaisedEvent) override;
40     };
41
42     //Send P-DATA indication primitive
43     //Next state:  eSta6TransferReady
44     class ULActionDT2 :  public ULAction {
45     public:
46       EStateID PerformAction(Subject *s, ULEvent& inEvent, ULConnection& inConnection,
47         bool& outWaitingForEvent, EEventID& outRaisedEvent) override;
48     };
49   }
50 }
51 #endif // GDCMULACTIONDT_H
```

11.583 gdcmULBasicCallback.h File Reference

```
#include "gdcmULConnectionCallback.h"
#include "gdcmDataSet.h"
#include <vector>
```

Include dependency graph for gdcmULBasicCallback.h:



Classes

- class gdcm::network::ULBasicCallback

  ULBasicCallback.

Namespaces

- namespace gdcm
- namespace gdcm::network

11.584 gdcmULBasicCallback.h

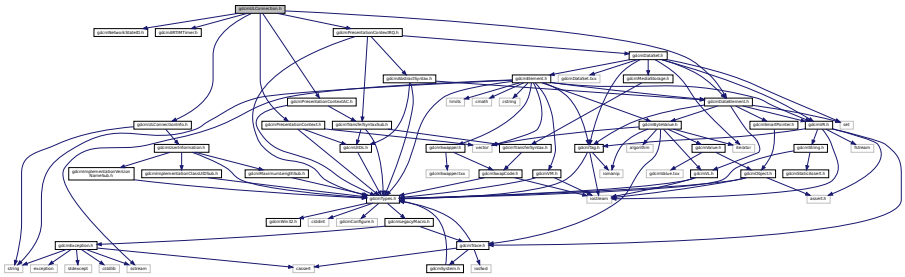
Go to the documentation of this file.

```
1 /*=========================================================================
2 *
3 *  Copyright NumFOCUS
4 *
5 *  Licensed under the Apache License, Version 2.0 (the "License");
6 *  you may not use this file except in compliance with the License.
7 *  You may obtain a copy of the License at
8 *
9 *         http://www.apache.org/licenses/LICENSE-2.0.txt
10 *
11 *  Unless required by applicable law or agreed to in writing, software
12 *  distributed under the License is distributed on an "AS IS" BASIS,
13 *  WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
14 *  See the License for the specific language governing permissions and
15 *  limitations under the License.
16 *
17 *=========================================================================*/
18 #ifndef GDCMULCONNECTIONBASICCALLBACK_H
19 #define GDCMULCONNECTIONBASICCALLBACK_H
20
21 #include "gdcmULConnectionCallback.h"
22 #include "gdcmDataSet.h"
23 #include <vector>
24
25 namespace gdcm
26 {
27   namespace network
28     {
38     class GDCM_EXPORT ULBasicCallback :  public ULConnectionCallback
39     {
40     std::vector<DataSet> mDataSets;
41     std::vector<DataSet> mResponses;
42   public:
43     ULBasicCallback() = default;
44     ~ULBasicCallback() override = default; //empty, for later inheritance
45
46     void HandleDataSet(const DataSet& inDataSet) override;
47     void HandleResponse(const DataSet& inDataSet) override;
48
49     std::vector<DataSet> const & GetDataSets() const;
50     std::vector<DataSet> const & GetResponses() const;
51     };
52     } // end namespace network
53 } // end namespace gdcm
54
55 #endif // GDCMULCONNECTIONBASICCALLBACK_H
```

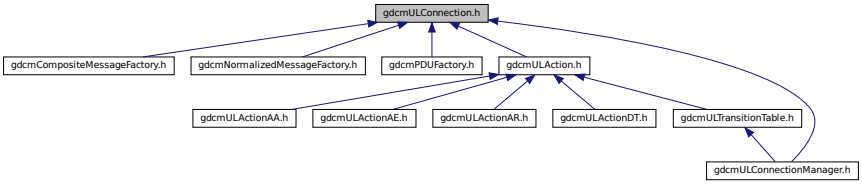
11.585 gdcmULConnection.h File Reference

```
#include "gdcmNetworkStateID.h"
#include "gdcmARTIMTimer.h"
#include "gdcmULConnectionInfo.h"
#include "gdcmPresentationContextRQ.h"
#include "gdcmDataElement.h"
#include "gdcmPresentationContextAC.h"
#include "gdcmPresentationContext.h"
```

Include dependency graph for gdcmULConnection.h:



This graph shows which files directly or indirectly include this file:



Classes

- class gdcm::network::ULConnection

  ULConnection.

Namespaces

- namespace gdcm
- namespace gdcm::network

11.586 gdcmULConnection.h

Go to the documentation of this file.

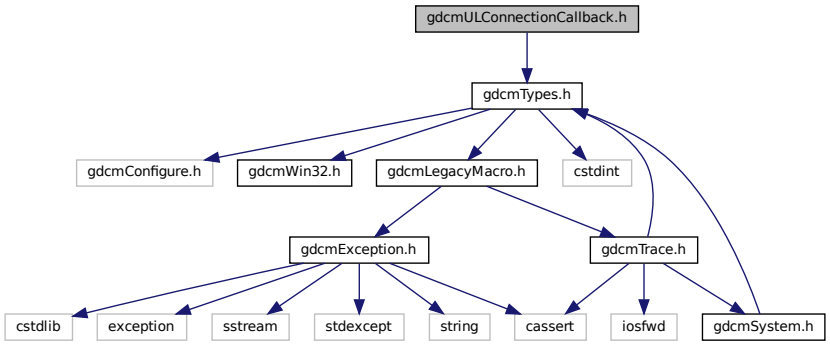
```
1 /*=========================================================================
2 *
3 *  Copyright NumFOCUS
4 *
5 *  Licensed under the Apache License, Version 2.0 (the "License");
6 *  you may not use this file except in compliance with the License.
7 *  You may obtain a copy of the License at
8 *
9 *         http://www.apache.org/licenses/LICENSE-2.0.txt
10 *
11 *  Unless required by applicable law or agreed to in writing, software
12 *  distributed under the License is distributed on an "AS IS" BASIS,
13 *  WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
14 *  See the License for the specific language governing permissions and
```

```
15 *  limitations under the License.
16 *
17 *=========================================================================*/
18 #ifndef GDCMULCONNECTION_H
19 #define GDCMULCONNECTION_H
20
21 #include "gdcmNetworkStateID.h"
22 #include "gdcmARTIMTimer.h"
23 #include "gdcmULConnectionInfo.h"
24 #include "gdcmPresentationContextRQ.h"
25 #include "gdcmDataElement.h"
26 #include "gdcmPresentationContextAC.h"
27 #include "gdcmPresentationContext.h"
28
29 class iosockinet;
30 class echo;
31 namespace gdcm{
32   namespace network{
33
57 class GDCM_EXPORT ULConnection
58 {
59       ULConnectionInfo mInfo;
60       //this is a dirty dirty hack
61       //but to establish an outgoing connection (scu), we need the echo service
62       //to establish incoming, we just need a port and localhost, so an iosockinet works while an
63       //echo would fail (probably because one already exists)
64       echo* mEcho;
65       iosockinet* mSocket;//of the three protocols offered by socket++-- echo, smtp, and ftp--
66       //echo most closely matches what the DICOM standard describes as a network connection
67       ARTIMTimer mTimer;
68
69       EStateID mCurrentState;
70
71       std::vector<PresentationContextRQ> mPresentationContexts;
72       //this is our list of presentation contexts of what we can send
73       uint32_t mMaxPDUSize;
74
75       std::vector<PresentationContextAC> mAcceptedPresentationContexts;//these come back from the server
76       //and tell us what can be sent over this connection
77
78       TransferSyntaxSub cstorets;
79
80       friend class ULActionAE6;
81       void SetCStoreTransferSyntax( TransferSyntaxSub const & ts );
82       friend class ULConnectionManager;
83       TransferSyntaxSub const & GetCStoreTransferSyntax( ) const;
84     public:
85
86       ULConnection(const ULConnectionInfo& inUserInformation);
87       //destructors are virtual to prevent memory leaks by inherited classes
88       virtual ~ULConnection();
89
90       EStateID GetState() const;
91       void SetState(const EStateID& inState);//must be able to update state...
92
93       //echo* GetProtocol();
94       std::iostream* GetProtocol();
95       void StopProtocol();
96
97       ARTIMTimer& GetTimer();
98
99       const ULConnectionInfo &GetConnectionInfo() const;
100
101       //when the connection is first associated, the connection is told
102       //the max packet/PDU size and the way in which to present data
103       //(presentation contexts, etc).  Store that here.
104       void SetMaxPDUSize(uint32_t inSize);
105       uint32_t GetMaxPDUSize() const;
106
107       const PresentationContextAC *GetPresentationContextACByID(uint8_t id) const;
108       const PresentationContextRQ *GetPresentationContextRQByID(uint8_t id) const;
109
111       uint8_t GetPresentationContextIDFromPresentationContext(PresentationContextRQ const & pc) const;
112
113       std::vector<PresentationContextRQ> const & GetPresentationContexts() const;
114       void SetPresentationContexts(const std::vector<PresentationContextRQ>& inContexts);
115
116       void SetPresentationContexts(const std::vector<PresentationContext>& inContexts);
117
118       //given a particular data element, presumably the SOP class,
119       //find the presentation context for that SOP
```

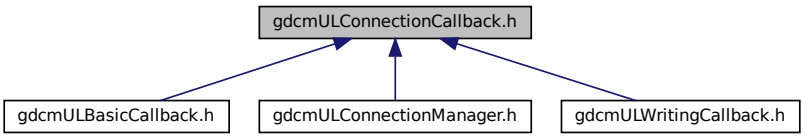
```
120       //NOT YET IMPLEMENTED
121       PresentationContextRQ FindContext(const DataElement& de) const;
122
123       std::vector<PresentationContextAC> const & GetAcceptedPresentationContexts() const;
124       std::vector<PresentationContextAC> & GetAcceptedPresentationContexts();
125       void AddAcceptedPresentationContext(const PresentationContextAC& inPC);
126
128       bool InitializeConnection();
129
131       bool InitializeIncomingConnection();
132
133       ULConnection(const ULConnection&) = delete;
134       void operator=(const ULConnection&) = delete;
135     };
136   }
137 }
138
139 #endif // ULCONNECTION_H
```

11.587 gdcmULConnectionCallback.h File Reference

`#include "gdcmTypes.h"`
Include dependency graph for gdcmULConnectionCallback.h:



This graph shows which files directly or indirectly include this file:



Classes

- class gdcm::network::ULConnectionCallback

Namespaces

- namespace gdcm
- namespace gdcm::network

11.588 gdcmULConnectionCallback.h

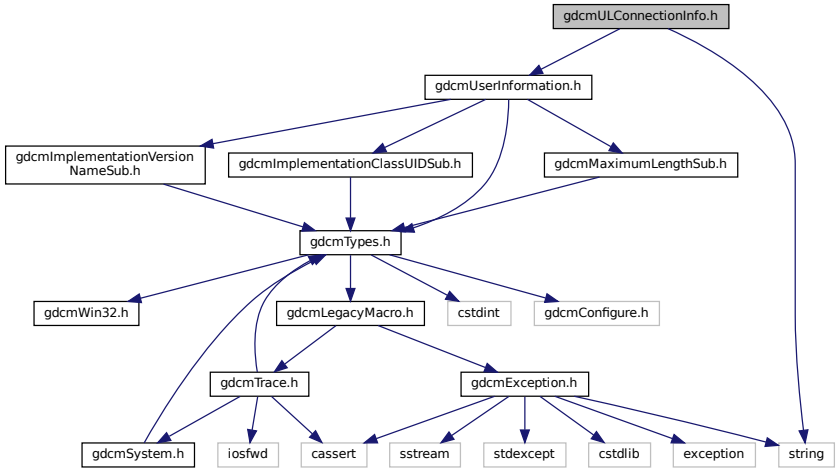
Go to the documentation of this file.

```
1 /*=========================================================================
2 *
3 *  Copyright NumFOCUS
4 *
5 *  Licensed under the Apache License, Version 2.0 (the "License");
6 *  you may not use this file except in compliance with the License.
7 *  You may obtain a copy of the License at
8 *
9 *         http://www.apache.org/licenses/LICENSE-2.0.txt
10 *
11 *  Unless required by applicable law or agreed to in writing, software
12 *  distributed under the License is distributed on an "AS IS" BASIS,
13 *  WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
14 *  See the License for the specific language governing permissions and
15 *  limitations under the License.
16 *
17 *=========================================================================*/
18 #ifndef GDCMULCONNECTIONCALLBACK_H
19 #define GDCMULCONNECTIONCALLBACK_H
20
21 #include "gdcmTypes.h" //to be able to export the class
22
23 namespace gdcm
24 {
25   class DataSet;
26   namespace network
27   {
39     class GDCM_EXPORT ULConnectionCallback {
40       bool mHandledDataSet;
41     protected:
42       bool mImplicit;
43       //inherited callbacks MUST call this function for the cmove loop to work properly
44       void DataSetHandled() { mHandledDataSet = true; }
45     public:
46       ULConnectionCallback():mHandledDataSet(false),mImplicit(true){}
47       virtual ~ULConnectionCallback() = default; //placeholder for inherited objects
48       virtual void HandleDataSet(const DataSet& inDataSet) = 0;
49       virtual void HandleResponse(const DataSet& inDataSet) = 0;
50
51       bool DataSetHandles()const { return mHandledDataSet; }
52       void ResetHandledDataSet() { mHandledDataSet = false; }
53
54       void SetImplicitFlag( const bool imp ) { mImplicit = imp; }
55     };
56   }
57 }
58 #endif //GDCMULCONNECTIONCALLBACK_H
```

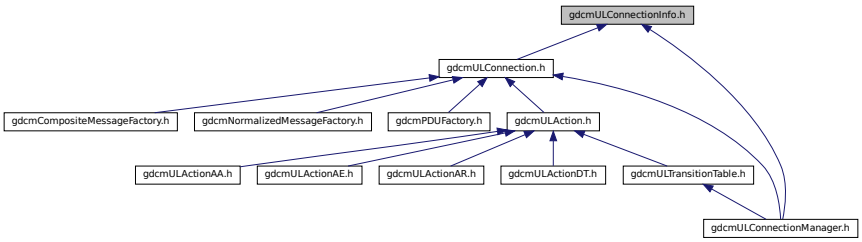
11.589 gdcmULConnectionInfo.h File Reference

```
#include "gdcmUserInformation.h"
#include <string>
```

Include dependency graph for gdcmULConnectionInfo.h:



This graph shows which files directly or indirectly include this file:



Classes

- class gdcm::network::ULConnectionInfo

  ULConnectionInfo.

Namespaces

- namespace gdcm
- namespace gdcm::network

11.590 gdcmULConnectionInfo.h

Go to the documentation of this file.

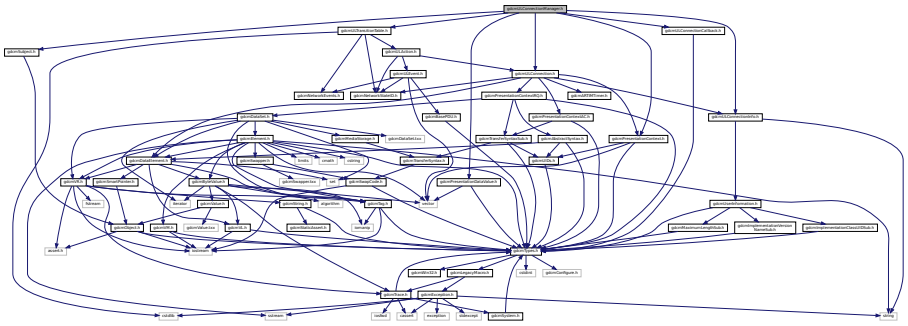
```
1 /*=========================================================================
2 *
3 *  Copyright NumFOCUS
4 *
5 *  Licensed under the Apache License, Version 2.0 (the "License");
6 *  you may not use this file except in compliance with the License.
7 *  You may obtain a copy of the License at
8 *
9 *         http://www.apache.org/licenses/LICENSE-2.0.txt
10 *
11 *  Unless required by applicable law or agreed to in writing, software
12 *  distributed under the License is distributed on an "AS IS" BASIS,
13 *  WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
14 *  See the License for the specific language governing permissions and
15 *  limitations under the License.
16 *
17 *=========================================================================*/
18 #ifndef GDCMULCONNECTIONINFO_H
19 #define GDCMULCONNECTIONINFO_H
20
21 #include "gdcmUserInformation.h"
22 #include <string>
23
24 namespace gdcm{
25   namespace network {
38 class ULConnectionInfo {
39       UserInformation mUserInformation;
40
41       std::string mCalledAETitle;
42       std::string mCallingAETitle;
43
44       unsigned long mCalledIPAddress;
45       int mCalledIPPort;
46       std::string mCalledComputerName; //either the IP or the name has to be filled in
47
48       unsigned long mMaxPDULength;
49     public:
50       ULConnectionInfo();
51
52       //it is possible to misinitialize this object, so
53       //have it return false if something breaks (ie, given AEs are bigger than 16 characters,
54       //no name or IP address).
55       bool Initialize(UserInformation const &inUserInformation,
56         const char *inCalledAETitle, const char *inCallingAETitle,
57         unsigned long inCalledIPAddress, int inCalledIPPort,
58         std::string inCalledComputerName);
59
60       //UserInformation GetUserInformation() const;
61       const char* GetCalledAETitle() const;
62       const char* GetCallingAETitle() const;
63
64       unsigned long GetCalledIPAddress() const;
65       int GetCalledIPPort() const;
66       std::string GetCalledComputerName() const;
67
68       //CStore needs to know the max pdu length, so the value gets initialized
69       //when a cstore connection is established (but not for the others).
70       void SetMaxPDULength(unsigned long inMaxPDULength);
71       unsigned long GetMaxPDULength() const;
72     };
73   }
74 }
75
76 #endif //GDCMULCONNECTIONINFO_H
```

11.591 gdcmULConnectionManager.h File Reference

```
#include "gdcmULTransitionTable.h"
#include "gdcmULConnection.h"
```

```
#include "gdcmULConnectionInfo.h"
#include "gdcmPresentationDataValue.h"
#include "gdcmULConnectionCallback.h"
#include "gdcmSubject.h"
#include "gdcmPresentationContext.h"
```

Include dependency graph for gdcmULConnectionManager.h:



Classes

- class gdcm::network::ULConnectionManager

  ULConnectionManager.

Namespaces

- namespace gdcm
- namespace gdcm::network

11.592 gdcmULConnectionManager.h

Go to the documentation of this file.

```
1 /*=========================================================================
2 *
3 *  Copyright NumFOCUS
4 *
5 *  Licensed under the Apache License, Version 2.0 (the "License");
6 *  you may not use this file except in compliance with the License.
7 *  You may obtain a copy of the License at
8 *
9 *         http://www.apache.org/licenses/LICENSE-2.0.txt
10 *
11 *  Unless required by applicable law or agreed to in writing, software
12 *  distributed under the License is distributed on an "AS IS" BASIS,
13 *  WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
14 *  See the License for the specific language governing permissions and
15 *  limitations under the License.
16 *
17 *=========================================================================*/
18 #ifndef GDCMULCONNECTIONMANAGER_H
19 #define GDCMULCONNECTIONMANAGER_H
20
21 #include "gdcmULTransitionTable.h"
22 #include "gdcmULConnection.h"
23 #include "gdcmULConnectionInfo.h"
24 #include "gdcmPresentationDataValue.h"
```

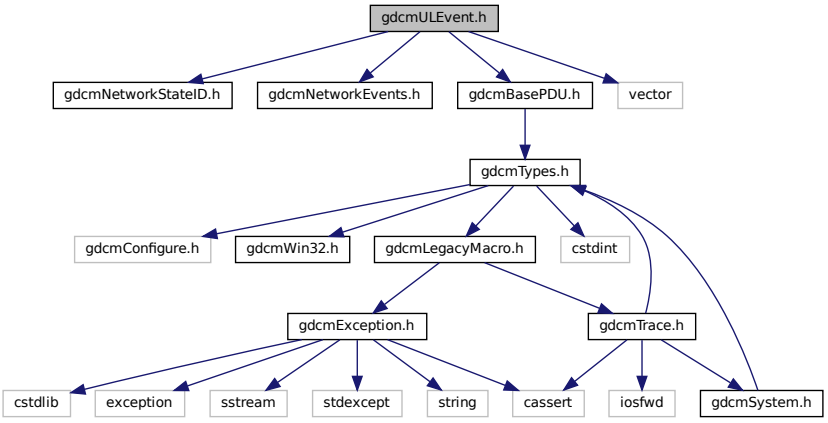
```
25 #include "gdcmULConnectionCallback.h"
26 #include "gdcmSubject.h"
27 #include "gdcmPresentationContext.h"
28
29 namespace gdcm {
30   class File;
31   class BaseRootQuery;
32   class BaseQuery;
33
34   namespace network {
35
45   class GDCM_EXPORT ULConnectionManager :  public Subject
46   {
47     protected:
48       ULConnection* mConnection;
49       ULConnection* mSecondaryConnection;
50       ULTransitionTable mTransitions;
51
52       //no copying
53       ULConnectionManager(const ULConnectionManager& inCM);
54
55       //event handler loop.
56       //will just keep running until the current event is nonexistent.
57       //at which point, it will return the current state of the connection
58       //this starts by initiating an action, but can be put into a passive mode
59       //for a cmove/cstore combination by setting startWaiting to true
60       EStateID RunEventLoop(ULEvent& inEvent, ULConnection* inWhichConnection,
61         ULConnectionCallback* inCallback, const bool& startWaiting);
62
63       //like the above, but will manage the event loop for a move event (which
64       //is basically two simultaneous connections interwoven, one inbound and
65       //the other outbound.    Note, for instance, that cmoversp's can be sent back
66       //during the other connection's operation.
67       EStateID RunMoveEventLoop(ULEvent& inEvent, ULConnectionCallback* inCallback);
68
69     public:
70       ULConnectionManager();
71       ~ULConnectionManager() override;
72
73       // NOTE: (MM) The following two functions are difficults to use, therefore marking
74       // them as internal for now.
75
76       // \internal
84       bool EstablishConnection(const std::string& inAETitle,
85         const std::string& inConnectAETitle,
86         const std::string& inComputerName, long inIPAddress,
87         uint16_t inConnectPort, double inTimeout,
88         std::vector<PresentationContext> const & pcVector );
89
92       bool EstablishConnectionMove(const std::string& inAETitle,
93         const std::string& inConnectAETitle,
94         const std::string& inComputerName, long inIPAddress,
95         uint16_t inConnectPort, double inTimeout,
96         uint16_t inReturnPort,
97         std::vector<PresentationContext> const & pcVector);
98       // \endinternal
99
100
101       //bool ReestablishConnection(const EConnectionType& inConnectionType,
102       //  const DataSet& inDS);
103
104       //allows for a connection to be broken, but waits for an acknowledgement
105       //of the breaking for a certain amount of time.    Returns true of the
106       //other side acknowledges the break
107       bool BreakConnection(const double& inTimeout);
108
109       //severs the connection, if it's open, without waiting for any kind of response.
110       //typically done if the program is going down.
111       void BreakConnectionNow();
112
113       //This function will send a given piece of data
114       //across the network connection.    It will return true if the
115       //sending worked, false otherwise.
116       //note that sending is asynchronous; as such, there's
117       //also a 'receive' option, but that requires a callback function.
118       //bool SendData();
119
120       //send the Data PDU associated with Echo (ie, a default DataPDU)
121       //this lets the user confirm that the connection is alive.
122       //the user should look to cout to see the response of the echo command
123       //returns the PresentationDataValue that was returned by the remote
```

```
124       //host.    Note that the PDV can be uninitialized, which would indicate failure.
125       //Echo does not use a callback for results.
126       std::vector<PresentationDataValue> SendEcho();
127
128       // \internal
129       // API will change...
130       std::vector<DataSet> SendStore(const File &file, std::istream * pStream = nullptr, std::streampos
      dataSetOffset = 0  );
131       std::vector<DataSet> SendFind(const BaseRootQuery* inRootQuery);
132       std::vector<DataSet> SendMove(const BaseRootQuery* inRootQuery);
133
134     std::vector<DataSet> SendNEventReport (const BaseQuery* inQuery);
135       std::vector<DataSet> SendNGet     (const BaseQuery* inQuery);
136       std::vector<DataSet> SendNSet     (const BaseQuery* inQuery);
137       std::vector<DataSet> SendNAction   (const BaseQuery* inQuery);
138       std::vector<DataSet> SendNCreate   (const BaseQuery* inQuery);
139       std::vector<DataSet> SendNDelete   (const BaseQuery* inQuery);
140       // \endinternal
141
143       void SendStore(const File & file, ULConnectionCallback* inCallback, std::istream * pStream = nullptr ,
      std::streampos dataSetOffset = 0 );
144       void SendFind(const BaseRootQuery* inRootQuery, ULConnectionCallback* inCallback);
146       bool SendMove(const BaseRootQuery* inRootQuery, ULConnectionCallback* inCallback);
147
148       void SendNEventReport (const BaseQuery* inQuery, ULConnectionCallback* inCallback);
149       void SendNGet     (const BaseQuery* inQuery, ULConnectionCallback* inCallback);
150       void SendNSet     (const BaseQuery* inQuery, ULConnectionCallback* inCallback);
151       void SendNAction   (const BaseQuery* inQuery, ULConnectionCallback* inCallback);
152       void SendNCreate   (const BaseQuery* inQuery, ULConnectionCallback* inCallback);
153       void SendNDelete   (const BaseQuery* inQuery, ULConnectionCallback* inCallback);
154
155     };
156   }
157 }
158
159 #endif // GDCMULCONNECTIONMANAGER_H
```

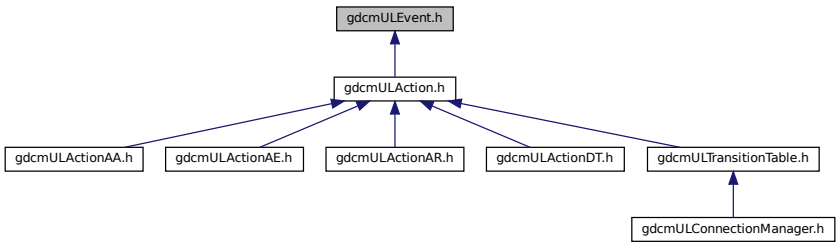
11.593 gdcmULEvent.h File Reference

```
#include "gdcmNetworkStateID.h"
#include "gdcmNetworkEvents.h"
#include "gdcmBasePDU.h"
#include <vector>
```

Include dependency graph for gdcmULEvent.h:



This graph shows which files directly or indirectly include this file:



Classes

- class gdcm::network::ULEvent

  ULEvent.

Namespaces

- namespace gdcm
- namespace gdcm::network

11.594 gdcmULEvent.h

Go to the documentation of this file.

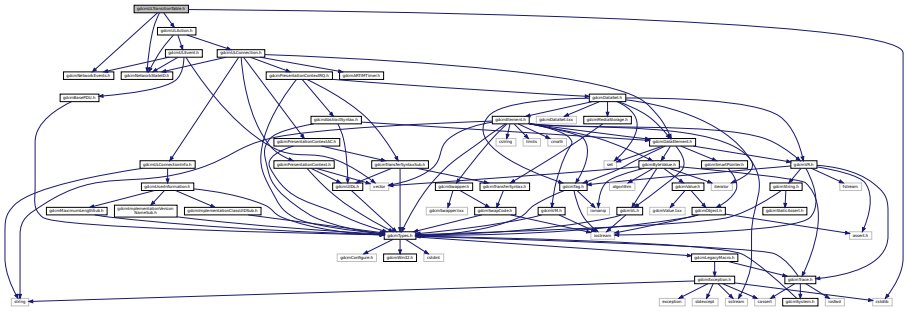
```
1 /*=========================================================================
2 *
3 *  Copyright NumFOCUS
4 *
5 *  Licensed under the Apache License, Version 2.0 (the "License");
6 *  you may not use this file except in compliance with the License.
7 *  You may obtain a copy of the License at
8 *
9 *         http://www.apache.org/licenses/LICENSE-2.0.txt
10 *
11 *  Unless required by applicable law or agreed to in writing, software
12 *  distributed under the License is distributed on an "AS IS" BASIS,
13 *  WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
14 *  See the License for the specific language governing permissions and
15 *  limitations under the License.
16 *
17 *=========================================================================*/
18 #ifndef GDCMULEVENT_H
19 #define GDCMULEVENT_H
20
21 #include "gdcmNetworkStateID.h"
22 #include "gdcmNetworkEvents.h"
23 #include "gdcmBasePDU.h"
24 #include <vector>
25
26 namespace gdcm {
27   namespace network {
28
37 class ULEvent {
38       EEventID mEvent;
39       std::vector<BasePDU*> mBasePDU;
```

```
40     std::istream * m_pStream ;
41     std::streampos m_posDataSet ;
42       void DeletePDUVector(){
43         std::vector<BasePDU*>::iterator baseItor;
44         for (baseItor = mBasePDU.begin(); baseItor < mBasePDU.end(); baseItor++){
45           if (*baseItor != NULL){
46             delete *baseItor;
47             *baseItor = NULL;
48           }
49         }
50       }
51
52     public:
53       ULEvent(const EEventID& inEventID, std::vector<BasePDU*> inBasePDU, std::istream * iStream = nullptr,
     std::streampos posDataSet = 0 ){
54         mEvent = inEventID;
55         mBasePDU = inBasePDU;
56     m_pStream = iStream ;
57     m_posDataSet = posDataSet ;
58       }
59       ULEvent(const EEventID& inEventID, BasePDU* inBasePDU, std::istream * iStream = nullptr, std::streampos
     posDataSet = 0 ){
60         mEvent = inEventID;
61         mBasePDU.push_back(inBasePDU);
62     m_pStream = iStream ;
63     m_posDataSet = posDataSet ;
64       }
65       ~ULEvent(){
66         DeletePDUVector();
67       }
68
69       EEventID GetEvent()const { return mEvent; }
70       std::vector<BasePDU*> const & GetPDUs()const { return mBasePDU; }
71     std::istream * GetIStream()const { return m_pStream; }
72     std::streampos GetDataSetPos()const { return m_posDataSet; }
73
74       void SetEvent(const EEventID& inEvent) { mEvent = inEvent; }
75       void SetPDU(std::vector<BasePDU*> const & inPDU) {
76         DeletePDUVector();
77         mBasePDU = inPDU;
78       }
79     };
80   }
81 }
82
83 #endif //GDCMULEVENT_H
```

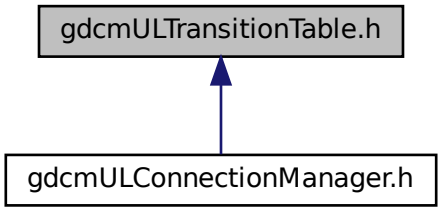
11.595 gdcmULTransitionTable.h File Reference

```
#include "gdcmNetworkStateID.h"
#include "gdcmNetworkEvents.h"
#include "gdcmULAction.h"
#include <cstdlib>
```

Include dependency graph for gdcmULTransitionTable.h:



This graph shows which files directly or indirectly include this file:



Classes

- class gdcm::network::TableRow
- struct gdcm::network::Transition
- class gdcm::network::ULTransitionTable

  ULTransitionTable The transition table of all the ULEvents, new ULActions, and ULStates.

Namespaces

- namespace gdcm
- namespace gdcm::network

11.596 gdcmULTransitionTable.h

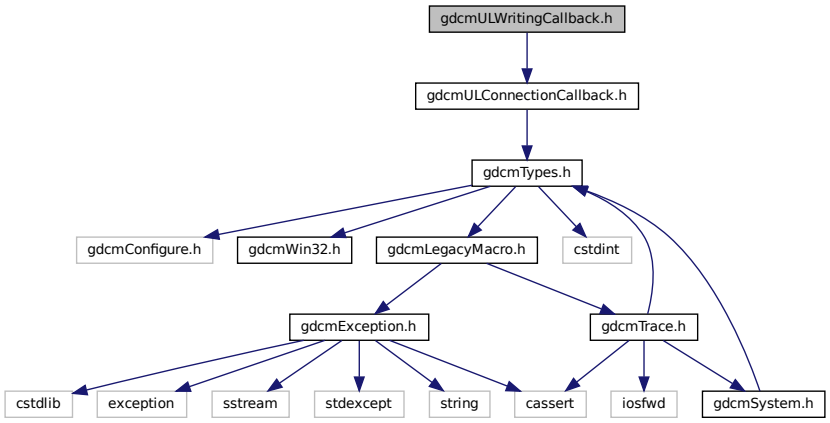
Go to the documentation of this file.

```
1 /*=========================================================================
2 *
3 *  Copyright NumFOCUS
4 *
5 *  Licensed under the Apache License, Version 2.0 (the "License");
6 *  you may not use this file except in compliance with the License.
7 *  You may obtain a copy of the License at
8 *
9 *         http://www.apache.org/licenses/LICENSE-2.0.txt
10 *
11 *  Unless required by applicable law or agreed to in writing, software
12 *  distributed under the License is distributed on an "AS IS" BASIS,
13 *  WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
14 *  See the License for the specific language governing permissions and
15 *  limitations under the License.
16 *
17 *=========================================================================*/
18 #ifndef GDCMULTRANSITIONTABLE_H
19 #define GDCMULTRANSITIONTABLE_H
20
21 #include "gdcmNetworkStateID.h"
22 #include "gdcmNetworkEvents.h"
23 #include "gdcmULAction.h"
24
25 #include <cstdlib>  // NULL
```

```
26
27 namespace gdcm {
28 class Subject;
29   namespace network{
30 class ULConnection;
31 class ULAction;
32 class ULEvent;
33
34     //The transition dictates the action that should be taken from the start state to the end state
35     struct Transition {
36       int mEnd;
37       ULAction* mAction;
38       Transition(){
39         mEnd = eStaDoesNotExist;
40         mAction = nullptr;
41       }
42       ~Transition(){
43         if (mAction != nullptr){
44           delete mAction;
45           mAction = nullptr;
46         }
47       }
48       Transition(int inEndState, ULAction* inAction){
49         mEnd = inEndState;
50         mAction = inAction;
51       }
52       static Transition* MakeNew(int inEndState, ULAction* inAction){
53         return new Transition(inEndState, inAction);
54       }
55     };
56
57     //used to define a row in table 9-10 of 3.8 2009
58     //the transition table is events, then state,
59     //then the transition itself (which has the event
60     //and start state implied by their starting locations)
61     //don't need to store the event; that's implicitly defined in the Table itself by location
62     class TableRow{
63     public:
64       TableRow() {
65         for(int stateIndex = 0; stateIndex < cMaxStateID; ++stateIndex)
66           {
67           transitions[stateIndex] = nullptr;
68           }
69       }
70       ~TableRow() {
71         for(int stateIndex = 0; stateIndex < cMaxStateID; ++stateIndex)
72           {
73           Transition *t = transitions[stateIndex];
74           delete t;
75           }
76        }
77       Transition *transitions[cMaxStateID];
78
79       //copy constructor for stl additions into the transition table below.
80     };
81
98 class ULTransitionTable
99 {
100     private:
101       TableRow mTable[cMaxEventID];
102     public:
103       ULTransitionTable();
104
105       void HandleEvent(Subject*s,ULEvent& inEvent, ULConnection& inConnection,
106         bool& outWaitingForEvent, EEventID& outRaisedEvent) const;
107
108       void PrintTable() const; //so that the table can be printed and verified against the DICOM standard
109     };
110   }
111 }
112 #endif // GDCMULTRANSITIONTABLE_H
```

11.597 gdcmULWritingCallback.h File Reference

`#include "gdcmULConnectionCallback.h"`
Include dependency graph for gdcmULWritingCallback.h:



Classes

- class gdcm::network::ULWritingCallback

Namespaces

- namespace gdcm
- namespace gdcm::network

11.598 gdcmULWritingCallback.h

Go to the documentation of this file.

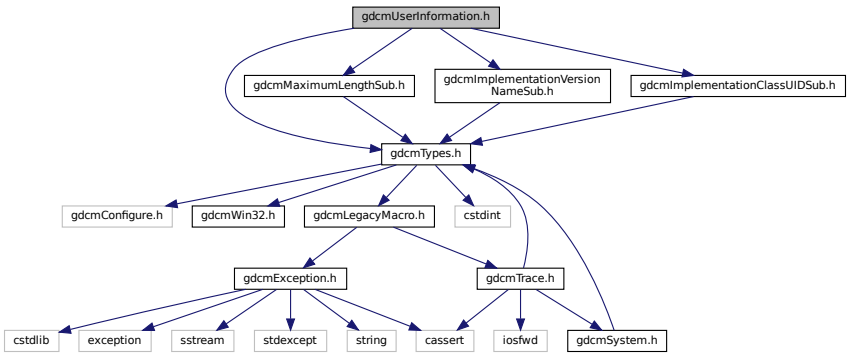
```
1 /*=========================================================================
2 *
3 *  Copyright NumFOCUS
4 *
5 *  Licensed under the Apache License, Version 2.0 (the "License");
6 *  you may not use this file except in compliance with the License.
7 *  You may obtain a copy of the License at
8 *
9 *         http://www.apache.org/licenses/LICENSE-2.0.txt
10 *
11 *  Unless required by applicable law or agreed to in writing, software
12 *  distributed under the License is distributed on an "AS IS" BASIS,
13 *  WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
14 *  See the License for the specific language governing permissions and
15 *  limitations under the License.
16 *
17 *=========================================================================*/
18 #ifndef GDCMULCONNECTIONWRITINGCALLBACK_H
19 #define GDCMULCONNECTIONWRITINGCALLBACK_H
```

```
20
21 #include "gdcmULConnectionCallback.h"
22
23 namespace gdcm
24 {
25 class DataSet;
26 namespace network
27 {
28 /* \brief ULWritingCallback
29 * \details This is the most basic of callbacks for how the ULConnectionManager handles
30 * incoming datasets.    DataSets are immediately written to disk as soon as they
31 * are received.    NOTE that if the incoming connection is faster than the disk
32 * writing speed, this callback could cause some pileups!
33 */
34 class GDCM_EXPORT ULWritingCallback :  public ULConnectionCallback
35 {
36   std::string mDirectoryName;
37 public:
38   ULWritingCallback() = default;
39   ~ULWritingCallback() override = default; //empty, for later inheritance
40
42   void SetDirectory(const std::string& inDirectoryName) { mDirectoryName = inDirectoryName; }
43
44   void HandleDataSet(const DataSet& inDataSet) override;
45   void HandleResponse(const DataSet& inDataSet) override;
46 };
47 } // end namespace network
48 } // end namespace gdcm
49
50 #endif //GDCMULCONNECTIONWRITINGCALLBACK_H
```

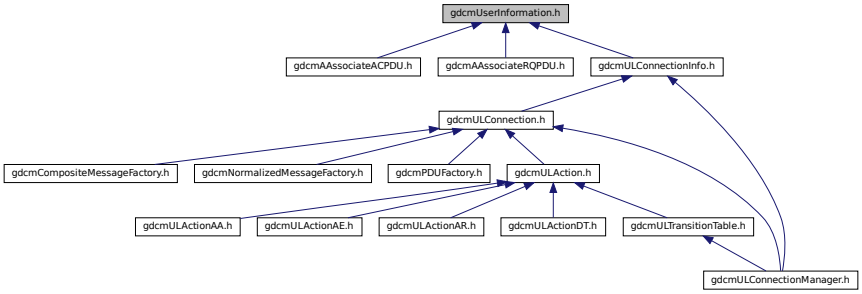
11.599 gdcmUserInformation.h File Reference

```
#include "gdcmTypes.h"
#include "gdcmMaximumLengthSub.h"
#include "gdcmImplementationVersionNameSub.h"
#include "gdcmImplementationClassUIDSub.h"
```

Include dependency graph for gdcmUserInformation.h:



This graph shows which files directly or indirectly include this file:



Classes

- class gdcm::network::UserInformation

  UserInformation.

Namespaces

- namespace gdcm
- namespace gdcm::network

11.600 gdcmUserInformation.h

Go to the documentation of this file.

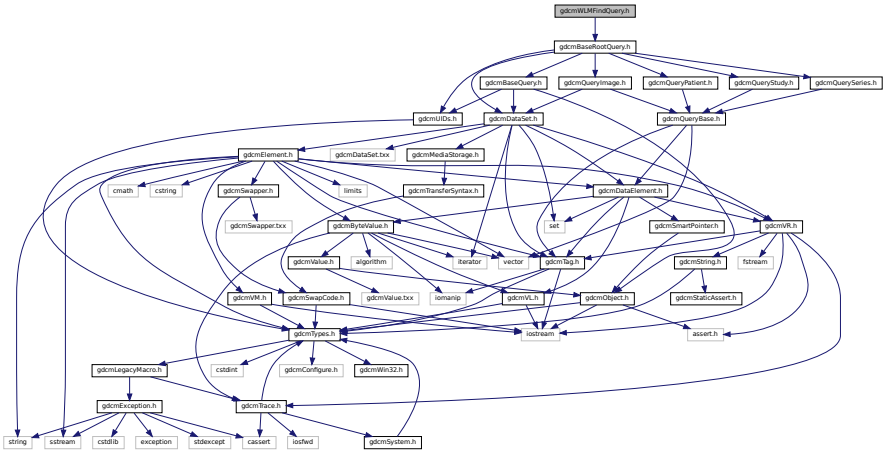
```
1 /*=========================================================================
2
3 Program:  GDCM (Grassroots DICOM). A DICOM library
4
5 Copyright (c) 2006-2011 Mathieu Malaterre
6 All rights reserved.
7 See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.
8
9 This software is distributed WITHOUT ANY WARRANTY; without even
10 the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
11 PURPOSE.  See the above copyright notice for more information.
12
13 =========================================================================*/
14 #ifndef GDCMUSERINFORMATION_H
15 #define GDCMUSERINFORMATION_H
16
17 #include "gdcmTypes.h"
18 #include "gdcmMaximumLengthSub.h"
19 #include "gdcmImplementationVersionNameSub.h"
20 #include "gdcmImplementationClassUIDSub.h"
21
22 namespace gdcm
23 {
24
25 namespace network
26 {
27
28 class AsynchronousOperationsWindowSub;
29 class RoleSelectionSub;
30 struct RoleSelectionSubItems;
```

```
31 class SOPClassExtendedNegociationSub;
32 struct SOPClassExtendedNegociationSubItems;
43 class UserInformation
44 {
45 public:
46   UserInformation();
47   ~UserInformation();
48   std::istream &Read(std::istream &is);
49   const std::ostream &Write(std::ostream &os) const;
50   size_t Size() const;
51
52   void Print(std::ostream &os) const;
53
54   const MaximumLengthSub &GetMaximumLengthSub()const { return MLS; }
55   MaximumLengthSub &GetMaximumLengthSub() { return MLS; }
56
57   void AddRoleSelectionSub( RoleSelectionSub const & r );
58   void AddSOPClassExtendedNegociationSub( SOPClassExtendedNegociationSub const & s );
59
60 private:
61   static const uint8_t ItemType;
62   static const uint8_t Reserved2;
63   uint16_t ItemLength; // len of
64   MaximumLengthSub MLS;
65   ImplementationClassUIDSub ICUID;
66   AsynchronousOperationsWindowSub *AOWS;
67   RoleSelectionSubItems *RSSI;
68   SOPClassExtendedNegociationSubItems *SOPCENSI;
69   ImplementationVersionNameSub IVNS;
70
71 public:
72   UserInformation(const UserInformation&) = delete;
73   UserInformation &operator=(const UserInformation&);
74 };
75
76 } // end namespace network
77
78 } // end namespace gdcm
79
80 #endif //GDCMUSERINFORMATION_H
```

11.601 gdcmWLMFindQuery.h File Reference

`#include "gdcmBaseRootQuery.h"`

Include dependency graph for gdcmWLMFindQuery.h:



Classes

- class gdcm::WLMFindQuery

  PatientRootQuery.

Namespaces

- namespace gdcm

11.602 gdcmWLMFindQuery.h

Go to the documentation of this file.

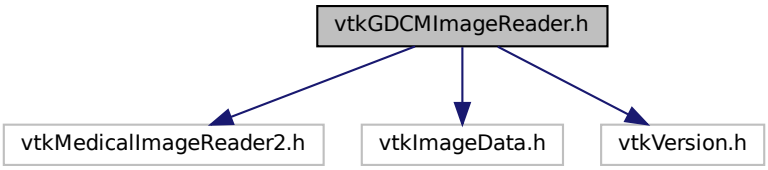
```
1 /*=========================================================================
2
3 Program:  GDCM (Grassroots DICOM). A DICOM library
4
5 Copyright (c) 2006-2011 Mathieu Malaterre
6 All rights reserved.
7 See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.
8
9 This software is distributed WITHOUT ANY WARRANTY; without even
10 the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
11 PURPOSE.  See the above copyright notice for more information.
12
13 =========================================================================*/
14 #ifndef GDCMWLMFindQuery_H
15 #define GDCMWLMFindQuery_H
16
17 #include "gdcmBaseRootQuery.h"
18
19 namespace gdcm
20 {
25 class GDCM_EXPORT WLMFindQuery :  public BaseRootQuery
26 {
27   friend class QueryFactory;
28 public:
29   WLMFindQuery();
30
31   // no sense here
32   void InitializeDataSet(const EQueryLevel& inQueryLevel) override;
33   std::vector<Tag> GetTagListByLevel(const EQueryLevel& inQueryLevel) override;
34   // validate query has required tag
35   bool ValidateQuery(bool inStrict = true) const override;
36
37   UIDs::TSName GetAbstractSyntaxUID() const override;
38 protected :
39   DataSet GetValidDataSet() const;
40 };
41
42 } // end namespace gdcm
43
44 #endif // GDCMWLMFindQuery_H
```

11.603 vtkGDCMImageReader.h File Reference

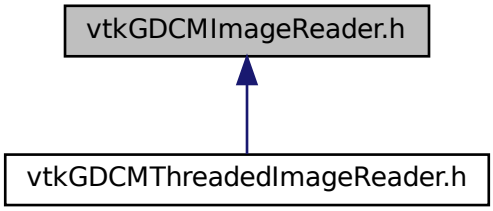
```
#include "vtkMedicalImageReader2.h"
#include "vtkImageData.h"
```

```
#include "vtkVersion.h"
```

Include dependency graph for vtkGDCMImageReader.h:



This graph shows which files directly or indirectly include this file:



Classes

- class vtkGDCMImageReader

Namespaces

- namespace gdcm

Macros

- #define VTK_CMYK 8
- #define VTK_INVERSE_LUMINANCE 5
- #define VTK_LOOKUP_TABLE 6
- #define VTK_YBR 7

11.603.1 Macro Definition Documentation

11.603.1.1 VTK_CMYK

```
#define VTK_CMYK 8
```

11.603.1.2 VTK_INVERSE_LUMINANCE

```
#define VTK_INVERSE_LUMINANCE 5
```

11.603.1.3 VTK_LOOKUP_TABLE

```
#define VTK_LOOKUP_TABLE 6
```

11.603.1.4 VTK_YBR

```
#define VTK_YBR 7
```

11.604 vtkGDCMImageReader.h

Go to the documentation of this file.

```
1 /*=========================================================================
2
3 Program:  GDCM (Grassroots DICOM). A DICOM library
4
5 Copyright (c) 2006-2011 Mathieu Malaterre
6 All rights reserved.
7 See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.
8
9 This software is distributed WITHOUT ANY WARRANTY; without even
10 the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
11 PURPOSE.  See the above copyright notice for more information.
12
13 =========================================================================*/
14 // .NAME vtkGDCMImageReader - read DICOM Image files (Pixel Data)
15 // .SECTION Description
16 // vtkGDCMImageReader is a source object that reads some DICOM files
17 // this reader is single threaded.
18 // .SECTION Implementation note:  when FileLowerLeft is set to on the image is not flipped
19 // upside down as VTK would expect, use this option only if you know what you are doing.
20 // .SECTION Implementation note:  when reading a series of 2D slices, user is
21 // expected to provide an ordered list of filenames.  No sorting will be applied afterward.
22 // .SECTION Implementation note:  Although 99% of the time the Zspacing as read
23 // from a tag in a 2D DICOM file should be correct, there has been reports that this
```

```
24 // value can be missing, or incorrect, in which case users are advised to override this
25 // value using the return value from gdcm::IPPSorter::GetZSpacing() and set it via
26 // vtkImageChangeInformation on the reader itself.
27 // .SECTION TODO
28 // This reader does not handle a series of 3D images, only a single 3D (multi frame) or a
29 // list of 2D files are supported for now.
30 // .SECTION TODO
31 // Did not implement SetFilePattern / SetFilePrefix API, move it to protected section for now.
32 // .SECTION BUG
33 // Overlay are assumed to have the same extent as image.  Right now if overlay origin is not
34 // 0,0 the overlay will have an offset...
35 // Only the very first overlay is loaded at the VTK level, for now (even if there are more than one in the
      file)
36 // .SECTION DataOrigin
37 // When the reader is instantiated with FileLowerLeftOn the DataOrigin and Image Position (Patient) are
38 // identical.  But when FileLowerLeft is Off, we have to reorder the Y-line of the image, and thus the
      DataOrigin
39 // is then translated to the other side of the image.
40 // .SECTION Spacing
41 // When reading a 3D volume, the spacing along the Z dimension might be negative (so as to respect
      up-side-down)
42 // as specified in the Image Orientation (Patient) tag.  When Z-spacing is 0, this means the multi-frame
      object
43 // contains image which do not represent uniform volume.
44 // .SECTION Warning
45 // When using vtkGDCMPolyDataReader in conjonction with vtkGDCMImageReader
46 // it is *required* that FileLowerLeft is set to ON as coordinate system
47 // would be inconsistent in between the two data structures.
48 // .SECTION Color Space mapping:
49 // * VTK_LUMINANCE         <-> MONOCHROME2
50 // * VTK_LUMINANCE_ALPHA   <-> Not supported
51 // * VTK_RGB               <-> RGB
52 // * VTK_RGBA              <-> ARGB (deprecated, DICOM 2008)
53 // * VTK_INVERSE_LUMINANCE <-> MONOCHROME1
54 // * VTK_LOOKUP_TABLE      <-> PALETTE COLOR
55 // * VTK_YBR               <-> YBR_FULL
56 //
57 // For detailed information on color space transformation and true lossless transformation see:
58 // http://gdcm.sourceforge.net/wiki/index.php/Color_Space_Transformations
59
60 // .SECTION See Also
61 // vtkMedicalImageReader2 vtkMedicalImageProperties vtkGDCMPolyDataReader vtkGDCMImageWriter
62 // vtkDICOMImageReader
63
64 #ifndef VTKGDCMIMAGEREADER_H
65 #define VTKGDCMIMAGEREADER_H
66
67 #include "vtkMedicalImageReader2.h"
68 #include "vtkImageData.h"
69 #include "vtkVersion.h"
70
71 #if (VTK_MAJOR_VERSION >= 5) || ( VTK_MAJOR_VERSION == 4 && VTK_MINOR_VERSION > 5 )
72 #else
73 class vtkMedicalImageProperties;
74 #endif
75 #if (VTK_MAJOR_VERSION > 5) || ( VTK_MAJOR_VERSION == 5 && VTK_MINOR_VERSION > 0 )
76 #else
77 class vtkStringArray;
78 #endif
79 class vtkPolyData;
80
81 // vtkSystemIncludes.h defines:
82 // #define VTK_LUMINANCE       1
83 // #define VTK_LUMINANCE_ALPHA 2
84 // #define VTK_RGB             3
85 // #define VTK_RGBA            4
86 #ifndef VTK_INVERSE_LUMINANCE
87 #define VTK_INVERSE_LUMINANCE 5
88 #endif
89 #ifndef VTK_LOOKUP_TABLE
90 #define VTK_LOOKUP_TABLE 6
91 #endif
92 #ifndef VTK_YBR
93 #define VTK_YBR 7
94 #endif
95 #ifndef VTK_CMYK
96 #define VTK_CMYK 8
97 #endif
98
99 //BTX
100 namespace gdcm { class ImageReader; }
```

```
101 //ETX
102 class vtkMatrix4x4;
103 class VTK_EXPORT vtkGDCMImageReader :  public vtkMedicalImageReader2
104 {
105 public:
106   static vtkGDCMImageReader *New();
107   vtkTypeMacro(vtkGDCMImageReader,vtkMedicalImageReader2);
108   virtual void PrintSelf(ostream& os, vtkIndent indent);
109
110   // Description:  is the given file name a DICOM file containing an image ?
111   virtual int CanReadFile(const char* fname);
112
113   // Description:
114   // Valid extensions
115   virtual const char* GetFileExtensions()
116     {
117     // I would like to get rid of ACR/NEMA/IMA so only allow dcm extension for now
118     return ".dcm .DCM";
119     }
120
121   // Description:
122   // A descriptive name for this format
123   virtual const char* GetDescriptiveName()
124     {
125     return "DICOM";
126     }
127
128   // Description:
129   // Get the Image Position (Patient) as stored in the DICOM file
130   // This is a read-only data member
131   vtkGetObjectMacro(DirectionCosines, vtkMatrix4x4);
132
133 #if (VTK_MAJOR_VERSION >= 5) || ( VTK_MAJOR_VERSION == 4 && VTK_MINOR_VERSION > 5 )
134 #else
135   // Description:
136   // Get the medical image properties object
137   vtkGetObjectMacro(MedicalImageProperties, vtkMedicalImageProperties);
138 #endif
139   virtual void SetMedicalImageProperties(vtkMedicalImageProperties *pd);
140
141 #if (VTK_MAJOR_VERSION > 5) || ( VTK_MAJOR_VERSION == 5 && VTK_MINOR_VERSION > 0 )
142 #else
143   virtual void SetFileNames(vtkStringArray*);
144   vtkGetObjectMacro(FileNames, vtkStringArray);
145 #endif
146
147   // Description:
148   // Specifically request to load the overlay into the gdcm-VTK layer (gdcm always loads them when found).
149   // If no overlay is found in the image, then the vtkImageData for the overlay will be empty.
150   vtkGetMacro(LoadOverlays,int);
151   vtkSetMacro(LoadOverlays,int);
152   vtkBooleanMacro(LoadOverlays,int);
153
154   // Description:
155   // Set/Get whether or not to load the Icon as vtkImageData (if found in the DICOM file)
156   vtkGetMacro(LoadIconImage,int);
157   vtkSetMacro(LoadIconImage,int);
158   vtkBooleanMacro(LoadIconImage,int);
159
160   // Description:
161   // Set/Get whether or not the image was compressed using a lossy compression algorithm
162   vtkGetMacro(LossyFlag,int);
163   vtkSetMacro(LossyFlag,int);
164   vtkBooleanMacro(LossyFlag,int);
165
166   // Description:
167   // Read only:  number of overlays as found in this image (multiple overlays per slice is allowed)
168   // Only valid when LoadOverlays is true
169   vtkGetMacro(NumberOfOverlays,int);
170
171   // Description:
172   // Read only:  number of icon image (there can only be zero or one icon per file)
173   // Only valid when LoadIconImage is true
174   vtkGetMacro(NumberOfIconImages,int);
175
176   // Description:
177   // Get Overlay/IconImage
178   // Remember to ALWAYS use those methods in your code, as the internal number for the output port
179   // is not garantee to remain the same, as features are added to the reader
180 #if (VTK_MAJOR_VERSION >= 5) || ( VTK_MAJOR_VERSION == 4 && VTK_MINOR_VERSION > 5 )
181 //FIXME: Need to get rid of BTX/ETX if only the Python Wrapper of VTK 4.2 would let me
```

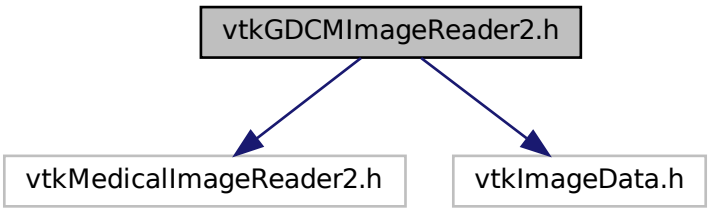
```
182 //BTX
183   vtkAlgorithmOutput* GetOverlayPort(int index);
184   vtkAlgorithmOutput* GetIconImagePort();
185 //ETX
186 #endif
187   vtkImageData* GetOverlay(int i);
188   vtkImageData* GetIconImage();
189
190   // Description:
191   // Load image with its associated Lookup Table
192   vtkGetMacro(ApplyLookupTable,int);
193   vtkSetMacro(ApplyLookupTable,int);
194   vtkBooleanMacro(ApplyLookupTable,int);
195
196   // Description:
197   // Load image as YBR
198   vtkGetMacro(ApplyYBRToRGB,int)
199   vtkSetMacro(ApplyYBRToRGB,int)
200   vtkBooleanMacro(ApplyYBRToRGB,int);
201
202   // Description:
203   // Return VTK_LUMINANCE, VTK_INVERSE_LUMINANCE, VTK_RGB, VTK_RGBA, VTK_LOOKUP_TABLE, VTK_YBR or VTK_CMYK
204   // or 0 when ImageFormat is not handled.
205   // Warning:  For color image, PlanarConfiguration need to be taken into account.
206   vtkGetMacro(ImageFormat,int);
207
208   // Description:
209   // Return the Planar Configuration.  This simply means that the internal DICOM image was stored
210   // using a particular planar configuration (most of the time:  0)
211   // For monochrome image, PlanarConfiguration is always 0
212   vtkGetMacro(PlanarConfiguration,int);
213
214   // Description:
215   // Return the 'raw' information stored in the DICOM file:
216   // In case of a series of multiple files, only the first file is considered.  The Image Orientation
      (Patient)
217   // is garantee to remain the same, and image Image Position (Patient) in other slice can be computed
218   // using the ZSpacing (3rd dimension)
219   // (0020,0032) DS [87.774866\-182.908510\168.629671]       #  32, 3 ImagePositionPatient
220   // (0020,0037) DS [0.001479\0.999989\-0.004376\-0.002039\-0.004372\-0.999988] #  58, 6
      ImageOrientationPatient
221   vtkGetVector3Macro(ImagePositionPatient,double);
222   vtkGetVector6Macro(ImageOrientationPatient,double);
223
224   // Description:
225   // Set/Get the first Curve Data:
226   vtkGetObjectMacro(Curve,vtkPolyData);
227   virtual void SetCurve(vtkPolyData *pd);
228
229   // Description:
230   // \DEPRECATED:
231   // Modality LUT
232   // Value returned by GetShift/GetScale might be inaccurate since Shift/Scale could be
233   // varying along the Series read.  Therefore user are advices not to use those functions
234   // anymore
235   vtkGetMacro(Shift,double);
236   vtkGetMacro(Scale,double);
237
238 protected:
239   vtkGDCMImageReader();
240   ~vtkGDCMImageReader();
241
242   vtkSetVector6Macro(ImageOrientationPatient,double);
243
244 //BTX
245   void FillMedicalImageInformation(const gdcm::ImageReader &reader);
246 //ETX
247   int RequestInformationCompat();
248   int RequestDataCompat();
249
250 #if (VTK_MAJOR_VERSION >= 5) || ( VTK_MAJOR_VERSION == 4 && VTK_MINOR_VERSION > 5 )
251   int ProcessRequest(vtkInformation* request,
252                                vtkInformationVector** inputVector,
253                                vtkInformationVector* outputVector);
254   int RequestInformation(vtkInformation *request,
255                          vtkInformationVector **inputVector,
256                          vtkInformationVector *outputVector);
257   int RequestData(vtkInformation *request,
258                   vtkInformationVector **inputVector,
259                   vtkInformationVector *outputVector);
260 #else /*(VTK_MAJOR_VERSION >= 5) || ( VTK_MAJOR_VERSION == 4 && VTK_MINOR_VERSION > 5 )*/
```

```
261   void ExecuteInformation();
262   void ExecuteData(vtkDataObject *out);
263 #endif /*(VTK_MAJOR_VERSION >= 5) || ( VTK_MAJOR_VERSION == 4 && VTK_MINOR_VERSION > 5 )*/
264
265 protected:
266 #if (VTK_MAJOR_VERSION >= 5) || ( VTK_MAJOR_VERSION == 4 && VTK_MINOR_VERSION > 5 )
267 #else
268   // Description:
269   // Medical Image properties
270   vtkMedicalImageProperties *MedicalImageProperties;
271 #endif
272 #if (VTK_MAJOR_VERSION > 5) || ( VTK_MAJOR_VERSION == 5 && VTK_MINOR_VERSION > 0 )
273 #else
274   vtkStringArray *FileNames;
275 #endif
276
277   vtkMatrix4x4 *DirectionCosines;
278   int LoadOverlays;
279   int NumberOfOverlays;
280   int LoadIconImage;
281   int NumberOfIconImages;
282   int IconImageDataExtent[6];
283   double ImagePositionPatient[3];
284   double ImageOrientationPatient[6];
285   vtkPolyData *Curve;
286
287   int ImageFormat;
288   // the following 3, should remain optional
289   int ApplyInverseVideo;
290   int ApplyLookupTable;
291   int ApplyYBRToRGB;
292   // I think that planar configuration need to always be applied as far as VTK is concerned
293   int ApplyPlanarConfiguration;
294   int ApplyShiftScale;
295
296   int LoadSingleFile(const char *filename, char *pointer, unsigned long &outlen);
297
298   double Shift;
299   double Scale;
300   int IconDataScalarType;
301   int IconNumberOfScalarComponents;
302   int PlanarConfiguration;
303   int LossyFlag;
304   int ForceRescale;
305
306 protected:
307   // TODO / FIXME
308   void SetFilePrefix(const char *) {}
309   vtkGetStringMacro(FilePrefix);
310   void SetFilePattern(const char *) {}
311   vtkGetStringMacro(FilePattern);
312
313 private:
314   vtkGDCMImageReader(const vtkGDCMImageReader&);  // Not implemented.
315   void operator=(const vtkGDCMImageReader&);  // Not implemented.
316 };
317 #endif
```

11.605 vtkGDCMImageReader2.h File Reference

```
#include "vtkMedicalImageReader2.h"
#include "vtkImageData.h"
```

Include dependency graph for vtkGDCMImageReader2.h:



Classes

- class vtkGDCMImageReader2

Namespaces

- namespace gdcm

Macros

- #define VTK_CMYK 8
- #define VTK_INVERSE_LUMINANCE 5
- #define VTK_LOOKUP_TABLE 6
- #define VTK_YBR 7

11.605.1 Macro Definition Documentation

11.605.1.1 VTK_CMYK

```
#define VTK_CMYK 8
```

11.605.1.2 VTK_INVERSE_LUMINANCE

```
#define VTK_INVERSE_LUMINANCE 5
```

11.605.1.3 VTK_LOOKUP_TABLE

```
#define VTK_LOOKUP_TABLE 6
```

11.605.1.4 VTK_YBR

```
#define VTK_YBR 7
```

11.606 vtkGDCMImageReader2.h

Go to the documentation of this file.

```
1 /*=========================================================================
2
3 Program:  GDCM (Grassroots DICOM). A DICOM library
4
5 Copyright (c) 2006-2011 Mathieu Malaterre
6 All rights reserved.
7 See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.
8
9 This software is distributed WITHOUT ANY WARRANTY; without even
10 the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
11 PURPOSE.  See the above copyright notice for more information.
12
13 =========================================================================*/
14 // .NAME vtkGDCMImageReader2 - read DICOM Image files (Pixel Data)
15 // .SECTION Description
16 // vtkGDCMImageReader2 is a source object that reads some DICOM files
17 // this reader is single threaded.
18 // .SECTION Implementation note:  when FileLowerLeft is set to on the image is not flipped
19 // upside down as VTK would expect, use this option only if you know what you are doing.
20 // .SECTION Implementation note:  when reading a series of 2D slices, user is
21 // expected to provide an ordered list of filenames.  No sorting will be applied afterward.
22 // .SECTION Implementation note:  Although 99% of the time the Zspacing as read
23 // from a tag in a 2D DICOM file should be correct, there has been reports that this
24 // value can be missing, or incorrect, in which case users are advised to override this
25 // value using the return value from gdcm::IPPSorter::GetZSpacing() and set it via
26 // vtkImageChangeInformation on the reader itself.
27 // .SECTION TODO
28 // This reader does not handle a series of 3D images, only a single 3D (multi frame) or a
29 // list of 2D files are supported for now.
30 // .SECTION TODO
31 // Did not implement SetFilePattern / SetFilePrefix API, move it to protected section for now.
32 // .SECTION BUG
33 // Overlay are assumed to have the same extent as image.  Right now if overlay origin is not
34 // 0,0 the overlay will have an offset...
35 // Only the very first overlay is loaded at the VTK level, for now (even if there are more than one in the
      file)
36 // .SECTION DataOrigin
37 // When the reader is instantiated with FileLowerLeftOn the DataOrigin and Image Position (Patient) are
38 // identical.  But when FileLowerLeft is Off, we have to reorder the Y-line of the image, and thus the
      DataOrigin
39 // is then translated to the other side of the image.
40 // .SECTION Spacing
41 // When reading a 3D volume, the spacing along the Z dimension might be negative (so as to respect
      up-side-down)
42 // as specified in the Image Orientation (Patient) tag.  When Z-spacing is 0, this means the multi-frame
      object
43 // contains image which do not represent uniform volume.
44 // .SECTION Warning
45 // When using vtkGDCMPolyDataReader in conjonction with vtkGDCMImageReader2
46 // it is *required* that FileLowerLeft is set to ON as coordinate system
47 // would be inconsistent in between the two data structures.
48 // .SECTION Color Space mapping:
49 // * VTK_LUMINANCE         <-> MONOCHROME2
50 // * VTK_LUMINANCE_ALPHA   <-> Not supported
```

```
51 // * VTK_RGB               <-> RGB
52 // * VTK_RGBA              <-> ARGB (deprecated, DICOM 2008)
53 // * VTK_INVERSE_LUMINANCE <-> MONOCHROME1
54 // * VTK_LOOKUP_TABLE      <-> PALETTE COLOR
55 // * VTK_YBR               <-> YBR_FULL
56 //
57 // For detailed information on color space transformation and true lossless transformation see:
58 // http://gdcm.sourceforge.net/wiki/index.php/Color_Space_Transformations
59
60 // .SECTION See Also
61 // vtkMedicalImageReader2 vtkMedicalImageProperties vtkGDCMPolyDataReader vtkGDCMImageWriter
62 // vtkDICOMImageReader
63
64 #ifndef VTKGDCMIMAGEREADER2_H
65 #define VTKGDCMIMAGEREADER2_H
66
67 #include "vtkMedicalImageReader2.h"
68 #include "vtkImageData.h"
69
70 class vtkPolyData;
71
72 // vtkSystemIncludes.h defines:
73 // #define VTK_LUMINANCE       1
74 // #define VTK_LUMINANCE_ALPHA 2
75 // #define VTK_RGB             3
76 // #define VTK_RGBA            4
77 #ifndef VTK_INVERSE_LUMINANCE
78 #define VTK_INVERSE_LUMINANCE 5
79 #endif
80 #ifndef VTK_LOOKUP_TABLE
81 #define VTK_LOOKUP_TABLE 6
82 #endif
83 #ifndef VTK_YBR
84 #define VTK_YBR 7
85 #endif
86 #ifndef VTK_CMYK
87 #define VTK_CMYK 8
88 #endif
89
90 //BTX
91 namespace gdcm { class ImageReader; }
92 //ETX
93 class vtkMatrix4x4;
94 class VTK_EXPORT vtkGDCMImageReader2 :  public vtkMedicalImageReader2
95 {
96 public:
97   static vtkGDCMImageReader2 *New();
98   vtkTypeMacro(vtkGDCMImageReader2,vtkMedicalImageReader2);
99   virtual void PrintSelf(ostream& os, vtkIndent indent);
100
101   // Description:  is the given file name a DICOM file containing an image ?
102   virtual int CanReadFile(const char* fname);
103
104   // Description:
105   // Valid extensions
106   virtual const char* GetFileExtensions()
107     {
108     // I would like to get rid of ACR/NEMA/IMA so only allow dcm extension for now
109     return ".dcm .DCM";
110     }
111
112   // Description:
113   // A descriptive name for this format
114   virtual const char* GetDescriptiveName()
115     {
116     return "DICOM";
117     }
118
119   // Description:
120   // Get the Image Position (Patient) as stored in the DICOM file
121   // This is a read-only data member
122   vtkGetObjectMacro(DirectionCosines, vtkMatrix4x4);
123
124   virtual void SetMedicalImageProperties(vtkMedicalImageProperties *pd);
125
126   // Description:
127   // Specifically request to load the overlay into the gdcm-VTK layer (gdcm always loads them when found).
128   // If no overlay is found in the image, then the vtkImageData for the overlay will be empty.
129   vtkGetMacro(LoadOverlays,int);
130   vtkSetMacro(LoadOverlays,int);
131   vtkBooleanMacro(LoadOverlays,int);
```

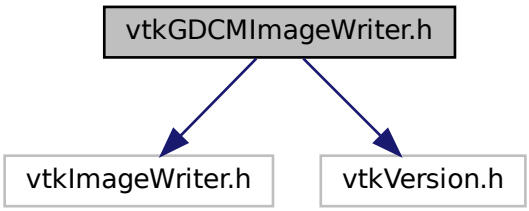
```
132
133   // Description:
134   // Set/Get whether or not to load the Icon as vtkImageData (if found in the DICOM file)
135   vtkGetMacro(LoadIconImage,int);
136   vtkSetMacro(LoadIconImage,int);
137   vtkBooleanMacro(LoadIconImage,int);
138
139   // Description:
140   // Set/Get whether or not the image was compressed using a lossy compression algorithm
141   vtkGetMacro(LossyFlag,int);
142   vtkSetMacro(LossyFlag,int);
143   vtkBooleanMacro(LossyFlag,int);
144
145   // Description:
146   // Read only:  number of overlays as found in this image (multiple overlays per slice is allowed)
147   // Only valid when LoadOverlays is true
148   vtkGetMacro(NumberOfOverlays,int);
149
150   // Description:
151   // Read only:  number of icon image (there can only be zero or one icon per file)
152   // Only valid when LoadIconImage is true
153   vtkGetMacro(NumberOfIconImages,int);
154
155   // Description:
156   // Get Overlay/IconImage
157   // Remember to ALWAYS use those methods in your code, as the internal number for the output port
158   // is not garantee to remain the same, as features are added to the reader
159   vtkAlgorithmOutput* GetOverlayPort(int index);
160   vtkAlgorithmOutput* GetIconImagePort();
161   vtkImageData* GetOverlay(int i);
162   vtkImageData* GetIconImage();
163
164   // Description:
165   // Load image with its associated Lookup Table
166   vtkGetMacro(ApplyLookupTable,int);
167   vtkSetMacro(ApplyLookupTable,int);
168   vtkBooleanMacro(ApplyLookupTable,int);
169
170   // Description:
171   // Load image as YBR
172   vtkGetMacro(ApplyYBRToRGB,int)
173   vtkSetMacro(ApplyYBRToRGB,int)
174   vtkBooleanMacro(ApplyYBRToRGB,int);
175
176   // Description:
177   // Return VTK_LUMINANCE, VTK_INVERSE_LUMINANCE, VTK_RGB, VTK_RGBA, VTK_LOOKUP_TABLE, VTK_YBR or VTK_CMYK
178   // or 0 when ImageFormat is not handled.
179   // Warning:  For color image, PlanarConfiguration need to be taken into account.
180   vtkGetMacro(ImageFormat,int);
181
182   // Description:
183   // Return the Planar Configuration.  This simply means that the internal DICOM image was stored
184   // using a particular planar configuration (most of the time:  0)
185   // For monochrome image, PlanarConfiguration is always 0
186   vtkGetMacro(PlanarConfiguration,int);
187
188   // Description:
189   // Return the 'raw' information stored in the DICOM file:
190   // In case of a series of multiple files, only the first file is considered.  The Image Orientation
      (Patient)
191   // is garantee to remain the same, and image Image Position (Patient) in other slice can be computed
192   // using the ZSpacing (3rd dimension)
193   // (0020,0032) DS [87.774866\-182.908510\168.629671]       #  32, 3 ImagePositionPatient
194   // (0020,0037) DS [0.001479\0.999989\-0.004376\-0.002039\-0.004372\-0.999988] #  58, 6
      ImageOrientationPatient
195   vtkGetVector3Macro(ImagePositionPatient,double);
196   vtkGetVector6Macro(ImageOrientationPatient,double);
197
198   // Description:
199   // Set/Get the first Curve Data:
200   vtkGetObjectMacro(Curve,vtkPolyData);
201   virtual void SetCurve(vtkPolyData *pd);
202
203   // Description:
204   // \DEPRECATED:
205   // Modality LUT
206   // Value returned by GetShift/GetScale might be inacurate since Shift/Scale could be
207   // varying along the Series read.  Therefore user are advices not to use those functions
208   // anymore
209   vtkGetMacro(Shift,double);
210   vtkGetMacro(Scale,double);
```

```
211
212 protected:
213   vtkGDCMImageReader2();
214   ~vtkGDCMImageReader2();
215
216   vtkSetVector6Macro(ImageOrientationPatient,double);
217
218 //BTX
219   void FillMedicalImageInformation(const gdcm::ImageReader &reader);
220 //ETX
221   int RequestInformationCompat();
222   int RequestDataCompat();
223
224   int ProcessRequest(vtkInformation* request,
225                                vtkInformationVector** inputVector,
226                                vtkInformationVector* outputVector);
227   int RequestInformation(vtkInformation *request,
228                          vtkInformationVector **inputVector,
229                          vtkInformationVector *outputVector);
230   int RequestData(vtkInformation *request,
231                   vtkInformationVector **inputVector,
232                   vtkInformationVector *outputVector);
233
234 protected:
235   vtkMatrix4x4 *DirectionCosines;
236   int LoadOverlays;
237   int NumberOfOverlays;
238   int LoadIconImage;
239   int NumberOfIconImages;
240   int IconImageDataExtent[6];
241   double ImagePositionPatient[3];
242   double ImageOrientationPatient[6];
243   vtkPolyData *Curve;
244
245   int ImageFormat;
246   // the following 3, should remain optional
247   int ApplyInverseVideo;
248   int ApplyLookupTable;
249   int ApplyYBRToRGB;
250   // I think that planar configuration need to always be applied as far as VTK is concerned
251   int ApplyPlanarConfiguration;
252   int ApplyShiftScale;
253
254   int LoadSingleFile(const char *filename, char *pointer, unsigned long &outlen);
255
256   double Shift;
257   double Scale;
258   int IconDataScalarType;
259   int IconNumberOfScalarComponents;
260   int PlanarConfiguration;
261   int LossyFlag;
262   int ForceRescale;
263
264 protected:
265   // TODO / FIXME
266   void SetFilePrefix(const char *) {}
267   vtkGetStringMacro(FilePrefix);
268   void SetFilePattern(const char *) {}
269   vtkGetStringMacro(FilePattern);
270
271 private:
272   vtkGDCMImageReader2(const vtkGDCMImageReader2&);  // Not implemented.
273   void operator=(const vtkGDCMImageReader2&);  // Not implemented.
274 };
275 #endif
```

11.607 vtkGDCMImageWriter.h File Reference

```
#include "vtkImageWriter.h"
#include "vtkVersion.h"
```

Include dependency graph for vtkGDCMImageWriter.h:



Classes

- class vtkGDCMImageWriter

11.608 vtkGDCMImageWriter.h

Go to the documentation of this file.

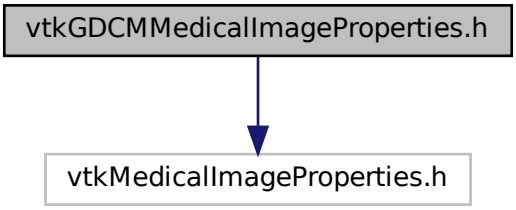
```
1 /*=========================================================================
2
3 Program:  GDCM (Grassroots DICOM). A DICOM library
4
5 Copyright (c) 2006-2011 Mathieu Malaterre
6 All rights reserved.
7 See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.
8
9 This software is distributed WITHOUT ANY WARRANTY; without even
10 the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
11 PURPOSE.  See the above copyright notice for more information.
12
13 =========================================================================*/
14 // .NAME vtkGDCMImageWriter - write DICOM files
15 // .SECTION Description
16 // vtkGDCMImageWriter is a sink object that write DICOM files
17 // this writer is single threaded (see vtkGDCMThreadedImageReader2 for multi-thread)
18 //
19 // .SECTION Warning:  vtkLookupTable from the vtkImageData object taken into account
20 // only if ImageFormat is set to VTK_LOOKUP_TABLE
21 //
22 // .SECTION NOTE We are not using the usual API SetFilePrefix / SetFilePattern,
23 // but instead a list of filenames:  see SetFileNames and class gdcm::FilenameGenerator
24 //
25 // .SECTION Warning
26 // You need to specify the correct ImageFormat (taken from the reader)
27 // You need to explicitly specify the DirectionCosines (taken from the reader)
28 // Since VTK 5.4 vtkMedicalImageProperties has its own DirectionCosine (no 's')
29 // user need to make sure the vtkMatrix4x4 is compatible with the 6-vector DirectionCosine.
30 //
31 // .SECTION NOTE Shift/Scale are global to all DICOM frames (=files) written
32 // as 2D slice, therefore the shift/scale operation might not be optimized for
33 // all slices.  This is not recommended for image with a large dynamic range.
34 //
35 // .SECTION See Also
36 // vtkImageWriter vtkMedicalImageProperties vtkGDCMImageReader
37
38 #ifndef VTKGDCMIMAGEWRITER_H
39 #define VTKGDCMIMAGEWRITER_H
```

```
40
41 #include "vtkImageWriter.h"
42 #include "vtkVersion.h"
43
44 class vtkLookupTable;
45 class vtkMedicalImageProperties;
46 class vtkMatrix4x4;
47 class vtkStringArray;
48 class VTK_EXPORT vtkGDCMImageWriter :  public vtkImageWriter
49 {
50 public:
51   static vtkGDCMImageWriter *New();
52   vtkTypeMacro(vtkGDCMImageWriter,vtkImageWriter);
53   virtual void PrintSelf(ostream& os, vtkIndent indent);
54
55   // Description:
56   // Pass in the vtkmedicalimageproperties object for medical information
57   // to be mapped to DICOM attributes.
58   vtkGetObjectMacro(MedicalImageProperties, vtkMedicalImageProperties);
59   virtual void SetMedicalImageProperties(vtkMedicalImageProperties*);
60
61   // Description:
62   // Pass in the list of filename to be used to write out the DICOM file(s)
63   virtual void SetFileNames(vtkStringArray*);
64   vtkGetObjectMacro(FileNames, vtkStringArray);
65
66   // Description:
67   // Set/Get whether or not the image was compressed using a lossy compression algorithm
68   vtkGetMacro(LossyFlag,int);
69   vtkSetMacro(LossyFlag,int);
70   vtkBooleanMacro(LossyFlag,int);
71
72   // I need that...
73   virtual void Write();
74
75   // Description:
76   // Get the entension for this file format.
77   virtual const char* GetFileExtensions() {
78     return ".dcm .DCM"; }
79
80   // Description:
81   // Get the name of this file format.
82   virtual const char* GetDescriptiveName() {
83     return "DICOM"; }
84
85   // Description:
86   // You need to manually specify the direction the image is in to write a valid DICOM file
87   // since vtkImageData do not contains one (eg.  MR Image Storage, CT Image Storage...)
88   virtual void SetDirectionCosines(vtkMatrix4x4 *matrix);
89   vtkGetObjectMacro(DirectionCosines, vtkMatrix4x4);
90   virtual void SetDirectionCosinesFromImageOrientationPatient(const double dircos[6]);
91
92   // Description:
93   // Modality LUT
94   vtkSetMacro(Shift, double);
95   vtkGetMacro(Shift, double);
96   vtkSetMacro(Scale, double);
97   vtkGetMacro(Scale, double);
98
99   // Description:
100   // See vtkGDCMImageReader for list of ImageFormat
101   vtkGetMacro(ImageFormat,int);
102   vtkSetMacro(ImageFormat,int);
103
104   // Description:
105   // Set/Get whether the data comes from the file starting in the lower left
106   // corner or upper left corner.
107   vtkBooleanMacro(FileLowerLeft, int);
108   vtkGetMacro(FileLowerLeft, int);
109   vtkSetMacro(FileLowerLeft, int);
110
111   // Description:
112   // For color image (more than a single comp) you can specify the planar configuration you prefer
113   vtkSetMacro(PlanarConfiguration,int);
114   vtkGetMacro(PlanarConfiguration,int);
115
116   // Description:
117   // Set/Get specific StudyUID / SeriesUID
118   vtkSetStringMacro(StudyUID);
119   vtkGetStringMacro(StudyUID);
120   vtkSetStringMacro(SeriesUID);
```

```
121   vtkGetStringMacro(SeriesUID);
122
123 //BTX
124   enum CompressionTypes {
125     NO_COMPRESSION = 0,   // raw (default)
126     JPEG_COMPRESSION,     // JPEG
127     JPEG2000_COMPRESSION, // J2K
128     JPEGLS_COMPRESSION,   // JPEG-LS
129     RLE_COMPRESSION       // RLE
130   };
131 //ETX
132   // Set/Get the compression type
133   vtkSetMacro(CompressionType, int);
134   vtkGetMacro(CompressionType, int);
135
136   //void SetCompressionTypeFromString(const char *);
137   //const char *GetCompressionTypeAsString();
138
139 protected:
140   vtkGDCMImageWriter();
141   ~vtkGDCMImageWriter();
142
143 #if (VTK_MAJOR_VERSION >= 5) || ( VTK_MAJOR_VERSION == 4 && VTK_MINOR_VERSION > 5 )
144   int FillInputPortInformation(int port, vtkInformation *info);
145   int RequestInformation(
146     vtkInformation *request,
147     vtkInformationVector **inputVector,
148     vtkInformationVector *outputVector);
149   int RequestUpdateExtent(
150     vtkInformation *request,
151     vtkInformationVector **inputVector,
152     vtkInformationVector *outputVector);
153   int RequestData(
154     vtkInformation *request,
155     vtkInformationVector **inputVector,
156     vtkInformationVector *outputVector);
157 #else
158   void WriteSlice(vtkImageData *data);
159 #endif /*(VTK_MAJOR_VERSION >= 5) || ( VTK_MAJOR_VERSION == 4 && VTK_MINOR_VERSION > 5 )*/
160   int WriteGDCMData(vtkImageData *data, int timeStep);
161
162 protected:
163   virtual /*const*/ char *GetFileName();
164
165 private:
166   vtkGDCMImageWriter(const vtkGDCMImageWriter&);  // Not implemented.
167   void operator=(const vtkGDCMImageWriter&);  // Not implemented.
168
169   // VTK structs:
170   //vtkLookupTable *LookupTable;
171   vtkMedicalImageProperties *MedicalImageProperties;
172   char *StudyUID;
173   char *SeriesUID;
174
175   int DataUpdateExtent[6];
176   int ImageFormat;
177
178   vtkStringArray *FileNames;
179   vtkMatrix4x4 *DirectionCosines;
180
181   double Shift;
182   double Scale;
183   int FileLowerLeft;
184   int PlanarConfiguration;
185   int LossyFlag;
186   int CompressionType;
187 };
188
189 #endif
```

11.609 vtkGDCMMedicalImageProperties.h File Reference

`#include "vtkMedicalImageProperties.h"`
Include dependency graph for vtkGDCMMedicalImageProperties.h:



Classes

- class vtkGDCMMedicalImageProperties

Namespaces

- namespace gdcm

11.610 vtkGDCMMedicalImageProperties.h

Go to the documentation of this file.

```
1 /*=========================================================================
2
3 Program:  GDCM (Grassroots DICOM). A DICOM library
4
5 Copyright (c) 2006-2011 Mathieu Malaterre
6 All rights reserved.
7 See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.
8
9 This software is distributed WITHOUT ANY WARRANTY; without even
10 the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
11 PURPOSE.  See the above copyright notice for more information.
12
13 =========================================================================*/
14 // .NAME vtkGDCMMedicalImageProperties - some medical image properties.
15 // .SECTION Description
16 // vtkGDCMMedicalImageProperties is a helper class that can be used by medical
17 // image readers and applications to encapsulate medical image/acquisition
18 // properties.  Later on, this should probably be extended to add
19 // any user-defined property.
20 // .SECTION See Also
21 // vtkMedicalImageReader2
22
23 #ifndef VTKGDCMMEDICALIMAGEPROPERTIES_H
24 #define VTKGDCMMEDICALIMAGEPROPERTIES_H
25
```

```
26 #include "vtkMedicalImageProperties.h"
27
28 class vtkGDCMMedicalImagePropertiesInternals;
29 //BTX
30 namespace gdcm { class File; }
31 //ETX
32
33 class VTK_EXPORT vtkGDCMMedicalImageProperties :  public vtkMedicalImageProperties
34 {
35 public:
36   static vtkGDCMMedicalImageProperties *New();
37   vtkTypeMacro(vtkGDCMMedicalImageProperties,vtkMedicalImageProperties);
38   void PrintSelf(ostream& os, vtkIndent indent);
39
40   // Description:
41   // Convenience method to reset all fields to an empty string/value
42   virtual void Clear();
43
44 /*
45 // Description:
46 // Patient name
47 // For ex:  DICOM (0010,0010) = DOE,JOHN
48 vtkSetStringMacro(PatientName);
49 vtkGetStringMacro(PatientName);
50
51 // Description:
52 // Patient ID
53 // For ex:  DICOM (0010,0020) = 1933197
54 vtkSetStringMacro(PatientID);
55 vtkGetStringMacro(PatientID);
56
57 // Description:
58 // Patient age
59 // Format:  nnnD, nnW, nnnM or nnnY (eventually nnD, nnW, nnY)
60 //          with D (day), M (month), W (week), Y (year)
61 // For ex:  DICOM (0010,1010) = 031Y
62 vtkSetStringMacro(PatientAge);
63 vtkGetStringMacro(PatientAge);
64
65 // Description:
66 // Take as input a string in VR=AS (DICOM PS3.5) and extract either
67 // different fields namely:  year month week day
68 // Return 0 on error, 1 on success
69 // One can test fields if they are different from -1 upon success
70 static int GetAgeAsFields(const char *age, int &year, int &month, int &week, int &day);
71
72 // For Tcl:
73 // From C++ use GetPatientAge + GetAgeAsField
74 // Those function parse a DICOM string, and return the value of the number expressed
75 // this is either expressed in year, month or days.  Thus if a string is expressed in years
76 // GetPatientAgeDay/GetPatientAgeWeek/GetPatientAgeMonth will return 0
77 int GetPatientAgeYear();
78 int GetPatientAgeMonth();
79 int GetPatientAgeWeek();
80 int GetPatientAgeDay();
81
82 // Description:
83 // Patient sex
84 // For ex:  DICOM (0010,0040) = M
85 vtkSetStringMacro(PatientSex);
86 vtkGetStringMacro(PatientSex);
87
88 // Description:
89 // Patient birth date
90 // Format:  yyyymmdd
91 // For ex:  DICOM (0010,0030) = 19680427
92 vtkSetStringMacro(PatientBirthDate);
93 vtkGetStringMacro(PatientBirthDate);
94
95 // For Tcl:
96 // From C++ use GetPatientBirthDate + GetDateAsFields
97 int GetPatientBirthDateYear();
98 int GetPatientBirthDateMonth();
99 int GetPatientBirthDateDay();
100
101 // Description:
102 // Study Date
103 // Format:  yyyymmdd
104 // For ex:  DICOM (0008,0020) = 20030617
105 vtkSetStringMacro(StudyDate);
106 vtkGetStringMacro(StudyDate);
```

```
107
108 // Description:
109 // Acquisition Date
110 // Format:  yyyymmdd
111 // For ex:  DICOM (0008,0022) = 20030617
112 vtkSetStringMacro(AcquisitionDate);
113 vtkGetStringMacro(AcquisitionDate);
114
115 // For Tcl:
116 // From C++ use GetAcquisitionDate + GetDateAsFields
117 int GetAcquisitionDateYear();
118 int GetAcquisitionDateMonth();
119 int GetAcquisitionDateDay();
120
121 // Description:
122 // Study Time
123 // Format:  hhmmss.frac (any trailing component(s) can be ommited)
124 // For ex:  DICOM (0008,0030) = 162552.0705 or 230012, or 0012
125 vtkSetStringMacro(StudyTime);
126 vtkGetStringMacro(StudyTime);
127
128 // Description:
129 // Acquisition time
130 // Format:  hhmmss.frac (any trailing component(s) can be ommited)
131 // For ex:  DICOM (0008,0032) = 162552.0705 or 230012, or 0012
132 vtkSetStringMacro(AcquisitionTime);
133 vtkGetStringMacro(AcquisitionTime);
134
135 // Description:
136 // Image Date aka Content Date
137 // Format:  yyyymmdd
138 // For ex:  DICOM (0008,0023) = 20030617
139 vtkSetStringMacro(ImageDate);
140 vtkGetStringMacro(ImageDate);
141
142 // For Tcl:
143 // From C++ use GetImageDate + GetDateAsFields
144 int GetImageDateYear();
145 int GetImageDateMonth();
146 int GetImageDateDay();
147
148 // Description:
149 // Take as input a string in ISO 8601 date (YYYY/MM/DD) and extract the
150 // different fields namely:  year month day
151 // Return 0 on error, 1 on success
152 static int GetDateAsFields(const char *date, int &year, int &month, int &day);
153
154 // Description:
155 // Take as input a string in ISO 8601 date (YYYY/MM/DD) and construct a
156 // locale date based on the different fields (see GetDateAsFields to extract
157 // different fields)
158 // Return 0 on error, 1 on success
159 static int GetDateAsLocale(const char *date, char *locale);
160
161 // Description:
162 // Image Time
163 // Format:  hhmmss.frac (any trailing component(s) can be ommited)
164 // For ex:  DICOM (0008,0033) = 162552.0705 or 230012, or 0012
165 vtkSetStringMacro(ImageTime);
166 vtkGetStringMacro(ImageTime);
167
168 // Description:
169 // Image number
170 // For ex:  DICOM (0020,0013) = 1
171 vtkSetStringMacro(ImageNumber);
172 vtkGetStringMacro(ImageNumber);
173
174 // Description:
175 // Series number
176 // For ex:  DICOM (0020,0011) = 902
177 vtkSetStringMacro(SeriesNumber);
178 vtkGetStringMacro(SeriesNumber);
179
180 // Description:
181 // Series Description
182 // User provided description of the Series
183 // For ex:  DICOM (0008,103e) = SCOUT
184 vtkSetStringMacro(SeriesDescription);
185 vtkGetStringMacro(SeriesDescription);
186
187 // Description:
```

```
188 // Study ID
189 // For ex:  DICOM (0020,0010) = 37481
190 vtkSetStringMacro(StudyID);
191 vtkGetStringMacro(StudyID);
192
193 // Description:
194 // Study description
195 // For ex:  DICOM (0008,1030) = BRAIN/C-SP/FACIAL
196 vtkSetStringMacro(StudyDescription);
197 vtkGetStringMacro(StudyDescription);
198
199 // Description:
200 // Modality
201 // For ex:  DICOM (0008,0060)= CT
202 vtkSetStringMacro(Modality);
203 vtkGetStringMacro(Modality);
204
205 // Description:
206 // Manufacturer
207 // For ex:  DICOM (0008,0070) = Siemens
208 vtkSetStringMacro(Manufacturer);
209 vtkGetStringMacro(Manufacturer);
210
211 // Description:
212 // Manufacturer's Model Name
213 // For ex:  DICOM (0008,1090) = LightSpeed QX/i
214 vtkSetStringMacro(ManufacturerModelName);
215 vtkGetStringMacro(ManufacturerModelName);
216
217 // Description:
218 // Station Name
219 // For ex:  DICOM (0008,1010) = LSPD_OC8
220 vtkSetStringMacro(StationName);
221 vtkGetStringMacro(StationName);
222
223 // Description:
224 // Institution Name
225 // For ex:  DICOM (0008,0080) = FooCity Medical Center
226 vtkSetStringMacro(InstitutionName);
227 vtkGetStringMacro(InstitutionName);
228
229 // Description:
230 // Convolution Kernel (or algorithm used to reconstruct the data)
231 // For ex:  DICOM (0018,1210) = Bone
232 vtkSetStringMacro(ConvolutionKernel);
233 vtkGetStringMacro(ConvolutionKernel);
234
235 // Description:
236 // Slice Thickness (Nominal reconstructed slice thickness, in mm)
237 // For ex:  DICOM (0018,0050) = 0.273438
238 vtkSetStringMacro(SliceThickness);
239 vtkGetStringMacro(SliceThickness);
240 virtual double GetSliceThicknessAsDouble();
241
242 // Description:
243 // Peak kilo voltage output of the (x-ray) generator used
244 // For ex:  DICOM (0018,0060) = 120
245 vtkSetStringMacro(KVP);
246 vtkGetStringMacro(KVP);
247
248 // Description:
249 // Gantry/Detector tilt (Nominal angle of tilt in degrees of the scanning
250 // gantry.)
251 // For ex:  DICOM (0018,1120) = 15
252 vtkSetStringMacro(GantryTilt);
253 vtkGetStringMacro(GantryTilt);
254 virtual double GetGantryTiltAsDouble();
255
256 // Description:
257 // Echo Time
258 // (Time in ms between the middle of the excitation pulse and the peak of
259 // the echo produced)
260 // For ex:  DICOM (0018,0081) = 105
261 vtkSetStringMacro(EchoTime);
262 vtkGetStringMacro(EchoTime);
263
264 // Description:
265 // Echo Train Length
266 // (Number of lines in k-space acquired per excitation per image)
267 // For ex:  DICOM (0018,0091) = 35
268 vtkSetStringMacro(EchoTrainLength);
```

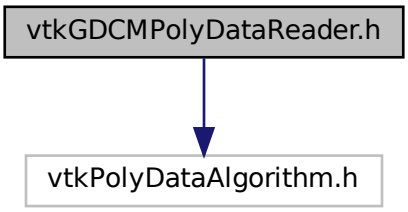
```
269 vtkGetStringMacro(EchoTrainLength);
270
271 // Description:
272 // Repetition Time
273 // The period of time in msec between the beginning of a pulse sequence and
274 // the beginning of the succeeding (essentially identical) pulse sequence.
275 // For ex:  DICOM (0018,0080) = 2040
276 vtkSetStringMacro(RepetitionTime);
277 vtkGetStringMacro(RepetitionTime);
278
279 // Description:
280 // Exposure time (time of x-ray exposure in msec)
281 // For ex:  DICOM (0018,1150) = 5
282 vtkSetStringMacro(ExposureTime);
283 vtkGetStringMacro(ExposureTime);
284
285 // Description:
286 // X-ray tube current (in mA)
287 // For ex:  DICOM (0018,1151) = 400
288 vtkSetStringMacro(XRayTubeCurrent);
289 vtkGetStringMacro(XRayTubeCurrent);
290
291 // Description:
292 // Exposure (The exposure expressed in mAs, for example calculated
293 // from Exposure Time and X-ray Tube Current)
294 // For ex:  DICOM (0018,1152) = 114
295 vtkSetStringMacro(Exposure);
296 vtkGetStringMacro(Exposure);
297
298 // Interface to allow insertion of user define values, for instance in DICOM one would want to
299 // store the Protocol Name (0018,1030), in this case one would do:
300 // AddUserDefinedValue( "Protocol Name", "T1W/SE/1024" );
301 void AddUserDefinedValue(const char *name, const char *value);
302 // Get a particular user value
303 const char *GetUserDefinedValue(const char *name);
304 // Get the number of user defined values
305 unsigned int GetNumberOfUserDefinedValues();
306 // Get a name/value by index
307 const char *GetUserDefinedNameByIndex(unsigned int idx);
308 const char *GetUserDefinedValueByIndex(unsigned int idx);
309
310 // Description:
311 // Copy the contents of p to this instance.
312 virtual void DeepCopy(vtkGDCMMedicalImageProperties *p);
313
314 // Description:
315 // Add/Remove/Query the window/level presets that may have been associated
316 // to a medical image.  Window is also known as 'width', level is also known
317 // as 'center'.  The same window/level pair can not be added twice.
318 // As a convenience, a comment (aka Explanation) can be associated to a preset.
319 // For ex:  DICOM Window Center (0028,1050) = 00045\000470
320 //          DICOM Window Width  (0028,1051) = 0106\03412
321 //          DICOM Window Center Width Explanation (0028,1055) = WINDOW1\WINDOW2
322 virtual void AddWindowLevelPreset(double w, double l);
323 virtual void RemoveWindowLevelPreset(double w, double l);
324 virtual void RemoveAllWindowLevelPresets();
325 virtual int GetNumberOfWindowLevelPresets();
326 virtual int HasWindowLevelPreset(double w, double l);
327 virtual int GetNthWindowLevelPreset(int idx, double *w, double *l);
328 virtual double* GetNthWindowLevelPreset(int idx);
329 virtual void SetNthWindowLevelPresetComment(int idx, const char *comment);
330 virtual const char* GetNthWindowLevelPresetComment(int idx);
331
332 // Description:
333 // Mapping from a sliceidx within a volumeidx into a DICOM Instance UID
334 // Some DICOM reader can populate this structure so that later on from a slice index
335 // in a vtkImageData volume we can backtrack and find out which 2d slice it was coming from
336 const char *GetInstanceUIDFromSliceID(int volumeidx, int sliceid);
337 void SetInstanceUIDFromSliceID(int volumeidx, int sliceid, const char *uid);
338
339 // Description:
340 // Provides the inverse mapping.  Returns -1 if a slice for this uid is
341 // not found.
342 int GetSliceIDFromInstanceUID(int &volumeidx, const char *uid);
343
344 //BTX
345 typedef enum {
346 AXIAL = 0,
347 CORONAL,
348 SAGITTAL
349 } OrientationType;
```

```
350 //ETX
351 int GetOrientationType(int volumeidx);
352 void SetOrientationType(int volumeidx, int orientation);
353 static const char *GetStringFromOrientationType(unsigned int type);
354 */
355 protected:
356   vtkGDCMMedicalImageProperties();
357   ~vtkGDCMMedicalImageProperties();
358
359 //BTX
360   friend class vtkGDCMImageReader;
361   friend class vtkGDCMImageReader2;
362   friend class vtkGDCMImageWriter;
363   void PushBackFile(gdcm::File const &f);
364   gdcm::File const & GetFile(unsigned int t);
365 //ETX
366
367 private:
368   vtkGDCMMedicalImagePropertiesInternals *Internals;
369
370   vtkGDCMMedicalImageProperties(const vtkGDCMMedicalImageProperties&); // Not implemented.
371   void operator=(const vtkGDCMMedicalImageProperties&); // Not implemented.
372 };
373
374 #endif
```

11.611 vtkGDCMPolyDataReader.h File Reference

`#include "vtkPolyDataAlgorithm.h"`

Include dependency graph for vtkGDCMPolyDataReader.h:



Classes

- class vtkGDCMPolyDataReader

Namespaces

- namespace gdcm

11.612 vtkGDCMPolyDataReader.h

Go to the documentation of this file.

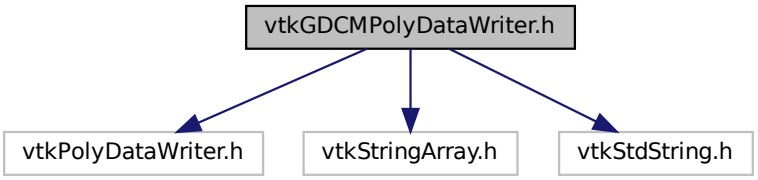
```
1 /*=========================================================================
2
3 Program:  GDCM (Grassroots DICOM). A DICOM library
4
5 Copyright (c) 2006-2011 Mathieu Malaterre
6 All rights reserved.
7 See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.
8
9 This software is distributed WITHOUT ANY WARRANTY; without even
10 the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
11 PURPOSE.  See the above copyright notice for more information.
12
13 =========================================================================*/
14 // .NAME vtkGDCMPolyDataReader - read DICOM PolyData files (Contour Data...)
15 // .SECTION Description
16 // For now only support RTSTRUCT (RT Structure Set Storage)
17 // .SECTION TODO
18 // Need to do the same job for DVH Sequence/DVH Data...
19 // .SECTION Warning
20 // When using vtkGDCMPolyDataReader in conjonction with vtkGDCMImageReader
21 // it is *required* that FileLowerLeft is set to ON as coordinate system
22 // would be inconsistent in between the two data structures.
23 //
24 // .SECTION See Also
25 // vtkGDCMImageReader vtkGDCMPolyDataWriter vtkRTStructSetProperties
26
27
28 #ifndef VTKGDCMPOLYDATAREADER_H
29 #define VTKGDCMPOLYDATAREADER_H
30
31 #include "vtkPolyDataAlgorithm.h"
32
33 class vtkMedicalImageProperties;
34 class vtkRTStructSetProperties;
35 //BTX
36 namespace gdcm { class Reader; }
37 //ETX
38 class VTK_EXPORT vtkGDCMPolyDataReader :  public vtkPolyDataAlgorithm
39 {
40 public:
41   static vtkGDCMPolyDataReader *New();
42   vtkTypeMacro(vtkGDCMPolyDataReader,vtkPolyDataAlgorithm);
43   virtual void PrintSelf(ostream& os, vtkIndent indent);
44
45   // Description:
46   // Set/Get the filename of the file to be read
47   vtkSetStringMacro(FileName);
48   vtkGetStringMacro(FileName);
49
50   // Description:
51   // Get the medical image properties object
52   vtkGetObjectMacro(MedicalImageProperties, vtkMedicalImageProperties);
53
54   vtkGetObjectMacro(RTStructSetProperties, vtkRTStructSetProperties);
55
56 protected:
57   vtkGDCMPolyDataReader();
58   ~vtkGDCMPolyDataReader();
59
60   char *FileName;
61   vtkMedicalImageProperties *MedicalImageProperties;
62   vtkRTStructSetProperties *RTStructSetProperties;
63 //BTX
64   void FillMedicalImageInformation(const gdcm::Reader &reader);
65 //ETX
66
67   int RequestData(vtkInformation *, vtkInformationVector **, vtkInformationVector *);
68   int RequestInformation(
69     vtkInformation *vtkNotUsed(request),
70     vtkInformationVector **vtkNotUsed(inputVector),
71     vtkInformationVector *outputVector);
72 //BTX
73   int RequestInformation_RTStructureSetStorage(gdcm::Reader const & reader);
74   int RequestData_RTStructureSetStorage(gdcm::Reader const &reader, vtkInformationVector *outputVector);
75   int RequestInformation_HemodynamicWaveformStorage(gdcm::Reader const & reader);
76   int RequestData_HemodynamicWaveformStorage(gdcm::Reader const &reader, vtkInformationVector *outputVector);
```

```
77 //ETX
78
79 private:
80   vtkGDCMPolyDataReader(const vtkGDCMPolyDataReader&);  // Not implemented.
81   void operator=(const vtkGDCMPolyDataReader&);  // Not implemented.
82 };
83
84 #endif
```

11.613 vtkGDCMPolyDataWriter.h File Reference

```
#include "vtkPolyDataWriter.h"
#include "vtkStringArray.h"
#include "vtkStdString.h"
```

Include dependency graph for vtkGDCMPolyDataWriter.h:



Classes

- class vtkGDCMPolyDataWriter

Namespaces

- namespace gdcm

11.614 vtkGDCMPolyDataWriter.h

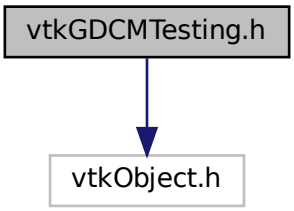
Go to the documentation of this file.

```
1 /*=========================================================================
2
3 Program:  GDCM (Grassroots DICOM). A DICOM library
4
5 Copyright (c) 2006-2011 Mathieu Malaterre
6 All rights reserved.
7 See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.
8
9 This software is distributed WITHOUT ANY WARRANTY; without even
10 the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
11 PURPOSE.  See the above copyright notice for more information.
12
13 =========================================================================*/
```

```
14 // .NAME vtkGDCMPolyDataWriter - writer DICOM PolyData files (Contour Data...)
15 // .SECTION Description
16 // For now only support RTSTRUCT (RT Structure Set Storage)
17 // .SECTION TODO
18 // Need to do the same job for DVH Sequence/DVH Data...
19 // .SECTION Warning
20 //
21 // .SECTION See Also
22 // vtkGDCMImageReader vtkGDCMPolyDataReader vtkRTStructSetProperties
23
24
25 #ifndef VTKGDCMPOLYDATAWRITER_H
26 #define VTKGDCMPOLYDATAWRITER_H
27
28 #include "vtkPolyDataWriter.h"
29 #include "vtkStringArray.h"
30 #include "vtkStdString.h"
31
32
33 class vtkMedicalImageProperties;
34 class vtkRTStructSetProperties;
35 //BTX
36 namespace gdcm { class File; }
37 //ETX
38 class VTK_EXPORT vtkGDCMPolyDataWriter :  public vtkPolyDataWriter
39 {
40 public:
41   static vtkGDCMPolyDataWriter *New();
42   vtkTypeMacro(vtkGDCMPolyDataWriter,vtkPolyDataWriter);
43   virtual void PrintSelf(ostream& os, vtkIndent indent);
44
45   // Description:
46   // Set/Get the filename of the file to be read
47 //  vtkSetStringMacro(FileName);
48 //  vtkGetStringMacro(FileName);
49
50   // Description:
51   // Get the medical image properties object
52 //  vtkGetObjectMacro(MedicalImageProperties, vtkMedicalImageProperties);
53   virtual void SetMedicalImageProperties(vtkMedicalImageProperties *pd);
54
55   virtual void SetRTStructSetProperties(vtkRTStructSetProperties *pd);
56
57
58   //this function will initialize the contained rtstructset with
59   //the inputs of the writer and the various extra information
60   //necessary for writing a complete rtstructset.
61   //NOTE: inputs must be set BEFORE calling this function!
62   //NOTE: the number of outputs for the appendpolydata MUST MATCH the ROI vectors!
63   void InitializeRTStructSet(vtkStdString inDirectory,
64      vtkStdString inStructLabel, vtkStdString inStructName,
65      vtkStringArray* inROINames,
66      vtkStringArray* inROIAlgorithmName,
67      vtkStringArray* inROIType);
68
69   // make parent class public...
70   void SetNumberOfInputPorts(int n);
71
72 protected:
73   vtkGDCMPolyDataWriter();
74   ~vtkGDCMPolyDataWriter();
75
76   vtkMedicalImageProperties *MedicalImageProperties;
77   vtkRTStructSetProperties *RTStructSetProperties;
78
79   void WriteData();
80 //BTX
81   void WriteRTSTRUCTInfo(gdcm::File &file);
82   void WriteRTSTRUCTData(gdcm::File &file, int num);
83 //ETX
84
85 private:
86   vtkGDCMPolyDataWriter(const vtkGDCMPolyDataWriter&);  // Not implemented.
87   void operator=(const vtkGDCMPolyDataWriter&);  // Not implemented.
88 };
89
90 #endif
```

11.615 vtkGDCMTesting.h File Reference

`#include "vtkObject.h"`
Include dependency graph for vtkGDCMTesting.h:



Classes

- class vtkGDCMTesting

11.616 vtkGDCMTesting.h

Go to the documentation of this file.

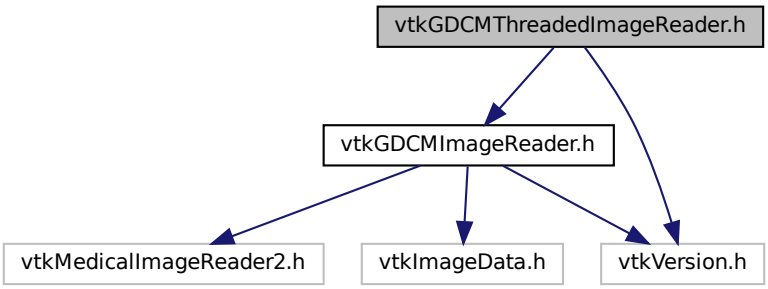
```
1 /*=========================================================================
2
3 Program:  GDCM (Grassroots DICOM). A DICOM library
4
5 Copyright (c) 2006-2011 Mathieu Malaterre
6 All rights reserved.
7 See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.
8
9 This software is distributed WITHOUT ANY WARRANTY; without even
10 the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
11 PURPOSE.  See the above copyright notice for more information.
12
13 =========================================================================*/
14 // .NAME vtkGDCMTesting - GDCM Testing
15 // .SECTION Description
16 // GDCM Testing
17
18 // .SECTION See Also
19 // vtkTesting
20
21 #ifndef VTKGDCMTESTING_H
22 #define VTKGDCMTESTING_H
23
24 #include "vtkObject.h"
25
26 class VTK_EXPORT vtkGDCMTesting :  public vtkObject
27 {
28 public:
29   static vtkGDCMTesting *New();
30   vtkTypeMacro(vtkGDCMTesting,vtkObject);
31   void PrintSelf(ostream& os, vtkIndent indent);
32
33   static const char *GetVTKDataRoot();
```

```
34   static const char *GetGDCMDataRoot();
35
36 //BTX
37   typedef const char* const (*MD5MetaImagesType)[3];
38   static const char * const * GetMD5MetaImage(unsigned int file);
39 //ETX
40   static unsigned int GetNumberOfMD5MetaImages();
41
42   static const char * GetMHDMD5FromFile(const char *filepath);
43   static const char * GetRAWMD5FromFile(const char *filepath);
44
45 protected:
46   vtkGDCMTesting();
47   ~vtkGDCMTesting();
48
49 private:
50   vtkGDCMTesting(const vtkGDCMTesting&);  // Not implemented.
51   void operator=(const vtkGDCMTesting&);  // Not implemented.
52 };
53
54 #endif
```

11.617 vtkGDCMThreadedImageReader.h File Reference

```
#include "vtkGDCMImageReader.h"
#include "vtkVersion.h"
```

Include dependency graph for vtkGDCMThreadedImageReader.h:



Classes

- class vtkGDCMThreadedImageReader

11.618 vtkGDCMThreadedImageReader.h

Go to the documentation of this file.

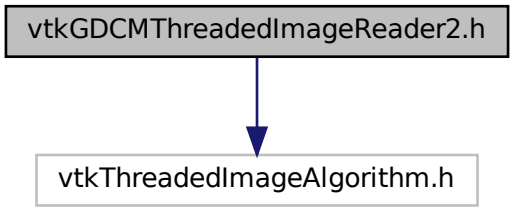
```
1 /*=========================================================================
2
3 Program:  GDCM (Grassroots DICOM). A DICOM library
4
5 Copyright (c) 2006-2011 Mathieu Malaterre
```

```
6 All rights reserved.
7 See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.
8
9 This software is distributed WITHOUT ANY WARRANTY; without even
10 the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
11 PURPOSE.  See the above copyright notice for more information.
12
13 =========================================================================*/
14 // .NAME vtkGDCMThreadedImageReader - read DICOM files with multiple threads
15 // .SECTION Description
16 // vtkGDCMThreadedImageReader is a source object that reads some DICOM files
17 // This reader is threaded.  Meaning that on a multiple core CPU with N cpu, it will
18 // read approx N times faster than when reading in a single thread.
19 //
20 // .SECTION Warning:  Advanced users only.  Do not use this class in the general case,
21 // you have to understand how physically medium works first (sequential reading for
22 // instance) before playing with this class
23 //
24 // .SECTION Implementation note:  when FileLowerLeft is set to on the image is not flipped
25 // upside down as VTK would expect, use this option only if you know what you are doing
26 //
27 // .SECTION FIXME: need to implement the other mode where FileLowerLeft is set to OFF
28 //
29 // .SECTION FIXME: you need to call SetFileName when reading a volume file (multiple slices DICOM)
30 // since SetFileNames expect each single file to be single slice (see parent class)
31 //
32 // .SECTION BUG: you should really consider using vtkGDCMThreadedImageReader2 instead !
33 //
34 // .SECTION See Also
35 // vtkMedicalImageReader2 vtkMedicalImageProperties vtkGDCMThreadedImageReader2
36
37 #ifndef VTKGDCMTHREADEDIMAGEREADER_H
38 #define VTKGDCMTHREADEDIMAGEREADER_H
39
40 #include "vtkGDCMImageReader.h"
41 #include "vtkVersion.h"
42
43 class VTK_EXPORT vtkGDCMThreadedImageReader :  public vtkGDCMImageReader
44 {
45 public:
46   static vtkGDCMThreadedImageReader *New();
47   vtkTypeMacro(vtkGDCMThreadedImageReader,vtkGDCMImageReader);
48   virtual void PrintSelf(ostream& os, vtkIndent indent);
49
50   // Description:
51   // Explicitly set the Rescale Intercept (0028,1052)
52   vtkSetMacro(Shift,double);
53
54   // Description:
55   // Explicitly get/set the Rescale Slope (0028,1053)
56   vtkSetMacro(Scale,double);
57
58   // Description:
59   // Determine whether or not reader should use value from Shift/Scale
60   // Default is 1
61   vtkSetMacro(UseShiftScale,int);
62   vtkGetMacro(UseShiftScale,int);
63   vtkBooleanMacro(UseShiftScale,int);
64
65   // Within this class this is allowed to set the Number of Overlays from outside
66   //vtkSetMacro(NumberOfOverlays,int);
67
68 protected:
69   vtkGDCMThreadedImageReader();
70   ~vtkGDCMThreadedImageReader();
71
72 #if (VTK_MAJOR_VERSION >= 5) || ( VTK_MAJOR_VERSION == 4 && VTK_MINOR_VERSION > 5 )
73   int RequestInformation(vtkInformation *request,
74                          vtkInformationVector **inputVector,
75                          vtkInformationVector *outputVector);
76   int RequestData(vtkInformation *request,
77                   vtkInformationVector **inputVector,
78                   vtkInformationVector *outputVector);
79 #else /*(VTK_MAJOR_VERSION >= 5) || ( VTK_MAJOR_VERSION == 4 && VTK_MINOR_VERSION > 5 )*/
80   void ExecuteInformation();
81   void ExecuteData(vtkDataObject *out);
82 #endif /*(VTK_MAJOR_VERSION >= 5) || ( VTK_MAJOR_VERSION == 4 && VTK_MINOR_VERSION > 5 )*/
83
84   void ReadFiles(unsigned int nfiles, const char *filenames[]);
85   void RequestDataCompat();
86
```

```
87 private:
88   vtkGDCMThreadedImageReader(const vtkGDCMThreadedImageReader&);  // Not implemented.
89   void operator=(const vtkGDCMThreadedImageReader&);  // Not implemented.
90
91   int UseShiftScale;
92 };
93
94 #endif
```

11.619 vtkGDCMThreadedImageReader2.h File Reference

`#include "vtkThreadedImageAlgorithm.h"`
Include dependency graph for vtkGDCMThreadedImageReader2.h:



Classes

- class vtkGDCMThreadedImageReader2

11.620 vtkGDCMThreadedImageReader2.h

Go to the documentation of this file.

```
1 /*=========================================================================
2
3 Program:  GDCM (Grassroots DICOM). A DICOM library
4
5 Copyright (c) 2006-2011 Mathieu Malaterre
6 All rights reserved.
7 See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.
8
9 This software is distributed WITHOUT ANY WARRANTY; without even
10 the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
11 PURPOSE.  See the above copyright notice for more information.
12
13 =========================================================================*/
14 // .NAME vtkGDCMThreadedImageReader2 - read DICOM files with multiple threads
15 // .SECTION Description
16 // vtkGDCMThreadedImageReader2 is a source object that reads some DICOM files
17 // This reader is threaded.  Meaning that on a multiple core CPU with N cpu, it will
18 // read approx N times faster than when reading in a single thread assuming the IO is
19 // not a bottleneck operation.
20 // If looking for a single threaded class see:  vtkGDCMImageReader
21 //
```

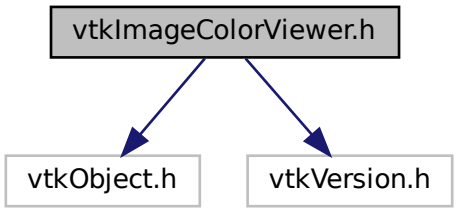
```
22 // .SECTION Warning:  Advanced users only.  Do not use this class in the general case,
23 // you have to understand how physically medium works first (sequential reading for
24 // instance) before playing with this class
25 //
26 // .SECTION Implementation note:  when FileLowerLeft is set to on the image is not flipped
27 // upside down as VTK would expect, use this option only if you know what you are doing
28 //
29 // .SECTION FIXME: need to implement the other mode where FileLowerLeft is set to OFF
30 //
31 // .SECTION FIXME: need to implement reading of series of 3D files
32 //
33 // .SECTION Implementation note:  this class is meant to superseed vtkGDCMThreadedImageReader
34 // because it had support for ProgressEvent support even from python layer.  There is a
35 // subtle trick down in the threading mechanism in VTK were the main thread (talking to the
36 // python interpreter) is also part of the execution process (and the N-1 other thread
37 // are just there to execute the remaining of ThreadedRequestData), this separation into
38 // two types of thread is necessary to achieve a working implementation of UpdateProgress
39
40 // .SECTION See Also
41 // vtkMedicalImageReader2 vtkMedicalImageProperties vtkGDCMImageReader
42
43 #ifndef VTKGDCMTHREADEDIMAGEREADER2_H
44 #define VTKGDCMTHREADEDIMAGEREADER2_H
45
46 #include "vtkThreadedImageAlgorithm.h"
47
48 class vtkStringArray;
49 class VTK_EXPORT vtkGDCMThreadedImageReader2 :  public vtkThreadedImageAlgorithm
50 {
51 public:
52   static vtkGDCMThreadedImageReader2 *New();
53   vtkTypeMacro(vtkGDCMThreadedImageReader2,vtkThreadedImageAlgorithm);
54   virtual void PrintSelf(ostream& os, vtkIndent indent);
55
56   vtkGetMacro(FileLowerLeft,int);
57   vtkSetMacro(FileLowerLeft,int);
58   vtkBooleanMacro(FileLowerLeft,int);
59
60   vtkGetMacro(NumberOfOverlays,int);
61
62   vtkSetMacro(DataScalarType,int);
63   vtkGetMacro(DataScalarType,int);
64
65   vtkSetMacro(NumberOfScalarComponents,int);
66   vtkGetMacro(NumberOfScalarComponents,int);
67
68   vtkGetMacro(LoadOverlays,int);
69   vtkSetMacro(LoadOverlays,int);
70   vtkBooleanMacro(LoadOverlays,int);
71
72   vtkSetVector6Macro(DataExtent,int);
73   vtkGetVector6Macro(DataExtent,int);
74
75   vtkSetVector3Macro(DataOrigin,double);
76   vtkGetVector3Macro(DataOrigin,double);
77
78   vtkSetVector3Macro(DataSpacing,double);
79   vtkGetVector3Macro(DataSpacing,double);
80
81   //vtkGetStringMacro(FileName);
82   //vtkSetStringMacro(FileName);
83   virtual const char *GetFileName(int i = 0);
84   virtual void SetFileName(const char *filename);
85
86   virtual void SetFileNames(vtkStringArray*);
87   vtkGetObjectMacro(FileNames, vtkStringArray);
88
89   int SplitExtent(int splitExt[6], int startExt[6],
90                   int num, int total);
91
92   // Description:
93   // Explicitly set the Rescale Intercept (0028,1052)
94   vtkSetMacro(Shift,double);
95   vtkGetMacro(Shift,double);
96
97   // Description:
98   // Explicitly get/set the Rescale Slope (0028,1053)
99   vtkSetMacro(Scale,double);
100   vtkGetMacro(Scale,double);
101
102   // Description:
```

```
103   // Determine whether or not reader should use value from Shift/Scale
104   // Default is 1
105   vtkSetMacro(UseShiftScale,int);
106   vtkGetMacro(UseShiftScale,int);
107   vtkBooleanMacro(UseShiftScale,int);
108
109 protected:
110   vtkGDCMThreadedImageReader2();
111   ~vtkGDCMThreadedImageReader2();
112
113   int RequestInformation(vtkInformation *request,
114                          vtkInformationVector **inputVector,
115                          vtkInformationVector *outputVector);
116
117 protected:
118   void ThreadedRequestData (
119     vtkInformation * request,
120     vtkInformationVector** inputVector,
121     vtkInformationVector * outputVector,
122     vtkImageData ***inData,
123     vtkImageData **outData,
124     int outExt[6], int id);
125
126 private:
127   int FileLowerLeft;
128   char *FileName;
129   vtkStringArray *FileNames;
130   int LoadIconImage;
131   int DataExtent[6];
132   int LoadOverlays;
133   int NumberOfOverlays;
134   int DataScalarType;
135
136   int NumberOfScalarComponents;
137   double DataSpacing[3];
138   double DataOrigin[3];
139   int IconImageDataExtent[6];
140
141   double Shift;
142   double Scale;
143   int UseShiftScale;
144
145 private:
146   vtkGDCMThreadedImageReader2(const vtkGDCMThreadedImageReader2&);  // Not implemented.
147   void operator=(const vtkGDCMThreadedImageReader2&);  // Not implemented.
148 };
149
150 #endif
```

11.621 vtkImageColorViewer.h File Reference

```
#include "vtkObject.h"
#include "vtkVersion.h"
```

Include dependency graph for vtkImageColorViewer.h:



Classes

- class vtkImageColorViewer

11.622 vtkImageColorViewer.h

Go to the documentation of this file.

```
1 /*=========================================================================
2
3 Program:  GDCM (Grassroots DICOM). A DICOM library
4
5 Copyright (c) 2006-2011 Mathieu Malaterre
6 All rights reserved.
7 See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.
8
9 This software is distributed WITHOUT ANY WARRANTY; without even
10 the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
11 PURPOSE.  See the above copyright notice for more information.
12
13 =========================================================================*/
14 // .NAME vtkImageColorViewer - Display a 2D image.
15 // .SECTION Description
16 // vtkImageColorViewer is a convenience class for displaying a 2D image.    It
17 // packages up the functionality found in vtkRenderWindow, vtkRenderer,
18 // vtkImageActor and vtkImageMapToWindowLevelColors into a single easy to use
19 // class.    This class also creates an image interactor style
20 // (vtkInteractorStyleImage) that allows zooming and panning of images, and
21 // supports interactive window/level operations on the image.  Note that
22 // vtkImageColorViewer is simply a wrapper around these classes.
23 //
24 // vtkImageColorViewer uses the 3D rendering and texture mapping engine
25 // to draw an image on a plane.    This allows for rapid rendering,
26 // zooming, and panning.  The image is placed in the 3D scene at a
27 // depth based on the z-coordinate of the particular image slice.  Each
28 // call to SetSlice() changes the image data (slice) displayed AND
29 // changes the depth of the displayed slice in the 3D scene.  This can
30 // be controlled by the AutoAdjustCameraClippingRange ivar of the
31 // InteractorStyle member.
32 //
33 // It is possible to mix images and geometry, using the methods:
34 //
35 // viewer->SetInput( myImage );
36 // viewer->GetRenderer()->AddActor( myActor );
37 //
38 // This can be used to annotate an image with a PolyData of "edges" or
39 // or highlight sections of an image or display a 3D isosurface
```

```
40 // with a slice from the volume, etc.  Any portions of your geometry
41 // that are in front of the displayed slice will be visible; any
42 // portions of your geometry that are behind the displayed slice will
43 // be obscured.  A more general framework (with respect to viewing
44 // direction) for achieving this effect is provided by the
45 // vtkImagePlaneWidget .
46 //
47 // Note that pressing 'r' will reset the window/level and pressing
48 // shift+'r' or control+'r' will reset the camera.
49 //
50 // .SECTION See Also
51 // vtkRenderWindow vtkRenderer vtkImageActor vtkImageMapToWindowLevelColors
52
53 #ifndef VTKIMAGECOLORVIEWER_H
54 #define VTKIMAGECOLORVIEWER_H
55
56 #include "vtkObject.h"
57 #include "vtkVersion.h"
58
59 class vtkAlgorithm;
60 class vtkAlgorithmOutput;
61 class vtkImageActor;
62 class vtkImageData;
63 class vtkImageMapToWindowLevelColors2;
64 class vtkInformation;
65 class vtkInteractorStyleImage;
66 class vtkRenderWindow;
67 class vtkRenderer;
68 class vtkRenderWindowInteractor;
69 class vtkPolyData;
70
71 class VTK_EXPORT vtkImageColorViewer :  public vtkObject
72 {
73 public:
74   static vtkImageColorViewer *New();
75   vtkTypeMacro(vtkImageColorViewer,vtkObject);
76   void PrintSelf(ostream& os, vtkIndent indent);
77
78   // Description:
79   // Get the name of rendering window.
80   virtual const char *GetWindowName();
81
82   // Description:
83   // Render the resulting image.
84   virtual void Render(void);
85
86   // Description:
87   // Set/Get the input image to the viewer.
88 #if (VTK_MAJOR_VERSION >= 6)
89   virtual void SetInputData(vtkImageData *in);
90 #else
91   virtual void SetInput(vtkImageData *in);
92 #endif
93   virtual vtkImageData *GetInput();
94   virtual void SetInputConnection(vtkAlgorithmOutput* input);
95   virtual void AddInputConnection(vtkAlgorithmOutput* input);
96   virtual void AddInput(vtkImageData * input);
97   //virtual void AddInput(vtkPolyData * input);
98
99   double GetOverlayVisibility();
100   void SetOverlayVisibility(double vis);
101
102   // Description:
103   // Set/get the slice orientation
104   //BTX
105   enum
106   {
107     SLICE_ORIENTATION_YZ = 0,
108     SLICE_ORIENTATION_XZ = 1,
109     SLICE_ORIENTATION_XY = 2
110   };
111   //ETX
112   vtkGetMacro(SliceOrientation, int);
113   virtual void SetSliceOrientation(int orientation);
114   virtual void SetSliceOrientationToXY()
115     { this->SetSliceOrientation(vtkImageColorViewer::SLICE_ORIENTATION_XY); };
116   virtual void SetSliceOrientationToYZ()
117     { this->SetSliceOrientation(vtkImageColorViewer::SLICE_ORIENTATION_YZ); };
118   virtual void SetSliceOrientationToXZ()
119     { this->SetSliceOrientation(vtkImageColorViewer::SLICE_ORIENTATION_XZ); };
120
```

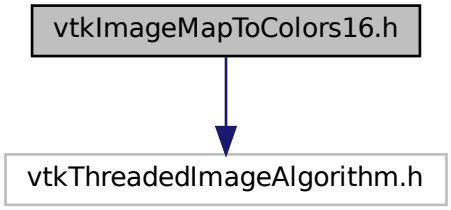
```
121   // Description:
122   // Set/Get the current slice to display (depending on the orientation
123   // this can be in X, Y or Z).
124   vtkGetMacro(Slice, int);
125   virtual void SetSlice(int s);
126
127   // Description:
128   // Update the display extent manually so that the proper slice for the
129   // given orientation is displayed.  It will also try to set a
130   // reasonable camera clipping range.
131   // This method is called automatically when the Input is changed, but
132   // most of the time the input of this class is likely to remain the same,
133   // i.e.  connected to the output of a filter, or an image reader.  When the
134   // input of this filter or reader itself is changed, an error message might
135   // be displayed since the current display extent is probably outside
136   // the new whole extent.  Calling this method will ensure that the display
137   // extent is reset properly.
138   virtual void UpdateDisplayExtent();
139
140   // Description:
141   // Return the minimum and maximum slice values (depending on the orientation
142   // this can be in X, Y or Z).
143   virtual int GetSliceMin();
144   virtual int GetSliceMax();
145   virtual void GetSliceRange(int range[2])
146     { this->GetSliceRange(range[0], range[1]); }
147   virtual void GetSliceRange(int &min, int &max);
148   virtual int* GetSliceRange();
149
150   // Description:
151   // Set window and level for mapping pixels to colors.
152   virtual double GetColorWindow();
153   virtual double GetColorLevel();
154   virtual void SetColorWindow(double s);
155   virtual void SetColorLevel(double s);
156
157   // Description:
158   // These are here when using a Tk window.
159   virtual void SetDisplayId(void *a);
160   virtual void SetWindowId(void *a);
161   virtual void SetParentId(void *a);
162
163   // Description:
164   // Set/Get the position in screen coordinates of the rendering window.
165   virtual int* GetPosition();
166   virtual void SetPosition(int a,int b);
167   virtual void SetPosition(int a[2]) { this->SetPosition(a[0],a[1]); }
168
169   // Description:
170   // Set/Get the size of the window in screen coordinates in pixels.
171   virtual int* GetSize();
172   virtual void SetSize(int a, int b);
173   virtual void SetSize(int a[2]) { this->SetSize(a[0],a[1]); }
174
175   // Description:
176   // Get the internal render window, renderer, image actor, and
177   // image map instances.
178   vtkGetObjectMacro(RenderWindow,vtkRenderWindow);
179   vtkGetObjectMacro(Renderer, vtkRenderer);
180   vtkGetObjectMacro(ImageActor,vtkImageActor);
181   vtkGetObjectMacro(WindowLevel,vtkImageMapToWindowLevelColors2);
182   vtkGetObjectMacro(InteractorStyle,vtkInteractorStyleImage);
183
184   // Description:
185   // Set your own renderwindow and renderer
186   virtual void SetRenderWindow(vtkRenderWindow *arg);
187   virtual void SetRenderer(vtkRenderer *arg);
188
189   // Description:
190   // Attach an interactor for the internal render window.
191   virtual void SetupInteractor(vtkRenderWindowInteractor*);
192
193   // Description:
194   // Create a window in memory instead of on the screen.  This may not
195   // be supported for every type of window and on some windows you may
196   // need to invoke this prior to the first render.
197   virtual void SetOffScreenRendering(int);
198   virtual int GetOffScreenRendering();
199   vtkBooleanMacro(OffScreenRendering,int);
200
201   // Description:
```

```
202   // @deprecated Replaced by vtkImageColorViewer::GetSliceMin() as of VTK 5.0.
203   VTK_LEGACY(int GetWholeZMin());
204
205   // Description:
206   // @deprecated Replaced by vtkImageColorViewer::GetSliceMax() as of VTK 5.0.
207   VTK_LEGACY(int GetWholeZMax());
208
209   // Description:
210   // @deprecated Replaced by vtkImageColorViewer::GetSlice() as of VTK 5.0.
211   VTK_LEGACY(int GetZSlice());
212
213   // Description:
214   // @deprecated Replaced by vtkImageColorViewer::SetSlice() as of VTK 5.0.
215   VTK_LEGACY(void SetZSlice(int));
216
217 protected:
218   vtkImageColorViewer();
219   ~vtkImageColorViewer();
220
221   virtual void InstallPipeline();
222   virtual void UnInstallPipeline();
223
224   vtkImageMapToWindowLevelColors2  *WindowLevel;
225   vtkRenderWindow                 *RenderWindow;
226   vtkRenderer                     *Renderer;
227   vtkImageActor                   *ImageActor;
228   vtkImageActor                   *OverlayImageActor;
229   vtkRenderWindowInteractor       *Interactor;
230   vtkInteractorStyleImage         *InteractorStyle;
231
232   int SliceOrientation;
233   int FirstRender;
234   int Slice;
235
236   virtual void UpdateOrientation();
237
238 #if (VTK_MAJOR_VERSION >= 6)
239   vtkAlgorithm* GetInputAlgorithm();
240   vtkInformation* GetInputInformation();
241 #endif
242
243   friend class vtkImageColorViewerCallback;
244
245 private:
246   vtkImageColorViewer(const vtkImageColorViewer&);  // Not implemented.
247   void operator=(const vtkImageColorViewer&);  // Not implemented.
248 };
249
250 #endif
```

11.623 vtkImageMapToColors16.h File Reference

`#include "vtkThreadedImageAlgorithm.h"`

Include dependency graph for vtkImageMapToColors16.h:



Classes

- class vtkImageMapToColors16

11.624 vtkImageMapToColors16.h

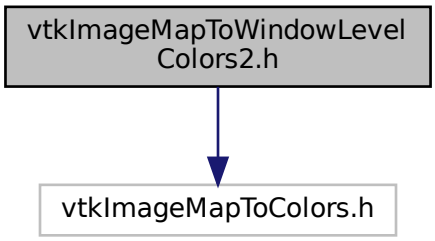
Go to the documentation of this file.

```
1 /*=========================================================================
2
3 Program:  GDCM (Grassroots DICOM). A DICOM library
4
5 Copyright (c) 2006-2011 Mathieu Malaterre
6 All rights reserved.
7 See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.
8
9 This software is distributed WITHOUT ANY WARRANTY; without even
10 the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
11 PURPOSE.  See the above copyright notice for more information.
12
13 =========================================================================*/
14 /*=========================================================================
15
16 Portions of this file are subject to the VTK Toolkit Version 3 copyright.
17
18 Program:      Visualization Toolkit
19 Module:        $RCSfile:  vtkImageMapToColors16.h,v $
20
21 Copyright (c) Ken Martin, Will Schroeder, Bill Lorensen
22 All rights reserved.
23 See Copyright.txt or http://www.kitware.com/Copyright.htm for details.
24
25 This software is distributed WITHOUT ANY WARRANTY; without even
26 the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
27 PURPOSE.  See the above copyright notice for more information.
28
29 =========================================================================*/
30 // .NAME vtkImageMapToColors16 - map the input image through a lookup table
31 // .SECTION Description
32 // The vtkImageMapToColors16 filter will take an input image of any valid
33 // scalar type, and map the first component of the image through a
34 // lookup table.    The result is an image of type VTK_UNSIGNED_CHAR.
35 // If the lookup table is not set, or is set to NULL, then the input
36 // data will be passed through if it is already of type VTK_UNSIGNED_CHAR.
37
38 // .SECTION See Also
39 // vtkLookupTable vtkScalarsToColors
40
41 #ifndef VTKIMAGEMAPTOCOLORS16_H
42 #define VTKIMAGEMAPTOCOLORS16_H
43
44
45 #include "vtkThreadedImageAlgorithm.h"
46
47 class vtkScalarsToColors;
48
49 class VTK_EXPORT vtkImageMapToColors16 :  public vtkThreadedImageAlgorithm
50 {
51 public:
52   static vtkImageMapToColors16 *New();
53   vtkTypeMacro(vtkImageMapToColors16,vtkThreadedImageAlgorithm);
54   void PrintSelf(ostream& os, vtkIndent indent);
55
56   // Description:
57   // Set the lookup table.
58   virtual void SetLookupTable(vtkScalarsToColors*);
59   vtkGetObjectMacro(LookupTable,vtkScalarsToColors);
60
61   // Description:
62   // Set the output format, the default is RGBA.
63   vtkSetMacro(OutputFormat,int);
64   vtkGetMacro(OutputFormat,int);
65   void SetOutputFormatToRGBA() { this->OutputFormat = VTK_RGBA; };
66   void SetOutputFormatToRGB() { this->OutputFormat = VTK_RGB; };
67   void SetOutputFormatToLuminanceAlpha() { this->OutputFormat = VTK_LUMINANCE_ALPHA; };
```

```
68   void SetOutputFormatToLuminance() { this->OutputFormat = VTK_LUMINANCE; };
69
70   // Description:
71   // Set the component to map for multi-component images (default:  0)
72   vtkSetMacro(ActiveComponent,int);
73   vtkGetMacro(ActiveComponent,int);
74
75   // Description:
76   // Use the alpha component of the input when computing the alpha component
77   // of the output (useful when converting monochrome+alpha data to RGBA)
78   vtkSetMacro(PassAlphaToOutput,int);
79   vtkBooleanMacro(PassAlphaToOutput,int);
80   vtkGetMacro(PassAlphaToOutput,int);
81
82   // Description:
83   // We need to check the modified time of the lookup table too.
84 #ifdef VTK_HAS_MTIME_TYPE
85   virtual vtkMTimeType GetMTime();
86 #else
87   virtual unsigned long GetMTime();
88 #endif
89
90 protected:
91   vtkImageMapToColors16();
92   ~vtkImageMapToColors16();
93
94   virtual int RequestInformation (vtkInformation *, vtkInformationVector **, vtkInformationVector *);
95
96   void ThreadedRequestData(vtkInformation *request,
97                            vtkInformationVector **inputVector,
98                            vtkInformationVector *outputVector,
99                            vtkImageData ***inData, vtkImageData **outData,
100                            int extent[6], int id);
101
102   virtual int RequestData(vtkInformation *request,
103                           vtkInformationVector **inputVector,
104                           vtkInformationVector *outputVector);
105
106   vtkScalarsToColors *LookupTable;
107   int OutputFormat;
108
109   int ActiveComponent;
110   int PassAlphaToOutput;
111
112   int DataWasPassed;
113 private:
114   vtkImageMapToColors16(const vtkImageMapToColors16&);  // Not implemented.
115   void operator=(const vtkImageMapToColors16&);  // Not implemented.
116 };
117
118 #endif
```

11.625 vtkImageMapToWindowLevelColors2.h File Reference

`#include "vtkImageMapToColors.h"`
Include dependency graph for vtkImageMapToWindowLevelColors2.h:



Classes

- class vtkImageMapToWindowLevelColors2

11.626 vtkImageMapToWindowLevelColors2.h

Go to the documentation of this file.

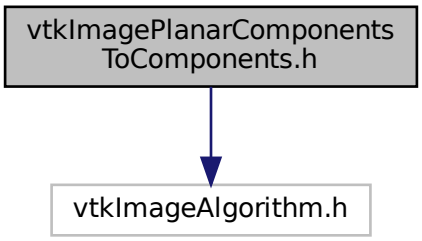
```
1 /*=========================================================================
2
3 Program:  GDCM (Grassroots DICOM). A DICOM library
4
5 Copyright (c) 2006-2011 Mathieu Malaterre
6 All rights reserved.
7 See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.
8
9 This software is distributed WITHOUT ANY WARRANTY; without even
10 the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
11 PURPOSE.  See the above copyright notice for more information.
12
13 =========================================================================*/
14 /*=========================================================================
15
16 Portions of this file are subject to the VTK Toolkit Version 3 copyright.
17
18 Program:      Visualization Toolkit
19 Module:        $RCSfile:  vtkImageMapToWindowLevelColors2.h,v $
20
21 Copyright (c) Ken Martin, Will Schroeder, Bill Lorensen
22 All rights reserved.
23 See Copyright.txt or http://www.kitware.com/Copyright.htm for details.
24
25 This software is distributed WITHOUT ANY WARRANTY; without even
26 the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
27 PURPOSE.  See the above copyright notice for more information.
28
29 =========================================================================*/
30 // .NAME vtkImageMapToWindowLevelColors2 - map the input image through a lookup table and window / level it
31 // .SECTION Description
32 // The vtkImageMapToWindowLevelColors2 filter will take an input image of any
```

```
33 // valid scalar type, and map the first component of the image through a
34 // lookup table.    This resulting color will be modulated with value obtained
35 // by a window / level operation.  The result is an image of type
36 // VTK_UNSIGNED_CHAR. If the lookup table is not set, or is set to NULL, then
37 // the input data will be passed through if it is already of type
38 // UNSIGNED_CHAR.
39 //
40 // .SECTION See Also
41 // vtkLookupTable vtkScalarsToColors
42
43 #ifndef VTKIMAGEMAPTOWINDOWLEVELCOLORS2_H
44 #define VTKIMAGEMAPTOWINDOWLEVELCOLORS2_H
45
46 #include "vtkImageMapToColors.h"
47
48 class VTK_EXPORT vtkImageMapToWindowLevelColors2 :  public vtkImageMapToColors
49 {
50 public:
51   static vtkImageMapToWindowLevelColors2 *New();
52   vtkTypeMacro(vtkImageMapToWindowLevelColors2,vtkImageMapToColors);
53   void PrintSelf(ostream& os, vtkIndent indent);
54
55   // Description:
56   // Set / Get the Window to use -> modulation will be performed on the
57   // color based on (S - (L - W/2))/W where S is the scalar value, L is
58   // the level and W is the window.
59   vtkSetMacro( Window, double );
60   vtkGetMacro( Window, double );
61
62   // Description:
63   // Set / Get the Level to use -> modulation will be performed on the
64   // color based on (S - (L - W/2))/W where S is the scalar value, L is
65   // the level and W is the window.
66   vtkSetMacro( Level, double );
67   vtkGetMacro( Level, double );
68
69 protected:
70   vtkImageMapToWindowLevelColors2();
71   ~vtkImageMapToWindowLevelColors2();
72
73   virtual int RequestInformation (vtkInformation *, vtkInformationVector **, vtkInformationVector *);
74   void ThreadedRequestData(vtkInformation *request,
75                            vtkInformationVector **inputVector,
76                            vtkInformationVector *outputVector,
77                            vtkImageData ***inData, vtkImageData **outData,
78                            int extent[6], int id);
79   virtual int RequestData(vtkInformation *request,
80                           vtkInformationVector **inputVector,
81                           vtkInformationVector *outputVector);
82
83   double Window;
84   double Level;
85
86 private:
87   vtkImageMapToWindowLevelColors2(const vtkImageMapToWindowLevelColors2&);  // Not implemented.
88   void operator=(const vtkImageMapToWindowLevelColors2&);  // Not implemented.
89 };
90
91 #endif
```

11.627 vtkImagePlanarComponentsToComponents.h File Reference

```
#include "vtkImageAlgorithm.h"
```

Include dependency graph for vtkImagePlanarComponentsToComponents.h:



Classes

- class vtkImagePlanarComponentsToComponents

11.628 vtkImagePlanarComponentsToComponents.h

Go to the documentation of this file.

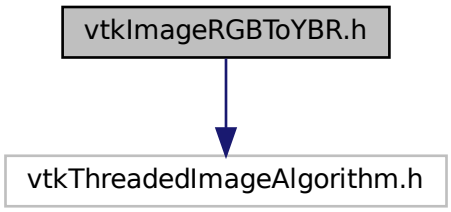
```
1 /*=========================================================================
2
3 Program:  GDCM (Grassroots DICOM). A DICOM library
4
5 Copyright (c) 2006-2011 Mathieu Malaterre
6 All rights reserved.
7 See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.
8
9 This software is distributed WITHOUT ANY WARRANTY; without even
10 the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
11 PURPOSE.  See the above copyright notice for more information.
12
13 =========================================================================*/
14 /*=========================================================================
15
16 Portions of this file are subject to the VTK Toolkit Version 3 copyright.
17
18 Program:      Visualization Toolkit
19 Module:        $RCSfile:  vtkImagePlanarComponentsToComponents.h,v $
20
21 Copyright (c) Ken Martin, Will Schroeder, Bill Lorensen
22 All rights reserved.
23 See Copyright.txt or http://www.kitware.com/Copyright.htm for details.
24
25 This software is distributed WITHOUT ANY WARRANTY; without even
26 the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
27 PURPOSE.  See the above copyright notice for more information.
28
29 =========================================================================*/
30 // .NAME vtkImagePlanarComponentsToComponents - Converts planar comp to pixel comp
31 // .SECTION Description
32
```

```
33 // .SECTION See Also
34 // TODO: Can I make this filter threaded ?
35 // TODO: How do I handle the VTK-flipping (FileLowerLeft)?
36
37 #ifndef VTKIMAGEPLANARCOMPONENTSTOCOMPONENTS_H
38 #define VTKIMAGEPLANARCOMPONENTSTOCOMPONENTS_H
39
40 #include "vtkImageAlgorithm.h"
41
42 // everything is now handled within the vtkGDCMImageReader as Planar Configuration can not
43 // be externalized (conflict with file lower left)
44
45 #error do not use this class
46
47 //class VTK_EXPORT vtkImagePlanarComponentsToComponents :  public vtkThreadedImageAlgorithm
48 class VTK_EXPORT vtkImagePlanarComponentsToComponents :  public vtkImageAlgorithm
49 {
50 public:
51   static vtkImagePlanarComponentsToComponents *New();
52   //vtkTypeMacro(vtkImagePlanarComponentsToComponents,vtkThreadedImageAlgorithm);
53   vtkTypeMacro(vtkImagePlanarComponentsToComponents,vtkImageAlgorithm);
54
55   void PrintSelf(ostream& os, vtkIndent indent);
56
57 protected:
58   vtkImagePlanarComponentsToComponents();
59   ~vtkImagePlanarComponentsToComponents() {};
60
61 //  void ThreadedExecute (vtkImageData *inData, vtkImageData *outData,
62 //                        int ext[6], int id);
63 //  virtual int RequestInformation (vtkInformation *, vtkInformationVector**, vtkInformationVector *);
64   virtual int RequestData(vtkInformation *, vtkInformationVector **, vtkInformationVector *);
65
66 private:
67   vtkImagePlanarComponentsToComponents(const vtkImagePlanarComponentsToComponents&);  // Not implemented.
68   void operator=(const vtkImagePlanarComponentsToComponents&);  // Not implemented.
69 };
70
71 #endif
```

11.629 vtkImageRGBToYBR.h File Reference

`#include "vtkThreadedImageAlgorithm.h"`

Include dependency graph for vtkImageRGBToYBR.h:



Classes

- class vtkImageRGBToYBR

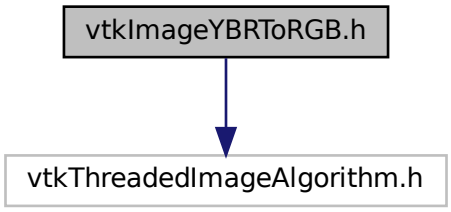
11.630 vtkImageRGBToYBR.h

Go to the documentation of this file.

```
1 /*=========================================================================
2
3 Program:  GDCM (Grassroots DICOM). A DICOM library
4
5 Copyright (c) 2006-2011 Mathieu Malaterre
6 All rights reserved.
7 See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.
8
9 This software is distributed WITHOUT ANY WARRANTY; without even
10 the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
11 PURPOSE.  See the above copyright notice for more information.
12
13 =========================================================================*/
14 /*=========================================================================
15
16 Portions of this file are subject to the VTK Toolkit Version 3 copyright.
17
18 Program:      Visualization Toolkit
19 Module:        $RCSfile:  vtkImageRGBToYBR.h,v $
20
21 Copyright (c) Ken Martin, Will Schroeder, Bill Lorensen
22 All rights reserved.
23 See Copyright.txt or http://www.kitware.com/Copyright.htm for details.
24
25 This software is distributed WITHOUT ANY WARRANTY; without even
26 the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
27 PURPOSE.  See the above copyright notice for more information.
28
29 =========================================================================*/
30 // .NAME vtkImageRGBToYBR - Converts YBR components to RGB.
31 // .SECTION Description
32 // For each pixel with hue, saturation and value components this filter
33 // outputs the color coded as red, green, blue.    Output type must be the same
34 // as input type.
35
36 // .SECTION See Also
37 // vtkImageRGBToHSV
38
39 #ifndef VTKIMAGERGBTOYBR_H
40 #define VTKIMAGERGBTOYBR_H
41
42 #include "vtkThreadedImageAlgorithm.h"
43
44 class VTK_EXPORT vtkImageRGBToYBR :  public vtkThreadedImageAlgorithm
45 {
46 public:
47   static vtkImageRGBToYBR *New();
48   vtkTypeMacro(vtkImageRGBToYBR,vtkThreadedImageAlgorithm);
49
50   void PrintSelf(ostream& os, vtkIndent indent);
51
52 protected:
53   vtkImageRGBToYBR();
54   ~vtkImageRGBToYBR() {};
55
56   void ThreadedExecute (vtkImageData *inData, vtkImageData *outData,
57                         int ext[6], int id);
58 private:
59   vtkImageRGBToYBR(const vtkImageRGBToYBR&);  // Not implemented.
60   void operator=(const vtkImageRGBToYBR&);  // Not implemented.
61 };
62
63 #endif
```

11.631 vtkImageYBRToRGB.h File Reference

`#include "vtkThreadedImageAlgorithm.h"`
Include dependency graph for vtkImageYBRToRGB.h:



Classes

- class vtkImageYBRToRGB

11.632 vtkImageYBRToRGB.h

Go to the documentation of this file.

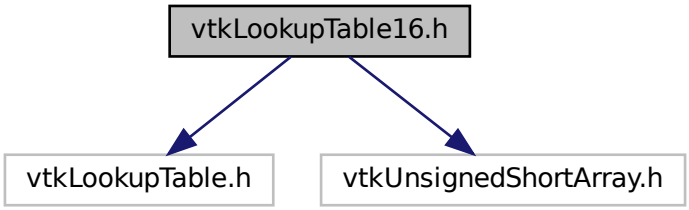
```
1 /*=========================================================================
2
3 Program:  GDCM (Grassroots DICOM). A DICOM library
4
5 Copyright (c) 2006-2011 Mathieu Malaterre
6 All rights reserved.
7 See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.
8
9 This software is distributed WITHOUT ANY WARRANTY; without even
10 the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
11 PURPOSE.  See the above copyright notice for more information.
12
13 =========================================================================*/
14 /*=========================================================================
15
16 Portions of this file are subject to the VTK Toolkit Version 3 copyright.
17
18 Program:      Visualization Toolkit
19 Module:        $RCSfile:  vtkImageYBRToRGB.h,v $
20
21 Copyright (c) Ken Martin, Will Schroeder, Bill Lorensen
22 All rights reserved.
23 See Copyright.txt or http://www.kitware.com/Copyright.htm for details.
24
25 This software is distributed WITHOUT ANY WARRANTY; without even
26 the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
27 PURPOSE.  See the above copyright notice for more information.
28
29 =========================================================================*/
30 // .NAME vtkImageYBRToRGB - Converts YBR components to RGB.
31 // .SECTION Description
32 // For each pixel with hue, saturation and value components this filter
33 // outputs the color coded as red, green, blue.    Output type must be the same
```

```
34 // as input type.
35
36 // .SECTION See Also
37 // vtkImageRGBToHSV
38
39 #ifndef VTKIMAGEYBRTORGB_H
40 #define VTKIMAGEYBRTORGB_H
41
42 #include "vtkThreadedImageAlgorithm.h"
43
44 class VTK_EXPORT vtkImageYBRToRGB :  public vtkThreadedImageAlgorithm
45 {
46 public:
47   static vtkImageYBRToRGB *New();
48   vtkTypeMacro(vtkImageYBRToRGB,vtkThreadedImageAlgorithm);
49
50   void PrintSelf(ostream& os, vtkIndent indent);
51
52 protected:
53   vtkImageYBRToRGB();
54   ~vtkImageYBRToRGB() {};
55
56   void ThreadedExecute (vtkImageData *inData, vtkImageData *outData,
57                         int ext[6], int id);
58 private:
59   vtkImageYBRToRGB(const vtkImageYBRToRGB&);  // Not implemented.
60   void operator=(const vtkImageYBRToRGB&);  // Not implemented.
61 };
62
63 #endif
```

11.633 vtkLookupTable16.h File Reference

```
#include "vtkLookupTable.h"
#include "vtkUnsignedShortArray.h"
```

Include dependency graph for vtkLookupTable16.h:



Classes

- class vtkLookupTable16

11.634 vtkLookupTable16.h

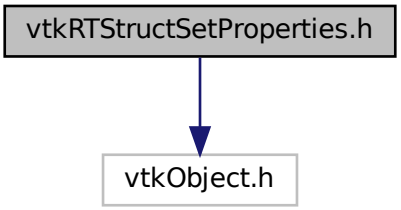
Go to the documentation of this file.

```
1 /*=========================================================================
2
3 Program:  GDCM (Grassroots DICOM). A DICOM library
4
5 Copyright (c) 2006-2011 Mathieu Malaterre
6 All rights reserved.
7 See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.
8
9 This software is distributed WITHOUT ANY WARRANTY; without even
10 the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
11 PURPOSE.  See the above copyright notice for more information.
12
13 =========================================================================*/
14 /*=========================================================================
15
16 Portions of this file are subject to the VTK Toolkit Version 3 copyright.
17
18 Program:      Visualization Toolkit
19 Module:        $RCSfile:  vtkLookupTable16.h,v $
20
21 Copyright (c) Ken Martin, Will Schroeder, Bill Lorensen
22 All rights reserved.
23 See Copyright.txt or http://www.kitware.com/Copyright.htm for details.
24
25 This software is distributed WITHOUT ANY WARRANTY; without even
26 the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
27 PURPOSE.  See the above copyright notice for more information.
28
29 =========================================================================*/
30 // .NAME vtkLookupTable16 -
31 // .SECTION Description
32 //
33 // .SECTION Caveats
34 //
35 // .SECTION See Also
36 // vtkLookupTable
37
38 #ifndef VTKLOOKUPTABLE16_H
39 #define VTKLOOKUPTABLE16_H
40
41 #include "vtkLookupTable.h"
42 #include "vtkUnsignedShortArray.h"
43
44 class VTK_EXPORT vtkLookupTable16 :  public vtkLookupTable
45 {
46 public:
47   static vtkLookupTable16 *New();
48
49   vtkTypeMacro(vtkLookupTable16,vtkLookupTable);
50   void PrintSelf(ostream& os, vtkIndent indent);
51
52   void Build();
53
54   void SetNumberOfTableValues(vtkIdType number);
55
56   unsigned char *WritePointer(const vtkIdType id, const int number);
57
58   unsigned short *GetPointer(const vtkIdType id) {
59     return this->Table16->GetPointer(4*id); };
60
61 protected:
62   vtkLookupTable16(int sze=256, int ext=256);
63   ~vtkLookupTable16();
64
65   vtkUnsignedShortArray *Table16;
66
67 void MapScalarsThroughTable2(void *input,
68                                           unsigned char *output,
69                                           int inputDataType,
70                                           int numberOfValues,
71                                           int inputIncrement,
72                                           int outputFormat);
73
74 private:
75   vtkLookupTable16(const vtkLookupTable16&);  // Not implemented.
76   void operator=(const vtkLookupTable16&);  // Not implemented.
```

```
77 };
78
79 //----------------------------------------------------------------------------
80 inline unsigned char *vtkLookupTable16::WritePointer(const vtkIdType id,
81                                                      const int number)
82 {
83   //this->InsertTime.Modified();
84   return (unsigned char*)this->Table16->WritePointer(4*id,4*number);
85 }
86
87 #endif
```

11.635 vtkRTStructSetProperties.h File Reference

`#include "vtkObject.h"`
Include dependency graph for vtkRTStructSetProperties.h:



Classes

- class vtkRTStructSetProperties

11.636 vtkRTStructSetProperties.h

Go to the documentation of this file.

```
1 /*=========================================================================
2
3 Program:  GDCM (Grassroots DICOM). A DICOM library
4
5 Copyright (c) 2006-2011 Mathieu Malaterre
6 All rights reserved.
7 See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.
8
9 This software is distributed WITHOUT ANY WARRANTY; without even
10 the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
11 PURPOSE.  See the above copyright notice for more information.
12
13 =========================================================================*/
14 // .NAME vtkRTStructSetProperties - some rtstruct properties.
15 // .SECTION Description
16 //
17 // .SECTION See Also
18 // vtkGDCMPolyDataReader vtkGDCMPolyDataWriter
```

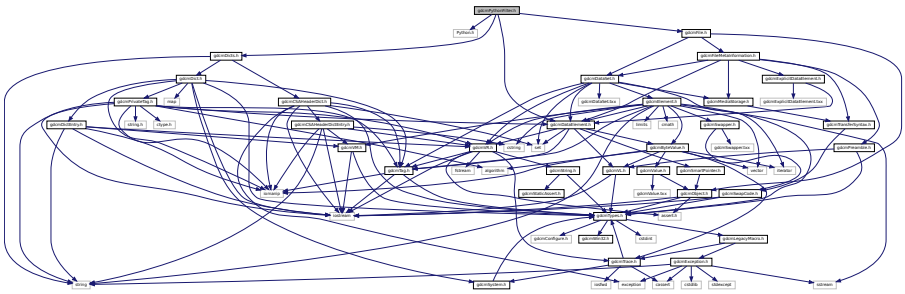
```
19
20 #ifndef VTKRTSTRUCTSETPROPERTIES_H
21 #define VTKRTSTRUCTSETPROPERTIES_H
22
23 #include "vtkObject.h"
24
25 class vtkRTStructSetPropertiesInternals;
26
27 class VTK_EXPORT vtkRTStructSetProperties :  public vtkObject
28 {
29 public:
30   static vtkRTStructSetProperties *New();
31   vtkTypeMacro(vtkRTStructSetProperties,vtkObject);
32   void PrintSelf(ostream& os, vtkIndent indent);
33
34   // Description:
35   // Convenience method to reset all fields to an empty string/value
36   virtual void Clear();
37
38   // Description:
39   //
40   vtkSetStringMacro(StructureSetLabel);
41   vtkGetStringMacro(StructureSetLabel);
42
43   vtkSetStringMacro(StructureSetName);
44   vtkGetStringMacro(StructureSetName);
45
46   vtkSetStringMacro(StructureSetDate);
47   vtkGetStringMacro(StructureSetDate);
48
49   vtkSetStringMacro(StructureSetTime);
50   vtkGetStringMacro(StructureSetTime);
51
52   vtkSetStringMacro(SOPInstanceUID);
53   vtkGetStringMacro(SOPInstanceUID);
54
55   vtkSetStringMacro(StudyInstanceUID);
56   vtkGetStringMacro(StudyInstanceUID);
57
58   vtkSetStringMacro(SeriesInstanceUID);
59   vtkGetStringMacro(SeriesInstanceUID);
60
61   vtkSetStringMacro(ReferenceSeriesInstanceUID);
62   vtkGetStringMacro(ReferenceSeriesInstanceUID);
63
64   vtkSetStringMacro(ReferenceFrameOfReferenceUID);
65   vtkGetStringMacro(ReferenceFrameOfReferenceUID);
66
67   // Description:
68   // Copy the contents of p to this instance.
69   virtual void DeepCopy(vtkRTStructSetProperties *p);
70
71   void AddContourReferencedFrameOfReference( vtkIdType pdnum, const char *classuid , const char * instanceuid
      );
72   const char *GetContourReferencedFrameOfReferenceClassUID( vtkIdType pdnum, vtkIdType id );
73   const char *GetContourReferencedFrameOfReferenceInstanceUID( vtkIdType pdnum, vtkIdType id );
74   vtkIdType GetNumberOfContourReferencedFrameOfReferences();
75   vtkIdType GetNumberOfContourReferencedFrameOfReferences(vtkIdType pdnum);
76
77   void AddReferencedFrameOfReference( const char *classuid , const char * instanceuid );
78   const char *GetReferencedFrameOfReferenceClassUID( vtkIdType id );
79   const char *GetReferencedFrameOfReferenceInstanceUID( vtkIdType id );
80   vtkIdType GetNumberOfReferencedFrameOfReferences();
81
82   void AddStructureSetROI( int roinumber,
83     const char* reframerefuid,
84     const char* roiname,
85     const char* ROIGenerationAlgorithm,
86     const char* ROIDescription = 0
87   );
88   void AddStructureSetROIObservation( int refnumber,
89     int observationnumber,
90     const char *rtroiinterpretedtype,
91     const char *roiinterpreter,
92     const char *roiobservationlabel = 0
93   );
94
95   vtkIdType GetNumberOfStructureSetROIs();
96   int GetStructureSetObservationNumber(vtkIdType id);
97   int GetStructureSetROINumber(vtkIdType id);
98   const char *GetStructureSetROIRefFrameRefUID(vtkIdType);
```

```
99   const char *GetStructureSetROIName(vtkIdType);
100   const char *GetStructureSetROIGenerationAlgorithm(vtkIdType);
101   const char *GetStructureSetROIDescription(vtkIdType id);
102   const char *GetStructureSetRTROIInterpretedType(vtkIdType id);
103   const char *GetStructureSetROIObservationLabel(vtkIdType id);
104
105 protected:
106   vtkRTStructSetProperties();
107   ~vtkRTStructSetProperties();
108
109   char *StructureSetLabel;
110   char *StructureSetName;
111   char *StructureSetDate;
112   char *StructureSetTime;
113
114   char *SOPInstanceUID;
115   char *StudyInstanceUID;
116   char *SeriesInstanceUID;
117
118   char *ReferenceSeriesInstanceUID;
119   char *ReferenceFrameOfReferenceUID;
120
121   // Description:
122   // PIMPL Encapsulation for STL containers
123   //BTX
124   vtkRTStructSetPropertiesInternals *Internals;
125   //ETX
126
127 private:
128   vtkRTStructSetProperties(const vtkRTStructSetProperties&); // Not implemented.
129   void operator=(const vtkRTStructSetProperties&); // Not implemented.
130 };
131
132 #endif
```

11.637 gdcmPythonFilter.h File Reference

```
#include <Python.h>
#include "gdcmDataElement.h"
#include "gdcmDicts.h"
#include "gdcmFile.h"
```

Include dependency graph for gdcmPythonFilter.h:



Classes

- class gdcm::PythonFilter

  PythonFilter PythonFilter is the class that make gdcm2.x looks more like gdcm1 and transform the binary blob contained in a DataElement into a string, typically this is a nice feature to have for wrapped language.

Namespaces

- namespace gdcm

11.638 gdcmPythonFilter.h

Go to the documentation of this file.

```
1 /*=========================================================================
2
3 Program:  GDCM (Grassroots DICOM). A DICOM library
4
5 Copyright (c) 2006-2011 Mathieu Malaterre
6 All rights reserved.
7 See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.
8
9 This software is distributed WITHOUT ANY WARRANTY; without even
10 the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
11 PURPOSE.  See the above copyright notice for more information.
12
13 =========================================================================*/
14 #ifndef GDCMPYTHONFILTER_H
15 #define GDCMPYTHONFILTER_H
16
17 #include <Python.h>
18
19 #include "gdcmDataElement.h"
20 #include "gdcmDicts.h"
21 #include "gdcmFile.h"
22
23 namespace gdcm
24 {
25
31 class GDCM_EXPORT PythonFilter
32 {
33 public:
34   PythonFilter();
35   ~PythonFilter();
36
37   void UseDictAlways(bool ) {}
38
39   // Allow user to pass in there own dicts
40   void SetDicts(const Dicts &dicts);
41
42   // Convert to string the ByteValue contained in a DataElement
43   PyObject *ToPyObject(const Tag& t) const;
44
45   void SetFile(const File& f);
46   File &GetFile();
47   const File &GetFile() const;
48
49 private:
50   SmartPointer<File> F;
51 };
52
53 } // end namespace gdcm
54
55 #endif //GDCMPYTHONFILTER_H
```

Chapter 12

Example Documentation

12.1 TestByteSwap.cxx

This is a C++ example on how to use gdcm::ByteSwap

```
/*=========================================================================

Program:  GDCM (Grassroots DICOM). A DICOM library

Copyright (c) 2006-2011 Mathieu Malaterre
All rights reserved.
See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.

This software is distributed WITHOUT ANY WARRANTY; without even
the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
PURPOSE.  See the above copyright notice for more information.

=========================================================================*/
#include "gdcmTypes.h"
#include "gdcmSwapCode.h"
#include "gdcmByteSwap.h"
#include <string.h> // memcpy
int myfunc()
{
  char vl_str[4];
  const char raw[] = "\000\000\000\004";
  memcpy(vl_str, raw, 4);
  uint32_t vl;
  gdcm::ByteSwap<uint32_t>::SwapRangeFromSwapCodeIntoSystem((uint32_t*)(&vl_str), gdcm::SwapCode::BigEndian, 1);
  memcpy(&vl, vl_str, 4);
  if( vl != 0x00000004 )
    {
    std::cerr « std::hex « "vl:  " « vl « std::endl;
    return 1;
    }
  gdcm::ByteSwap<uint32_t>::SwapFromSwapCodeIntoSystem(vl, gdcm::SwapCode::LittleEndian);
  if( vl != 0x00000004 )
    {
    std::cerr « std::hex « "vl:  " « vl « std::endl;
    return 1;
    }
  gdcm::ByteSwap<uint32_t>::SwapFromSwapCodeIntoSystem(vl, gdcm::SwapCode::BigEndian);
  if( vl != 0x4000000 )
    {
    std::cerr « std::hex « "vl:  " « vl « std::endl;
    return 1;
    }
  return 0;
}
int TestByteSwap(int , char *[])
{
  gdcm::SwapCode sc = gdcm::SwapCode::Unknown;
  if ( gdcm::ByteSwap<uint16_t>::SystemIsBigEndian() )
```

```
  {
  sc = gdcm::SwapCode::BigEndian;
  }
else if ( gdcm::ByteSwap<uint16_t>::SystemIsLittleEndian() )
  {
  sc = gdcm::SwapCode::LittleEndian;
  }
if( sc == gdcm::SwapCode::Unknown )
  {
  std::cerr « "unk" « std::endl;
  return 1;
  }
//std::cout « "sc:  " « sc « std::endl;
uint16_t t = 0x1234;
gdcm::ByteSwap<uint16_t>::SwapFromSwapCodeIntoSystem(t, sc);
if( sc == gdcm::SwapCode::BigEndian )
  {
  if( t != 0x3412 )
    {
    std::cerr « std::hex « "t:  " « t « std::endl;
    return 1;
    }
  // ok test pass rest value to old one
  t = 0x1234;
  }
else if ( sc == gdcm::SwapCode::LittleEndian )
  {
  if( t != 0x1234 )
    {
    std::cerr « std::hex « "t:  " « t « std::endl;
    return 1;
    }
  }
union { char n[2]; uint16_t tn; } u16;
memcpy(u16.n, &t, 2 );
gdcm::ByteSwap<uint16_t>::SwapRangeFromSwapCodeIntoSystem(&u16.tn, sc, 1);
uint16_t tn = u16.tn;
if( sc == gdcm::SwapCode::BigEndian )
  {
  if( tn != 0x3412 )
    {
    std::cerr « std::hex « "tn:  " « tn « std::endl;
    return 1;
    }
  // ok test pass rest value to old one
  t = 0x1234;
  }
else if ( sc == gdcm::SwapCode::LittleEndian )
  {
  if( tn != 0x1234 )
    {
    std::cerr « std::hex « "tn:  " « tn « std::endl;
    return 1;
    }
  }
gdcm::ByteSwap<uint16_t>::SwapRangeFromSwapCodeIntoSystem(&u16.tn, gdcm::SwapCode::BigEndian, 1);
tn = u16.tn;
if( sc == gdcm::SwapCode::LittleEndian )
  {
  if( tn != 0x3412 )
    {
    std::cerr « std::hex « "tn:  " « tn « std::endl;
    return 1;
    }
  }
else if ( sc == gdcm::SwapCode::BigEndian )
  {
  if( tn != 0x1234 )
    {
    std::cerr « std::hex « "tn:  " « tn « std::endl;
    return 1;
    }
  }
if( myfunc() )
  {
  return 1;
  }
uint16_t array[] = { 0x1234 };
gdcm::ByteSwap<uint16_t>::SwapRangeFromSwapCodeIntoSystem(array,
  gdcm::SwapCode::BigEndian,1);
 if ( array[0] != 0x3412 )
```

```
    {
    std::cerr « std::hex « "array:  " « array[0] « std::endl;
    return 1;
    }
  return 0;
}
```

12.2 PatchFile.cxx

This is a C++ example on how to use gdcm::Attribute

```
/*=========================================================================

Program:  GDCM (Grassroots DICOM). A DICOM library

Copyright (c) 2006-2011 Mathieu Malaterre
All rights reserved.
See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.

This software is distributed WITHOUT ANY WARRANTY; without even
the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
PURPOSE.  See the above copyright notice for more information.

=========================================================================*/
/*
* The image was a broken file where the Pixel Data element was 8 times too big
* Apparently multiplying the BitsAllocated to 4 and multiplying the number of
* frames by 2 would solve the problem
*
* This C++ code can be used to patch the header.
*/
#include "gdcmReader.h"
#include "gdcmImageReader.h"
#include "gdcmWriter.h"
#include "gdcmDataSet.h"
#include "gdcmAttribute.h"
int main(int argc, char *argv[])
{
  if( argc < 3 )
    {
    return 1;
    }
  const char *f = argv[1];
  const char *out = argv[2];
  gdcm::Reader r;
  r.SetFileName( f );
  if( !r.Read() )
    {
    return 1;
    }
  gdcm::File &file = r.GetFile();
  gdcm::DataSet& ds = file.GetDataSet();
  // (0028,0100) US 16                                       #   2, 1 BitsAllocated
  // (0028,0101) US 16                                       #   2, 1 BitsStored
  // (0028,0102) US 15                                       #   2, 1 HighBit
  //
    {
    gdcm::Attribute<0x28,0x100> at;
    at.SetFromDataElement( ds.GetDataElement( at.GetTag() ) );
    if( at.GetValue() != 8 )
      {
      return 1;
      }
    at.SetValue( 32 );
    ds.Replace( at.GetAsDataElement() );
    }
    {
    gdcm::Attribute<0x28,0x101> at;
    at.SetFromDataElement( ds.GetDataElement( at.GetTag() ) );
    if( at.GetValue() != 8 )
      {
      return 1;
      }
    at.SetValue( 32 );
    ds.Replace( at.GetAsDataElement() );
    }
```

```
    {
    gdcm::Attribute<0x28,0x102> at;
    at.SetFromDataElement( ds.GetDataElement( at.GetTag() ) );
    if( at.GetValue() != 7 )
      {
      return 1;
      }
    at.SetValue( 31 );
    ds.Replace( at.GetAsDataElement() );
    }
  // (0028,0008) IS [56]                                    #   2, 1 NumberOfFrames
    {
    gdcm::Attribute<0x28,0x8> at;
    at.SetFromDataElement( ds.GetDataElement( at.GetTag() ) );
    at.SetValue( at.GetValue() * 2 );
    ds.Replace( at.GetAsDataElement() );
    }
  gdcm::Writer w;
  w.SetFile( file );
  w.SetCheckFileMetaInformation( false );
  w.SetFileName( out );
  if( !w.Write() )
    {
    return 1;
    }
  // Now let's see if we can read it as an image:
  gdcm::ImageReader ir;
  ir.SetFileName( out );
  if(!ir.Read())
    {
    return 1;
    }
  gdcm::Image &image = ir.GetImage();
  unsigned long len = image.GetBufferLength();
  const gdcm::ByteValue *bv = ir.GetFile().GetDataSet().GetDataElement( gdcm::Tag(0x7fe0,0x0010)
      ).GetByteValue();
  if( !bv || len != bv->GetLength() )
    {
    return 1;
    }
  std::cout « bv->GetLength() « " " « len « std::endl;
  std::cout « "Sucess to rewrite image !" « std::endl;
  image.Print( std::cout );
  return 0;
}
```

12.3 SimplePrint.cs

This is a C# example on how to use gdcm::SWIGDataSet

```
/*=========================================================================

Program:  GDCM (Grassroots DICOM). A DICOM library

Copyright (c) 2006-2011 Mathieu Malaterre
All rights reserved.
See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.

This software is distributed WITHOUT ANY WARRANTY; without even
the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
PURPOSE.  See the above copyright notice for more information.

=========================================================================*/
/*
Convertor convertor = new Convertor();
int a = convertor.Convert<int>( some_int_blob );
double b = convertor.Convert<double>( some_double_blob );
*/
/*
* Usage:
* $ export LD_LIBRARY_PATH=$HOME/Projects/gdcm/debug-gcc/bin
* $ mono bin/SimplePrint.exe gdcmData/012345.002.050.dcm
*/
using System;
using gdcm;
public class SimplePrint
```

```
{
  public static void RecurseDataSet(File f, DataSet ds, string indent)
    {
    CSharpDataSet cds = new CSharpDataSet(ds);
    while(!cds.IsAtEnd())
      {
      DataElement de = cds.GetCurrent();
      // Compute VR from the toplevel file, and the currently processed dataset:
      VR vr = DataSetHelper.ComputeVR(f, ds, de.GetTag() );
      if( vr.Compatible( new VR(VR.VRType.SQ) ) )
        {
        uint uvl = (uint)de.GetVL(); // Test cast is ok
        System.Console.WriteLine( indent + de.GetTag().toString() + ":" + uvl ); // why not ?
        //SequenceOfItems sq = de.GetSequenceOfItems();
        // GetValueAsSQ handle more cases than GetSequenceOfItems
        SmartPtrSQ sq = de.GetValueAsSQ();
        uint n = sq.GetNumberOfItems();
        for( uint i = 1; i <= n; i++) // item starts at 1, not 0
          {
          Item item = sq.GetItem( i );
          DataSet nested = item.GetNestedDataSet();
          RecurseDataSet( f, nested, indent + "  " );
          }
        }
      else
        {
        System.Console.WriteLine( indent + de.toString() );
        }
      cds.Next();
      }
    }
  public static int Main(string[] args)
    {
    string filename = args[0];
    Reader reader = new Reader();
    reader.SetFileName( filename );
    bool ret = reader.Read();
    if( !ret )
      {
      return 1;
      }
    File f = reader.GetFile();
    DataSet ds = f.GetDataSet();
    RecurseDataSet( f, ds, "" );
    return 0;
    }
}
```

12.4 TestReader.cxx

```
/*=========================================================================

Program:  GDCM (Grassroots DICOM). A DICOM library

Copyright (c) 2006-2011 Mathieu Malaterre
All rights reserved.
See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.

This software is distributed WITHOUT ANY WARRANTY; without even
the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
PURPOSE.  See the above copyright notice for more information.

=========================================================================*/
#include "gdcmReader.h"
#include "gdcmFileMetaInformation.h"
#include "gdcmFile.h"
#include "gdcmTesting.h"
#include "gdcmMediaStorage.h"
int TestRead(const char* filename, bool verbose = false)
{
  if( verbose )
  std::cout « "TestRead:  " « filename « std::endl;
  gdcm::Reader reader;
  reader.SetFileName( filename );
  if ( !reader.Read() )
    {
    std::cerr « "TestReadError:  Failed to read:  " « filename « std::endl;
```

```
    return 1;
    }
//commenting out the fmi and ds to avoid warnings
  //const gdcm::FileMetaInformation &h = reader.GetFile().GetHeader();
  //std::cout « h « std::endl;
  //const gdcm::DataSet &ds = reader.GetFile().GetDataSet();
  //std::cout « ds « std::endl;
  const char *ref = gdcm::Testing::GetMediaStorageFromFile(filename);
  gdcm::MediaStorage ms;
  ms.SetFromFile( reader.GetFile() );
  if( !ref )
    {
    std::cerr « "TestReadError:  Missing MediaStorage:  " « filename « std::endl;
    std::cerr « "It should be:  " « ms « std::endl;
    return 1;
    }
  if( ms.IsUndefined() && ref && *ref != 0 )
    {
    std::cerr « "TestReadError:  MediaStorage:  " « filename « std::endl;
    std::cerr « "It should be instead:  " « ref « std::endl;
    return 1;
    }
  // Make sure it is the right one:
  if( ref && *ref != 0 && ms != gdcm::MediaStorage::GetMSType(ref) )
    {
    std::cerr « "Error:  Found MediaStorage:  " « ms « " for " « filename « std::endl;
    std::cerr « "It should be instead:  " « ref « std::endl;
    return 1;
    }
  return 0;
}
int TestReader(int argc, char *argv[])
{
  if( argc == 2 )
    {
    const char *filename = argv[1];
    return TestRead(filename, true);
    }
  // else
  gdcm::Trace::DebugOff();
  gdcm::Trace::WarningOff();
  int r = 0, i = 0;
  const char *filename;
  const char * const *filenames = gdcm::Testing::GetFileNames();
  while( (filename = filenames[i]) )
    {
    r += TestRead( filename );
    ++i;
    }
  return r;
}
```

12.5 TestReader.py

This is a C++ example on how to use gdcm::Reader

```
1
14
15 import os,sys
16 import gdcm
17
18 def TestRead(filename, verbose = False):
19   r = gdcm.Reader()
20   r.SetFileName( filename )
21   success = r.Read()
22   #if verbose:  print r.GetFile()
23   if verbose:  print(r.GetFile().GetDataSet())
24   return success
25
26 if __name__ == "__main__":
27   success = 0
28   try:
29     filename = os.sys.argv[1]
30     success += TestRead( filename, True )
31   except:
32     # loop over all files:
```

```
33      gdcm.Trace.DebugOff()
34      gdcm.Trace.WarningOff()
35      t = gdcm.Testing()
36      nfiles = t.GetNumberOfFileNames()
37      for i in range(0,nfiles):
38        filename = t.GetFileName(i)
39        success += TestRead( filename )
40
41
42    # Test succeed ?
43    sys.exit(success == 0)
```

12.6 DecompressJPEGFile.cs

This is a C# example on how to use gdcm::SequenceOfFragments

```
/*=========================================================================

Program:  GDCM (Grassroots DICOM). A DICOM library

Copyright (c) 2006-2011 Mathieu Malaterre
All rights reserved.
See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.

This software is distributed WITHOUT ANY WARRANTY; without even
the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
PURPOSE.  See the above copyright notice for more information.

=========================================================================*/
/*
* Usage:
* $ export LD_LIBRARY_PATH=$HOME/Projects/gdcm/debug-gcc/bin
* $ mono bin/DecompressJPEGFile.exe somejpegfile.jpg
*/
using System;
using gdcm;
public class DecompressJPEGFile
{
  public static int Main(string[] args)
    {
    string file1 = args[0];
    System.IO.FileStream infile =
      new System.IO.FileStream(file1, System.IO.FileMode.Open, System.IO.FileAccess.Read);
    uint fsize = gdcm.PosixEmulation.FileSize(file1);
    byte[] jstream  = new byte[fsize];
    infile.Read(jstream, 0 , jstream.Length);
    Trace.DebugOn();
    Image image = new Image();
    image.SetNumberOfDimensions( 2 ); // important for now
    DataElement pixeldata = new DataElement( new gdcm.Tag(0x7fe0,0x0010) );
    // DO NOT set a ByteValue here, JPEG is a particular kind of encapsulated syntax
    // in which can one cannot use a simple byte array for storage.  Instead, see
    // gdcm.SequenceOfFragments
    //pixeldata.SetByteValue( jstream, new gdcm.VL( (uint)jstream.Length ) );
    // Create a new SequenceOfFragments C++ object, store it as a SmartPointer :
    SmartPtrFrag sq = SequenceOfFragments.New();
    Fragment frag = new Fragment();
    frag.SetByteValue( jstream, new gdcm.VL( (uint)jstream.Length) );
    // Single file => single fragment
    sq.AddFragment( frag );
    // Pass by reference:
    pixeldata.SetValue( sq.__ref__() );
    // insert:
    image.SetDataElement( pixeldata );
    // JPEG use YBR to achieve better compression ratio by default (not RGB)
    // FIXME hardcoded:
    PhotometricInterpretation pi = new PhotometricInterpretation( PhotometricInterpretation.PIType.YBR_FULL );
    image.SetPhotometricInterpretation( pi );
    // FIXME hardcoded:
    PixelFormat pixeltype = new PixelFormat(3,8,8,7);
    image.SetPixelFormat( pixeltype );
    // FIXME hardcoded:
    image.SetTransferSyntax( new TransferSyntax( TransferSyntax.TSType.JPEGLosslessProcess14_1 ) );
    image.SetDimension(0, 692);
    image.SetDimension(1, 721);
    // Decompress !
```

```
    byte[] decompressedData = new byte[(int)image.GetBufferLength()];
    image.GetBuffer(decompressedData);
    // Write out the decompressed bytes
    System.Console.WriteLine(image.toString());
    using (System.IO.Stream stream =
      System.IO.File.Open(@"/tmp/dd.raw",
        System.IO.FileMode.Create))
      {
      System.IO.BinaryWriter writer = new System.IO.BinaryWriter(stream);
      writer.Write(decompressedData);
      }
    return 0;
    }
}
```

12.7 ManipulateFile.cs

```
/*=========================================================================

Program:  GDCM (Grassroots DICOM). A DICOM library

Copyright (c) 2006-2011 Mathieu Malaterre
All rights reserved.
See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.

This software is distributed WITHOUT ANY WARRANTY; without even
the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
PURPOSE.  See the above copyright notice for more information.

=========================================================================*/
/*
* Usage:
* $ export LD_LIBRARY_PATH=$HOME/Projects/gdcm/debug-gcc/bin
* $ mono bin/ManipulateFile.exe gdcmData/012345.002.050.dcm out.dcm
*/
using System;
using gdcm;
public class ManipulateFile
{
  public static int Main(string[] args)
    {
    string file1 = args[0];
    string file2 = args[1];
    Reader reader = new Reader();
    reader.SetFileName( file1 );
    bool ret = reader.Read();
    if( !ret )
      {
      return 1;
      }
    Anonymizer ano = new Anonymizer();
    ano.SetFile( reader.GetFile() );
    ano.RemovePrivateTags();
    ano.RemoveGroupLength();
    Tag t = new Tag(0x10,0x10);
    ano.Replace( t, "GDCM^Csharp^Test^Hello^World" );
    UIDGenerator g = new UIDGenerator();
    ano.Replace( new Tag(0x0008,0x0018), g.Generate() );
    ano.Replace( new Tag(0x0020,0x000d), g.Generate() );
    ano.Replace( new Tag(0x0020,0x000e), g.Generate() );
    ano.Replace( new Tag(0x0020,0x0052), g.Generate() );
    Writer writer = new Writer();
    writer.SetFileName( file2 );
    writer.SetFile( ano.GetFile() );
    ret = writer.Write();
    if( !ret )
      {
      return 1;
      }
    return 0;
    }
}
```

12.8 ClinicalTrialIdentificationWorkflow.cs

This is a C# example on how to use Anonymizer

```
/*=========================================================================

Program:  GDCM (Grassroots DICOM). A DICOM library

Copyright (c) 2006-2011 Mathieu Malaterre
All rights reserved.
See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.

This software is distributed WITHOUT ANY WARRANTY; without even
the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
PURPOSE.  See the above copyright notice for more information.

=========================================================================*/
/*
* Typical usage on UNIX:
* $ export LD_LIBRARY_PATH=$HOME/Projects/gdcm/debug-gcc/bin
* $ mono bin/ClinicalTrialIdentificationWorkflow.exe input_dir output_dir
*/
using System;
using gdcm;
public class MyWatcher :  SimpleSubjectWatcher
{
  public MyWatcher(Subject s):base(s,"Override String"){}
  protected override void StartFilter() {
    System.Console.WriteLine( "This is my start" );
  }
  protected override void EndFilter(){
    System.Console.WriteLine( "This is my end" );
  }
  protected override void ShowProgress(Subject caller, Event evt){
    ProgressEvent pe = ProgressEvent.Cast(evt);
    System.Console.WriteLine( "This is my progress:  " + pe.GetProgress() );
  }
  protected override void ShowIteration(){
    System.Console.WriteLine( "This is my iteration" );
  }
  protected override void ShowAnonymization(Subject caller, Event evt){
/*
* A couple of explanation are necessary here to understand how SWIG work
*  http://www.swig.org/Doc1.3/Java.html#adding_downcasts
*
*  System.Console.WriteLine( "This is my Anonymization.  Type:  " + evt.GetEventName() );
*  System.Type type = evt.GetType();
*  System.Console.WriteLine( "This is my Anonymization.  System.Type:  " + type.ToString() );
*  System.Console.WriteLine( "This is my Anonymization.  CheckEvent:  " + ae.CheckEvent( evt ) );
*  System.Console.WriteLine( "This is my Anonymization.  Processing Tag #" + ae.GetTag().toString() );
*/
    AnonymizeEvent ae = AnonymizeEvent.Cast(evt);
    if( ae != null )
      {
      Tag t = ae.GetTag();
      System.Console.WriteLine( "This is my Anonymization.  Processing Tag #" + t.toString() );
      }
    else
      {
      System.Console.WriteLine( "This is my Anonymization.  Unhandled Event type:  " + evt.GetEventName() );
      }
  }
  protected override void ShowAbort(){
    System.Console.WriteLine( "This is my abort" );
  }
}
public class ClinicalTrialIdentificationWorkflow
{
  public static bool ProcessOneFile( gdcm.Anonymizer ano , string filename, string outfilename )
    {
    Reader reader = new Reader();
    reader.SetFileName( filename );
    bool ret = reader.Read();
    if( !ret )
      {
      return false;
      }
    // Pass in the file:
    ano.SetFile( reader.GetFile() );
    // First step, let's protect all Patient information as per
```

```
    // PS 3.15 / E.1 / Basic Application Level Confidentiality Profile
    if( !ano.BasicApplicationLevelConfidentialityProfile() )
      {
      return false;
      }
    // Now let's pass in all Clinical Trial fields
    // PS 3.3 - 2008 / C.7.1.3 Clinical Trial Subject Module
    /*
Clinical Trial Sponsor Name (0012,0010) 1 The name of the clinical trial sponsor.  See C.7.1.3.1.1.
Clinical Trial Protocol ID (0012,0020) 1 Identifier for the noted protocol.  See C.7.1.3.1.2.
Clinical Trial Protocol Name (0012,0021) 2 The name of the clinical trial protocol.  See C.7.1.3.1.3.
Clinical Trial Site ID (0012,0030) 2 The identifier of the site responsible for submitting clinical trial data.
      See C.7.1.3.1.4.
Clinical Trial Site Name (0012,0031) 2 Name of the site responsible for submitting clinical trial data.  See
      C.7.1.3.1.5
Clinical Trial Subject ID (0012,0040) 1C The assigned identifier for the clinical trial subject.  See
      C.7.1.3.1.6.  Shall be present if Clinical Trial Subject Reading ID (0012,0042) is absent.  May be present
      otherwise.
Clinical Trial Subject Reading ID (0012,0042) 1C Identifies the subject for blinded evaluations.  Shall be
      present if Clinical Trial Subject ID (0012,0040) is absent.    May be present otherwise.  See C.7.1.3.1.7.
*/
    ano.Replace( new gdcm.Tag(0x0012,0x0010), "MySponsorName");
    ano.Replace( new gdcm.Tag(0x0012,0x0020), "MyProtocolID");
    ano.Replace( new gdcm.Tag(0x0012,0x0021), "MyProtocolName");
    ano.Replace( new gdcm.Tag(0x0012,0x0030), "MySiteId");
    ano.Replace( new gdcm.Tag(0x0012,0x0031), "MySiteName");
    ano.Replace( new gdcm.Tag(0x0012,0x0040), "MySponsorId");
    ano.Replace( new gdcm.Tag(0x0012,0x0050), "MyTPId");
    ano.Replace( new gdcm.Tag(0x0012,0x0051), "MyTPDescription");
    // The following two are not required as they are guaranteed to be filled in by the
    // Basic Application Level Confidentiality Profile.  Only override if you understand what
    // you are doing
    //ano.Replace( new gdcm.Tag(0x0012,0x0062), "YES");
    //ano.Replace( new gdcm.Tag(0x0012,0x0063), "My Super Duper Anonymization Overload");
    // We might be generating a subdirectory.  Let's make sure the subdir exist:
    gdcm.Filename fn = new gdcm.Filename( outfilename );
    string subdir = fn.GetPath();
    if( !gdcm.PosixEmulation.MakeDirectory( subdir ) )
      {
      return false;
      }
    gdcm.FileMetaInformation fmi = ano.GetFile().GetHeader();
    // The following three lines make sure to regenerate any value:
    fmi.Remove( new gdcm.Tag(0x0002,0x0012) );
    fmi.Remove( new gdcm.Tag(0x0002,0x0013) );
    fmi.Remove( new gdcm.Tag(0x0002,0x0016) );
    Writer writer = new Writer();
    writer.SetFileName( outfilename );
    writer.SetFile( ano.GetFile() );
    ret = writer.Write();
    if( !ret )
      {
      return false;
      }
    return true;
    }
  public static int Main(string[] args)
    {
    gdcm.FileMetaInformation.SetSourceApplicationEntityTitle( "My ClinicalTrial App" );
    // http://www.oid-info.com/get/1.3.6.1.4.17434
    string THERALYS_ORG_ROOT = "1.3.6.1.4.17434";
    gdcm.UIDGenerator.SetRoot( THERALYS_ORG_ROOT );
    System.Console.WriteLine( "Root dir is now:  " + gdcm.UIDGenerator.GetRoot() );
    gdcm.Global global = gdcm.Global.GetInstance();
    if( !global.LoadResourcesFiles() )
      {
      System.Console.WriteLine( "Could not LoadResourcesFiles" );
      return 1;
      }
    if( args.Length != 2 )
      {
      System.Console.WriteLine( "Usage:" );
      System.Console.WriteLine( "ClinicalTrialIdentificationWorkflow input_dir output_dir" );
      return 1;
      }
    string dir1 = args[0];
    string dir2 = args[1];
    // Check input is valid:
    if( !gdcm.PosixEmulation.FileIsDirectory(dir1) )
      {
      System.Console.WriteLine( "Input directory:  " + dir1 + " does not exist.  Sorry" );
```

```
      return 1;
      }
    if( !gdcm.PosixEmulation.FileIsDirectory(dir2) )
      {
      System.Console.WriteLine( "Output directory:  " + dir2 + " does not exist.  Sorry" );
      return 1;
      }
    // Recursively search all file within this toplevel directory:
    Directory d = new Directory();
    uint nfiles = d.Load( dir1, true );
    if(nfiles == 0) return 1;
    // Let's use the pre-shipped certificate of GDCM.
    string certpath = gdcm.Filename.Join(gdcm.Testing.GetSourceDirectory(),
      "/Testing/Source/Data/certificate.pem" );
    gdcm.CryptoFactory fact = gdcm.CryptoFactory.GetFactoryInstance();
    gdcm.CryptographicMessageSyntax cms = fact.CreateCMSProvider();
    if( !cms.ParseCertificateFile( certpath ) )
      {
      System.Console.WriteLine( "PEM Certificate :  " + certpath + " could not be read.  Sorry" );
      return 1;
      }
    //Anonymizer ano = new Anonymizer();
    // A reference to an actual C++ instance is required here:
    SmartPtrAno sano = Anonymizer.New();
    Anonymizer ano = sano.__ref__();
    //SimpleSubjectWatcher watcher = new SimpleSubjectWatcher(ano, "Anonymizer");
    MyWatcher watcher = new MyWatcher(ano);
    // Explicitly specify the Cryptographic Message Syntax to use:
    ano.SetCryptographicMessageSyntax( cms );
    // Process all filenames:
    FilenamesType filenames = d.GetFilenames();
    for( uint i = 0; i < nfiles; ++i )
      {
      string filename = filenames[ (int)i ];
      string outfilename = filename.Replace( dir1, dir2 );
      System.Console.WriteLine( "Filename:  " + filename );
      System.Console.WriteLine( "Out Filename:  " + outfilename );
      if( !ProcessOneFile( ano , filename, outfilename ) )
        {
        System.Console.WriteLine( "Could not process filename:  " + filename );
        return 1;
        }
      }
    return 0;
    }
}
```

12.9 GenerateDICOMDIR.cs

This is a C# example on how to use DICOMDIRGenerator

```
/*=========================================================================

Program:  GDCM (Grassroots DICOM). A DICOM library

Copyright (c) 2006-2011 Mathieu Malaterre
All rights reserved.
See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.

This software is distributed WITHOUT ANY WARRANTY; without even
the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
PURPOSE.  See the above copyright notice for more information.

=========================================================================*/
/*
* Simple C# example to show how to use DICOMDIRGenerator
*
* Usage:
* $ export LD_LIBRARY_PATH=$HOME/Projects/gdcm/debug-gcc/bin
* $ mono bin/GenerateDICOMDIR.exe path output_filename
*/
using System;
using gdcm;
public class GenerateDICOMDIR
{
  public static int Main(string[] args)
```

```
    {
    string directory = args[0];
    string outfilename = args[1];
    Directory d = new Directory();
    uint nfiles = d.Load( directory, true );
    if(nfiles == 0) return 1;
    //System.Console.WriteLine( "Files:\n" + d.toString() );
    // Implement fast path ?
    // Scanner s = new Scanner();
    string descriptor = "My_Descriptor";
    FilenamesType filenames = d.GetFilenames();
    gdcm.DICOMDIRGenerator gen = new DICOMDIRGenerator();
    gen.SetFilenames( filenames );
    gen.SetDescriptor( descriptor );
    if( !gen.Generate() )
      {
      return 1;
      }
    gdcm.FileMetaInformation.SetSourceApplicationEntityTitle( "GenerateDICOMDIR" );
    gdcm.Writer writer = new Writer();
    writer.SetFile( gen.GetFile() );
    writer.SetFileName( outfilename );
    if( !writer.Write() )
      {
      return 1;
      }
    return 0;
    }
}
```

12.10 GenFakeImage.cxx

```
/*=========================================================================

Program:  GDCM (Grassroots DICOM). A DICOM library

Copyright (c) 2006-2011 Mathieu Malaterre
All rights reserved.
See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.

This software is distributed WITHOUT ANY WARRANTY; without even
the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
PURPOSE.  See the above copyright notice for more information.

=========================================================================*/
#include "gdcmImage.h"
#include "gdcmImageWriter.h"
#include "gdcmFileDerivation.h"
#include "gdcmUIDGenerator.h"
//#include "gdcmImageChangePhotometricInterpretation.h"
/*
* This example shows two things:
* 1.  How to create an image ex-nihilo
* 2.  How to use the gdcm.FileDerivation filter.  This filter is meant to create "DERIVED" image
* object.  FileDerivation has a simple API where you can reference *all* the input image that have been
* used to generate the image.  The API also allows user to specify the purpose of reference (see CID 7202,
* PS 3.16 - 2008), and the image derivation type (CID 7203, PS 3.16 - 2008).
*/
int main(int, char *[])
{
  // Step 1:  Fake Image
  gdcm::SmartPointer<gdcm::Image> im = new gdcm::Image;
  char * buffer = new char[ 256 * 256 * 3];
  char * p = buffer;
  int b = 128;
  //int ybr[3];
  int ybr2[3];
  //int rgb[3];
  for(int r = 0; r < 256; ++r)
    for(int g = 0; g < 256; ++g)
      //for(int b = 0; b < 256; ++b)
      {
      //rgb[0] = r;
      //rgb[1] = g;
      //rgb[1] = 128;
      //rgb[2] = b;
      //ybr[0] = r;
```

```
      //ybr[1] = g;
      //ybr[1] = 128;
      //ybr[2] = b;
      ybr2[0] = r;
      ybr2[1] = g;
      ybr2[1] = 128;
      ybr2[2] = b;
      //gdcm::ImageChangePhotometricInterpretation::YBR2RGB(rgb, ybr);
      //gdcm::ImageChangePhotometricInterpretation::RGB2YBR(ybr2, rgb);
      *p++ = (char)ybr2[0];
      *p++ = (char)ybr2[1];
      *p++ = (char)ybr2[2];
      }
  im->SetNumberOfDimensions( 2 );
  im->SetDimension(0, 256 );
  im->SetDimension(1, 256 );
  im->GetPixelFormat().SetSamplesPerPixel(3);
  //im->SetPhotometricInterpretation( gdcm::PhotometricInterpretation::RGB );
  im->SetPhotometricInterpretation( gdcm::PhotometricInterpretation::YBR_FULL );
  unsigned long l = im->GetBufferLength();
  if( l != 256 * 256 * 3 )
    {
    return 1;
    }
  gdcm::DataElement pixeldata( gdcm::Tag(0x7fe0,0x0010) );
  pixeldata.SetByteValue( buffer, (uint32_t)l );
  delete[] buffer;
  im->SetDataElement( pixeldata );
  gdcm::UIDGenerator uid; // helper for uid generation
  gdcm::SmartPointer<gdcm::File> file = new gdcm::File; // empty file
  // Step 2:  DERIVED object
  gdcm::FileDerivation fd;
  // For the pupose of this execise we will pretend that this image is referencing
  // two source image (we need to generate fake UID for that).
  const char ReferencedSOPClassUID[] = "1.2.840.10008.5.1.4.1.1.7"; // Secondary Capture
  fd.AddReference( ReferencedSOPClassUID, uid.Generate() );
  fd.AddReference( ReferencedSOPClassUID, uid.Generate() );
  // Again for the purpose of the exercise we will pretend that the image is a
  // multiplanar reformat (MPR):
  // CID 7202 Source Image Purposes of Reference
  // {"DCM",121322,"Source image for image processing operation"},
  fd.SetPurposeOfReferenceCodeSequenceCodeValue( 121322 );
  // CID 7203 Image Derivation
  // { "DCM",113072,"Multiplanar reformatting" },
  fd.SetDerivationCodeSequenceCodeValue( 113072 );
  fd.SetFile( *file );
  // If all Code Value are ok the filter will execute properly
  if( !fd.Derive() )
    {
    std::cerr « "Sorry could not derive using input info" « std::endl;
    return 1;
    }
  // We pass both :
  // 1.  the fake generated image
  // 2.  the 'DERIVED' dataset object
  // to the writer.
  gdcm::ImageWriter w;
  w.SetImage( *im );
  w.SetFile( fd.GetFile() );
  // Set the filename:
  w.SetFileName( "ybr2.dcm" );
  if( !w.Write() )
    {
    return 1;
    }
  return 0;
}
```

12.11 ReformatFile.cs

This is a C++ example on how to use FileDerivation

```
/*=========================================================================

Program:  GDCM (Grassroots DICOM). A DICOM library
```

```
Copyright (c) 2006-2011 Mathieu Malaterre
All rights reserved.
See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.

This software is distributed WITHOUT ANY WARRANTY; without even
the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
PURPOSE.  See the above copyright notice for more information.

=========================================================================*/
/*
* Simple C# example
*
* Usage:
* $ export LD_LIBRARY_PATH=$HOME/Projects/gdcm/debug-gcc/bin
* $ mono bin/ReformatFile.exe input.dcm output.dcm
*/
using System;
using gdcm;
public class ReformatFile
{
  public static int Main(string[] args)
    {
    gdcm.FileMetaInformation.SetSourceApplicationEntityTitle( "My Reformat App" );
    // http://www.oid-info.com/get/1.3.6.1.4.17434
    string THERALYS_ORG_ROOT = "1.3.6.1.4.17434";
    gdcm.UIDGenerator.SetRoot( THERALYS_ORG_ROOT );
    System.Console.WriteLine( "Root dir is now:  " + gdcm.UIDGenerator.GetRoot() );
    string filename = args[0];
    string outfilename = args[1];
    Reader reader = new Reader();
    reader.SetFileName( filename );
    if( !reader.Read() )
      {
      System.Console.WriteLine( "Could not read:  " + filename );
      return 1;
      }
    UIDGenerator uid = new UIDGenerator(); // helper for uid generation
    FileDerivation fd = new FileDerivation();
    // For the pupose of this execise we will pretend that this image is referencing
    // two source image (we need to generate fake UID for that).
    string ReferencedSOPClassUID = "1.2.840.10008.5.1.4.1.1.7"; // Secondary Capture
    fd.AddReference( ReferencedSOPClassUID, uid.Generate() );
    fd.AddReference( ReferencedSOPClassUID, uid.Generate() );
    // Again for the purpose of the exercise we will pretend that the image is a
    // multiplanar reformat (MPR):
    // CID 7202 Source Image Purposes of Reference
    // {"DCM",121322,"Source image for image processing operation"},
    fd.SetPurposeOfReferenceCodeSequenceCodeValue( 121322 );
    // CID 7203 Image Derivation
    // { "DCM",113072,"Multiplanar reformatting" },
    fd.SetDerivationCodeSequenceCodeValue( 113072 );
    fd.SetFile( reader.GetFile() );
    // If all Code Value are ok the filter will execute properly
    if( !fd.Derive() )
      {
      return 1;
      }
    gdcm.FileMetaInformation fmi = reader.GetFile().GetHeader();
    // The following three lines make sure to regenerate any value:
    fmi.Remove( new gdcm.Tag(0x0002,0x0012) );
    fmi.Remove( new gdcm.Tag(0x0002,0x0013) );
    fmi.Remove( new gdcm.Tag(0x0002,0x0016) );
    Writer writer = new Writer();
    writer.SetFileName( outfilename );
    writer.SetFile( fd.GetFile() );
    if( !writer.Write() )
      {
      System.Console.WriteLine( "Could not write:  " + outfilename );
      return 1;
      }
    return 0;
    }
}
```

12.12 DecompressImage.cs

This is a C# example on how to use Image

```
/*=========================================================================

Program:  GDCM (Grassroots DICOM). A DICOM library

Copyright (c) 2006-2011 Mathieu Malaterre
All rights reserved.
See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.

This software is distributed WITHOUT ANY WARRANTY; without even
the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
PURPOSE.  See the above copyright notice for more information.

=========================================================================*/
/*
* Usage:
* $ export LD_LIBRARY_PATH=$HOME/Projects/gdcm/debug-gcc/bin
* $ mono bin/DecompressImage.exe gdcmData/012345.002.050.dcm decompress.dcm
*/
using System;
using gdcm;
public class DecompressImage
{
  public static int Main(string[] args)
    {
    string file1 = args[0];
    string file2 = args[1];
    ImageReader reader = new ImageReader();
    reader.SetFileName( file1 );
    bool ret = reader.Read();
    if( !ret )
      {
      return 1;
      }
    // check that one can access a Fragment from C#:
    var de = reader.GetFile().GetDataSet().GetDataElement(new Tag(0x7fe0, 0x0010));
    var sq = de.GetSequenceOfFragments();
    sq.GetFragment(0);
    Image image = new Image();
    Image ir = reader.GetImage();
    image.SetNumberOfDimensions( ir.GetNumberOfDimensions() );
    //Just for fun:
    //int dircos =  ir.GetDirectionCosines();
    //t = gdcm.Orientation.GetType(dircos);
    //int l = gdcm.Orientation.GetLabel(t);
    //System.Console.WriteLine( "Orientation label:" + l );
    // Set the dimensions,
    // 1.  either one at a time
    //image.SetDimension(0, ir.GetDimension(0) );
    //image.SetDimension(1, ir.GetDimension(1) );
    // 2.  the array at once
    uint[] dims = {0, 0};
    // Just for fun let's invert the dimensions:
    dims[0] = ir.GetDimension(1);
    dims[1] = ir.GetDimension(0);
    ir.SetDimensions( dims );
    PixelFormat pixeltype = ir.GetPixelFormat();
    image.SetPixelFormat( pixeltype );
    PhotometricInterpretation pi = ir.GetPhotometricInterpretation();
    image.SetPhotometricInterpretation( pi );
    DataElement pixeldata = new DataElement( new Tag(0x7fe0,0x0010) );
    byte[] str1 = new byte[ ir.GetBufferLength()];
    ir.GetBuffer( str1 );
    //System.Console.WriteLine( ir.GetBufferLength() );
    pixeldata.SetByteValue( str1, new VL( (uint)str1.Length ) );
    //image.SetDataElement( pixeldata );
    ir.SetDataElement( pixeldata );
    ImageWriter writer = new ImageWriter();
    writer.SetFileName( file2 );
    writer.SetFile( reader.GetFile() );
    writer.SetImage( ir );
    ret = writer.Write();
    if( !ret )
      {
      return 1;
      }
    return 0;
    }
}
```

12.13 StandardizeFiles.cs

This is a C++ example on how to use ImageChangeTransferSyntax

```
/*=========================================================================

Program:  GDCM (Grassroots DICOM). A DICOM library

Copyright (c) 2006-2011 Mathieu Malaterre
All rights reserved.
See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.

This software is distributed WITHOUT ANY WARRANTY; without even
the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
PURPOSE.  See the above copyright notice for more information.

=========================================================================*/
/*
* Simple C# example to show how one would 'Standardize' a DICOM File-Set
*
* Usage:
* $ export LD_LIBRARY_PATH=$HOME/Projects/gdcm/debug-gcc/bin
* $ mono bin/StandardizeFiles.exe input_path output_path
*/
using System;
using gdcm;
public class StandardizeFiles
{
  public static bool ProcessOneFile( string filename, string outfilename )
    {
    PixmapReader reader = new PixmapReader();
    reader.SetFileName( filename );
    if( !reader.Read() )
      {
      System.Console.WriteLine( "Could not read:  " + filename );
      return false;
      }
    ImageChangeTransferSyntax change = new ImageChangeTransferSyntax();
    change.SetForce( false ); // do we really want to recompress when input is alread compressed in same alg ?
    change.SetCompressIconImage( false ); // Keep it simple
    change.SetTransferSyntax( new TransferSyntax( TransferSyntax.TSType.JPEG2000Lossless ) );
    change.SetInput( reader.GetPixmap() );
    if( !change.Change() )
      {
      System.Console.WriteLine( "Could not change:  " + filename );
      return false;
      }
    gdcm.FileMetaInformation fmi = reader.GetFile().GetHeader();
    // The following three lines make sure to regenerate any value:
    fmi.Remove( new gdcm.Tag(0x0002,0x0012) );
    fmi.Remove( new gdcm.Tag(0x0002,0x0013) );
    fmi.Remove( new gdcm.Tag(0x0002,0x0016) );
    PixmapWriter writer = new PixmapWriter();
    writer.SetFileName( outfilename );
    writer.SetFile( reader.GetFile() );
    gdcm.Pixmap pixout = ((PixmapToPixmapFilter)change).GetOutput();
    writer.SetPixmap( pixout );
    if( !writer.Write() )
      {
      System.Console.WriteLine( "Could not write:  " + outfilename );
      return false;
      }
    return true;
    }
  public static int Main(string[] args)
    {
    gdcm.FileMetaInformation.SetSourceApplicationEntityTitle( "My Standardize App" );
    // http://www.oid-info.com/get/1.3.6.1.4.17434
    string THERALYS_ORG_ROOT = "1.3.6.1.4.17434";
    gdcm.UIDGenerator.SetRoot( THERALYS_ORG_ROOT );
    System.Console.WriteLine( "Root dir is now:  " + gdcm.UIDGenerator.GetRoot() );
    string dir1 = args[0];
    string dir2 = args[1];
    // Check input is valid:
    if( !gdcm.PosixEmulation.FileIsDirectory(dir1) )
      {
      System.Console.WriteLine( "Input directory:  " + dir1 + " does not exist.  Sorry" );
      return 1;
      }
    if( !gdcm.PosixEmulation.FileIsDirectory(dir2) )
```

```
      {
      System.Console.WriteLine( "Output directory:  " + dir2 + " does not exist.  Sorry" );
      return 1;
      }
    Directory d = new Directory();
    uint nfiles = d.Load( dir1, true );
    if(nfiles == 0) return 1;
    // Process all filenames:
    FilenamesType filenames = d.GetFilenames();
    for( uint i = 0; i < nfiles; ++i )
      {
      string filename = filenames[ (int)i ];
      string outfilename = filename.Replace( dir1, dir2 );
      System.Console.WriteLine( "Filename:  " + filename );
      System.Console.WriteLine( "Out Filename:  " + outfilename );
      if( !ProcessOneFile( filename, outfilename ) )
        {
        System.Console.WriteLine( "Could not process filename:  " + filename );
        //return 1;
        }
      }
    return 0;
    }
}
```

12.14 ScanDirectory.cs

This is a C# example on how to use Scanner

```
/*=========================================================================

Program:  GDCM (Grassroots DICOM). A DICOM library

Copyright (c) 2006-2011 Mathieu Malaterre
All rights reserved.
See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.

This software is distributed WITHOUT ANY WARRANTY; without even
the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
PURPOSE.  See the above copyright notice for more information.

=========================================================================*/
/*
* Usage:
* $ bin/ScanDirectory.exe /path/to/gdcmData/
*/
using System;
using gdcm;
// We will print each filename being processed
public class MyWatcher :  SimpleSubjectWatcher
{
  public MyWatcher(Subject s):base(s,"Override String"){}
  protected override void ShowFileName(Subject caller, Event evt){
    FileNameEvent fne = FileNameEvent.Cast(evt);
    if( fne != null )
      {
      string fn = fne.GetFileName();
      System.Console.WriteLine( "This is my Scanner.  Processing FileName:  " + fn );
      }
    else
      {
      System.Console.WriteLine( "This is my Anonymization.  Unhandled Event type:  " + evt.GetEventName() );
      }
  }
}
public class ScanDirectory
{
  public static int Main(string[] args)
    {
    string directory = args[0];
    Tag t = new Tag(0x8,0x80);
    Directory d = new Directory();
    uint nfiles = d.Load( directory );
    if(nfiles == 0) return 1;
    //System.Console.WriteLine( "Files:\n" + d.toString() );
    // Use a StrictScanner, need to use a reference to pass the C++ pointer to
```

```
    // MyWatcher implementation
    SmartPtrStrictScan sscan = StrictScanner.New();
    StrictScanner s = sscan.__ref__();
    MyWatcher watcher = new MyWatcher(s);
    s.AddTag( t );
    bool b = s.Scan( d.GetFilenames() );
    if(!b) return 1;
    for(int i = 0; i < (int)nfiles; ++i)
      {
      if( !s.IsKey( d.GetFilenames()[i] ) )
        {
        System.Console.WriteLine( "File is not DICOM or could not be read:  " + d.GetFilenames()[i] );
        }
      }
    System.Console.WriteLine( "Scan:\n" + s.toString() );
    System.Console.WriteLine( "success" );
    return 0;
    }
}
```

12.15 BasicAnonymizer.cs

```
/*=========================================================================

Program:  GDCM (Grassroots DICOM). A DICOM library

Copyright (c) 2006-2011 Mathieu Malaterre
All rights reserved.
See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.

This software is distributed WITHOUT ANY WARRANTY; without even
the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
PURPOSE.  See the above copyright notice for more information.

=========================================================================*/
/*
* Usage:
* $ export LD_LIBRARY_PATH=$HOME/Projects/gdcm/debug-gcc/bin
* $ mono bin/BasicAnonymizer.exe gdcmData/012345.002.050.dcm out.dcm
*/
using System;
using gdcm;
public class MyWatcher :  SimpleSubjectWatcher
{
  public MyWatcher(Subject s):base(s,"Override String"){}
  protected override void StartFilter() {
    System.Console.WriteLine( "This is my start" );
  }
  protected override void EndFilter(){
    System.Console.WriteLine( "This is my end" );
  }
  protected override void ShowProgress(Subject caller, Event evt){
    ProgressEvent pe = ProgressEvent.Cast(evt);
    System.Console.WriteLine( "This is my progress:  " + pe.GetProgress() );
  }
  protected override void ShowIteration(){
    System.Console.WriteLine( "This is my iteration" );
  }
  protected override void ShowAnonymization(Subject caller, Event evt){
/*
* A couple of explanation are necessary here to understand how SWIG work
*  http://www.swig.org/Doc1.3/Java.html#adding_downcasts
*
*  System.Console.WriteLine( "This is my Anonymization.  Type:  " + evt.GetEventName() );
*  System.Type type = evt.GetType();
*  System.Console.WriteLine( "This is my Anonymization.  System.Type:  " + type.ToString() );
*  System.Console.WriteLine( "This is my Anonymization.  CheckEvent:  " + ae.CheckEvent( evt ) );
*  System.Console.WriteLine( "This is my Anonymization.  Processing Tag #" + ae.GetTag().toString() );
*/
    AnonymizeEvent ae = AnonymizeEvent.Cast(evt);
    if( ae != null )
      {
      Tag t = ae.GetTag();
      System.Console.WriteLine( "This is my Anonymization.  Processing Tag #" + t.toString() );
      }
    else
      {
```

```
      System.Console.WriteLine( "This is my Anonymization.  Unhandled Event type:  " + evt.GetEventName() );
      }
  }
  protected override void ShowAbort(){
    System.Console.WriteLine( "This is my abort" );
  }
}
public class BasicAnonymizer
{
  public static int Main(string[] args)
    {
    gdcm.Global global = gdcm.Global.GetInstance();
    if( !global.LoadResourcesFiles() )
      {
      System.Console.WriteLine( "Could not LoadResourcesFiles" );
      return 1;
      }
    string file1 = args[0];
    string file2 = args[1];
    Reader reader = new Reader();
    reader.SetFileName( file1 );
    bool ret = reader.Read();
    if( !ret )
      {
      return 1;
      }
    string certpath = gdcm.Filename.Join(gdcm.Testing.GetSourceDirectory(),
      "/Testing/Source/Data/certificate.pem" );
    gdcm.CryptoFactory fact = gdcm.CryptoFactory.GetFactoryInstance();
    gdcm.CryptographicMessageSyntax cms = fact.CreateCMSProvider();
    if( !cms.ParseCertificateFile( certpath ) )
      {
      return 1;
      }
    //Anonymizer ano = new Anonymizer();
    SmartPtrAno sano = Anonymizer.New();
    Anonymizer ano = sano.__ref__();
    //SimpleSubjectWatcher watcher = new SimpleSubjectWatcher(ano, "Anonymizer");
    MyWatcher watcher = new MyWatcher(ano);
    ano.SetFile( reader.GetFile() );
    ano.SetCryptographicMessageSyntax( cms );
    if( !ano.BasicApplicationLevelConfidentialityProfile() )
      {
      return 1;
      }
    Writer writer = new Writer();
    writer.SetFileName( file2 );
    writer.SetFile( ano.GetFile() );
    ret = writer.Write();
    if( !ret )
      {
      return 1;
      }
    return 0;
    }
}
```

12.16 BasicImageAnonymizer.cs

```
/*=========================================================================

Program:  GDCM (Grassroots DICOM). A DICOM library

Copyright (c) 2006-2011 Mathieu Malaterre
All rights reserved.
See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.

This software is distributed WITHOUT ANY WARRANTY; without even
the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
PURPOSE.  See the above copyright notice for more information.

=========================================================================*/
/*
*/
using System;
using gdcm;
public class BasicImageAnonymizer
```

```
{
  public static int Main(string[] args)
    {
    string filename = args[0];
    // instanciate the reader:
    gdcm.ImageReader reader = new gdcm.ImageReader();
    reader.SetFileName( filename );
    if (!reader.Read()) return 1;
    Image ir = reader.GetImage();
    uint[] dims = {0, 0, 0};
    dims[0] = ir.GetDimension(0);
    dims[1] = ir.GetDimension(1);
    dims[2] = ir.GetDimension(2);
    System.Console.WriteLine( "Dim:" + dims[0] );
    System.Console.WriteLine( "Dim:" + dims[1] );
    System.Console.WriteLine( "Dim:" + dims[2] );
    // buffer to get the pixels
    byte[] buffer = new byte[ ir.GetBufferLength()];
    System.Console.WriteLine( "Dim:" + ir.GetBufferLength() );
    ir.GetBuffer( buffer );
    for (uint z = 0; z < dims[2]; z++)
      {
      for (uint y = 0; y < dims[1] / 2; y++) // only half Y
        {
        for (uint x = 0; x < dims[0] / 2; x++) // only half X
          {
          buffer[ (z * dims[1] + y) * dims[0] + x ] = 0; // works when pixel type == UINT8
          }
        }
      }
    DataElement pixeldata = new DataElement( new Tag(0x7fe0,0x0010) );
    pixeldata.SetByteValue( buffer, new VL( (uint)buffer.Length ) );
    ir.SetDataElement( pixeldata );
    ir.SetTransferSyntax( new TransferSyntax( TransferSyntax.TSType.ExplicitVRLittleEndian ) );
    ImageChangeTransferSyntax change = new ImageChangeTransferSyntax();
    change.SetTransferSyntax( new TransferSyntax( TransferSyntax.TSType.JPEGLSLossless ) );
    change.SetInput( ir );
    if( !change.Change() )
      {
      System.Console.WriteLine( "Could not change:  " + filename );
      return 1;
      }
    ImageWriter writer = new ImageWriter();
    writer.SetFileName( "out.dcm" );
    writer.SetFile( reader.GetFile() );
    writer.SetImage( change.GetOutput() );
    bool ret = writer.Write();
    if( !ret )
      {
      return 1;
      }
    return 0;
    }
}
```

12.17 CompressLossyJPEG.cs

```
/*=========================================================================

Program:  GDCM (Grassroots DICOM). A DICOM library

Copyright (c) 2006-2011 Mathieu Malaterre
All rights reserved.
See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.

This software is distributed WITHOUT ANY WARRANTY; without even
the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
PURPOSE.  See the above copyright notice for more information.

=========================================================================*/
/*
* Usage:
* $ export LD_LIBRARY_PATH=$HOME/Perso/gdcm/debug-gcc/bin
* $ mono bin/CompressLossyJPEG.exe input.dcm output.dcm
*/
using System;
using gdcm;
```

```
public class CompressLossyJPEG
{
  public static int Main(string[] args)
    {
    if( args.Length < 2 )
      {
      System.Console.WriteLine( " input.dcm output.dcm" );
      return 1;
      }
    string filename = args[0];
    string outfilename = args[1];
    ImageReader reader = new ImageReader();
    reader.SetFileName( filename );
    if( !reader.Read() )
      {
      System.Console.WriteLine( "Could not read:  " + filename );
      return 1;
      }
    // The output of gdcm::Reader is a gdcm::File
    File file = reader.GetFile();
    // the dataset is the the set of element we are interested in:
    DataSet ds = file.GetDataSet();
    Image image = reader.GetImage();
    //image.Print( cout );
    ImageChangeTransferSyntax change = new ImageChangeTransferSyntax();
    TransferSyntax targetts =  new TransferSyntax( TransferSyntax.TSType.JPEGBaselineProcess1 );
    change.SetTransferSyntax( targetts );
    // Setup our JPEGCodec, warning it should be compatible with JPEGBaselineProcess1
    JPEGCodec jpegcodec = new JPEGCodec();
    if( !jpegcodec.CanCode( targetts ) )
      {
      System.Console.WriteLine( "Something went really wrong, JPEGCodec cannot handle JPEGBaselineProcess1" );
      return 1;
      }
    jpegcodec.SetLossless( false );
    jpegcodec.SetQuality( 50 ); // poor quality !
    change.SetUserCodec( jpegcodec ); // specify the codec to use to the ImageChangeTransferSyntax
    change.SetInput( image );
    bool b = change.Change();
    if( !b )
      {
      System.Console.WriteLine( "Could not change the Transfer Syntax" );
      return 1;
      }
    ImageWriter writer = new ImageWriter();
    writer.SetImage( (gdcm.Image)change.GetOutput() );
    writer.SetFile( reader.GetFile() );
    writer.SetFileName( outfilename );
    if( !writer.Write() )
      {
      System.Console.WriteLine( "Could not write:  " + outfilename );
      return 1;
      }
    return 0;
    }
}
```

12.18 DecompressImageMultiframe.cs

```
/*=========================================================================

Program:  GDCM (Grassroots DICOM). A DICOM library

Copyright (c) 2006-2011 Mathieu Malaterre
All rights reserved.
See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.

This software is distributed WITHOUT ANY WARRANTY; without even
the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
PURPOSE.  See the above copyright notice for more information.

=========================================================================*/
/*
$ gdcminfo ~/Desktop/angiogram-06.dcm
MediaStorage is 1.2.840.10008.5.1.4.1.1.12.1 [X-Ray Angiographic Image Storage]
TransferSyntax is 1.2.840.10008.1.2.4.50 [JPEG Baseline (Process 1):  Default Transfer Syntax for Lossy JPEG 8
      Bit Image Compression]
```

```
NumberOfDimensions:  3
Dimensions:  (512,512,355)
Origin:  (0,0,0)
Spacing:  (1,1,40)
DirectionCosines:  (1,0,0,0,1,0)
Rescale Intercept/Slope:  (0,1)
SamplesPerPixel    :1
BitsAllocated      :8
BitsStored         :8
HighBit            :7
PixelRepresentation:0
ScalarType found   :UINT8
PhotometricInterpretation:  MONOCHROME2
PlanarConfiguration:  0
TransferSyntax:  1.2.840.10008.1.2.4.50
Orientation Label:  AXIAL
*/
/*
* Description:
*
* Assume we have a file angiogram-06.dcm as described above.
* the following program will decompress directly from the extracted jpeg stream.
*
* First step extract the jpeg stream (but not the Basic Offset Table):
*
* $ gdcmraw -i angiogram-06.dcm -o /tmp/output/chris --split-frags --pattern %d.jpg
*
* Check that indeed there are 355 files, while there are 356 fragments in the original DICOM file, since
* gdcmraw always skip the first fragment (Basic Offset Table).
*
* Now from those individual jpeg stream, recreate a fake gdcm.DataElement...
*
* Usage:
*
* $ export LD_LIBRARY_PATH=$HOME/Projects/gdcm/debug-gcc/bin
* $ mono ./bin/DecompressImageMultiframe.exe /tmp/output
*/
using System;
using gdcm;
public class DecompressImageMultiframe
{
  public static int Main(string[] args)
    {
    string directory = args[0];
    gdcm.Directory dir = new gdcm.Directory();
    uint nfiles = dir.Load(directory);
    //System.Console.WriteLine(dir.toString());
    gdcm.FilenamesType filenames = dir.GetFilenames();
    Image image = new Image();
    image.SetNumberOfDimensions( 3 ); // important for now
    DataElement pixeldata = new DataElement( new gdcm.Tag(0x7fe0,0x0010) );
    // Create a new SequenceOfFragments C++ object, store it as a SmartPointer :
    SmartPtrFrag sq = SequenceOfFragments.New();
    // Yeah, the file are not garantee to be in order, please adapt...
    for(uint i = 0; i < nfiles; ++i)
      {
      System.Console.WriteLine( filenames[(int)i] );
      string file = filenames[(int)i];
      System.IO.FileStream infile =
        new System.IO.FileStream(file, System.IO.FileMode.Open, System.IO.FileAccess.Read);
      uint fsize = gdcm.PosixEmulation.FileSize(file);
      byte[] jstream  = new byte[fsize];
      infile.Read(jstream, 0 , jstream.Length);
      Fragment frag = new Fragment();
      frag.SetByteValue( jstream, new gdcm.VL( (uint)jstream.Length) );
      sq.AddFragment( frag );
      }
    // Pass by reference:
    pixeldata.SetValue( sq.__ref__() );
    // insert:
    image.SetDataElement( pixeldata );
    // JPEG use YBR to achieve better compression ratio by default (not RGB)
    // FIXME hardcoded:
    PhotometricInterpretation pi = new PhotometricInterpretation( PhotometricInterpretation.PIType.MONOCHROME2
      );
    image.SetPhotometricInterpretation( pi );
    // FIXME hardcoded:
    PixelFormat pixeltype = new PixelFormat(1,8,8,7);
    image.SetPixelFormat( pixeltype );
    // FIXME hardcoded:
    image.SetTransferSyntax( new TransferSyntax( TransferSyntax.TSType.JPEGLosslessProcess14_1 ) );
```

```
    image.SetDimension(0, 512);
    image.SetDimension(1, 512);
    image.SetDimension(2, 355);
    // Decompress !
    byte[] decompressedData = new byte[(int)image.GetBufferLength()];
    image.GetBuffer(decompressedData);
    // Write out the decompressed bytes
    System.Console.WriteLine(image.toString());
    using (System.IO.Stream stream =
      System.IO.File.Open(@"/tmp/dd.raw",
        System.IO.FileMode.Create))
      {
      System.IO.BinaryWriter writer = new System.IO.BinaryWriter(stream);
      writer.Write(decompressedData);
      }
    return 0;
    }
}
```

12.19 DumpCSA.cs

```
/*=========================================================================

Program:  GDCM (Grassroots DICOM). A DICOM library

Copyright (c) 2006-2011 Mathieu Malaterre
All rights reserved.
See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.

This software is distributed WITHOUT ANY WARRANTY; without even
the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
PURPOSE.  See the above copyright notice for more information.

=========================================================================*/
/*
* Usage:
* $ bin/DumpCSA.exe input.dcm
*/
using System;
using gdcm;
public class DumpCSA
{
  public static int Main(string[] args)
    {
    string filename = args[0];
    gdcm.Reader reader = new gdcm.Reader();
    reader.SetFileName( filename );
    if (!reader.Read()) return 1;
    gdcm.File f = reader.GetFile();
    gdcm.DataSet ds = f.GetDataSet();
    string[] expectedSiemensTags = new string[] { "B_value" , "AcquisitionMatrixText" };
    using (PrivateTag gtag = CSAHeader.GetCSAImageHeaderInfoTag())
    {
        if (ds.FindDataElement(gtag))
        {
            using (DataElement de = ds.GetDataElement(gtag))
            {
                if (de != null && !de.IsEmpty())
                {
                    using (CSAHeader csa = new CSAHeader())
                    {
                        if (csa.LoadFromDataElement(de))
                        {
                            foreach (string str in expectedSiemensTags)
                            {
                                if (csa.FindCSAElementByName(str))
                                {
                                    using (CSAElement elem = csa.GetCSAElementByName(str))
                                    {
                                        if (elem != null)
                                        {
                                            System.Console.WriteLine( elem.toString() );
                                        }
                                    }
                                }
                            }
                        }
```

```
                    }
                }
            }
        }
    }
    return 0;
    }
}
```

12.20 ExtractEncapsulatedFile.cs

```
/*=========================================================================

Program:  GDCM (Grassroots DICOM). A DICOM library

Copyright (c) 2006-2011 Mathieu Malaterre
All rights reserved.
See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.

This software is distributed WITHOUT ANY WARRANTY; without even
the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
PURPOSE.  See the above copyright notice for more information.

=========================================================================*/
/*
* This example shows how one from C# context can extract a binary blob
* and write out as a file.
* This example is meant for pdf encapsulated file, but can be adapted for other type
* of binary blob.
*
* DICOM file is:
* ...
* (0042,0010) ST (no value available)                     #   0, 0 DocumentTitle
* (0042,0011) OB 25\50\44\46\2d\31\2e\32\20\0d\25\e2\e3\cf\d3\20\0d\31\30\20\30\20...  # 40718, 1
      EncapsulatedDocument
* (0042,0012) LO [application/pdf]                       #  16, 1 MIMETypeOfEncapsulatedDocument
* ...
*
* Usage:
* $ export LD_LIBRARY_PATH=$HOME/Projects/gdcm/debug-gcc/bin
* $ mono bin/ExtractEncapsulatedFile.exe some_pdf_encapsulated.dcm
*/
using System;
using gdcm;
public class ExtractEncapsulatedFile
{
  public static int Main(string[] args)
    {
    string file = args[0];
    Reader reader = new Reader();
    reader.SetFileName( file );
    bool ret = reader.Read();
    if( !ret )
      {
      return 1;
      }
    File f = reader.GetFile();
    DataSet ds = f.GetDataSet();
    Tag tencapsulated_stream = new Tag(0x0042,0x0011); // Encapsulated Document
    if( !ds.FindDataElement( tencapsulated_stream ) )
      {
      return 1;
      }
    // else
    DataElement de = ds.GetDataElement( tencapsulated_stream );
    ByteValue bv = de.GetByteValue();
    uint len = bv.GetLength();
    byte[] encapsulated_stream = new byte[len];
    bv.GetBuffer( encapsulated_stream, len );
    // Write out the decompressed bytes
    //System.Console.WriteLine(image.toString());
    using (System.IO.Stream stream =
      System.IO.File.Open(@"/tmp/dd.pdf",
        System.IO.FileMode.Create))
      {
      System.IO.BinaryWriter writer = new System.IO.BinaryWriter(stream);
      writer.Write( encapsulated_stream );
```

```
      }
    return 0;
    }
}
```

12.21 ExtractImageRegion.cs

```
/*=========================================================================

Program:  GDCM (Grassroots DICOM). A DICOM library

Copyright (c) 2006-2011 Mathieu Malaterre
All rights reserved.
See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.

This software is distributed WITHOUT ANY WARRANTY; without even
the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
PURPOSE.  See the above copyright notice for more information.

=========================================================================*/
/*
* This small code shows how to use the gdcm.ImageRegionReader API
* In this example we are taking each frame by frame and dump them to
* /tmp/frame.raw.
*
* Usage:
* $ bin/ExtractImageRegion.exe input.dcm
*
* Example:
* $ bin/ExtractImageRegion.exe gdcmData/012345.002.050.dcm
* $ md5sum /tmp/frame.raw
* d594a5e2fde12f32b6633ca859b4d4a6  /tmp/frame.raw
* $ gdcminfo --md5sum gdcmData/012345.002.050.dcm
* [...]
* md5sum:  d594a5e2fde12f32b6633ca859b4d4a6
*/
using System;
using gdcm;
public class ExtractImageRegion
{
  public static int Main(string[] args)
    {
    string filename = args[0];
    uint file_size = gdcm.PosixEmulation.FileSize(filename);
    // instantiate the reader:
    gdcm.ImageRegionReader reader = new gdcm.ImageRegionReader();
    reader.SetFileName( filename );
    // pull DICOM info:
    if (!reader.ReadInformation()) return 1;
    // store current offset:
    uint cur_pos = reader.GetStreamCurrentPosition();
    uint remaining = file_size - cur_pos;
    Console.WriteLine("Remaining bytes to read (Pixel Data):  " + remaining.ToString() );
    // Get file infos
    gdcm.File f = reader.GetFile();
    // get some info about image
    UIntArrayType dims = ImageHelper.GetDimensionsValue(f);
    PixelFormat pf = ImageHelper.GetPixelFormatValue (f);
    int pixelsize = pf.GetPixelSize();
    PhotometricInterpretation pi = ImageHelper.GetPhotometricInterpretationValue(f);
    Console.WriteLine( pi.toString() );
    // buffer to get the pixels
    byte[] buffer = new byte[ dims[0] * dims[1] * pixelsize ];
    // define a simple box region.
    BoxRegion box = new BoxRegion();
    for (uint z = 0; z < dims[2]; z++)
      {
      // Define that I want the image 0, full size (dimx x dimy pixels)
      // and do that for each z:
      box.SetDomain(0, dims[0] - 1, 0, dims[1] - 1, z, z);
      //System.Console.WriteLine( box.toString() );
      reader.SetRegion( box );
      // reader will try to load the uncompressed image region into buffer.
      // the call returns an error when buffer.Length is too small.  For instance
      // one can call:
      // uint buf_len = reader.ComputeBufferLength(); // take into account pixel size
      // to get the exact size of minimum buffer
```

```
      if (reader.ReadIntoBuffer(buffer, (uint)buffer.Length))
        {
        using (System.IO.Stream stream =
          System.IO.File.Open(@"/tmp/frame.raw",
            System.IO.FileMode.Create))
          {
          System.IO.BinaryWriter writer = new System.IO.BinaryWriter(stream);
          writer.Write(buffer);
          }
        }
      else
        {
        throw new Exception("can't read pixels error");
        }
      }
    return 0;
    }
}
```

12.22 ExtractImageRegionWithLUT.cs

```
/*=========================================================================

Program:  GDCM (Grassroots DICOM). A DICOM library

Copyright (c) 2006-2011 Mathieu Malaterre
All rights reserved.
See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.

This software is distributed WITHOUT ANY WARRANTY; without even
the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
PURPOSE.  See the above copyright notice for more information.

=========================================================================*/
/*
* This small code shows how to use the gdcm.ImageRegionReader API
* In this example we are taking each frame by frame and dump them to
* /tmp/frame.raw.
* Furthermore we are applying the LUT on this image.
* Special care should be taken in case the image is not PALETTE COLOR
*
* Usage:
* $ bin/ExtractImageRegionWithLUT.exe input.dcm
*
* Example:
* $ bin/ExtractImageRegionWithLUT.exe gdcmData/rle16loo.dcm
* $ md5sum /tmp/frame_rgb.raw
* 73bf61325fdb6e2830244a2b7b0c4ae2  /tmp/frame_rgb.raw
* $ gdcmimg --depth 16 --spp 3 --size 600,430 /tmp/frame_rgb.raw rgb.dcm
* $ gdcmviewer rgb.dcm
*/
using System;
using gdcm;
public class ExtractImageRegion
{
  public static int Main(string[] args)
    {
    string filename = args[0];
    // instantiate the reader:
    gdcm.ImageRegionReader reader = new gdcm.ImageRegionReader();
    reader.SetFileName( filename );
    // pull DICOM info:
    if (!reader.ReadInformation()) return 1;
    // Get file infos
    gdcm.File f = reader.GetFile();
    gdcm.LookupTable lut = reader.GetImage().GetLUT();
    // get some info about image
    UIntArrayType dims = ImageHelper.GetDimensionsValue(f);
    PixelFormat pf = ImageHelper.GetPixelFormatValue (f);
    int pixelsize = pf.GetPixelSize();
    // buffer to get the pixels
    byte[] buffer = new byte[ dims[0] * dims[1] * pixelsize ];
    // output buffer for the RGB decoded image:
    byte[] buffer2 = new byte[ dims[0] * dims[1] * pixelsize * 3 ];
    // define a simple box region.
    BoxRegion box = new BoxRegion();
    for (uint z = 0; z < dims[2]; z++)
```

```
      {
      // Define that I want the image 0, full size (dimx x dimy pixels)
      // and do that for each z:
      box.SetDomain(0, dims[0] - 1, 0, dims[1] - 1, z, z);
      //System.Console.WriteLine( box.toString() );
      reader.SetRegion( box );
      // reader will try to load the uncompressed image region into buffer.
      // the call returns an error when buffer.Length is too small.  For instance
      // one can call:
      // uint buf_len = reader.ComputeBufferLength(); // take into account pixel size
      // to get the exact size of minimum buffer
      if (reader.ReadIntoBuffer(buffer, (uint)buffer.Length))
        {
        if( !lut.Decode( buffer2, (uint)buffer2.Length, buffer, (uint)buffer.Length ) )
          {
          throw new Exception("can't decode");
          }
        using (System.IO.Stream stream =
          System.IO.File.Open(@"/tmp/frame_rgb.raw",
            System.IO.FileMode.Create))
          {
          System.IO.BinaryWriter writer = new System.IO.BinaryWriter(stream);
          writer.Write(buffer2);
          }
        }
      else
        {
        throw new Exception("can't read pixels error");
        }
      }
    return 0;
    }
}
```

12.23 ExtractOneFrame.cs

```
/*=========================================================================

Program:  GDCM (Grassroots DICOM). A DICOM library

Copyright (c) 2006-2011 Mathieu Malaterre
All rights reserved.
See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.

This software is distributed WITHOUT ANY WARRANTY; without even
the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
PURPOSE.  See the above copyright notice for more information.

=========================================================================*/
/*
* This small code shows how to use the gdcm.StreamImageReader API
* to read a single (whole) frame at a time
* The API allow extracting a smaller extent of the frame of course.
* It will write out the extracted frame in /tmp/frame.raw
*
* Usage:
* $ bin/ExtractOneFrame.exe input.dcm
*/
using System;
using gdcm;
public class ExtractOneFrame
{
  public static int Main(string[] args)
    {
    string filename = args[0];
    gdcm.StreamImageReader reader = new gdcm.StreamImageReader();
    reader.SetFileName( filename );
    if (!reader.ReadImageInformation()) return 1;
    // Get file infos
    gdcm.File f = reader.GetFile();
    // get some info about image
    UIntArrayType extent = ImageHelper.GetDimensionsValue(f);
    //System.Console.WriteLine( extent[0] );
    uint dimx = extent[0];
    //System.Console.WriteLine( extent[1] );
    uint dimy = extent[1];
    //System.Console.WriteLine( extent[2] );
```

```
    uint dimz = extent[2];
    PixelFormat pf = ImageHelper.GetPixelFormatValue (f);
    int pixelsize = pf.GetPixelSize();
    //System.Console.WriteLine( pixelsize );
    // buffer to get the pixels
    byte[] buffer = new byte[ dimx * dimy * pixelsize ];
    for (int i = 0; i < dimz; i++)
      {
      // Define that I want the image 0, full size (dimx x dimy pixels)
      reader.DefinePixelExtent(0, (ushort)dimx, 0, (ushort)dimy, (ushort)i, (ushort)(i+1));
      uint buf_len = reader.DefineProperBufferLength(); // take into account pixel size
      //System.Console.WriteLine( buf_len );
      if( buf_len > buffer.Length )
        {
        throw new Exception("buffer is too small for target");
        }
      if (reader.Read(buffer, (uint)buffer.Length))
        {
        using (System.IO.Stream stream =
          System.IO.File.Open(@"/tmp/frame.raw",
            System.IO.FileMode.Create))
          {
          System.IO.BinaryWriter writer = new System.IO.BinaryWriter(stream);
          writer.Write(buffer);
          }
        }
      else
        {
        throw new Exception("can't read pixels error");
        }
      }
    return 0;
    }
}
```

12.24 FileAnonymize.cs

```
/*=========================================================================

Program:  GDCM (Grassroots DICOM). A DICOM library

Copyright (c) 2006-2011 Mathieu Malaterre
All rights reserved.
See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.

This software is distributed WITHOUT ANY WARRANTY; without even
the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
PURPOSE.  See the above copyright notice for more information.

=========================================================================*/
/*
* Simple C# example
*
* Usage:
* $ mono bin/FileAnonymize.exe input.dcm output.dcm
*/
using System;
using gdcm;
public class FileAnonymize
{
  public static int Main(string[] args)
    {
    string filename = args[0];
    string outfilename = args[1];
    gdcm.FileAnonymizer fa = new gdcm.FileAnonymizer();
    fa.SetInputFileName( filename );
    fa.SetOutputFileName( outfilename );
    // Empty Operations
    // It will create elements, since those tags are non-registered public elements (2011):
    fa.Empty( new Tag(0x0008,0x1313) );
    fa.Empty( new Tag(0x0008,0x1317) );
    // Remove Operations
    // The following Tag are actually carefully chosen, since they refer to SQ:
    fa.Remove( new Tag(0x0008,0x2112) );
    fa.Remove( new Tag(0x0008,0x9215) );
    // Replace Operations
    // do not call replace operation on SQ attribute !
```

```
    fa.Replace( new Tag(0x0018,0x5100), "MYVALUE " );
    fa.Replace( new Tag(0x0008,0x1160), "MYOTHERVAL" );
    if( !fa.Write() )
      {
      System.Console.WriteLine( "Could not write" );
      return 1;
      }
    return 0;
    }
}
```

12.25 FileChangeTS.cs

```
/*=========================================================================

Program:  GDCM (Grassroots DICOM). A DICOM library

Copyright (c) 2006-2011 Mathieu Malaterre
All rights reserved.
See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.

This software is distributed WITHOUT ANY WARRANTY; without even
the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
PURPOSE.  See the above copyright notice for more information.

=========================================================================*/
/*
* Simple C# example
*
* Shows multiple steps:
* Steps 1.
* Create a fake (dummy) DICOM file, with size 512 x 512 x 2 We use a small
* image to be able to create the volume in memory Of course you can use any
* existing DICOM instead
*
* Step 2.
* Hack the DICOM file to pretend the number of frames is 1000 (instead of 2)
* At this point in time this makes the DICOM file invalid (truncated).  But the
* next step will fix this.
*
* Step 3.
* Use C# to create a binary data which will represent our source object for
* image.
*
* Step 4.
* We use gdcm.FileStreamer to merge the template DICOM file from Step 2, with
* the binary data from Step 3.  We decide to read a scanline at a time, but
* this can be read with any number of bytes.  AppendToDataElement() will always
* do the proper computation.
*
* Step 5.
* We compress this gigantic file, into [JPEG Lossless, Non-Hierarchical,
* First-Order Prediction (Process 14 [Selection Value 1])]
*
* Usage:
* $ mono bin/FileChangeTS.exe small.dcm big.dcm raw.data merge.dcm jpeg.dcm
*/
using System;
using System.IO;
using gdcm;
public class FileChangeTS
{
  public static byte[] StrToByteArray(string str)
    {
    System.Text.ASCIIEncoding  encoding=new System.Text.ASCIIEncoding();
    return encoding.GetBytes(str);
    }
  // Create a 256 x 256 Secondary Capture Image Storage
  static private void CreateSmallDICOM(string fileName)
    {
    using( var writer = new gdcm.PixmapWriter() )
      {
      gdcm.Pixmap img = writer.GetImage();
      img.SetNumberOfDimensions( 3 );
      img.SetDimension(0, 512 );
      img.SetDimension(1, 512 );
      img.SetDimension(2, 2 ); // fake a 3d volume
```

```
    PhotometricInterpretation pi = new PhotometricInterpretation( PhotometricInterpretation.PIType.MONOCHROME2
    );
    img.SetPhotometricInterpretation( pi );
    gdcm.DataElement pixeldata = new gdcm.DataElement( new gdcm.Tag(0x7fe0,0x0010) );
    byte[] buffer = new byte[ 512 * 512 * 2 ];
    pixeldata.SetByteValue( buffer, new gdcm.VL((uint)buffer.Length) );
    img.SetDataElement( pixeldata );
    gdcm.File file = writer.GetFile();
    gdcm.DataSet ds = file.GetDataSet();
    gdcm.DataElement ms = new gdcm.DataElement(new gdcm.Tag(0x0008,0x0016));
    string mediastorage = "1.2.840.10008.5.1.4.1.1.7.2"; // Multi-frame Grayscale Byte Secondary Capture Image
    Storage
    byte[] val = StrToByteArray(mediastorage);
    ms.SetByteValue( val, new gdcm.VL( (uint)val.Length) );
    ds.Insert( ms );
    writer.SetFileName( fileName );
    writer.Write();
    }
  }
static private void CreateBigDICOM(string fileName, string outfilename)
  {
  using( var ano = new gdcm.FileAnonymizer() )
    {
    // The following is somewhat dangerous, do not try at home:
    string nframes = "1000";
    ano.Replace( new gdcm.Tag(0x0028,0x0008), nframes );
    ano.SetInputFileName(fileName);
    ano.SetOutputFileName(outfilename);
    ano.Write(); // at this point the DICOM is invalid !
    }
  }
static private void CreateDummyFile(string fileName, long length)
  {
  using (var fileStream = new FileStream(fileName, FileMode.Create, FileAccess.Write, FileShare.None))
    {
    // Looks like C# always init to 0 (fallocate ?)
    // For the purpose of the test we could add some random noise
    fileStream.SetLength(length);
    }
  }
static private void ReadBytesIntoArray( byte[] array, FileStream source )
  {
  int numBytesToRead = array.Length;
  int numBytesRead = 0;
  while (numBytesToRead > 0)
    {
    // According to spec:  Read() may return anything from 0 to numBytesToRead.
    int n = source.Read(array, numBytesRead, numBytesToRead);
    // Break when the end of the file is reached.
    if (n == 0)
      break;
    numBytesRead += n;
    numBytesToRead -= n;
    }
  }
static private void AssembleDICOMAndRaw(string dicomfn, string rawdata, string outfn)
  {
  using ( var fs = new gdcm.FileStreamer() )
    {
    fs.SetTemplateFileName(dicomfn);
    fs.SetOutputFileName(outfn);
    gdcm.Tag pixeldata = new gdcm.Tag(0x7fe0, 0x0010);
    // FileStreamer support automatic checking of pixel data length
    // based on DICOM attributes, only if we say so:
    fs.CheckDataElement( pixeldata );
    // Declare we are working on Pixel Data attribute:
    fs.StartDataElement( pixeldata );
    using (FileStream rawSource = new FileStream(rawdata,
        FileMode.Open, FileAccess.Read))
      {
      byte[] bytes = new byte[512];
      // Only read one scanline at a time
      // We could have been reading more at once, if this is more efficient,
      // AppendToDataElement will do the logic in all cases.
      for( int i = 0; i < 512 * 1000; ++i )
        {
        // Read the source file into a byte array.
        ReadBytesIntoArray( bytes, rawSource );
        fs.AppendToDataElement( pixeldata, bytes, (uint)bytes.Length );
        }
      }
```

```
      if( !fs.StopDataElement( pixeldata ) )
        {
        // Most likely an issue with Pixel Data Length computation:
        throw new Exception("StopDataElement failed");
        }
      }
    }
  static private void CompressIntoJPEG(string rawdicom, string jpegdicom)
    {
    using( var sfcts = FileChangeTransferSyntax.New() )
      {
      // Need to retrieve the actual C++ reference, to pass to
      // SimpleSubjectWatcher:
      FileChangeTransferSyntax fcts = sfcts.__ref__();
      SimpleSubjectWatcher watcher = new SimpleSubjectWatcher(fcts, "FileChangeTransferSyntax");
      gdcm.TransferSyntax ts = new TransferSyntax( TransferSyntax.TSType.JPEGLosslessProcess14_1 );
      fcts.SetTransferSyntax( ts );
      fcts.SetInputFileName( rawdicom );
      fcts.SetOutputFileName( jpegdicom );
      fcts.Change();
      }
    }
  public static int Main(string[] args)
    {
    string filename = args[0];
    string outfilename = args[1];
    string rawfilename = args[2];
    string mergefn = args[3];
    string jpegfn = args[4];
    CreateSmallDICOM(filename);
    CreateBigDICOM(filename, outfilename);
    CreateDummyFile(rawfilename, 512 * 512 * 1000 );
    AssembleDICOMAndRaw(outfilename, rawfilename, mergefn);
    CompressIntoJPEG(mergefn, jpegfn);
    return 0;
    }
}
```

12.26 FileChangeTSLossy.cs

```
/*=========================================================================

Program:  GDCM (Grassroots DICOM). A DICOM library

Copyright (c) 2006-2011 Mathieu Malaterre
All rights reserved.
See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.

This software is distributed WITHOUT ANY WARRANTY; without even
the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
PURPOSE.  See the above copyright notice for more information.

=========================================================================*/
/*
* Simple C# example
*
* Shows multiple steps:
* Steps 1.
* Create a fake (dummy) DICOM file, with size 512 x 512 x 2 We use a small
* image to be able to create the volume in memory Of course you can use any
* existing DICOM instead
*
* Step 2.
* Hack the DICOM file to pretend the number of frames is 1000 (instead of 2)
* At this point in time this makes the DICOM file invalid (truncated).  But the
* next step will fix this.
*
* Step 3.
* Use C# to create a binary data which will represent our source object for
* image.
*
* Step 4.
* We use gdcm.FileStreamer to merge the template DICOM file from Step 2, with
* the binary data from Step 3.  We decide to read a scanline at a time, but
* this can be read with any number of bytes.  AppendToDataElement() will always
* do the proper computation.
*
```

```
* Step 5.
* We compress this gigantic file, into [JPEG Baseline (Process 1):  Default Transfer Syntax for Lossy JPEG 8 Bit
      Image Compression]
*
* Usage:
* $ bin/FileChangeTSLossy.exe small.dcm big.dcm raw.data merge.dcm jpeg.dcm
*/
using System;
using System.IO;
using gdcm;
public class FileChangeTS
{
  public static byte[] StrToByteArray(string str)
    {
    System.Text.ASCIIEncoding  encoding=new System.Text.ASCIIEncoding();
    return encoding.GetBytes(str);
    }
  // Create a 256 x 256 Secondary Capture Image Storage
  static private void CreateSmallDICOM(string fileName)
    {
    using( var writer = new gdcm.PixmapWriter() )
      {
      gdcm.Pixmap img = writer.GetImage();
      img.SetNumberOfDimensions( 3 );
      img.SetDimension(0, 512 );
      img.SetDimension(1, 512 );
      img.SetDimension(2, 2 ); // fake a 3d volume
      PhotometricInterpretation pi = new PhotometricInterpretation( PhotometricInterpretation.PIType.MONOCHROME2
      );
      img.SetPhotometricInterpretation( pi );
      gdcm.DataElement pixeldata = new gdcm.DataElement( new gdcm.Tag(0x7fe0,0x0010) );
      byte[] buffer = new byte[ 512 * 512 * 2 ];
      pixeldata.SetByteValue( buffer, new gdcm.VL((uint)buffer.Length) );
      img.SetDataElement( pixeldata );
      gdcm.File file = writer.GetFile();
      gdcm.DataSet ds = file.GetDataSet();
      gdcm.DataElement ms = new gdcm.DataElement(new gdcm.Tag(0x0008,0x0016));
      string mediastorage = "1.2.840.10008.5.1.4.1.1.7.2"; // Multi-frame Grayscale Byte Secondary Capture Image
      Storage
      byte[] val = StrToByteArray(mediastorage);
      ms.SetByteValue( val, new gdcm.VL( (uint)val.Length) );
      ds.Insert( ms );
      writer.SetFileName( fileName );
      writer.Write();
      }
    }
  static private void CreateBigDICOM(string fileName, string outfilename)
    {
    using( var ano = new gdcm.FileAnonymizer() )
      {
      // The following is somewhat dangerous, do not try at home:
      string nframes = "1000";
      ano.Replace( new gdcm.Tag(0x0028,0x0008), nframes );
      ano.SetInputFileName(fileName);
      ano.SetOutputFileName(outfilename);
      ano.Write(); // at this point the DICOM is invalid !
      }
    }
  static private void CreateDummyFile(string fileName, long length)
    {
    using (var fileStream = new FileStream(fileName, FileMode.Create, FileAccess.Write, FileShare.None))
      {
      // Looks like C# always init to 0 (fallocate ?)
      // For the purpose of the test we could add some random noise
      fileStream.SetLength(length);
      }
    }
  static private void ReadBytesIntoArray( byte[] array, FileStream source )
    {
    int numBytesToRead = array.Length;
    int numBytesRead = 0;
    while (numBytesToRead > 0)
      {
      // According to spec:  Read() may return anything from 0 to numBytesToRead.
      int n = source.Read(array, numBytesRead, numBytesToRead);
      // Break when the end of the file is reached.
      if (n == 0)
        break;
      numBytesRead += n;
      numBytesToRead -= n;
      }
```

```
    }
  static private void AssembleDICOMAndRaw(string dicomfn, string rawdata, string outfn)
    {
    using ( var fs = new gdcm.FileStreamer() )
      {
      fs.SetTemplateFileName(dicomfn);
      fs.SetOutputFileName(outfn);
      gdcm.Tag pixeldata = new gdcm.Tag(0x7fe0, 0x0010);
      // FileStreamer support automatic checking of pixel data length
      // based on DICOM attributes, only if we say so:
      fs.CheckDataElement( pixeldata );
      // Declare we are working on Pixel Data attribute:
      fs.StartDataElement( pixeldata );
      using (FileStream rawSource = new FileStream(rawdata,
          FileMode.Open, FileAccess.Read))
        {
        byte[] bytes = new byte[512];
        // Only read one scanline at a time
        // We could have been reading more at once, if this is more efficient,
        // AppendToDataElement will do the logic in all cases.
        for( int i = 0; i < 512 * 1000; ++i )
          {
          // Read the source file into a byte array.
          ReadBytesIntoArray( bytes, rawSource );
          fs.AppendToDataElement( pixeldata, bytes, (uint)bytes.Length );
          }
        }
      if( !fs.StopDataElement( pixeldata ) )
        {
        // Most likely an issue with Pixel Data Length computation:
        throw new Exception("StopDataElement failed");
        }
      }
    }
  static private void CompressIntoJPEG(string rawdicom, string jpegdicom)
    {
    using( var sfcts = FileChangeTransferSyntax.New() )
      {
      // Need to retrieve the actual C++ reference, to pass to
      // SimpleSubjectWatcher:
      FileChangeTransferSyntax fcts = sfcts.__ref__();
      SimpleSubjectWatcher watcher = new SimpleSubjectWatcher(fcts, "FileChangeTransferSyntax");
      gdcm.TransferSyntax ts = new TransferSyntax( TransferSyntax.TSType.JPEGBaselineProcess1 );
      fcts.SetTransferSyntax( ts );
      ImageCodec ic = fcts.GetCodec();
      JPEGCodec jpeg = JPEGCodec.Cast( ic );
      jpeg.SetLossless( false );
      jpeg.SetQuality( 50 ); // poor quality !
      fcts.SetInputFileName( rawdicom );
      fcts.SetOutputFileName( jpegdicom );
      fcts.Change();
      }
    }
  public static int Main(string[] args)
    {
    string filename = args[0];
    string outfilename = args[1];
    string rawfilename = args[2];
    string mergefn = args[3];
    string jpegfn = args[4];
    CreateSmallDICOM(filename);
    CreateBigDICOM(filename, outfilename);
    CreateDummyFile(rawfilename, 512 * 512 * 1000 );
    AssembleDICOMAndRaw(outfilename, rawfilename, mergefn);
    CompressIntoJPEG(mergefn, jpegfn);
    return 0;
    }
}
```

12.27 FileStreaming.cs

```
/*=========================================================================

Program:  GDCM (Grassroots DICOM). A DICOM library

Copyright (c) 2006-2011 Mathieu Malaterre
All rights reserved.
```

```
See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.

This software is distributed WITHOUT ANY WARRANTY; without even
the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
PURPOSE.  See the above copyright notice for more information.

=========================================================================*/
/*
* Simple C# example
*
* Usage:
* $ mono bin/FileStreaming.exe gdcmData/CT_16b_signed-UsedBits13.dcm output.dcm
*
* The class will take care of group handling and will use the first available group:
* (0009,0012) ??  (LO) [MYTEST]                            # 6,1 Private Creator
*/
using System;
using gdcm;
public class FileStreaming
{
  public static int Main(string[] args)
    {
    string filename = args[0];
    string outfilename = args[1];
    gdcm.PrivateTag pt = new gdcm.PrivateTag( new gdcm.Tag(0x9,0x10), "MYTEST" );
    gdcm.FileStreamer fs = new gdcm.FileStreamer();
    fs.SetTemplateFileName( filename );
    fs.SetOutputFileName( outfilename );
    byte[] buffer = new byte[ 8192 ];
    uint len = (uint)buffer.Length;
    // In this example, we want that each newly created Private Attribute
    // contains at most 1000 bytes of incoming dataset.
    // We are also calling the function twice to check that appending mode is
    // working from one call to the other.  The last element will have a length
    // of (2 * 8192) % 1000 = 384
    if( !fs.StartGroupDataElement( pt, 1000, 1 )
      || !fs.AppendToGroupDataElement( pt, buffer, len )
      || !fs.AppendToGroupDataElement( pt, buffer, len )
      || !fs.StopGroupDataElement( pt ) )
      {
      System.Console.WriteLine( "Could not change private group" );
      return 1;
      }
    return 0;
    }
}
```

12.28 GetArray.cs

```
/*=========================================================================

Program:  GDCM (Grassroots DICOM). A DICOM library

Copyright (c) 2006-2011 Mathieu Malaterre
All rights reserved.
See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.

This software is distributed WITHOUT ANY WARRANTY; without even
the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
PURPOSE.  See the above copyright notice for more information.

=========================================================================*/
/*
* Usage:
* $ export LD_LIBRARY_PATH=$HOME/Projects/gdcm/debug-gcc/bin
* $ mono bin/GetArray.exe gdcmData/012345.002.050.dcm
*/
using System;
using gdcm;
public class GetArray
{
  public static int Main(string[] args)
    {
    string file1 = args[0];
    ImageReader reader = new ImageReader();
    reader.SetFileName( file1 );
    bool ret = reader.Read();
```

```
    if( !ret )
      {
      return 1;
      }
    Image image = reader.GetImage();
    PixelFormat pixeltype = image.GetPixelFormat();
    if( image.GetNumberOfDimensions() != 2 )
      {
      // For the purpose of the test, exit early on
      return 1;
      }
    uint dimx = image.GetDimension(0);
    uint dimy = image.GetDimension(1);
    uint npixels = dimx * dimy;
    //LookupTable lut = image.GetLUT();
    //uint rl = lut.GetLUTLength( LookupTable.LookupTableType.RED );
    //byte[] rbuf = new byte[ rl ];
    //uint rl2 = lut.GetLUT( LookupTable.LookupTableType.RED, rbuf );
    //assert rl == rl2;
    //byte[] str1 = new byte[ image.GetBufferLength()];
    //image.GetBuffer( str1 );
    if( pixeltype.GetScalarType() == PixelFormat.ScalarType.UINT8 )
      {
      System.Console.WriteLine( "Processing UINT8 image type" );
      byte[] str1 = new byte[ npixels ];
      image.GetArray( str1 );
      }
    else if( pixeltype.GetScalarType() == PixelFormat.ScalarType.INT16 )
      {
      System.Console.WriteLine( "Processing INT16 image type" );
      short[] str1 = new short[ npixels ];
      image.GetArray( str1 );
      }
    else if( pixeltype.GetScalarType() == PixelFormat.ScalarType.UINT16 )
      {
      System.Console.WriteLine( "Processing UINT16 image type" );
      ushort[] str1 = new ushort[ npixels ];
      image.GetArray( str1 );
      }
    else
      {
      //System.Console.WriteLine( "Default (unhandled pixel format):  " + pixeltype.toString() );
      System.Console.WriteLine( "Default (unhandled pixel format):  " + pixeltype.GetScalarTypeAsString() );
      // Get bytes
      byte[] str1 = new byte[ image.GetBufferLength()];
      image.GetBuffer( str1 );
      }
    return 0;
    }
}
```

12.29 MpegVideoInfo.cs

```
/*=========================================================================

Program:  GDCM (Grassroots DICOM). A DICOM library

Copyright (c) 2006-2011 Mathieu Malaterre
All rights reserved.
See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.

This software is distributed WITHOUT ANY WARRANTY; without even
the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
PURPOSE.  See the above copyright notice for more information.

=========================================================================*/
/*
* This examples takes in a MPEG2 and write out a Video Endoscopic Imagae Storage
* encoded using MPEG2 @ Main Profile
* ref:  http://chrisa.wordpress.com/2007/11/21/decoding-mpeg2-information/
* See also:
* http://dvd.sourceforge.net/dvdinfo/mpeghdrs.html#gop
* http://cvs.linux.hr/cgi-bin/viewcvs.cgi/mpeg_mod/README.infompeg?view=markup
* http://www.guru-group.fi/~too/sw/m2vmp2cut/mpeg2info.c
*/
/*
* Provides information about an MPEG2 file, including the duration, frame rate, aspect
```

```
* ratio, and resolution.    Good information about the MPEG2 file structure that helps
* explain parts of the code can be found here:
* http://dvd.sourceforge.net/dvdinfo/mpeghdrs.html#gop
*
* Copyright (c) 2007 Chris Anderson (chrisa@wordpress.com)
*
* This library is free software; you can redistribute it and/or
* modify it under the terms of the GNU Lesser General Public
* License as published by the Free Software Foundation; either
* version 2 of the License, or (at your option) any later version.
*
* This library is distributed in the hope that it will be useful,
* but WITHOUT ANY WARRANTY; without even the implied warranty of
* MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.  See the GNU
* Lesser General Public License for more details.
*/
using System;
using System.IO;
using gdcm;
public class Mpeg2VideoInfo
{
#region Member Variables
    private TimeSpan        m_startTime     = TimeSpan.Zero;
    private TimeSpan        m_endTime       = TimeSpan.Zero;
    private TimeSpan        m_duration      = TimeSpan.Zero;
    private eAspectRatios   m_aspectRatio   = eAspectRatios.Invalid;
    private eFrameRates     m_frameRate     = 0;
    private int             m_pictureWidth  = 0;
    private int             m_pictureHeight = 0;
#endregion
#region Constants
    private const byte PADDING_PACKET = 0xBE;
    private const byte VIDEO_PACKET = 0xE0;
    private const byte AUDIO_PACKET = 0xC0;
    private const byte SYSTEM_PACKET = 0xBB;
    private const byte TIMESTAMP_PACKET = 0xB8;
    private const byte HEADER_PACKET = 0xB3;
    private const int BUFFER_SIZE = 8162; // 8K buffer
    private readonly static TimeSpan EMPTY_TIMESPAN = new TimeSpan(0, 0, -1);
#endregion
#region Enumerations
    public enum eFrameRates
    {
        Invalid,
        PulldownNTSC,           // 24000d/1001d = 23.976 Hz
        Film,                   // 24 Hz
        PAL,                    // 25 Hz
        NTSC,                   // 30000d/1001d = 29.97 Hz
        DropFrameNTSC,          // 30 Hz
        DoubleRatePAL,          // 50 Hz
        DoubleRateNTSC,         // 59.97 Hz
        DoubleRateDropFrameNTSC // 60 Hz
    }
    public enum eAspectRatios
    {
        Invalid,
        VGA,        // 1/1
        StandardTV, // 4/3
        LargeTV,    // 16/9
        Cinema      // 2.21/1
    }
#endregion
#region Constructor
    public Mpeg2VideoInfo(string file)
    {
        ParseMpeg(file);
    }
#endregion
#region Public Properties
    public TimeSpan StartTime
    {
        get { return m_startTime; }
    }
    public TimeSpan EndTime
    {
        get { return m_endTime; }
    }
    public TimeSpan Duration
    {
        get { return m_duration; }
    }
```

```
    public eAspectRatios AspectRatio
    {
        get { return m_aspectRatio; }
    }
    public eFrameRates FrameRate
    {
        get { return m_frameRate; }
    }
    public int PictureWidth
    {
        get { return m_pictureWidth; }
    }
    public int PictureHeight
    {
        get { return m_pictureHeight; }
    }
#endregion
#region Private Functions
    private void ParseMpeg(string file)
    {
        FileStream fs = new FileStream(file, FileMode.Open, FileAccess.Read, FileShare.ReadWrite);
        BinaryReader br = new BinaryReader(fs);
        m_startTime = GetStartTimeStampInfo(br);
        m_endTime = GetEndTimeStampInfo(br);
        m_duration = m_endTime.Subtract(m_startTime);
        GetHeaderInfo(br);
        br.Close();
        fs.Close();
    }
    private TimeSpan GetStartTimeStampInfo(BinaryReader br)
    {
        TimeSpan startTime = EMPTY_TIMESPAN;
        byte[] buffer = new byte[BUFFER_SIZE];
        br.BaseStream.Seek(0, SeekOrigin.Begin);
        while (startTime == EMPTY_TIMESPAN && br.BaseStream.Position < br.BaseStream.Length)
        {
            int readBytes = br.Read(buffer, 0, BUFFER_SIZE);
            for (int offset = 0; offset < readBytes - 8; offset++)
            {
                if (IsStreamMarker(ref buffer, offset, TIMESTAMP_PACKET))
                {
                    offset += 4; // Move to the data position which follows the stream header
                    uint timeStampEncoded = GetData(ref buffer, offset);
                    startTime = DecodeTimeStamp(timeStampEncoded);
                    if (startTime != EMPTY_TIMESPAN)
                        break;
                }
            }
        }
        return startTime;
    }
    private TimeSpan GetEndTimeStampInfo(BinaryReader br)
    {
        TimeSpan endTime = EMPTY_TIMESPAN;
        byte[] buffer = new byte[BUFFER_SIZE];
        br.BaseStream.Seek(-BUFFER_SIZE, SeekOrigin.End);
        while (endTime == EMPTY_TIMESPAN && br.BaseStream.Position > BUFFER_SIZE)
        {
            int readBytes = br.Read(buffer, 0, BUFFER_SIZE);
            for (int offset = readBytes - 8; offset >= 0; offset--)
            {
                if (IsStreamMarker(ref buffer, offset, TIMESTAMP_PACKET))
                {
                    offset += 4; // Move to the data position which follows the stream header
                    uint timeStampEncoded = GetData(ref buffer, offset);
                    endTime = DecodeTimeStamp(timeStampEncoded);
                    if (endTime != EMPTY_TIMESPAN)
                        break;
                }
            }
            br.BaseStream.Seek(-BUFFER_SIZE * 2, SeekOrigin.Current);
        }
        return endTime;
    }
    private TimeSpan DecodeTimeStamp(uint timeStampEncoded)
    {
        TimeSpan timeStamp = EMPTY_TIMESPAN;
        // Mask out the bits containing the property we are after, then
        // shift the data to the right to get its value
        int hour = (int)(timeStampEncoded & 0x7C000000) » 26;   // Bits 31 -> 27
        int minute = (int)(timeStampEncoded & 0x03F00000) » 20; // Bits 26 -> 21
```

```
        int second = (int)(timeStampEncoded & 0x0007E000) » 13; // Bits 19 -> 14
        int frame = (int)(timeStampEncoded & 0x00001F80) » 7;   // Bits 13 -> 8 - not used, but included for
      completeness
        timeStamp = new TimeSpan(hour, minute, second);
        return timeStamp;
    }
    private void GetHeaderInfo(BinaryReader br)
    {
        byte[] buffer = new byte[BUFFER_SIZE];
        br.BaseStream.Seek(0, SeekOrigin.Begin);
        br.Read(buffer, 0, BUFFER_SIZE);
        for (int offset = 0; offset < buffer.Length - 4; offset++)
        {
            if (IsStreamMarker(ref buffer, offset, HEADER_PACKET))
            {
                offset += 4; // Move to the data position which follows the stream header
                uint headerData = GetData(ref buffer, offset);
                // Mask out the bits containing the property we are after, then
                // shift the data to the right to get its value
                m_pictureWidth = (int)(headerData & 0xFFF00000) » 20;
                m_pictureHeight = (int)(headerData & 0x000FFF00) » 8;
                uint aspectRatioIndex = (headerData & 0x000000F0) » 4;
                uint fpsIndex = headerData & 0x0000000F;
                m_aspectRatio = (eAspectRatios)fpsIndex;
                m_frameRate = (eFrameRates)fpsIndex;
                break;
            }
        }
    }
    private uint GetData(ref byte[] buffer, int offset)
    {
        return (uint)  ((buffer[offset] « 24) |
                        (buffer[offset + 1] « 16) |
                        (buffer[offset + 2] « 8) |
                        (buffer[offset + 3]));
    }
    private bool IsStreamMarker(ref byte[] buffer, int offset, byte markerType)
    {
        return (buffer[offset] == 0x00 &&
                buffer[offset + 1] == 0x00 &&
                buffer[offset + 2] == 0x01 &&
                buffer[offset + 3] == markerType);
    }
#endregion
  public static int Main(string[] args)
    {
    string file1 = args[0];
    Mpeg2VideoInfo info = new Mpeg2VideoInfo(file1);
    System.Console.WriteLine( info.StartTime );
    System.Console.WriteLine( info.EndTime );
    System.Console.WriteLine( info.Duration );
    System.Console.WriteLine( info.AspectRatio );
    System.Console.WriteLine( info.FrameRate );
    System.Console.WriteLine( info.PictureWidth );
    System.Console.WriteLine( info.PictureHeight );
    ImageReader r = new ImageReader();
    //Image image = new Image();
    Image image = r.GetImage();
    image.SetNumberOfDimensions( 3 );
    DataElement pixeldata = new DataElement( new gdcm.Tag(0x7fe0,0x0010) );
    System.IO.FileStream infile =
      new System.IO.FileStream(file1, System.IO.FileMode.Open, System.IO.FileAccess.Read);
    uint fsize = gdcm.PosixEmulation.FileSize(file1);
    byte[] jstream  = new byte[fsize];
    infile.Read(jstream, 0 , jstream.Length);
    SmartPtrFrag sq = SequenceOfFragments.New();
    Fragment frag = new Fragment();
    frag.SetByteValue( jstream, new gdcm.VL( (uint)jstream.Length) );
    sq.AddFragment( frag );
    pixeldata.SetValue( sq.__ref__() );
    // insert:
    image.SetDataElement( pixeldata );
    PhotometricInterpretation pi = new PhotometricInterpretation(
      PhotometricInterpretation.PIType.YBR_PARTIAL_420 );
    image.SetPhotometricInterpretation( pi );
    // FIXME hardcoded:
    PixelFormat pixeltype = new PixelFormat(3,8,8,7);
    image.SetPixelFormat( pixeltype );
    // FIXME hardcoded:
    TransferSyntax ts = new TransferSyntax( TransferSyntax.TSType.MPEG2MainProfile);
    image.SetTransferSyntax( ts );
```

```
    image.SetDimension(0, (uint)info.PictureWidth);
    image.SetDimension(1, (uint)info.PictureHeight);
    image.SetDimension(2, 721);
    ImageWriter writer = new ImageWriter();
    gdcm.File file = writer.GetFile();
    file.GetHeader().SetDataSetTransferSyntax( ts );
    Anonymizer anon = new Anonymizer();
    anon.SetFile( file );
    MediaStorage ms = new MediaStorage( MediaStorage.MSType.VideoEndoscopicImageStorage);
    UIDGenerator gen = new UIDGenerator();
    anon.Replace( new Tag(0x0008,0x16), ms.GetString() );
    anon.Replace( new Tag(0x0018,0x40), "25" );
    anon.Replace( new Tag(0x0018,0x1063), "40.000000" );
    anon.Replace( new Tag(0x0028,0x34), "4\\3" );
    anon.Replace( new Tag(0x0028,0x2110), "01" );
    writer.SetImage( image );
    writer.SetFileName( "dummy.dcm" );
    if( !writer.Write() )
      {
      System.Console.WriteLine( "Could not write" );
      return 1;
      }
    return 0;
    }
}
```

12.30 NewSequence.cs

```
/*=========================================================================

Program:  GDCM (Grassroots DICOM). A DICOM library

Copyright (c) 2006-2011 Mathieu Malaterre
All rights reserved.
See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.

This software is distributed WITHOUT ANY WARRANTY; without even
the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
PURPOSE.  See the above copyright notice for more information.

=========================================================================*/
/*
* Usage:
* $ export LD_LIBRARY_PATH=$HOME/Projects/gdcm/debug-gcc/bin
* $ mono bin/NewSequence.exe gdcmData/012345.002.050.dcm out.dcm
*/
using System;
//using gdcm;
public class NewSequence
{
  public static byte[] StrToByteArray(string str)
    {
    System.Text.ASCIIEncoding  encoding=new System.Text.ASCIIEncoding();
    return encoding.GetBytes(str);
    }
  public static int Main(string[] argv)
    {
    string file1 = argv[0];
    string file2 = argv[1];
    gdcm.Reader r = new gdcm.Reader();
    r.SetFileName( file1 );
    if ( !  r.Read() )
      {
      return 1;
      }
    gdcm.File f = r.GetFile();
    gdcm.DataSet ds = f.GetDataSet();
    // tsis = gdcm.Tag(0x0008,0x2112) # SourceImageSequence
    // Create a dataelement
    gdcm.DataElement de = new gdcm.DataElement(new gdcm.Tag(0x0010, 0x2180));
    string occ = "Occupation";
    de.SetByteValue( StrToByteArray(occ), new gdcm.VL((uint)occ.Length));
    de.SetVR(new gdcm.VR(gdcm.VR.VRType.SH));
    // Create an item
    gdcm.Item it = new gdcm.Item();
    it.SetVLToUndefined();      // Needed to not popup error message
    //it.InsertDataElement(de)
```

```
    gdcm.DataSet nds = it.GetNestedDataSet();
    nds.Insert(de);
    // Create a Sequence
    gdcm.SmartPtrSQ sq = gdcm.SequenceOfItems.New();
    sq.SetLengthToUndefined();
    sq.AddItem(it);
    // Insert sequence into data set
    gdcm.DataElement des = new gdcm.DataElement(new gdcm.Tag(0x0400,0x0550));
    des.SetVR(new gdcm.VR(gdcm.VR.VRType.SQ));
    des.SetValue(sq.__ref__());
    des.SetVLToUndefined();
    ds.Insert(des);
    gdcm.Writer w = new gdcm.Writer();
    w.SetFile( f );
    w.SetFileName( file2 );
    if ( !w.Write() )
      return 1;
    return 0;
    }
}
```

12.31 RescaleImage.cs

```
/*=========================================================================

Program:  GDCM (Grassroots DICOM). A DICOM library

Copyright (c) 2006-2011 Mathieu Malaterre
All rights reserved.
See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.

This software is distributed WITHOUT ANY WARRANTY; without even
the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
PURPOSE.  See the above copyright notice for more information.

=========================================================================*/
/*
* Usage:
* $ export LD_LIBRARY_PATH=$HOME/Projects/gdcm/debug-gcc/bin
* $ mono bin/DecompressImage.exe gdcmData/012345.002.050.dcm rescaled.dcm
*/
using System;
using gdcm;
public class DecompressImage
{
  public static int Main(string[] args)
    {
    string file1 = args[0];
    ImageReader reader = new ImageReader();
    reader.SetFileName( file1 );
    bool ret = reader.Read();
    if( !ret )
      {
      return 1;
      }
    Image image = reader.GetImage();
    PixelFormat pixeltype = image.GetPixelFormat();
    Rescaler r = new Rescaler();
    r.SetIntercept( 0 );
    r.SetSlope( 1.2 );
    r.SetPixelFormat( pixeltype );
    PixelFormat outputpt = new PixelFormat( r.ComputeInterceptSlopePixelType() );
    System.Console.WriteLine( "pixeltype" );
    System.Console.WriteLine( pixeltype.toString() );
    System.Console.WriteLine( "outputpt" );
    System.Console.WriteLine( outputpt.toString() );
    uint len = image.GetBufferLength();
    short[] input = new short[ len / 2 ]; // sizeof(short) == 2
    image.GetArray( input );
    double[] output = new double[ len / 2 ];
    r.Rescale( output, input, len );
    // First Pixel is:
    System.Console.WriteLine( "Input:" );
    System.Console.WriteLine( input[0] );
    System.Console.WriteLine( "Output:" );
    System.Console.WriteLine( output[0] );
    return 0;
```

```
    }
}
```

12.32 SendFileSCU.cs

```
/*=========================================================================

Program:  GDCM (Grassroots DICOM). A DICOM library

Copyright (c) 2006-2011 Mathieu Malaterre
All rights reserved.
See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.

This software is distributed WITHOUT ANY WARRANTY; without even
the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
PURPOSE.  See the above copyright notice for more information.

=========================================================================*/
/*
* Usage:
* $ export LD_LIBRARY_PATH=$HOME/Perso/gdcm-gcc/bin
* $ mono bin/SendFileSCU.exe server port input.dcm
*/
using System;
using gdcm;
public class SendFileSCU
{
  public static int Main(string[] args)
    {
    string server = args[0];
    ushort port = ushort.Parse(args[1]);
    string filename = args[2];
    bool b = CompositeNetworkFunctions.CEcho( server, port );
    if( !b ) return 1;
    FilenamesType files = new FilenamesType();
    files.Add( filename );
    b = CompositeNetworkFunctions.CStore( server, port, files );
    if( !b ) return 1;
    return 0;
    }
}
```

12.33 SimplePrintPatientName.cs

```
/*=========================================================================

Program:  GDCM (Grassroots DICOM). A DICOM library

Copyright (c) 2006-2011 Mathieu Malaterre
All rights reserved.
See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.

This software is distributed WITHOUT ANY WARRANTY; without even
the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
PURPOSE.  See the above copyright notice for more information.

=========================================================================*/
/*
* Usage:
* $ export LD_LIBRARY_PATH=$HOME/Perso/gdcm/debug-gcc/bin
* $ mono bin/SimplePrintPatientName.exe gdcmData/012345.002.050.dcm
*/
/*
This example was provided by Jonathan Morra /jonmorra gmail com/
on the gdcm mailing list (Fri, 28 May 2010)
*/
using System;
using gdcm;
namespace GDCMTest
{
  class SimplePrintPatientName
    {
    static int Main(string[] args)
```

```
      {
      if (args.Length != 1)
        {
        Console.WriteLine("This program prints the patient name of a dicom file with gdcm");
        Console.WriteLine("Usage:  [input.dcm]");
        return 1;
        }
      gdcm.Reader reader = new gdcm.Reader();
      reader.SetFileName(args[0]);
      bool ret = reader.Read();
      //TagSetType tst = new TagSetType();
      //tst.Add( new Tag(0x7fe0,0x10) );
      //bool ret = reader.ReadUpToTag( new Tag(0x88,0x200), tst );
      if( !ret )
        {
        return 1;
        }
      gdcm.File file = reader.GetFile();
      gdcm.StringFilter filter = new gdcm.StringFilter();
      filter.SetFile(file);
      string value = filter.ToString(new gdcm.Tag(0x0010, 0x0010));
      Console.WriteLine("Patient Name:  " + value);
      return 0;
      }
    }
}
```

12.34 SortImage2.cs

```
/*=========================================================================

Program:  GDCM (Grassroots DICOM). A DICOM library

Copyright (c) 2006-2011 Mathieu Malaterre
All rights reserved.
See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.

This software is distributed WITHOUT ANY WARRANTY; without even
the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
PURPOSE.  See the above copyright notice for more information.

=========================================================================*/
/*
* Usage:
* $ export LD_LIBRARY_PATH=$HOME/Projects/gdcm/debug-gcc/bin
* $ mono bin/SortImage.exe gdcmData/012345.002.050.dcm out.dcm
*/
using System;
using gdcm;
public class SortImage2
{
  bool mysort(DataSet ds1, DataSet ds2)
    {
    return false;
    }
  public static int Main(string[] args)
    {
    Sorter sorter = new Sorter();
    sorter.SetSortFunction( mysort );
    return 0;
    }
}
```

12.35 CStoreQtProgress.cxx

```
/*=========================================================================

Program:  GDCM (Grassroots DICOM). A DICOM library

Copyright (c) 2006-2011 Mathieu Malaterre
All rights reserved.
See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.
```

```
This software is distributed WITHOUT ANY WARRANTY; without even
the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
PURPOSE.  See the above copyright notice for more information.

=========================================================================*/
/*
* This small example show how one can use the virtual function
* mechanism of the SimpleSubjectWatcher class to redirect progress
* report to a custom Qt classes
*
* http://doc.qt.nokia.com/latest/qprogressdialog.html
*
* Usage:
* CStoreQtProgress dicom.example.com 11112 gdcmData/MR_Spectroscopy_SIEMENS_OF.dcm
*
*/
#include "gdcmServiceClassUser.h"
#include "gdcmSimpleSubjectWatcher.h"
#include "gdcmProgressEvent.h"
#include "gdcmDirectory.h"
#include "gdcmPresentationContextGenerator.h"
#include <QApplication>
#include <QProgressDialog>
#include <QVBoxLayout>
namespace gdcm {
/*
* This class is a little more complicated than what this example demonstrate
* This watcher is capable of handling nested progress.  Since the Progress
* grows from [0 to 1] on a per file basis and we only have one instance of a
* watcher per association, we need some calculation to compute the global
* (total) progress
* In fact we simply divide the per-file progress by the number of files.
*
* This QtWatcher class will then update the progress bar according to the
* progress.
*/
class MyQtWatcher :  public SimpleSubjectWatcher
{
  size_t nfiles;
  double progress;
  size_t index;
  double refprogress;
  QWidget* win;
  QProgressDialog* qtprogress;
public:
  MyQtWatcher(Subject * s, const char *comment = "", QWidget *w = NULL, QProgressDialog* p = NULL, size_t n =
      1):
    SimpleSubjectWatcher(s,comment),nfiles(n),progress(0),index(0),refprogress(0),win(w),qtprogress(p){}
  void ShowIteration()
    {
    index++;
    assert( index <= nfiles );
    // update refprogess (we are moving to the next file)
    refprogress = progress;
    }
  void ShowProgress(Subject *, const Event &evt)
    {
    // Retrieve the ProgressEvent:
    const ProgressEvent &pe = dynamic_cast<const ProgressEvent&>(evt);
    // compute global progress:
    progress = refprogress + (1.  / (double)nfiles ) * pe.GetProgress();
    // Print Global and local progress to stdout:
    std::cout « "Global Progress:  " « progress « " per file progress " « pe.GetProgress() « std::endl;
    //set progress value in the QtProgress bar
    int i = (int)(progress * 100 + 0.5); // round to next int
    qtprogress->setValue(i);
    win->show();
    }
  virtual void ShowDataSet(Subject *caller, const Event &evt)
    {
    (void)caller;
    (void)evt;
    }
};
} // end namespace gdcm
int main(int argc, char *argv[])
{
  if( argc < 4 )
    {
    std::cerr « argv[0] « " remote_server port filename" « std::endl;
    return 1;
```

```
    }
  QApplication a(argc, argv);
  std::ostringstream error_log;
  gdcm::Trace::SetErrorStream( error_log );
  const char *remote = argv[1];
  int portno = atoi(argv[2]);
  const char *filename = argv[3];
  QVBoxLayout* layout = new QVBoxLayout;
  QWidget*  win = new QWidget;
  QProgressDialog* progress = new QProgressDialog("Sending data...", "Cancel", 0, 100);
  progress->setWindowModality(Qt::WindowModal);
  layout->addWidget(progress,Qt::AlignCenter);
  win->setLayout(layout);
  gdcm::SmartPointer<gdcm::ServiceClassUser> scup = new gdcm::ServiceClassUser;
  gdcm::ServiceClassUser &scu = *scup;
  //gdcm::SimpleSubjectWatcher w( &scu, "TestServiceClassUser" );
  // let's use a more complicated progress reported in this example
  gdcm::MyQtWatcher w( &scu, "QtWatcher", win, progress );
  scu.SetHostname( remote );
  scu.SetPort( (uint16_t)portno );
  scu.SetTimeout( 1000 );
  scu.SetCalledAETitle( "GDCM_STORE" );
  if( !scu.InitializeConnection() )
    {
    std::cerr << "Could not InitializeConnection" << std::endl;
    return 1;
    }
  gdcm::Directory::FilenamesType filenames;
  filenames.push_back( filename );
  // setup the PC(s) based on the filenames:
  gdcm::PresentationContextGenerator generator;
  if( !generator.GenerateFromFilenames(filenames) )
    {
    std::cerr << "Could not GenerateFromFilenames" << std::endl;
    return 1;
    }
  // Setup PresentationContext(s)
  scu.SetPresentationContexts( generator.GetPresentationContexts() );
  // Start ASSOCIATION
  if( !scu.StartAssociation() )
    {
    std::cerr << "Could not Start" << std::endl;
    return 1;
    }
  // Send C-STORE
  if( !scu.SendStore( filename ) )
    {
    std::cerr << "Could not Store" << std::endl;
    std::cerr << "Error log is:" << std::endl;
    std::cerr << error_log.str() << std::endl;
    return 1;
    }
  // Stop ASSOCIATION
  if( !scu.StopAssociation() )
    {
    std::cerr << "Could not Stop" << std::endl;
    return 1;
    }
  win->show();
  return a.exec();
}
```

12.36 ChangePrivateTags.cxx

```
/*=========================================================================

Program:  GDCM (Grassroots DICOM). A DICOM library

Copyright (c) 2006-2011 Mathieu Malaterre
All rights reserved.
See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.

This software is distributed WITHOUT ANY WARRANTY; without even
the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
PURPOSE.  See the above copyright notice for more information.

=========================================================================*/
```

```
#include "gdcmReader.h"
#include "gdcmWriter.h"
#include "gdcmPrivateTag.h"
int main(int argc, char* argv[] )
{
  if( argc < 3 )
    {
    std::cerr « argv[0] « " path/to/05148044-mr-siemens-avanto-syngo.dcm output.dcm" « std::endl;
    return 1;
    }
  const char *filename = argv[1];
  const char *outfilename = argv[2];
  gdcm::Reader reader;
  reader.SetFileName( filename );
  if (!  reader.Read() )
    {
    return 1;
    }
  // (0029,0010) LO [SIEMENS CSA HEADER]                                  # 18,1 Private Creator
  // (0029,0011) LO [SIEMENS MEDCOM HEADER ]                              # 22,1 Private Creator
  // (0029,0012) LO [SIEMENS MEDCOM HEADER2]                              # 22,1 Private Creator
  // [...]
  // (0029,1018) CS [MR]                                                  # 2,1 CSA Series Header Type
  // (0029,1134) CS [DB TO DICOM ]                                        # 12,1 PMTF Information 4
  // (0029,1260) LO [com ]                                                # 4,1 Series Workflow Status
  gdcm::File &file = reader.GetFile();
  gdcm::DataSet &ds = file.GetDataSet();
  // Declare private tag we need to find:
  gdcm::PrivateTag pt1( 0x29,0x18, "SIEMENS CSA HEADER" );
  gdcm::PrivateTag pt2( 0x29,0x34, "SIEMENS MEDCOM HEADER" );
  gdcm::PrivateTag pt3( 0x29,0x60, "SIEMENS MEDCOM HEADER2" );
  const char str1[] = "GDCM was here 3!";
  if( !ds.FindDataElement( pt1 ) ) return 1;
  gdcm::DataElement de1 = ds.GetDataElement( pt1 ); // Convert Private tag, into actual DataElement
  std::cout « de1 « std::endl;
  de1.SetByteValue( str1, (uint32_t)strlen(str1) );
  ds.Replace( de1 );
  const char str2[] = "GDCM was here 2!";
  if( !ds.FindDataElement( pt2 ) ) return 1;
  gdcm::DataElement de2 = ds.GetDataElement( pt2 );
  std::cout « de2 « std::endl;
  de2.SetByteValue( str2, (uint32_t)strlen(str2) );
  ds.Replace( de2 );
  const char str3[] = "GDCM was here 3!";
  if( !ds.FindDataElement( pt3 ) ) return 1;
  gdcm::DataElement de3 = ds.GetDataElement( pt3 );
  std::cout « de3 « std::endl;
  de3.SetByteValue( str3, (uint32_t)strlen(str3) );
  ds.Replace( de3 );
  gdcm::Writer writer;
  writer.SetFile( file );
  writer.SetFileName( outfilename );
  if ( !writer.Write() )
    {
    return 1;
    }
  return 0;
}
```

12.37 ChangeSequenceUltrasound.cxx

```
/*=========================================================================

Program:  GDCM (Grassroots DICOM). A DICOM library

Copyright (c) 2006-2011 Mathieu Malaterre
All rights reserved.
See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.

This software is distributed WITHOUT ANY WARRANTY; without even
the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
PURPOSE.  See the above copyright notice for more information.

=========================================================================*/
#include "gdcmReader.h"
#include "gdcmWriter.h"
#include "gdcmSmartPointer.h"
```

```
#include "gdcmDataSetHelper.h"
/*
./ChangeSequenceUltrasound gdcmData/D_CLUNIE_CT1_J2KI.dcm myoutput.dcm

This is the exact C++ translation of the original python example:  ManipulateSequence.py
*/
int main(int argc, char* argv[] )
{
  if( argc < 3 )
    {
    std::cerr « argv[0] « " input.dcm output.dcm" « std::endl;
    return 1;
    }
  const char *filename = argv[1];
  const char *outfilename = argv[2];
  gdcm::Reader reader;
  reader.SetFileName( filename );
  if (!  reader.Read() )
    {
    return 1;
    }
  gdcm::File &file = reader.GetFile();
  gdcm::DataSet &ds = file.GetDataSet();
  gdcm::Tag tsis(0x0008,0x2112); // SourceImageSequence
  if ( ds.FindDataElement( tsis ) )
    {
    const gdcm::DataElement &sis = ds.GetDataElement( tsis );
    gdcm::SmartPointer<gdcm::SequenceOfItems> sqsis = sis.GetValueAsSQ();
    if ( sqsis && sqsis->GetNumberOfItems() )
      {
      gdcm::Item &item1 = sqsis->GetItem(1);
      gdcm::DataSet &nestedds = item1.GetNestedDataSet();
      gdcm::Tag tprcs(0x0040,0xa170); // PurposeOfReferenceCodeSequence
      if( nestedds.FindDataElement( tprcs ) )
        {
        const gdcm::DataElement &prcs = nestedds.GetDataElement( tprcs );
        gdcm::SmartPointer<gdcm::SequenceOfItems> sqprcs = prcs.GetValueAsSQ();
        if ( sqprcs && sqprcs->GetNumberOfItems() )
          {
          gdcm::Item &item2 = sqprcs->GetItem(1);
          gdcm::DataSet &nestedds2 = item2.GetNestedDataSet();
          // (0008,0104) LO [Uncompressed predecessor]               #  24, 1 CodeMeaning
          gdcm::Tag tcm(0x0008,0x0104);
          if( nestedds2.FindDataElement( tcm ) )
            {
            gdcm::DataElement cm = nestedds2.GetDataElement( tcm );
            std::string mystr = "GDCM was here";
            cm.SetByteValue( mystr.c_str(), (uint32_t)mystr.size() );
            nestedds2.Replace( cm );
            }
          }
        }
      }
    }
  gdcm::Writer writer;
  writer.SetFile( file );
  writer.SetFileName( outfilename );
  if ( !writer.Write() )
    {
    return 1;
    }
  return 0;
}
```

12.38 CheckBigEndianBug.cxx

```
/*=========================================================================

Program:  GDCM (Grassroots DICOM). A DICOM library

Copyright (c) 2006-2011 Mathieu Malaterre
All rights reserved.
See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.

This software is distributed WITHOUT ANY WARRANTY; without even
the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
PURPOSE.  See the above copyright notice for more information.
```

```
============================================================================*/
/*
* WARNING: This is a dev tool, do not use !
*
* Usage:  after a gdcmconv, you would like to know if the conversion process is acceptable
* sometime a vbindiff is acceptable, sometime it is not.  In the case of the famous Philips
* Little/Big Endian Explicit Transfer Syntax it is not easy to compare two files.  However
* this only impact byte ordering, thus we can compute byte-indenpendant information to still
* compare the files.
*/
#include "gdcmImageReader.h"
#include "gdcmImage.h"
#include "gdcmWriter.h"
#include "gdcmAttribute.h"
#include "gdcmSystem.h"
#include <iostream>
#include <fstream>
int main(int argc, char *argv[])
{
  if( argc < 3 )
    {
    std::cerr « argv[0] « " input1.dcm input2.dcm" « std::endl;
    return 1;
    }
  const char *filename1 = argv[1];
  const char *filename2 = argv[2];
  gdcm::ImageReader reader1;
  reader1.SetFileName( filename1 );
  if( !reader1.Read() )
    {
    std::cerr « "Could not read:  " « filename1 « std::endl;
    return 1;
    }
  gdcm::ImageReader reader2;
  reader2.SetFileName( filename2 );
  if( !reader2.Read() )
    {
    std::cerr « "Could not read:  " « filename2 « std::endl;
    return 1;
    }
  // TODO: need a DataSet== operator implementation
  std::cout « "Both files can be read and looks like DICOM" « std::endl;
  size_t s1 = gdcm::System::FileSize(filename1);
  size_t s2 = gdcm::System::FileSize(filename2);
  if( s1 != s2 )
    {
    std::cout « "Size mismatch:  " « s1 « " != " « s2 « std::endl;
    return 1;
    }
  else
    {
    std::cout « "Size match:  " « s1 « " = " « s2 « std::endl;
    }
  std::ifstream is1( filename1, std::ios::binary );
  char *buffer1 = new char[s1];
  is1.read(buffer1, s1);
  std::ifstream is2( filename2, std::ios::binary );
  char *buffer2 = new char[s2];
  is2.read(buffer2, s2);
  assert( s1 == s2 );
  if( memcmp(buffer1, buffer2, s1 ) == 0 )
    {
    std::cout « "memcmp succeed !  File are bit identical" « std::endl;
    }
  else
    {
    std::cout « "memcmp failed!" « std::endl;
    }
  // Hum...memcmp failed, for big endian/ little endian inversion the histogram of bytes
  // should still be the same.  So let's compute it
  // buffer2[0] = 1; // let's make the test fail
  std::multiset<char> set1( buffer1, buffer1 + s1 );
  std::multiset<char> set2( buffer2, buffer2 + s2 );
  if( set1 == set2 )
    {
    std::cout « "set1 == set2.  Byte histogram seems valid" « std::endl;
    }
  else
    {
    std::cout « "set1 != set2" « std::endl;
```

```
    }
  delete[] buffer1;
  delete[] buffer2;
  return 0;
}
```

12.39 ClinicalTrialAnnotate.cxx

```
/*=========================================================================

Program:  GDCM (Grassroots DICOM). A DICOM library

Copyright (c) 2006-2011 Mathieu Malaterre
All rights reserved.
See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.

This software is distributed WITHOUT ANY WARRANTY; without even
the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
PURPOSE.  See the above copyright notice for more information.

=========================================================================*/
/*
* Dummy implementation of C.7.1.3 Clinical Trial Subject Module
*
* Usage:
*  ClinicalTrialAnnotate gdcmData/012345.002.050.dcm out.dcm
*/
#include "gdcmReader.h"
#include "gdcmWriter.h"
#include "gdcmAnonymizer.h"
int main(int argc, char *argv[])
{
  if( argc < 3 )
    {
    std::cerr « argv[0] « " input.dcm output.dcm" « std::endl;
    return 1;
    }
  const char *filename = argv[1];
  const char *outfilename = argv[2];
  gdcm::Reader reader;
  reader.SetFileName( filename );
  if( !reader.Read() )
    {
    std::cerr « "Could not read:  " « filename « std::endl;
    return 1;
    }
  // The output of gdcm::Reader is a gdcm::File
  //gdcm::File &file = reader.GetFile();
  // the dataset is the the set of element we are interested in:
  //gdcm::DataSet &ds = file.GetDataSet();
  gdcm::Anonymizer ano;
  ano.SetFile( reader.GetFile() );
  ano.RemoveGroupLength();
  ano.RemovePrivateTags();
  // PS 3.3 - 2008
  // C.7.1.3 Clinical Trial Subject Module
  // <entry group="0012" element="0010" vr="LO" vm="1" name="Clinical Trial Sponsor Name"/>
  ano.Replace( gdcm::Tag(0x12,0x10), "BigCompany name" );
  // <entry group="0012" element="0020" vr="LO" vm="1" name="Clinical Trial Protocol ID"/>
  ano.Replace( gdcm::Tag(0x12,0x20), "My Clinical Trial Protocol ID" );
  // <entry group="0012" element="0021" vr="LO" vm="1" name="Clinical Trial Protocol Name"/>
  ano.Replace( gdcm::Tag(0x12,0x21), "My Clinical Trial Protocol Name" );
  // <entry group="0012" element="0030" vr="LO" vm="1" name="Clinical Trial Site ID"/>
  ano.Replace( gdcm::Tag(0x12,0x30), "My Clinical Trial Site ID" );
  // <entry group="0012" element="0031" vr="LO" vm="1" name="Clinical Trial Site Name"/>
  ano.Replace( gdcm::Tag(0x12,0x31), "My Clinical Trial Site Name" );
  // <entry group="0012" element="0040" vr="LO" vm="1" name="Clinical Trial Subject ID"/>
  ano.Replace( gdcm::Tag(0x12,0x40), "My Clinical Trial Subject ID" );
  // <entry group="0012" element="0042" vr="LO" vm="1" name="Clinical Trial Subject Reading ID"/>
  ano.Replace( gdcm::Tag(0x12,0x42), "My Clinical Trial Subject Reading ID" );
  gdcm::Writer writer;
  writer.SetFile( reader.GetFile() );
  writer.SetFileName( outfilename );
  if( !writer.Write() )
    {
    return 1;
    }
```

```
  return 0;
}
```

12.40 CompressImage.cxx

```
/*=========================================================================

Program:  GDCM (Grassroots DICOM). A DICOM library

Copyright (c) 2006-2011 Mathieu Malaterre
All rights reserved.
See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.

This software is distributed WITHOUT ANY WARRANTY; without even
the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
PURPOSE.  See the above copyright notice for more information.

=========================================================================*/
/*
*
*/
#include "gdcmImageReader.h"
#include "gdcmImage.h"
#include "gdcmWriter.h"
#include "gdcmAttribute.h"
#include "gdcmImageWriter.h"
#include "gdcmImageChangeTransferSyntax.h"
#include <iostream>
#include <fstream>
int main(int argc, char *argv[])
{
  if( argc < 3 )
    {
    std::cerr « argv[0] « " input.dcm output.dcm" « std::endl;
    return 1;
    }
  const char *filename = argv[1];
  const char *outfilename = argv[2];
  gdcm::ImageReader reader;
  reader.SetFileName( filename );
  if( !reader.Read() )
    {
    std::cerr « "Could not read:  " « filename « std::endl;
    return 1;
    }
  // The output of gdcm::Reader is a gdcm::File
  //gdcm::File &file = reader.GetFile();
  // the dataset is the the set of element we are interested in:
  //gdcm::DataSet &ds = file.GetDataSet();
  gdcm::Image &image = reader.GetImage();
  // image.SetSpacing(0, 0.1);
  // image.SetSpacing(1, 0.2);
  image.Print( std::cout );
  gdcm::ImageChangeTransferSyntax change;
  change.SetTransferSyntax( gdcm::TransferSyntax::JPEG2000Lossless );
  change.SetTransferSyntax( gdcm::TransferSyntax::JPEGLosslessProcess14_1 );
  //change.SetTransferSyntax( gdcm::TransferSyntax::JPEGBaselineProcess1 );
  //change.SetTransferSyntax( image.GetTransferSyntax() );
  change.SetInput( image );
  bool b = change.Change();
  if( !b )
    {
    std::cerr « "Could not change the Transfer Syntax" « std::endl;
    return 1;
    }
  //std::ofstream out( outfilename, std::ios::binary );
  //image.GetBuffer2(out);
  //out.close();
  gdcm::ImageWriter writer;
  writer.SetImage( change.GetOutput() );
  writer.SetFile( reader.GetFile() );
  writer.SetFileName( outfilename );
  if( !writer.Write() )
    {
    return 1;
    }
  return 0;
}
```

12.41 ConvertToQImage.cxx

```
/*=========================================================================

Program:  GDCM (Grassroots DICOM). A DICOM library

Copyright (c) 2006-2011 Mathieu Malaterre
All rights reserved.
See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.

This software is distributed WITHOUT ANY WARRANTY; without even
the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
PURPOSE.  See the above copyright notice for more information.

=========================================================================*/
/*
* This example shows how to setup the pipeline from a gdcm::ImageReader into a
* Qt QImage data structure.
* It only handles 2D image.
*
* Ref:
* http://doc.trolltech.com/4.5/qimage.html
*
* Usage:
*  ConvertToQImage gdcmData/012345.002.050.dcm output.png

* Thanks:
*   Sylvain ADAM (sylvain51 hotmail com) for contributing this example
*/
#include "gdcmImageReader.h"
#include <QImage>
#include <QImageWriter>
bool ConvertToFormat_RGB888(gdcm::Image const & gimage, char *buffer, QImage* &imageQt)
{
  const unsigned int* dimension = gimage.GetDimensions();
  unsigned int dimX = dimension[0];
  unsigned int dimY = dimension[1];
  gimage.GetBuffer(buffer);
  // Let's start with the easy case:
  if( gimage.GetPhotometricInterpretation() == gdcm::PhotometricInterpretation::RGB )
    {
    if( gimage.GetPixelFormat() != gdcm::PixelFormat::UINT8 )
      {
      return false;
      }
    unsigned char *ubuffer = (unsigned char*)buffer;
    // QImage::Format_RGB888  13  The image is stored using a 24-bit RGB format (8-8-8).
    imageQt = new QImage((unsigned char *)ubuffer, dimX, dimY, 3*dimX, QImage::Format_RGB888);
    }
  else if( gimage.GetPhotometricInterpretation() == gdcm::PhotometricInterpretation::MONOCHROME2 )
    {
    if( gimage.GetPixelFormat() == gdcm::PixelFormat::UINT8 )
      {
      // We need to copy each individual 8bits into R / G and B:
      unsigned char *ubuffer = new unsigned char[dimX*dimY*3];
      unsigned char *pubuffer = ubuffer;
      for(unsigned int i = 0; i < dimX*dimY; i++)
        {
        *pubuffer++ = *buffer;
        *pubuffer++ = *buffer;
        *pubuffer++ = *buffer++;
        }
      imageQt = new QImage(ubuffer, dimX, dimY, QImage::Format_RGB888);
      }
    else if( gimage.GetPixelFormat() == gdcm::PixelFormat::INT16 )
      {
      // We need to copy each individual 16bits into R / G and B (truncate value)
      short *buffer16 = (short*)buffer;
      unsigned char *ubuffer = new unsigned char[dimX*dimY*3];
      unsigned char *pubuffer = ubuffer;
      for(unsigned int i = 0; i < dimX*dimY; i++)
        {
        // Scalar Range of gdcmData/012345.002.050.dcm is [0,192], we could simply do:
        // *pubuffer++ = *buffer16;
        // *pubuffer++ = *buffer16;
        // *pubuffer++ = *buffer16;
        // instead do it right:
        *pubuffer++ = (unsigned char)std::min(255, (32768 + *buffer16) / 255);
        *pubuffer++ = (unsigned char)std::min(255, (32768 + *buffer16) / 255);
        *pubuffer++ = (unsigned char)std::min(255, (32768 + *buffer16) / 255);
```

```
        buffer16++;
        }
      imageQt = new QImage(ubuffer, dimX, dimY, QImage::Format_RGB888);
      }
    else
      {
      std::cerr « "Pixel Format is:  " « gimage.GetPixelFormat() « std::endl;
      return false;
      }
    }
  else
    {
    std::cerr « "Unhandled PhotometricInterpretation:  " « gimage.GetPhotometricInterpretation() « std::endl;
    return false;
    }
  return true;
}
int main(int argc, char *argv[])
{
  if( argc < 2 )
    {
    return 1;
    }
  const char *filename = argv[1];
  const char *outfilename = argv[2];
  gdcm::ImageReader ir;
  ir.SetFileName( filename );
  if(!ir.Read())
    {
    //Read failed
    return 1;
    }
  std::cout«"Getting image from ImageReader..."«std::endl;
  const gdcm::Image &gimage = ir.GetImage();
  std::vector<char> vbuffer;
  vbuffer.resize( gimage.GetBufferLength() );
  char *buffer = &vbuffer[0];
  QImage *imageQt = NULL;
  if( !ConvertToFormat_RGB888( gimage, buffer, imageQt ) )
    {
    return 1;
    }
  QImageWriter writer;
  writer.setFormat("png");
  writer.setFileName( outfilename );
  if( !writer.write( *imageQt ) )
    {
    return 1;
    }
  return 0;
}
```

12.42 CreateARGBImage.cxx

```
/*=========================================================================

Program:  GDCM (Grassroots DICOM). A DICOM library

Copyright (c) 2006-2011 Mathieu Malaterre
All rights reserved.
See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.

This software is distributed WITHOUT ANY WARRANTY; without even
the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
PURPOSE.  See the above copyright notice for more information.

=========================================================================*/
/*
* http://www.w3.org/Graphics/PNG/inline-alpha.html
* alphatest.png:  PNG image data, 380 x 287, 8-bit/color RGBA, non-interlaced
*
* $ convert alphatest.png alphatest.rgba
*/
#include "gdcmImageReader.h"
#include "gdcmSequenceOfFragments.h"
#include "gdcmSystem.h"
#include "gdcmImageWriter.h"
```

```
#include <iostream>
#include <fstream>
int main(int argc, char *argv[])
{
  if( argc < 3 )
    {
    std::cerr « argv[0] « " input.rgba output.dcm" « std::endl;
    return 1;
    }
  const char *filename = argv[1];
  const char *outfilename = argv[2];
  size_t len = gdcm::System::FileSize(filename);
  std::ifstream is(filename, std::ios::binary);
  char * buf = new char[len];
  is.read(buf, len);
  gdcm::ImageWriter writer;
  gdcm::Image &image = writer.GetImage();
  image.SetNumberOfDimensions( 2 );
  unsigned int dims[3] = {};
  dims[0] = 380;
  dims[1] = 287;
  image.SetDimensions( dims );
  gdcm::PixelFormat pf = gdcm::PixelFormat::UINT8;
  pf.SetSamplesPerPixel( 4 );
  image.SetPixelFormat( pf );
  gdcm::PhotometricInterpretation pi = gdcm::PhotometricInterpretation::ARGB;
  image.SetPhotometricInterpretation( pi );
  image.SetTransferSyntax( gdcm::TransferSyntax::ExplicitVRLittleEndian );
  gdcm::DataElement pixeldata( gdcm::Tag(0x7fe0,0x0010) );
  pixeldata.SetByteValue( buf, (uint32_t)len );
  image.SetDataElement( pixeldata );
  writer.SetFileName( outfilename );
  if( !writer.Write() )
    {
    return 1;
    }
  delete[] buf;
  return 0;
}
```

12.43 CreateCMYKImage.cxx

```
/*=========================================================================

Program:  GDCM (Grassroots DICOM). A DICOM library

Copyright (c) 2006-2011 Mathieu Malaterre
All rights reserved.
See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.

This software is distributed WITHOUT ANY WARRANTY; without even
the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
PURPOSE.  See the above copyright notice for more information.

=========================================================================*/
/*
* http://www.w3.org/Graphics/PNG/inline-alpha.html
* alphatest.png:  PNG image data, 380 x 287, 8-bit/color RGBA, non-interlaced
*
* $ convert alphatest.png alphatest.cmyk
*/
#include "gdcmImageReader.h"
#include "gdcmSequenceOfFragments.h"
#include "gdcmSystem.h"
#include "gdcmImageWriter.h"
#include <iostream>
#include <fstream>
int main(int argc, char *argv[])
{
  if( argc < 3 )
    {
    std::cerr « argv[0] « " input.cmyk output.dcm" « std::endl;
    return 1;
    }
  const char *filename = argv[1];
  const char *outfilename = argv[2];
  size_t len = gdcm::System::FileSize(filename);
```

```
  std::ifstream is(filename, std::ios::binary);
  char * buf = new char[len];
  is.read(buf, len);
  gdcm::ImageWriter writer;
  gdcm::Image &image = writer.GetImage();
  image.SetNumberOfDimensions( 2 );
  unsigned int dims[3] = {};
  dims[0] = 380;
  dims[1] = 287;
  image.SetDimensions( dims );
  gdcm::PixelFormat pf = gdcm::PixelFormat::UINT8;
  pf.SetSamplesPerPixel( 4 );
  image.SetPixelFormat( pf );
  gdcm::PhotometricInterpretation pi = gdcm::PhotometricInterpretation::CMYK;
  image.SetPhotometricInterpretation( pi );
  image.SetTransferSyntax( gdcm::TransferSyntax::ExplicitVRLittleEndian );
  gdcm::DataElement pixeldata( gdcm::Tag(0x7fe0,0x0010) );
  pixeldata.SetByteValue( buf, (uint32_t)len );
  image.SetDataElement( pixeldata );
  writer.SetFileName( outfilename );
  if( !writer.Write() )
    {
    return 1;
    }
  delete[] buf;
  return 0;
}
```

12.44 CreateJPIPDataSet.cxx

```
/*=========================================================================

Program:  GDCM (Grassroots DICOM). A DICOM library

Copyright (c) 2006-2011 Mathieu Malaterre
All rights reserved.
See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.

This software is distributed WITHOUT ANY WARRANTY; without even
the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
PURPOSE.  See the above copyright notice for more information.

=========================================================================*/
/*
* This example was created during the GSOC 2011 project for
* JPIP
*/
#include "gdcmAnonymizer.h"
#include "gdcmWriter.h"
#include "gdcmUIDGenerator.h"
#include "gdcmFile.h"
#include "gdcmTag.h"
#include "gdcmSystem.h"
#include "gdcmAttribute.h"
int main(int argc, char *argv[])
{
  if( argc < 2 )
    {
    std::cerr « argv[0] « " output.dcm" « std::endl;
    return 1;
    }
  const char *outfilename = argv[1];
  gdcm::Writer w;
  gdcm::File &file = w.GetFile();
  gdcm::DataSet &ds = file.GetDataSet();
  //w.SetCheckFileMetaInformation( true );
  w.SetFileName( outfilename );
  file.GetHeader().SetDataSetTransferSyntax( gdcm::TransferSyntax::JPIPReferenced );
  gdcm::Anonymizer anon;
  anon.SetFile( file );
  gdcm::MediaStorage ms = gdcm::MediaStorage::SecondaryCaptureImageStorage;
  gdcm::UIDGenerator gen;
  anon.Replace( gdcm::Tag(0x0008,0x16), ms.GetString() );
  std::cout « ms.GetString() « std::endl;
  anon.Replace( gdcm::Tag(0x0008,0x18), gen.Generate() );
  //
  anon.Replace( gdcm::Tag(0x0010,0x10), "JPIP^EXAMPLE" );
```

```
  anon.Replace( gdcm::Tag(0x0010,0x20), "012345" );
  anon.Empty( gdcm::Tag(0x0010,0x30) );
  anon.Empty( gdcm::Tag(0x0010,0x40) );
  anon.Empty( gdcm::Tag(0x0008,0x20) );
  anon.Empty( gdcm::Tag(0x0008,0x30) );
  anon.Empty( gdcm::Tag(0x0008,0x90) );
  anon.Empty( gdcm::Tag(0x0020,0x10) );
  anon.Empty( gdcm::Tag(0x0020,0x11) );
  anon.Empty( gdcm::Tag(0x0008,0x50) );
  anon.Empty( gdcm::Tag(0x0020,0x0013) );
  anon.Replace( gdcm::Tag(0x0020,0xd), gen.Generate() );
  anon.Replace( gdcm::Tag(0x0020,0xe), gen.Generate() );
  anon.Replace( gdcm::Tag(0x0008,0x64), "WSD " );
  anon.Replace( gdcm::Tag(0x0008,0x60), "OT" );
  gdcm::Attribute<0x0028,0x7FE0> at;
  at.SetValue( "http://dicom.example.com/jpipserver.cgi?target=img.jp2" );
  ds.Insert( at.GetAsDataElement() );
  // Need to retrieve the PixelFormat information from the given file
  if (!w.Write() )
    {
    std::cerr « "Could not write:  " « outfilename « std::endl;
    return 1;
    }
  return 0;
}
```

12.45 DeriveSeries.cxx

```
/*=========================================================================

Program:  GDCM (Grassroots DICOM). A DICOM library

Copyright (c) 2006-2011 Mathieu Malaterre
All rights reserved.
See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.

This software is distributed WITHOUT ANY WARRANTY; without even
the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
PURPOSE.  See the above copyright notice for more information.

=========================================================================*/
#include "gdcmReader.h"
#include "gdcmWriter.h"
#include "gdcmAttribute.h"
#include "gdcmFileDerivation.h"
#include "gdcmUIDGenerator.h"
int main(int argc, char *argv[])
{
  if( argc < 3 )
  {
    return 1;
  }
  const char * ref = argv[1];
  const char * in  = argv[2];
  gdcm::Reader r1;
  r1.SetFileName( ref );
  if( !r1.Read() ) return 1;
  gdcm::Reader r2;
  r2.SetFileName( in );
  if( !r2.Read() ) return 1;
  // Fix Spatial info:
  gdcm::DataSet & ds1 = r1.GetFile().GetDataSet();
  gdcm::File & file2 = r2.GetFile();
  gdcm::DataSet & ds2 = file2.GetDataSet();
  //gdcm::Attribute<0x8,0x8> img_type = { "ORIGINAL", "PRIMARY" };
  ds2.Replace( ds1.GetDataElement( gdcm::Tag(0x0008,0x0008) ));
  ds2.Replace( ds1.GetDataElement( gdcm::Tag(0x0020,0x0032) ));
  ds2.Replace( ds1.GetDataElement( gdcm::Tag(0x0020,0x0037) ));
  ds2.Replace( ds1.GetDataElement( gdcm::Tag(0x0018,0x0088) )); // Spacing between slices
  ds2.Replace( ds1.GetDataElement( gdcm::Tag(0x0020,0x0013) )); // Instance Number
  ds2.Replace( ds1.GetDataElement( gdcm::Tag(0x0018,0x5100) )); // Patient Position
  ds2.Replace( ds1.GetDataElement( gdcm::Tag(0x0018,0x0050) )); // Slice Thickness
  ds2.Replace( ds1.GetDataElement( gdcm::Tag(0x0008,0x0070) )); // Manufacturer
  ds2.Replace( ds1.GetDataElement( gdcm::Tag(0x0018,0x0081) )); // Echo Time
  ds2.Replace( ds1.GetDataElement( gdcm::Tag(0x0020,0x1041) )); // Slice Location
  gdcm::Attribute<0x8,0x16> sopclassuid;
  sopclassuid.SetFromDataSet( ds1 );
```

```
  gdcm::Attribute<0x8,0x18> sopinstanceuid;
  sopinstanceuid.SetFromDataSet( ds1 );
  // Step 2:  DERIVED object
  gdcm::FileDerivation fd;
  fd.AddReference( sopclassuid.GetValue(), sopinstanceuid.GetValue() );
  // http://dicom.nema.org/MEDICAL/dicom/current/output/chtml/part16/chapter_D.html#DCM_121321
  // CID 7202 "Source Image Purposes of Reference"
  // DCM 121321 "Mask image for image processing operation"
  fd.SetPurposeOfReferenceCodeSequenceCodeValue( 121321 );
  // CID 7203 "Image Derivation"
  // DCM 113047 "Pixel by pixel mask"
  fd.SetDerivationCodeSequenceCodeValue( 113047 );
  fd.SetFile( file2 );
  // If all Code Value are ok the filter will execute properly
  if( !fd.Derive() )
    {
    std::cerr « "Sorry could not derive using input info" « std::endl;
    return 1;
    }
  gdcm::Writer w;
  w.SetFile( r2.GetFile() );
  w.SetFileName( "derived.dcm" );
  if( !w.Write() )
    {
    return 1;
    }
  return 0;
}
```

12.46 DiffFile.cxx

```
/*=========================================================================

Program:  GDCM (Grassroots DICOM). A DICOM library

Copyright (c) 2006-2011 Mathieu Malaterre
All rights reserved.
See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.

This software is distributed WITHOUT ANY WARRANTY; without even
the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
PURPOSE.  See the above copyright notice for more information.

=========================================================================*/
#include "gdcmReader.h"
int main(int argc, char *argv[])
{
  if( argc < 3 )
    {
    std::cerr « argv[0] « " input1.dcm input2.dcm" « std::endl;
    return 1;
    }
  const char *filename1 = argv[1];
  const char *filename2 = argv[2];
  gdcm::Reader reader1;
  reader1.SetFileName( filename1 );
  if( !reader1.Read() )
    {
    return 1;
    }
  gdcm::Reader reader2;
  reader2.SetFileName( filename2 );
  if( !reader2.Read() )
    {
    return 1;
    }
  const gdcm::File &file1 = reader1.GetFile();
  const gdcm::File &file2 = reader2.GetFile();
  const gdcm::DataSet &ds1 = file1.GetDataSet();
  const gdcm::DataSet &ds2 = file2.GetDataSet();
  gdcm::DataSet::ConstIterator it1 = ds1.Begin();
  gdcm::DataSet::ConstIterator it2 = ds2.Begin();
  const gdcm::DataElement &de1 = *it1;
  const gdcm::DataElement &de2 = *it2;
  if( de1 == de2 )
    {
    }
```

```
  while( it1 != ds1.End() && it2 != ds2.End() && *it1 == *it2 )
    {
    ++it1;
    ++it2;
    }
  if( it1 != ds1.End() || it2 != ds2.End() )
    {
    std::cerr « "Problem with:" « std::endl;
    if( it1 != ds1.End() )
      {
      std::cerr « "ds1:  " « *it1 « std::endl;
      }
    if( it2 != ds2.End() )
      {
      std::cerr « "ds2:  " « *it2 « std::endl;
      }
    return 1;
    }
  return 0;
}
```

12.47 DiscriminateVolume.cxx

```
/*=========================================================================

Program:  GDCM (Grassroots DICOM). A DICOM library

Copyright (c) 2006-2011 Mathieu Malaterre
All rights reserved.
See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.

This software is distributed WITHOUT ANY WARRANTY; without even
the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
PURPOSE.  See the above copyright notice for more information.

=========================================================================*/
#include "gdcmScanner.h"
#include "gdcmTesting.h"
#include "gdcmIPPSorter.h"
#include "gdcmDirectionCosines.h"
#include <cmath>
/*
* The following example is a basic sorted which should work in generic cases.
* It sort files based on:
* Study Instance UID
*   Series Instance UID
*     Frame of Reference UID
*       Image Orientation (Patient)
*         Image Position (Patient) (Sorting based on IPP + IOP)
*/
namespace gdcm {
  const Tag t1(0x0020,0x000d); // Study Instance UID
  const Tag t2(0x0020,0x000e); // Series Instance UID
  const Tag t3(0x0020,0x0052); // Frame of Reference UID
  const Tag t4(0x0020,0x0037); // Image Orientation (Patient)
class DiscriminateVolume
{
private:
  std::vector< Directory::FilenamesType > SortedFiles;
  std::vector< Directory::FilenamesType > UnsortedFiles;
  Directory::FilenamesType GetAllFilenamesFromTagToValue(
    Scanner const & s, Directory::FilenamesType const &filesubset, Tag const &t, const char *valueref)
{
  Directory::FilenamesType theReturn;
  if( valueref )
    {
    size_t len = strlen( valueref );
    Directory::FilenamesType::const_iterator file = filesubset.begin();
    for(; file != filesubset.end(); ++file)
      {
      const char *filename = file->c_str();
      const char * value = s.GetValue(filename, t);
      if( value && strncmp(value, valueref, len ) == 0 )
        {
        theReturn.push_back( filename );
        }
      }
```

```
    }
  return theReturn;
}
void ProcessAIOP(Scanner const & , Directory::FilenamesType const & subset, const char *iopval)
{
  std::cout « "IOP: " « iopval « std::endl;
  IPPSorter ipp;
  ipp.SetComputeZSpacing( true );
  ipp.SetZSpacingTolerance( 1e-3 ); // ??
  bool b = ipp.Sort( subset );
  if( !b )
    {
    // If you reach here this means you need one more parameter to discriminiat this
    // series.  Eg.  T1 / T2 intertwinted.  Multiple Echo (0018,0081)
    std::cerr « "Failed to sort:  " « subset.begin()->c_str() « std::endl;
    for(
      Directory::FilenamesType::const_iterator file = subset.begin();
      file != subset.end(); ++file)
      {
      std::cerr « *file « std::endl;
      }
    UnsortedFiles.push_back( subset );
    return ;
    }
  ipp.Print( std::cout );
  SortedFiles.push_back( ipp.GetFilenames() );
}
void ProcessAFrameOfRef(Scanner const & s, Directory::FilenamesType const & subset, const char * frameuid)
{
  // In this subset of files (belonging to same series), let's find those
  // belonging to the same Frame ref UID:
  Directory::FilenamesType files = GetAllFilenamesFromTagToValue(
    s, subset, t3, frameuid);
  std::set< std::string > iopset;
  for(
    Directory::FilenamesType::const_iterator file = files.begin();
    file != files.end(); ++file)
    {
    //std::cout « *file « std::endl;
    const char * value = s.GetValue(file->c_str(), gdcm::t4 );
    assert( value );
    iopset.insert( value );
    }
  size_t n = iopset.size();
  if ( n == 0 )
    {
    assert( files.empty() );
    return;
    }
  std::cout « "Frame of Ref:  " « frameuid « std::endl;
  if ( n == 1 )
    {
    ProcessAIOP(s, files, iopset.begin()->c_str() );
    }
  else
    {
    const char *f = files.begin()->c_str();
    std::cerr « "More than one IOP: " « f « std::endl;
    // Make sure that there is actually 'n' different IOP
    gdcm::DirectionCosines ref;
    gdcm::DirectionCosines dc;
    for(
      std::set< std::string >::const_iterator it = iopset.begin();
      it != iopset.end(); ++it )
      {
      ref.SetFromString( it->c_str() );
      for(
        Directory::FilenamesType::const_iterator file = files.begin();
        file != files.end(); ++file)
        {
        std::string value = s.GetValue(file->c_str(), gdcm::t4 );
        if( value != it->c_str() )
          {
          dc.SetFromString( value.c_str() );
          const double crossdot = ref.CrossDot(dc);
          const double eps = std::fabs( 1.  - crossdot );
          if( eps < 1e-6 )
            {
            std::cerr « "Problem with IOP discrimination:  " « file->c_str()
              « " " « it->c_str() « std::endl;
            return;
```

```
            }
          }
        }
      }
      // If we reach here this means there is actually 'n' different IOP
    for(
      std::set< std::string >::const_iterator it = iopset.begin();
      it != iopset.end(); ++it )
      {
      const char *iopvalue = it->c_str();
      Directory::FilenamesType iopfiles = GetAllFilenamesFromTagToValue(
        s, files, t4, iopvalue );
      ProcessAIOP(s, iopfiles, iopvalue );
      }
    }
}
void ProcessASeries(Scanner const & s, const char * seriesuid)
{
  std::cout « "Series:  " « seriesuid « std::endl;
  // let's find all files belonging to this series:
  Directory::FilenamesType seriesfiles = GetAllFilenamesFromTagToValue(
    s, s.GetFilenames(), t2, seriesuid);
  gdcm::Scanner::ValuesType vt3 = s.GetValues(t3);
  for(
    gdcm::Scanner::ValuesType::const_iterator it = vt3.begin()
    ; it != vt3.end(); ++it )
    {
    ProcessAFrameOfRef(s, seriesfiles, it->c_str());
    }
}
void ProcessAStudy(Scanner const & s, const char * studyuid)
{
  std::cout « "Study:  " « studyuid « std::endl;
  gdcm::Scanner::ValuesType vt2 = s.GetValues(t2);
  for(
    gdcm::Scanner::ValuesType::const_iterator it = vt2.begin()
    ; it != vt2.end(); ++it )
    {
    ProcessASeries(s, it->c_str());
    }
}
public:
void Print( std::ostream & os )
{
  os « "Sorted Files:  " « std::endl;
  for(
    std::vector< Directory::FilenamesType >::const_iterator it = SortedFiles.begin();
    it != SortedFiles.end(); ++it )
    {
    os « "Group:  " « std::endl;
    for(
      Directory::FilenamesType::const_iterator file = it->begin();
      file != it->end(); ++file)
      {
      os « *file « std::endl;
      }
    }
  os « "Unsorted Files:  " « std::endl;
  for(
    std::vector< Directory::FilenamesType >::const_iterator it = UnsortedFiles.begin();
    it != UnsortedFiles.end(); ++it )
    {
    os « "Group:  " « std::endl;
    for(
      Directory::FilenamesType::const_iterator file = it->begin();
      file != it->end(); ++file)
      {
      os « *file « std::endl;
      }
    }
}
  std::vector< Directory::FilenamesType > const & GetSortedFiles()const { return SortedFiles; }
  std::vector< Directory::FilenamesType > const & GetUnsortedFiles()const { return UnsortedFiles; }
void ProcessIntoVolume( Scanner const & s )
{
  gdcm::Scanner::ValuesType vt1 = s.GetValues( gdcm::t1 );
  for(
    gdcm::Scanner::ValuesType::const_iterator it = vt1.begin()
    ; it != vt1.end(); ++it )
    {
    ProcessAStudy( s, it->c_str() );
```

```
    }
}
};
} // namespace gdcm
int main(int argc, char *argv[])
{
  std::string dir1;
  if( argc < 2 )
    {
    const char *extradataroot = nullptr;
#ifdef GDCM_BUILD_TESTING
    extradataroot = gdcm::Testing::GetDataExtraRoot();
#endif
    if( !extradataroot )
      {
      return 1;
      }
    dir1 = extradataroot;
    dir1 += "/gdcmSampleData/ForSeriesTesting/VariousIncidences/ST1";
    }
  else
    {
    dir1 = argv[1];
    }
  gdcm::Directory d;
  d.Load( dir1.c_str(), true ); // recursive !
  gdcm::Scanner s;
  s.AddTag( gdcm::t1 );
  s.AddTag( gdcm::t2 );
  s.AddTag( gdcm::t3 );
  s.AddTag( gdcm::t4 );
  bool b = s.Scan( d.GetFilenames() );
  if( !b )
    {
    std::cerr « "Scanner failed" « std::endl;
    return 1;
    }
  gdcm::DiscriminateVolume dv;
  dv.ProcessIntoVolume( s );
  dv.Print( std::cout );
  return 0;
}
```

12.48 DumpADAC.cxx

```
/*=========================================================================

Program:  GDCM (Grassroots DICOM). A DICOM library

Copyright (c) 2006-2011 Mathieu Malaterre
All rights reserved.
See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.

This software is distributed WITHOUT ANY WARRANTY; without even
the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
PURPOSE.  See the above copyright notice for more information.

=========================================================================*/
/*
* the goal of this example is to mimic the behavior of disp_img_header
* see http://www.gmecorp-usa.com/IM/NM/GC/ADAC/SV/adactechtips/Released_01Q3.pdf
*/
#include "gdcmReader.h"
#include "gdcmPrivateTag.h"
#include "gdcmAttribute.h"
#include "gdcmImageWriter.h"
#include <iostream>
#include <fstream>
#include <vector>
#include <string.h>
#include <assert.h>
#include <stdint.h>
struct dict
{
  uint16_t key;
  const char *name;
};
```

```
dict Array[] = {
  { 0x01, "Patient name" },
  { 0x02, "Patient ID" },
  { 0x03, "Patient sex" },
  { 0x04, "Patient age" },
  { 0x05, "Patient height" },
  { 0x06, "Patient weight" },
  { 0x07, "Exam date" },
  { 0x08, "Dose admin.  time" },
  { 0x09, "Unique exam key" },
  { 0x0a, "Exam procedure" },
  { 0x0b, "Referring physician" },
  { 0x0c, "Attending physician" },
  { 0x0d, "Imaging modality" },
  { 0x0e, "Hospital ID" },
  { 0x0f, "Histogram crv file" },
  { 0x10, "Acq.  start time" },
  { 0x11, "Object data type" },
  { 0x12, "Image viewid" },
  { 0x13, "Imaging device name" },
  { 0x14, "Device serial number" },
  { 0x15, "Collimator" },
  { 0x16, "Software version" },
  { 0x17, "Radiopharmaceutical #1" },
  { 0x18, "Energy window #1 center" },
  { 0x19, "Radiopharmaceutical #2" },
  { 0x1a, "Energy window #1 width" },
  { 0x1b, "Isotope imaging mode" },
  { 0x1c, "Energy window #2 center" },
  { 0x1d, "Energy window #2 width" },
  { 0x1e, "Energy window #3 center" },
  { 0x1f, "Energy window #3 width" },
  { 0x20, "Energy window #4 center" },
  { 0x21, "Energy window #4 width" },
  { 0x22, "??Energy window #5 center" },
  { 0x23, "??Energy window #5 width" },
  { 0x24, "Patient orientation" },
  { 0x25, "Spatial resolution" },
  { 0x26, "Slice thickness" },
  { 0x27, "Image X dimension" },
  { 0x28, "Image Y dimension" },
  { 0x29, "Image Z dimension" },
  { 0x2a, "Image pixel width" },
  { 0x2b, "Uniformity corr.  file" },
  { 0x2c, "Acquisition zoom factor" },
  { 0x2d, "Total counts in set" },
  { 0x2e, "Time / frame" },
  { 0x2f, "Total acq.  time" },
  { 0x30, "Maximum pixel value" },
  { 0x31, "Minimum pixel value" },
  { 0x32, "R-R interval time" },
  { 0x33, "Percent of cycle imaged" },
  { 0x34, "# of cycles accepted" },
  { 0x35, "# of cycles rejected" },
  { 0x36, "Approximate ED frame" },
  { 0x37, "Approximate ES frame" },
  { 0x38, "Approximate EF" },
  { 0x39, "Starting angle" },
  { 0x3a, "Degrees of rotation" },
  { 0x3b, "Direction of rotation" },
  { 0x3c, "Cont.  or step/shoot" },
  { 0x3d, "Lim recon start frame" },
  { 0x3e, "Upper window grey shade" },
  { 0x3f, "Lower lvl grey shade" },
  { 0x40, "Associated color map" },
  { 0x41, "Custom color map file" },
  { 0x42, "Manipulated image" },
  { 0x43, "Axis of rotation corr." },
  { 0x44, "Reorientation azimuth" },
  { 0x45, "Reorientation elevation" },
  { 0x46, "Filter type" },
  { 0x47, "Filter order" },
  { 0x48, "Filter cutoff frequency" },
  { 0x49, "Reconstruction type" },
  { 0x4a, "Attenuation coefficient" },
  { 0x4b, "Associated parent file" },
  { 0x4c, "Unique patient key" },
  { 0x52, "Normalization crv file" },
  { 0x53, "Unique object key" },
  { 0x54, "This phase of VFR is" },
  { 0x55, "True color value" },
```

```
  { 0x56, "# of sets of x,y,z grps" },
  { 0x57, "Scale factor of set" },
  { 0x6d, "Date of birth" },
  { 0x6e, "Directional orientation" },
  { 0x6f, "Number of VFR studies" },
  { 0x70, "R-R low tolerance" },
  { 0x71, "R-R high tolerance" },
  { 0x72, "Prog specific results:" },
  { 0x99, nullptr }
};
void printname( int , int , uint16_t v )
{
  if( v == 0x1 )
    {
    std::cout « "DATABASE PARAMETERS" « std::endl;
    std::cout « "____________________" « std::endl;
    }
  else if( v == 0x27 )
    {
    std::cout « "IMAGE PARAMETERS" « std::endl;
    std::cout « "_________________" « std::endl;
    }
  else if( v == 0x13 )
    {
    std::cout « "EXTRA PARAMETERS" « std::endl;
    std::cout « "_________________" « std::endl;
    }
  else if( v == 0x2e )
    {
    std::cout « "*** NOT CURRENTLY USED :" « std::endl;
    }
  static const unsigned int n = sizeof( Array ) / sizeof( *Array ) - 1;
  for( unsigned int i = 0; i < n; ++i )
    {
    if( v == Array[i].key )
      {
      std::cout « /*"" « std::dec « len « "," « mult « " " « */ Array[i].name;
      std::cout « " :  ";
      return;
      }
    }
  std::cout « /*"\t# " « std::dec « len « "," « mult « */ std::hex « v « "\t:  ";
}
uint16_t readint16(std::istream &is )
{
  uint16_t val;
  is.read( (char*)&val, sizeof( val ));
  return (uint16_t)((val»8) | (val«8));
}
uint32_t readint32(std::istream &is )
{
  uint32_t val;
  is.read( (char*)&val, sizeof( val ));
  val= ((val«8)&0xFF00FF00) | ((val»8)&0x00FF00FF);
  return (val»16) | (val«16);
}
float readfloat32(std::istream &is )
{
  union { uint32_t val; float f;} dual;
  dual.val = readint32(is);
  return dual.f;
}
struct el
{
  uint16_t v1;
  uint16_t v2;
  uint16_t v3;
  void read( std::istream & is )
    {
    v1 = readint16(is);
    v2 = readint16(is);
    v3 = readint16(is);
    }
  void print( std::ostream & os )
    {
    os « std::hex « v1 « "\t" « v2 « "\t" « v3 « std::endl;
    }
};
std::vector<el> Vel;
void readelement( std::istream & is )
{
```

```
  el e;
  e.read( is );
  Vel.push_back( e );
}
void printascii( uint16_t tag, const char *buffer, size_t len )
{
  std::ostream & os = std::cout;
  if( tag == 0x72 )
    {
    os « "\n  ";
    for(size_t i = 0; i < len; ++i)
      {
      const char &c = buffer[i];
      if( c == 0x0 ) os « "!";
      else if( c == 0x0f ) os « " ";
      else if( c == 0x17 ) os « ":";
      else if( c == 0x14 ) os « ":";
      else if( c == 0x10 ) os « ":";
      else if( c == 0x16 ) os « ":";
      else if( c == 0x08 ) os « ":";
      else if( c == 0x0b ) os « ":";
      else if( c == 0x0e ) os « ":";
      else if( c == 0x07 ) os « ":";
      else os « c;
      }
    os « "";
    }
  else
    {
    (void)len;
    os « "" « buffer « "";
    }
}
bool DumpADAC( std::istream & is )
{
  std::ostream &os = std::cout;
  char magic[6 + 1];
  magic[6] = 0;
  is.read( magic, 6);
//  std::cout « magic « " ";
  assert( strcmp( magic, "adac01" ) == 0 );
  int c = is.get();
  assert( c == 0 ); (void)c;
  c = is.get();
  assert( c == 'X' );
  uint16_t v;
  v = readint16(is);
//  std::cout « v « std::endl;
  assert( v == 512 ); (void)v; // ??
  int nel = 87;
  for (int i = 0; i <= nel; ++i )
    {
    readelement( is );
    }
  char buffer[512];
  for( int i = 0; i <= nel; ++i )
    {
    const el &e = Vel[i];
    int diff;
    if( i == nel )
      {
      diff = 2048 - e.v3;
      if( diff > 512 ) diff = 512;
      }
    else
      {
      const el &enext = Vel[i+1];
      diff = enext.v3 - e.v3;
      }
    is.seekg( e.v3, std::ios::beg );
    //std::cout « "(" « std::hex « std::setw( 2 ) « std::setfill( '0' ) « e.v1 « ") " « std::hex « std::setw( 3
      ) « std::setfill( '0' ) « e.v2 « " ";
    printname( diff, 0, e.v1 );
    int mult = 1;
    if( e.v2 == 0 )
      {
      is.read( buffer, diff);
      buffer[ diff ] = 0;
      printascii( e.v1, buffer, diff);
      }
    else if( e.v2 == 0x100 )
```

```
      {
      mult = diff / 2;
      assert( diff == 2 * mult );
      for ( int ii = 0; ii < mult; ++ii )
        {
        if ( ii ) os « "\\";
        uint16_t val = readint16(is);
        os « "" « std::dec « val « "";
        }
      }
    else if( e.v2 == 0x200 )
      {
      assert( diff == 4 );
      uint32_t val = readint32(is);
      os « "" « std::dec « val « "";
      }
    else if( e.v2 == 0x300 )
      {
      assert( diff == 4 );
      float val = readfloat32(is);
      os « "" « std::dec « val « "";
      }
     else
      {
      assert( 0 );
      }
    os « std::endl;
    }
  return true;
}
int main(int argc, char *argv[])
{
  if( argc < 2 ) return 1;
  const char *filename = argv[1];
  gdcm::Reader reader;
  reader.SetFileName( filename );
  if( !reader.Read() )
    {
    std::cerr « "Failed to read:  " « filename « std::endl;
    return 1;
    }
  const gdcm::DataSet& ds = reader.GetFile().GetDataSet();
  // (0019,1061) UN (OB) 61\64\61\63\30         # 2048,1 Ver200 ADAC Pegasys Headers
  const gdcm::PrivateTag tver200adacpegasysheaders(0x0019,0x61,"ADAC_IMG");
  if( !ds.FindDataElement( tver200adacpegasysheaders ) ) return 1;
  const gdcm::DataElement& ver200adacpegasysheaders = ds.GetDataElement( tver200adacpegasysheaders );
  if ( ver200adacpegasysheaders.IsEmpty() ) return 1;
  const gdcm::ByteValue * bv = ver200adacpegasysheaders.GetByteValue();
  // (0019,1021) US 1               # 2,1 Ver200 Number of ADAC Headers
  // TODO
  // (0019,1041) IS [2048\221184 ]  # 12,1-n Ver200 ADAC Header/Image Size
  if( bv->GetLength() != 2048 ) return 1;
  gdcm::Element<gdcm::VR::IS,gdcm::VM::VM2> el;
  const gdcm::PrivateTag tver200adacheaderimagesize(0x0019,0x41,"ADAC_IMG");
  if( !ds.FindDataElement( tver200adacheaderimagesize ) ) return 1;
  const gdcm::DataElement& ver200adacheaderimagesize = ds.GetDataElement( tver200adacheaderimagesize );
  el.SetFromDataElement( ver200adacheaderimagesize );
  if( el.GetValue(0) != 2048 ) return 1;
  std::istringstream is;
  std::string dup( bv->GetPointer(), bv->GetLength() );
  is.str( dup );
  bool b = DumpADAC( is );
  if( !b ) return 1;
  return 0;
}
```

12.49 DumpExamCard.cxx

```
/*=========================================================================

Program:  GDCM (Grassroots DICOM). A DICOM library

Copyright (c) 2006-2011 Mathieu Malaterre
All rights reserved.
See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.

This software is distributed WITHOUT ANY WARRANTY; without even
```

```
the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
PURPOSE.  See the above copyright notice for more information.

=========================================================================*/
/*

Try to extract contents of Philips RAW storage class:

(0002,0002) UI [1.2.840.10008.5.1.4.1.1.66]                       # 26,1 Media Storage SOP Class UID
(0002,0003) UI [1.3.46.670589.11.17240.5.23.4.1.3012.20100324094825680 18]         # 56,1 Media Storage SOP
      Instance UID
(0002,0010) UI [1.2.840.10008.1.2.1]                              # 20,1 Transfer Syntax UID
(0002,0012) UI [1.3.46.670589.11.0.0.51.4.4.1]                    # 30,1 Implementation Class UID
(0002,0013) SH [MR DICOM 4.1]                                     # 12,1 Implementation Version Name

* Everything done in this code is for the sole purpose of writing interoperable
* software under Sect.  1201 (f) Reverse Engineering exception of the DMCA.
* If you believe anything in this code violates any law or any of your rights,
* please contact us (gdcm-developers@lists.sourceforge.net) so that we can
* find a solution.
*
* Everything you do with this code is at your own risk, since decompression
* algorithm was not written from specification documents.
*
* Special thanks to:
* Triplett,William T for bringing to your attention on this ExamCard stuff
*/
#include "gdcmReader.h"
#include "gdcmDataSet.h"
#include "gdcmPrivateTag.h"
#include "gdcmBase64.h"
#include <iomanip>
static bool compfn(const char *s1, const char *s2)
{
  return strcmp(s1,s2) < 0 ?  true :  false;
}
static const char *PDFStrings[] = { // Keep me ordered please
  "COILSTATE", // series of string ?
  "HARDWARE_CONFIG", // series of number ?
  "PDF_CONTROL_GEN_PARS",
  "PDF_CONTROL_PREP_PARS",
  "PDF_CONTROL_RECON_PARS",
  "PDF_CONTROL_SCAN_PARS",
  "PDF_EXAM_PARS",
  "PDF_HARDWARE_PARS",
  "PDF_PREP_PARS",
  "PDF_PRESCAN_COIL_PARS",
  "PDF_SPT_PARS",
};
static bool isvalidpdfstring( const char *pdfstring )
{
  assert( pdfstring );
  static const size_t n = sizeof( PDFStrings ) / sizeof( *PDFStrings );
  static const char **begin = PDFStrings;
  static const char **end = begin + n;
  return std::binary_search(begin, end, pdfstring, compfn);
}
typedef enum
{
  param_float = 0,
  param_integer = 1, // 1 « 0
  param_string = 2, // 1 « 1
  param_3, // ??
  param_enum = 4  // 1 « 2
} param_type;
static const char *gettypenamefromtype( int i)
{
  const char *ret = nullptr;
  param_type e = (param_type)i;
  switch( e )
    {
  case param_float:
    ret = "float";
    break;
  case param_integer:
    ret = "int";
    break;
  case param_string:
    ret = "string";
    break;
  case param_3:
```

```
    ret = "??";
    break;
  case param_enum:
    ret = "enum";
    break;
    }
  assert( ret );
  return ret;
}
struct header
{
/*
* TODO:
* Looks as if we could read all int*, float* and string* at once...
*/
  int32_t v1; // offset to int pointer array ?
  uint16_t nints; // number of ints (max number?)
  uint16_t v3; // always 0 ?
  int32_t v4; // offset to float pointer array ?
  uint32_t nfloats;
  int32_t v6; // offset to string pointer array ?
  uint32_t nstrings;
  int32_t v8; // always 8 ??
  uint32_t numparams;
  uint32_t getnints()const { return nints; }
  uint32_t getnfloats()const { return nfloats; }
  uint32_t getnstrings()const { return nstrings; }
  uint32_t getnparams()const { return numparams; }
  void read( std::istream & is )
    {
    is.read( (char*)&v1,sizeof(v1));
    if( v1 == 0x01 ) {
      // direct (FIXME how should we detect this, much like TIFF ???)
      nints = 0;
      v3 = 0;
      v4 = 0;
      nfloats = 0;
      v6 = 0;
      nstrings = 0;
      v8 = 0;
      numparams = 0;
      uint32_t bla;
      is.read( (char*)&bla, sizeof(bla) );
      assert( bla == 0x2 || bla == 0x3 );
      nstrings = 1;
      numparams = 1;
    } else {
      // indirect
      is.read( (char*)&nints,sizeof(nints));
      is.read( (char*)&v3,sizeof(v3));
      assert( v3 == 0 ); // looks like this is always 0
      is.read( (char*)&v4,sizeof(v4));
      is.read( (char*)&nfloats,sizeof(nfloats));
      is.read( (char*)&v6,sizeof(v6));
      is.read( (char*)&nstrings,sizeof(nstrings));
      is.read( (char*)&v8,sizeof(v8));
      assert( v8 == 8 );
      is.read( (char*)&numparams,sizeof(numparams));
    }
    }
  void print( std::ostream & os )
    {
    os « v1 « ",";
    os « nints « ",";
    os « v3 « ",";
    os « v4 « ",";
    os « nfloats « ",";
    os « v6 « ",";
    os « nstrings « ",";
    os « v8 « ",";
    os « numparams « std::endl;
    }
};
struct param
{
  char name[32+1];
  uint8_t boolean;
  int32_t type;
  uint32_t dim;
  union {
  uint32_t val;
```

```
  char * ptr; } v4;
  int32_t /*std::streamoff*/ offset;
  param_type gettype()const { return (param_type)type; }
  uint32_t getdim()const { return dim; }
  void read_direct_int( std::istream & is ) {
    uint32_t bla;
    int max = 9;
    std::vector<uint32_t> v;
    for( int i = 0; i < max; ++i ) {
      is.read( (char*)&bla, sizeof(bla) );
      v.push_back( bla );
    }
    is.read( (char*)&bla, sizeof(bla) );
    char name0[32];
    memset(name0,0,sizeof(name0));
    assert( bla < sizeof(name0) );
    is.read( name0, bla);
    size_t l = strlen(name0);
    assert( l == bla ); (void)l;
    char * ptr = strdup( name0 );
    v4.ptr = ptr;
    type = param_string;
    dim = 1;
    offset = 0; // important !
  }
  void read_direct_string( std::istream & is ) {
    uint32_t bla;
    is.read( (char*)&bla, sizeof(bla) );
    char name0[32];
    memset(name0,0,sizeof(name0));
    assert( bla < sizeof(name0) );
    is.read( name0, bla);
    size_t l = strlen(name0);
    assert( l == bla ); (void)l;
    memcpy( this->name, name0, bla );
    is.read( (char*)&bla, sizeof(bla) );
    assert( bla == 0x1 );
    is.read( (char*)&bla, sizeof(bla) );
    char value[32];
    memset(value,0,sizeof(value));
    assert( bla < sizeof(value) );
    is.read( value, bla);
    is.read( (char*)&bla, sizeof(bla) );
    assert( bla == 0 ); // trailing stuff ?
    is.read( (char*)&bla, sizeof(bla) );
    assert( bla == 0 ); // trailing stuff ?
    const uint32_t cur = (uint32_t)is.tellg();
    std::cerr « "offset:" « cur « std::endl;
    if( cur == 65 )
      is.read( (char*)&bla, 1 );
    else if( cur == 66 )
      is.read( (char*)&bla, 1 );
    else if( cur == 122 )
      is.read( (char*)&bla, 2 );
    else
      assert(0);
    type = param_string;
    dim = 1;
    // FIXME: store the value in v4 for now:
    char * ptr = strdup( value );
    v4.ptr = ptr;
    offset = 0; // important !
  }
  void read( std::istream & is )
    {
    is.read( name, 32 + 1);
    // This is always the same issue the string can contains garbarge from previous run,
    // we need to print only until the first \0 character:
    assert( strlen( name ) <= 32 );
    is.read( (char*)&boolean,1);
    assert( boolean == 0 || boolean == 1 || boolean == 0x69 ); // some kind of bool, or digital trash ?
    is.read( (char*)&type, sizeof( type ) );
    assert( gettypenamefromtype( type ) );
    is.read( (char*)&dim, sizeof( dim ) ); // number of elements
    is.read( (char*)&v4.val, sizeof( v4.val ) );
    //assert( v4.val == 0 ); // always 0 ?  sometimes not...
    const uint32_t cur = (uint32_t)is.tellg();
    is.read( (char*)&offset, sizeof( offset ) );
    assert( offset != 0 );
    offset += cur;
    }
```

```
  void print( std::ostream & os )const
{
    os « name « ",";
    os « (int)boolean « ",";
    os « type « ",";
    os « dim « ",";
    os « v4.val « ",";
    os « offset « std::endl;
    }
  void printvalue( std::ostream & os, std::istream & is )const
{
    if( offset ) {
    is.seekg( offset );
    switch( type )
      {
    case param_float:
        {
        os.precision(2);
        os « std::fixed;
        for( uint32_t idx = 0; idx < dim; ++idx )
          {
          if( idx ) os « ",";
          float v;
          is.read( (char*)&v, sizeof(v) );
          os « v; // what if the string contains \0 ?
          }
        }
      break;
    case param_integer:
        {
        int32_t v;
        for( uint32_t idx = 0; idx < dim; ++idx )
          {
          if( idx ) os « ",";
          is.read( (char*)&v, sizeof(v) );
          os « v;
          }
        }
      break;
    case param_string:
        {
        int size = 81;
        std::string v;
        v.resize( size );
        for( uint32_t idx = 0; idx < dim; ++idx )
          {
          if( idx ) os « ";";
          is.read( &v[0], size );
          os « v.c_str();
          }
        }
      break;
    case param_enum:
        {
        int32_t v;
        for( uint32_t idx = 0; idx < dim; ++idx )
          {
          if( idx ) os « ",";
          is.read( (char*)&v, sizeof(v) );
          os « v;
          }
        }
      break;
      }
    } else {
#if 1
      // direct
      assert ( type == param_string );
      char * ptr = v4.ptr;
        //std::string v;
        //v.resize( dim );
        //is.read( &v[0], dim );
        os « ptr;
#endif
    }
    }
  void printxml( std::ostream & os, std::istream & is )const
{
    // <Attribute Name="CGEN_force_par_mode" Type="enum">0</Attribute>
    os « "  <Attribute";
    os « " Name=\"" « name « "\"";
```

```
    os « " Type=\"" « gettypenamefromtype(type) « "\"";
    if( dim != 1 )
      {
      os « " ArraySize=\"" « dim « "\"";
      }
    os « ">";
    printvalue( os, is );
    os « "</Attribute>\n";
    }
  void printcsv( std::ostream & os, std::istream & is )const
{
    os « std::setw(32) « std::left « name « ",";
    os « std::setw(7) « std::right « gettypenamefromtype(type) « ",";
    os « std::setw(4) « dim « ",";
    os « " ";
    printvalue( os, is );
    os « ",\n";
    }
};
static bool ProcessNested( gdcm::DataSet & ds )
{
  /*
TODO:
Looks like the real length of the blob is stored here:
(2005,1132) SQ                                                  # u/l,1 ?
(fffe,e000) na (Item with undefined length)
(2005,0011) LO [Philips MR Imaging DD 002 ]                   # 26,1 Private Creator
(2005,1143) SL 3103                                           # 4,1 ?

Wotsit ?
(2005,1132) SQ                                                  # u/l,1 ?
(fffe,e000) na (Item with undefined length)
(2005,0011) LO [Philips MR Imaging DD 002 ]                   # 26,1 Private Creator
(2005,1147) CS [Y ]                                           # 2,1 ?
*/
  bool ret = false;
  //  (2005,1137) PN (LO) [PDF_CONTROL_GEN_PARS]                    # 20,1 Protocol Data Name
  const gdcm::PrivateTag pt0(0x2005,0x37,"Philips MR Imaging DD 002");
  if( !ds.FindDataElement( pt0 ) ) return false;
  const gdcm::DataElement &de0 = ds.GetDataElement( pt0 );
  if( de0.IsEmpty() ) return false;
  const gdcm::ByteValue * bv0 = de0.GetByteValue();
  std::string s0( bv0->GetPointer() , bv0->GetLength() );
  // (2005,1139) LO [IEEE_PDF]                                    # 8,1 Protocol Data Type
  const gdcm::PrivateTag pt1(0x2005,0x39,"Philips MR Imaging DD 002");
  if( !ds.FindDataElement( pt1 ) ) return false;
  const gdcm::DataElement &de1 = ds.GetDataElement( pt1 );
  // (2005,1143) SL 53                                            # 4,1 Protocol Data Block Length (non-padded)
  const gdcm::PrivateTag pt2(0x2005,0x43,"Philips MR Imaging DD 002");
  if( !ds.FindDataElement( pt2 ) ) return false;
  const gdcm::DataElement &de2 = ds.GetDataElement( pt2 );
  // (2005,1147) CS [Y ]                                          # 2,1 Protocol Data Boolean
  const gdcm::PrivateTag pt3(0x2005,0x47,"Philips MR Imaging DD 002");
  if( !ds.FindDataElement( pt3 ) ) return false;
  const gdcm::DataElement &de3 = ds.GetDataElement( pt3 );
  (void)de3;
  // (2005,1144) OW 00\00\00\00\05\00\00\00\35\2e\31\2e\37\00         # 54,1 Protocol Data Block
  const gdcm::PrivateTag pt(0x2005,0x44,"Philips MR Imaging DD 002");
  if( !ds.FindDataElement( pt ) ) return false;
  const gdcm::DataElement &de = ds.GetDataElement( pt );
  if( de.IsEmpty() ) return false;
  const gdcm::ByteValue * bv = de.GetByteValue();
  if( s0 == "ExamCardBlob" )
    {
    assert( de1.IsEmpty() );
    std::string fn = gdcm::LOComp::Trim( s0.c_str() ); // remove trailing space
    fn += ".xml";
    std::ofstream out( fn.c_str() );
    // remove trailing \0
    size_t len = strlen( bv->GetPointer() );
    out.write( bv->GetPointer() , len );
    out.close();
    // Extract binary64 thingy (this is a ugly hack, better use an XML parser)
    std::string dup( bv->GetPointer(), len );
    std::string::size_type pos1 = dup.find( "<ExamCardBlob>" );
    std::string::size_type pos2 = dup.find( "</ExamCardBlob>" );
    std::string b64( bv->GetPointer() + pos1 + 14, pos2 - (pos1 + 14) );
    // ulgy hack to remove \r\n from input base64:
    std::string::iterator r_pos = std::remove(b64.begin(), b64.end(), '\r');
    b64.erase(r_pos, b64.end());
    std::string::iterator n_pos = std::remove(b64.begin(), b64.end(), '\n');
```

```
    b64.erase(n_pos, b64.end());
#if 0
    std::ofstream out2( "debug" );
    out2.write( b64.c_str(), b64.size() );
    out2.close();
#endif
    const size_t dlen = gdcm::Base64::GetDecodeLength(b64.c_str(), b64.size() );
    std::string decoded;
    decoded.resize( dlen );
    gdcm::Base64::Decode( &decoded[0], decoded.size(), b64.c_str(), b64.size() );
    std::ofstream f64( "soap.xml" );
    f64.write( decoded.c_str(), decoded.size() );
    f64.close();
    ret = true;
    }
  else
    {
    if( de1.IsEmpty() ) return false;
    const gdcm::ByteValue * bv1 = de1.GetByteValue();
    gdcm::Element<gdcm::VR::SL,gdcm::VM::VM1> dlen = {{0l}};
    dlen.SetFromDataElement( de2 );
    std::string s1( bv1->GetPointer() , bv1->GetLength() );
    if( s1 == "IEEE_PDF" )
      {
      std::istringstream is;
      assert( bv->GetLength() == (size_t)dlen.GetValue() || bv->GetLength() == (size_t)(dlen.GetValue() + 1) );
      std::string dup( bv->GetPointer(), dlen.GetValue() /*bv->GetLength()*/ );
      is.str( dup );
      header h;
      h.read( is );
      //assert( is.peek() && is.eof() );
#if 1
      static int c = 0;
      std::string fn0 = gdcm::LOComp::Trim( s1.c_str() ); // remove trailing space
      std::stringstream ss;
      ss << fn0 << "_" << c++;
      if( h.v1 == 0x01 )
        ss << ".direct";
      else
        ss << ".indirect";
      std::cout << "fn0=" << ss.str() << " Len= " << bv->GetLength() << std::endl;
      std::ofstream out( ss.str().c_str() );
      out.write( bv->GetPointer(), bv->GetLength() );
      out.close();
#endif
#if 1
      std::cout << dup.c_str() << std::endl;
      h.print( std::cout );
#endif
      std::vector< param > params;
      if( h.v1 == 0x01 ) {
        for( uint32_t i = 0; i < 1 /* h.getnparams()*/; ++i ) {
          param p;
          if( s0 == "HARDWARE_CONFIG " )
            {
            p.read_direct_int( is );
            }
          else if( s0 == "COILSTATE " )
            {
            p.read_direct_string( is );
            }
          else
            {
            assert(0);
            }
          params.push_back( p );
        }
      } else {
        assert( is.tellg() == std::streampos(0x20) );
        is.seekg( 0x20 );
        param p;
        for( uint32_t i = 0; i < h.getnparams(); ++i )
          {
          p.read( is );
          //p.print( std::cout );
          params.push_back( p );
          }
      }
      std::string fn = gdcm::LOComp::Trim( s0.c_str() ); // remove trailing space
      bool b1 = isvalidpdfstring( fn.c_str() );
      assert( b1 ); (void)b1;
```

```
      fn += ".csv";
      //fn += ".xml";
      std::ofstream csv( fn.c_str() );
      // let's do some bookeeping:
      uint32_t nfloats = 0;
      uint32_t nints = 0;
      uint32_t nstrings = 0;
      for( std::vector<param>::const_iterator it = params.begin();
        it != params.end(); ++it )
        {
        param_type type = it->gettype();
        switch( type )
          {
        case param_float:
          nfloats += it->getdim();
          break;
        case param_integer:
          nints += it->getdim();
          break;
        case param_string:
          nstrings += it->getdim();
          break;
        default:
          ;
          }
        }
#if 0
      std::cout « "Stats:" « std::endl;
      std::cout « "nfloats:" « nfloats « std::endl;
      std::cout « "nints:" « nints « std::endl;
      std::cout « "nstrings:" « nstrings « std::endl;
#endif
      assert( h.getnints() >= nints );
      assert( h.getnfloats() >= nfloats );
      assert( h.getnstrings() >= nstrings);
      for( uint32_t i = 0; i < h.getnparams(); ++i )
        {
        params[i].printcsv( csv, is );
        //params[i].printxml( csv, is );
        }
      csv.close();
      ret = true;
      }
    else if( s1 == "ASCII " )
      {
#if 0
      std::cerr « "ASCII is not handled" « std::endl;
      std::string fn = gdcm::LOComp::Trim( s0.c_str() ); // remove trailing space
      fn += ".asc";
      std::ofstream out( fn.c_str() );
      out.write( bv->GetPointer() , bv->GetLength() );
      out.close();
#endif
      std::string fn = gdcm::LOComp::Trim( s0.c_str() ); // remove trailing space
      fn += ".sin";
      std::ofstream sin( fn.c_str() );
      const char *beg = bv->GetPointer();
      const char *end = beg + bv->GetLength();
      assert( *beg == 0 );
      const char *p = beg + 1; // skip first \0
      size_t prev = 0;
      for( ; p != end; ++p )
        {
        if( *p == 0 )
          {
          const char *s = beg + prev + 1;
          if( *s )
            {
            sin « s « std::endl;
            }
          else
            {
            sin « std::endl;
            }
          prev = p - beg;
          }
        }
      sin.close();
      ret = true;
      }
    else if( s1 == "BINARY" )
```

```
      {
      std::cerr « "BINARY is not handled" « std::endl;
      std::string fn = gdcm::LOComp::Trim( s0.c_str() ); // remove trailing space
      fn += ".bin";
      std::ofstream out( fn.c_str() );
      //out.write( bv->GetPointer() + 512, bv->GetLength() - 512);
      out.write( bv->GetPointer() , bv->GetLength() );
      out.close();
#if 0
      int array[ 128 ];
      memcpy( array, bv->GetPointer(), 512 );
      for( int i = 0; i < 14; ++i )
        {
        std::cout « array[i] « std::endl;
        }
#endif
      ret = true;
      }
    }
  // else -> ret == false
  assert( ret );
  return ret;
}
int main(int argc, char *argv[])
{
  if( argc < 2 ) return 1;
  const char *filename = argv[1];
  gdcm::Reader reader;
  reader.SetFileName( filename );
  if( !reader.Read() )
    {
    std::cerr « "Failed to read:  " « filename « std::endl;
    return 1;
    }
  const gdcm::DataSet& ds = reader.GetFile().GetDataSet();
/*
(2005,1132) SQ                                                  # u/l,1 ?
(fffe,e000) na (Item with undefined length)
(2005,0011) LO [Philips MR Imaging DD 002 ]                   # 26,1 Private Creator
(2005,1137) PN (LO) [PDF_CONTROL_GEN_PARS]                    # 20,1 ?
(2005,1138) PN (LO) (no value)                                # 0,1 ?
(2005,1139) PN (LO) [IEEE_PDF]                                # 8,1 ?
(2005,1140) PN (LO) (no value)                                # 0,1 ?
(2005,1141) PN (LO) (no value)                                # 0,1 ?
(2005,1143) SL 3103                                           # 4,1 ?
(2005,1144) OW
      66\05\00\00\3b\01\00\00\4a\0a\00\00\0e\00\00\00\7a\0a\00\00\95\01\00\00\08\00\00\00\1b\00\00\00\43\47\45\4e\5f\75\73\65\72\5
      # 3104,1 ?
(2005,1147) CS [Y ]                                           # 2,1 ?
(fffe,e00d)
*/
  const gdcm::PrivateTag pt(0x2005,0x32,"Philips MR Imaging DD 002");
  if( !ds.FindDataElement( pt ) ) return 1;
  const gdcm::DataElement &de = ds.GetDataElement( pt );
  if( de.IsEmpty() ) return 1;
  gdcm::SequenceOfItems *sqi = de.GetValueAsSQ();
  if ( !sqi ) return 1;
  gdcm::SequenceOfItems::SizeType s = sqi->GetNumberOfItems();
  for( gdcm::SequenceOfItems::SizeType i = 1; i <= s; ++i )
    {
    gdcm::Item &item = sqi->GetItem(i);
    gdcm::DataSet &nestedds = item.GetNestedDataSet();
    if( !ProcessNested( nestedds ) ) {
      std::cerr « "Error processing Item #" « i « std::endl;
    }
    }
  return 0;
}
```

12.50 DumpGEMSMovieGroup.cxx

```
/*=========================================================================

Program:  GDCM (Grassroots DICOM). A DICOM library

Copyright (c) 2006-2011 Mathieu Malaterre
All rights reserved.
```

```
See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.

This software is distributed WITHOUT ANY WARRANTY; without even
the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
PURPOSE.  See the above copyright notice for more information.

=========================================================================*/
#include "gdcmReader.h"
#include "gdcmImage.h"
#include "gdcmImageWriter.h"
#include "gdcmDataElement.h"
#include "gdcmPrivateTag.h"
#include "gdcmUIDGenerator.h"
#include <iostream>
#include <string>
#include <map>
bool PrintNameValueMapping( gdcm::SequenceOfItems *sqi_values,
gdcm::SequenceOfItems *sqi_names, std::string const & indent )
{
  using namespace gdcm;
  // prepare names mapping:
  typedef VRToType<VR::UL>::Type UL;
  std::map< UL, std::string > names;
  assert( sqi_names );
  assert( sqi_values );
  SequenceOfItems::SizeType s = sqi_names->GetNumberOfItems();
  PrivateTag tindex(0x7fe1,0x71,"GEMS_Ultrasound_MovieGroup_001");
  PrivateTag tname (0x7fe1,0x72,"GEMS_Ultrasound_MovieGroup_001");
  // First sequence contains all possible names (this is a dict)
  for( SequenceOfItems::SizeType i = 1; i <= s; ++i )
    {
    const Item & item = sqi_names->GetItem( i );
    const DataSet & ds = item.GetNestedDataSet();
    if( !ds.FindDataElement( tindex )
      || !ds.FindDataElement( tname ) )
      {
      assert( 0 );
      return false;
      }
    const DataElement & index = ds.GetDataElement( tindex );
    const DataElement & name = ds.GetDataElement( tname );
    if( index.IsEmpty() || name.IsEmpty() )
      {
      assert( 0 );
      return false;
      }
    gdcm::Element<VR::UL, VM::VM1> el1;
    el1.SetFromDataElement( index );
    gdcm::Element<VR::LO, VM::VM1> el2;
    el2.SetFromDataElement( name );
//    std::cout « el1.GetValue() « " " « el2.GetValue() « std::endl;
    names.insert( std::make_pair( el1.GetValue(), el2.GetValue() ) );
    }
  SequenceOfItems::SizeType s2 = sqi_values->GetNumberOfItems();
  assert( s2 <= s );
  PrivateTag tindex2(0x7fe1,0x48,"GEMS_Ultrasound_MovieGroup_001");
  for( SequenceOfItems::SizeType i = 1; i <= s2; ++i )
    {
    const Item & item = sqi_values->GetItem( i );
    const DataSet & ds = item.GetNestedDataSet();
    if( !ds.FindDataElement( tindex2 ) )
      {
      assert( 0 );
      return false;
      }
    const DataElement & index2 = ds.GetDataElement( tindex2 );
    if( index2.IsEmpty() )
      {
      assert( 0 );
      return false;
      }
    gdcm::Element<VR::FD, VM::VM1_2> el1;
    el1.SetFromDataElement( index2 );
    UL copy = (UL)el1.GetValue();
#if 1
    std::cout « indent;
    std::cout « "( " « names[ copy ];
#endif
    // (7fe1,1052) FD 1560                                     # 8,1 ?
    // (7fe1,1057) LT [MscSkelSup]                             # 10,1 ?
    //PrivateTag tvalue(0x7fe1,0x52,"GEMS_Ultrasound_MovieGroup_001");
```

```
    PrivateTag tvalueint(0x7fe1,0x49,"GEMS_Ultrasound_MovieGroup_001"); // UL
    PrivateTag tvaluefloat1(0x7fe1,0x51,"GEMS_Ultrasound_MovieGroup_001"); // FL
    PrivateTag tvaluefloat(0x7fe1,0x52,"GEMS_Ultrasound_MovieGroup_001"); // FD
    PrivateTag tvalueul(0x7fe1,0x53,"GEMS_Ultrasound_MovieGroup_001"); // UL
    PrivateTag tvaluesl(0x7fe1,0x54,"GEMS_Ultrasound_MovieGroup_001"); // SL
    PrivateTag tvalueob(0x7fe1,0x55,"GEMS_Ultrasound_MovieGroup_001"); // OB
    PrivateTag tvaluetext(0x7fe1,0x57,"GEMS_Ultrasound_MovieGroup_001");  // LT
    PrivateTag tvaluefd(0x7fe1,0x77,"GEMS_Ultrasound_MovieGroup_001");  // FD / 1-N
    PrivateTag tvaluesl3(0x7fe1,0x79,"GEMS_Ultrasound_MovieGroup_001");  // SL / 1-N
    PrivateTag tvaluesl2(0x7fe1,0x86,"GEMS_Ultrasound_MovieGroup_001");  // SL ??
    PrivateTag tvaluefd1(0x7fe1,0x87,"GEMS_Ultrasound_MovieGroup_001");  // FD / 1-N
    PrivateTag tvaluefloat2(0x7fe1,0x88,"GEMS_Ultrasound_MovieGroup_001");  // FD ??
#if 1
    std::cout « " ) = ";
#endif
    if( ds.FindDataElement( tvalueint ) )
      {
      const DataElement & value = ds.GetDataElement( tvalueint );
      gdcm::Element<VR::UL,VM::VM1> el2;
      el2.SetFromDataElement( value );
      std::cout « el2.GetValue() « std::endl;
      }
    else if( ds.FindDataElement( tvaluefloat1 ) )
      {
      const DataElement & value = ds.GetDataElement( tvaluefloat1 );
      gdcm::Element<VR::FL,VM::VM1> el2;
      el2.SetFromDataElement( value );
      std::cout « el2.GetValue() « std::endl;
      }
    else if( ds.FindDataElement( tvaluefloat ) )
      {
      const DataElement & value = ds.GetDataElement( tvaluefloat );
      gdcm::Element<VR::FD,VM::VM1> el2;
      el2.SetFromDataElement( value );
      std::cout « el2.GetValue() « std::endl;
      }
    else if( ds.FindDataElement( tvaluesl ) )
      {
      const DataElement & value = ds.GetDataElement( tvaluesl );
      gdcm::Element<VR::SL,VM::VM1> el2;
      el2.SetFromDataElement( value );
      std::cout « el2.GetValue() « std::endl;
      }
    else if( ds.FindDataElement( tvalueul ) )
      {
      const DataElement & value = ds.GetDataElement( tvalueul );
      gdcm::Element<VR::UL,VM::VM1_n> el2;
      el2.SetFromDataElement( value );
      assert( el2.GetLength() == 1 );
      std::cout « el2.GetValue() « std::endl;
      }
    else if( ds.FindDataElement( tvalueob ) )
      {
      const DataElement & value = ds.GetDataElement( tvalueob );
//      gdcm::Element<VR::SL,VM::VM1> el2;
//      el2.SetFromDataElement( value );
//      std::cout « el2.GetValue() « std::endl;
      std::cout « value « std::endl;
      }
    else if( ds.FindDataElement( tvaluetext ) )
      {
      const DataElement & value = ds.GetDataElement( tvaluetext );
      gdcm::Element<VR::LT,VM::VM1> el2;
      el2.SetFromDataElement( value );
      std::cout « el2.GetValue() « std::endl;
      }
    else if( ds.FindDataElement( tvaluesl2 ) )
      {
      const DataElement & value = ds.GetDataElement( tvaluesl2 );
      gdcm::Element<VR::SL,VM::VM1_n> el2;
      el2.SetFromDataElement( value );
      el2.Print( std::cout );
      assert( el2.GetLength() == 4 );
      std::cout « std::endl;
      }
    else if( ds.FindDataElement( tvaluesl3 ) )
      {
      const DataElement & value = ds.GetDataElement( tvaluesl3 );
      gdcm::Element<VR::SL,VM::VM1_n> el2;
      el2.SetFromDataElement( value );
      el2.Print( std::cout );
```

```
//      assert( el2.GetLength() == 4 );
      std::cout « std::endl;
      }
    else if( ds.FindDataElement( tvaluefd ) )
      {
      const DataElement & value = ds.GetDataElement( tvaluefd );
      gdcm::Element<VR::FD,VM::VM1_n> el2;
      el2.SetFromDataElement( value );
      el2.Print( std::cout );
//      assert( el2.GetLength() == 4 || el2.GetLength() == 3 || el2.GetLength() == 8 );
      std::cout « std::endl;
      }
    else if( ds.FindDataElement( tvaluefloat2 ) )
      {
      const DataElement & value = ds.GetDataElement( tvaluefloat2 );
      gdcm::Element<VR::FD,VM::VM1_n> el2;
      el2.SetFromDataElement( value );
      el2.Print( std::cout );
      assert( el2.GetLength() == 2 );
      std::cout « std::endl;
      }
    else if( ds.FindDataElement( tvaluefd1 ) )
      {
      const DataElement & value = ds.GetDataElement( tvaluefd1 );
      gdcm::Element<VR::FD,VM::VM1_n> el2;
      el2.SetFromDataElement( value );
      el2.Print( std::cout );
      assert( el2.GetLength() == 4 );
      std::cout « std::endl;
      }
    else
      {
      std::cout « "(no value)" « std::endl;
//      std::cout « ds « std::endl;
      assert( ds.Size() == 2 );
      }
    }
  return true;
}
bool PrintNameValueMapping2( gdcm::PrivateTag const & privtag, const gdcm::DataSet & ds ,
  gdcm::SequenceOfItems *sqi_names, std::string const & indent )
{
  if( !ds.FindDataElement( privtag ) ) return false;
  const gdcm::DataElement& seq_values = ds.GetDataElement( privtag );
  gdcm::SmartPointer<gdcm::SequenceOfItems> sqi = seq_values.GetValueAsSQ();
  return PrintNameValueMapping( sqi, sqi_names, indent);
}
bool PrintNameValueMapping3( gdcm::PrivateTag const & privtag1, gdcm::PrivateTag const & privtag2, const
      gdcm::DataSet & ds ,
gdcm::SequenceOfItems *sqi_names, std::string const & indent )
{
  if( !ds.FindDataElement( privtag1 ) )
    {
    assert( 0 );
    return false;
    }
  const gdcm::DataElement& values10name = ds.GetDataElement( privtag1 );
  gdcm::Element<gdcm::VR::LO,gdcm::VM::VM1> el;
  el.SetFromDataElement( values10name );
  std::cout « std::endl;
  std::cout « " <" « el.GetValue().c_str() « ">" « std::endl;
  return PrintNameValueMapping2( privtag2, ds, sqi_names, indent);
}
bool print73( gdcm::DataSet const & ds10, gdcm::SequenceOfItems *sqi_dict, std::string const & indent )
{
  const gdcm::PrivateTag tseq_values73(0x7fe1,0x73,"GEMS_Ultrasound_MovieGroup_001");
  if( !ds10.FindDataElement( tseq_values73 ) )
    {
    std::cout « indent « "No group 73" « std::endl;
    return false;
    }
  const gdcm::DataElement& seq_values73 = ds10.GetDataElement( tseq_values73 );
  gdcm::SmartPointer<gdcm::SequenceOfItems> sqi_values73 = seq_values73.GetValueAsSQ();
  size_t ni3 = sqi_values73->GetNumberOfItems();
  for( size_t i3 = 1; i3 <= ni3; ++i3 )
    {
    gdcm::Item &item_73 = sqi_values73->GetItem(i3);
    gdcm::DataSet &ds73 = item_73.GetNestedDataSet();
    assert( ds73.Size() == 3 );
    const gdcm::PrivateTag tseq_values74name(0x7fe1,0x74,"GEMS_Ultrasound_MovieGroup_001");
    const gdcm::PrivateTag tseq_values75(0x7fe1,0x75,"GEMS_Ultrasound_MovieGroup_001");
```

```
    PrintNameValueMapping3( tseq_values74name, tseq_values75, ds73, sqi_dict, indent);
    std::cout « std::endl;
    }
  return true;
}
bool print36( gdcm::DataSet const & ds10, gdcm::SequenceOfItems *sqi_dict, std::string const & indent )
{
  (void)sqi_dict;
  const gdcm::PrivateTag tseq_values36(0x7fe1,0x36,"GEMS_Ultrasound_MovieGroup_001");
  if( !ds10.FindDataElement( tseq_values36 ) )
    {
    std::cout « indent « "No group 36" « std::endl;
    return false;
    }
  const gdcm::DataElement& seq_values36 = ds10.GetDataElement( tseq_values36 );
  gdcm::SmartPointer<gdcm::SequenceOfItems> sqi_values36 = seq_values36.GetValueAsSQ();
  size_t ni3 = sqi_values36->GetNumberOfItems();
  assert( ni3 >= 1 );
  for( size_t i3 = 1; i3 <= ni3; ++i3 )
    {
    gdcm::Item &item_36 = sqi_values36->GetItem(i3);
    gdcm::DataSet &ds36 = item_36.GetNestedDataSet();
    assert( ds36.Size() == 4 );
    // (7fe1,1037) UL 47  # 4,1 US MovieGroup Number of Frames
    // (7fe1,1043) OB 40\00\1c\c4\67\2f\0b\11\40          # 376,1 ?
    // (7fe1,1060) OB 4e\4e\49\4f\4e\47\46\43\2a          # 4562714,1 US MovieGroup Image Data
    //
    const gdcm::PrivateTag timagedata(0x7fe1,0x60,"GEMS_Ultrasound_MovieGroup_001");
    assert( ds36.FindDataElement( timagedata ) );
    gdcm::DataElement const & imagedata = ds36.GetDataElement( timagedata );
      const gdcm::ByteValue * bv = imagedata.GetByteValue();
  assert( bv );
      static int c = 0;
      std::stringstream ss;
      ss « "/tmp/debug";
      ss « c++;
      std::ofstream os( ss.str().c_str(), std::ios::binary );
      os.write( bv->GetPointer(), bv->GetLength() );
      os.close();
    //const gdcm::PrivateTag tseq_values85(0x7fe1,0x85,"GEMS_Ultrasound_MovieGroup_001");
    //PrintNameValueMapping3( tseq_values84name, tseq_values85, ds83, sqi_dict, indent);
    //std::cout « std::endl;
    }
  return true;
}
bool print83( gdcm::DataSet const & ds10, gdcm::SequenceOfItems *sqi_dict, std::string const & indent )
{
  const gdcm::PrivateTag tseq_values83(0x7fe1,0x83,"GEMS_Ultrasound_MovieGroup_001");
  if( !ds10.FindDataElement( tseq_values83 ) )
    {
    std::cout « indent « "No group 83" « std::endl;
    return false;
    }
  const gdcm::DataElement& seq_values83 = ds10.GetDataElement( tseq_values83 );
  gdcm::SmartPointer<gdcm::SequenceOfItems> sqi_values83 = seq_values83.GetValueAsSQ();
  size_t ni3 = sqi_values83->GetNumberOfItems();
  for( size_t i3 = 1; i3 <= ni3; ++i3 )
    {
    gdcm::Item &item_83 = sqi_values83->GetItem(i3);
    gdcm::DataSet &ds83 = item_83.GetNestedDataSet();
    assert( ds83.Size() == 3 );
    const gdcm::PrivateTag tseq_values84name(0x7fe1,0x84,"GEMS_Ultrasound_MovieGroup_001");
    const gdcm::PrivateTag tseq_values85(0x7fe1,0x85,"GEMS_Ultrasound_MovieGroup_001");
    PrintNameValueMapping3( tseq_values84name, tseq_values85, ds83, sqi_dict, indent);
    std::cout « std::endl;
    }
  return true;
}
bool PrintNameValueMapping4( gdcm::PrivateTag const & privtag0, const gdcm::DataSet & subds, gdcm::PrivateTag
      const & privtag1, gdcm::PrivateTag const & privtag2,
gdcm::SequenceOfItems *sqi_dict, std::string const & indent )
{
  (void)indent;
  if( !subds.FindDataElement( privtag0 ) )
    {
    assert( 0 );
    return false;
    }
  const gdcm::DataElement& seq_values10 = subds.GetDataElement( privtag0 );
  gdcm::SmartPointer<gdcm::SequenceOfItems> sqi_values10 = seq_values10.GetValueAsSQ();
  size_t ni1 = sqi_values10->GetNumberOfItems();
```

```
//  assert( ni1 == 1 );
  for( size_t i1 = 1; i1 <= ni1; ++i1 )
    {
    gdcm::Item &item_10 = sqi_values10->GetItem(i1);
    gdcm::DataSet &ds10 = item_10.GetNestedDataSet();
    assert( ds10.Size() == 2 + 3 );
    // (7fe1,0010)
    // (7fe1,1012)
    // (7fe1,1018)
    // (7fe1,1020)
    // (7fe1,1083)
    PrintNameValueMapping3( privtag1, privtag2, ds10, sqi_dict, "  " );
    std::cout « std::endl;
    const gdcm::PrivateTag tseq_values20(0x7fe1,0x20,"GEMS_Ultrasound_MovieGroup_001");
    if( !ds10.FindDataElement( tseq_values20 ) )
      {
      assert( 0 );
      return false;
      }
    const gdcm::DataElement& seq_values20 = ds10.GetDataElement( tseq_values20 );
    gdcm::SmartPointer<gdcm::SequenceOfItems> sqi_values20 = seq_values20.GetValueAsSQ();
    size_t ni2 = sqi_values20->GetNumberOfItems();
    //assert( ni == 1 );
    for( size_t i2 = 1; i2 <= ni2; ++i2 )
      {
      gdcm::Item &item_20 = sqi_values20->GetItem(i2);
      gdcm::DataSet &ds20 = item_20.GetNestedDataSet();
      size_t count = ds20.Size(); (void)count;
      assert( ds20.Size() == 2 + 3 || ds20.Size() == 2 + 2 );
      // (7fe1,0010)
      // (7fe1,1024)
      // (7fe1,1026)
      // (7fe1,1036)
      // (7fe1,103a)
      // (7fe1,1083) (*)
      const gdcm::PrivateTag tseq_values20name(0x7fe1,0x24,"GEMS_Ultrasound_MovieGroup_001");
      const gdcm::PrivateTag tseq_values26(0x7fe1,0x26,"GEMS_Ultrasound_MovieGroup_001");
      PrintNameValueMapping3( tseq_values20name, tseq_values26, ds20, sqi_dict, "   ");
      std::cout « std::endl;
      print36(ds20, sqi_dict, "    ");
      print83(ds20, sqi_dict, "    ");
      }
    print83(ds10, sqi_dict, "   ");
    }
  return true;
}
int main(int argc, char *argv[])
{
  if( argc < 2 ) return 1;
  using namespace gdcm;
  const char *filename = argv[1];
  gdcm::Reader reader;
  reader.SetFileName( filename );
  if( !reader.Read() ) return 1;
  gdcm::File &file = reader.GetFile();
  gdcm::DataSet &ds = file.GetDataSet();
  const PrivateTag tseq(0x7fe1,0x1,"GEMS_Ultrasound_MovieGroup_001");
  if( !ds.FindDataElement( tseq ) ) return 1;
  const DataElement& seq = ds.GetDataElement( tseq );
  SmartPointer<SequenceOfItems> sqi = seq.GetValueAsSQ();
  assert( sqi->GetNumberOfItems() == 1 );
  Item &item = sqi->GetItem(1);
  DataSet &subds = item.GetNestedDataSet();
  const PrivateTag tseq_dict(0x7fe1,0x70,"GEMS_Ultrasound_MovieGroup_001");
  if( !subds.FindDataElement( tseq_dict ) ) return 1;
  const DataElement& seq_dict = subds.GetDataElement( tseq_dict );
  SmartPointer<SequenceOfItems> sqi_dict = seq_dict.GetValueAsSQ();
  const PrivateTag tseq_values8(0x7fe1,0x8,"GEMS_Ultrasound_MovieGroup_001");
  if( !subds.FindDataElement( tseq_values8 ) ) return 1;
  const DataElement& seq_values8 = subds.GetDataElement( tseq_values8 );
  SmartPointer<SequenceOfItems> sqi_values8 = seq_values8.GetValueAsSQ();
  const PrivateTag tseq_values8name(0x7fe1,0x2,"GEMS_Ultrasound_MovieGroup_001");
  if( !subds.FindDataElement( tseq_values8name ) ) return 1;
  const DataElement& values8name = subds.GetDataElement( tseq_values8name );
{
  Element<VR::LO,VM::VM1> el;
  el.SetFromDataElement( values8name );
  std::cout « el.GetValue() « std::endl;
}
  size_t count = subds.Size(); (void)count;
  assert( subds.Size() == 3 + 2 + 1 || subds.Size() == 3 + 2 + 2);
```

```
//  (7fe1,0010) # 30,1 Private Creator
//  (7fe1,1002) # 8,1 US MovieGroup Value 0008 Name
//  (7fe1,1003) # 4,1 ?
//  (7fe1,1008) # 8140,1 US MovieGroup Value 0008 Sequence
//  (7fe1,1010) # 1372196,1 ?
//  (7fe1,1070) # 33684,1 US MovieGroup Dict
//  (7fe1,1073) (*)
  PrintNameValueMapping( sqi_values8, sqi_dict, " ");
  const PrivateTag tseq_values10(0x7fe1,0x10,"GEMS_Ultrasound_MovieGroup_001");
  const PrivateTag tseq_values10name(0x7fe1,0x12,"GEMS_Ultrasound_MovieGroup_001");
  const PrivateTag tseq_values18(0x7fe1,0x18,"GEMS_Ultrasound_MovieGroup_001");
  PrintNameValueMapping4( tseq_values10, subds, tseq_values10name, tseq_values18, sqi_dict,"  ");
  print73( subds, sqi_dict, "  " );
#if 0
  gdcm::DataSet::ConstIterator it = subds.Begin();
  for( ; it != subds.End(); ++it )
    {
  const gdcm::DataElement &de = *it;
    std::cout « de.GetTag() « std::endl;
    }
#endif
  return 0;
}
```

12.51 DumpImageHeaderInfo.cxx

```
/*=========================================================================

Program:  GDCM (Grassroots DICOM). A DICOM library

Copyright (c) 2006-2011 Mathieu Malaterre
All rights reserved.
See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.

This software is distributed WITHOUT ANY WARRANTY; without even
the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
PURPOSE.  See the above copyright notice for more information.

=========================================================================*/
/*
* Dump TOSHIBA MDW HEADER / Image Header Info
*/
#include "gdcmReader.h"
#include "gdcmPrivateTag.h"
#include "gdcmAttribute.h"
#include "gdcmImageWriter.h"
#include <iostream>
#include <fstream>
#include <vector>
#include <string.h>
#include <assert.h>
#include <stdint.h>
struct element
{
  std::istream & read( std::istream & is );
};
std::istream & element::read( std::istream & is )
{
  static const uint32_t ref = 0xe000fffe;
  std::ostream &os = std::cout;
  if( is.eof() )
    {
    return is;
    }
  uint32_t magic;
  if( !is.read( (char*)&magic, sizeof(magic) ) )
    {
    return is;
    }
  //os « magic « std::endl;
  assert( magic == ref ); (void)ref;
  uint32_t l;
  is.read( (char*)&l, sizeof(l) );
  //os « l « std::endl;
  char str[17];
  str[16] = 0;
  is.read( str, 16 );
```

```
  os « str « " (" « l « ")" « std::endl;
  std::vector<char> bytes;
  bytes.resize( l - 16 );
  if( !bytes.empty() )
    {
    is.read( &bytes[0], l - 16 );
    }
  //os « "pos:" « is.tellg() « std::endl;
  if( strcmp(str, "TUSREMEASUREMENT" ) == 0 )
    {
    const char *p = &bytes[0];
    uint32_t val;
    memcpy( (char*)&val, p, sizeof(val) );
    os « " " « val « std::endl;
    p += sizeof(val);
    memcpy( (char*)&val, p, sizeof(val) );
    os « " " « val « std::endl;
    p += sizeof(val);
    memcpy( (char*)&val, p, sizeof(val) );
    os « " " « val « std::endl;
    p += sizeof(val);
    memcpy( (char*)&val, p, sizeof(val) );
    os « " " « val « std::endl;
    p += sizeof(val);
    memcpy( (char*)&val, p, sizeof(val) );
    os « " " « val « std::endl;
    p += sizeof(val);
    memcpy( (char*)&val, p, sizeof(val) );
    os « " " « val « std::endl;
    p += sizeof(val);
#if 0
    float f;
    memcpy( (char*)&f, p, sizeof(f) );
    os « " " « f « std::endl;
    p += sizeof(f);
#else
    memcpy( (char*)&val, p, sizeof(val) );
    os « " " « val « std::endl;
    p += sizeof(val);
#endif
    memcpy( (char*)&val, p, sizeof(val) );
    os « " " « val « std::endl;
    p += sizeof(val);
    char str2[17];
    memcpy( str2, p, 16 );
    str2[16] = 0;
    os « " " « str2 « std::endl;
    }
#if 0
  std::ofstream out( str, std::ios::binary );
  out.write( (char*)&magic, sizeof( magic ) );
  out.write( (char*)&l, sizeof( l ) );
  out.write( str, 16 );
  out.write( &bytes[0], bytes.size() );
#endif
  return is;
}
static bool DumpImageHeaderInfo( std::istream & is, size_t reflen )
{
  // TUSNONIMAGESTAM (5176)
  // TUSREMEASUREMEN (1352)
  // TUSBSINGLELAYOU (16)
  // TUSCLIPPARAMETE (104)
  element el;
  while( el.read( is ) )
    {
    }
  //size_t pos = is.tellg();
  //assert( pos == reflen );
  (void)reflen;
  return true;
}
int main(int argc, char *argv[])
{
  if( argc < 2 ) return 1;
  const char *filename = argv[1];
  gdcm::Reader reader;
  reader.SetFileName( filename );
  if( !reader.Read() )
    {
    std::cerr « "Failed to read:  " « filename « std::endl;
```

```
    return 1;
    }
  const gdcm::DataSet& ds = reader.GetFile().GetDataSet();
  const gdcm::PrivateTag timageheaderinfo(0x0029,0x10,"TOSHIBA MDW HEADER");
  if( !ds.FindDataElement( timageheaderinfo) ) return 1;
  const gdcm::DataElement& imageheaderinfo = ds.GetDataElement( timageheaderinfo );
  if ( imageheaderinfo.IsEmpty() ) return 1;
  const gdcm::ByteValue * bv = imageheaderinfo.GetByteValue();
  std::istringstream is;
  std::string dup( bv->GetPointer(), bv->GetLength() );
  is.str( dup );
  bool b = DumpImageHeaderInfo( is, bv->GetLength() );
  if( !b ) return 1;
#if 0
  const float d1 = 0.0041666668839752674; // 89 88 88 3B // 0x44c
  //const float d1 = 0.053231674455417881;
  const float d2 = 0.10828025639057159; // 0A C2 DD 3D // 0x1ac
  //const float d1 = 0.17869562069272813;
  //const unsigned int d2 = 4294967280;
  const float d3 = 0.10828025639057159; // 0A C2 DD 3D // 0x15c
  const int32_t d4 = 134;
  const uint32_t d5 = 1153476;
  std::ofstream t("/tmp/debug", std::ios::binary );
  //t.write( (char*)&d0, sizeof( d0 ) );
  t.write( (char*)&d1, sizeof( d1 ) );
  t.write( (char*)&d2, sizeof( d2 ) );
  t.write( (char*)&d3, sizeof( d3 ) );
  t.write( (char*)&d4, sizeof( d4 ) );
  t.write( (char*)&d5, sizeof( d5 ) );
  t.close();
#endif
  return 0;
}
```

12.52 DumpPhilipsECHO.cxx

```
/*=========================================================================

Program:  GDCM (Grassroots DICOM). A DICOM library

Copyright (c) 2006-2011 Mathieu Malaterre
All rights reserved.
See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.

This software is distributed WITHOUT ANY WARRANTY; without even
the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
PURPOSE.  See the above copyright notice for more information.

=========================================================================*/
#include "gdcmReader.h"
#include "gdcmDeflateStream.h"
#include "gdcm_zlib.h"
/*
* This example extract the ZLIB compressed US image from a Philips private tag
*
* Everything done in this code is for the sole purpose of writing interoperable
* software under Sect.  1201 (f) Reverse Engineering exception of the DMCA.
* If you believe anything in this code violates any law or any of your rights,
* please contact us (gdcm-developers@lists.sourceforge.net) so that we can
* find a solution.
*
* Everything you do with this code is at your own risk, since decompression
* algorithm was not written from specification documents.
*
* Usage:
*
* $ DumpPhilipsECHO private_us.dcm raw_us_img.raw
* $ gdcmimg --sop-class-uid 1.2.840.10008.5.1.4.1.1.3.1 --size 608,427,88 raw_us_img.raw raw_us_img.dcm
*/
// header:
struct hframe
{
  uint32_t val0; // 800 increment ?
  uint16_t val1[2];
  uint16_t val2[2];
  uint32_t imgsize;
  bool operator==(const hframe &h)const
```

```
{
    return val0 == h.val0 &&
      val1[0] == h.val1[0] &&
      val1[1] == h.val1[1] &&
      val2[0] == h.val2[0] &&
      val2[1] == h.val2[1] &&
      imgsize == h.imgsize;
    }
};
static bool ProcessDeflate( const char *outfilename, const int nslices, const
  int buf_size, const char *buf, const std::streampos len,
  const char *crcbuf, const size_t crclen )
{
  std::vector< hframe > crcheaders;
  crcheaders.reserve( nslices );
    {
    std::istringstream is;
    is.str( std::string( crcbuf, crclen ) );
    hframe header;
    for( int r = 0; r < nslices; ++r )
      {
      is.read( (char*)&header, sizeof( header ));
#if 0
      std::cout « header.val0
         « " " « header.val1[0]
         « " " « header.val1[1]
         « " " « header.val2[0]
         « " " « header.val2[1]
         « " " « header.imgsize « std::endl;
#endif
      crcheaders.push_back( header );
      }
    }
  std::istringstream is;
  is.str( std::string( buf, (size_t)len ) );
  std::streamoff totalsize;
  is.read( (char*)&totalsize, sizeof( totalsize ));
  assert( totalsize == len );
  uint32_t nframes;
  is.read( (char*)&nframes, sizeof( nframes ));
  assert( nframes == (uint32_t)nslices );
  std::vector< std::streamoff > offsets;
  offsets.reserve( nframes );
  for( uint32_t frame = 0; frame < nframes ; ++frame )
    {
    uint32_t offset;
    is.read( (char*)&offset, sizeof( offset ));
    offsets.push_back( offset );
    }
  std::vector<char> outbuf;
  const int size[2] = { 608, 427 }; // FIXME: where does it comes from ?
  std::stringstream ss;
  ss « outfilename;
  ss « '_';
  //ss « crcheaders[0].imgsize; // FIXME: Assume all header are identical !
  ss « size[0];
  ss « '_';
  ss « size[1];
  ss « '_';
  ss « nframes;
  ss « ".raw";
  std::ofstream os( ss.str().c_str(), std::ios::binary );
  assert( buf_size >= size[0] * size[1] );
  outbuf.resize( buf_size );
  hframe header;
  //uint32_t prev = 0;
  for( unsigned int r = 0; r < nframes; ++r )
    {
    is.read( (char*)&header, sizeof( header ));
    assert( header == crcheaders[r] );
    assert( header.val1[0] == 2000 );
    assert( header.val1[1] == 3 );
    assert( header.val2[0] == 1 );
    assert( header.val2[1] == 1280 );
    uLongf destLen = buf_size; // >= 608,427
    Bytef *dest = (Bytef*)&outbuf[0];
    assert( is.tellg() == offsets[r] + 16 );
    const Bytef *source = (const Bytef*)buf + offsets[r] + 16;
    uLong sourceLen;
    if( r + 1 == nframes )
      sourceLen = (uLong)totalsize - (uLong)offsets[r] - 16;
```

```
    else
      sourceLen = (uLong)offsets[r+1] - (uLong)offsets[r] - 16;
    // FIXME: in-memory decompression:
    int ret = uncompress (dest, &destLen, source, sourceLen);
    assert( ret == Z_OK ); (void)ret;
    assert( destLen >= (uLongf)size[0] * size[1] ); // 16bytes padding ?
    assert( header.imgsize == (uint32_t)size[0] * size[1] );
    //os.write( &outbuf[0], outbuf.size() );
    os.write( &outbuf[0], size[0] * size[1] );
    // skip data:
    is.seekg( sourceLen, std::ios::cur );
    }
  os.close();
  assert( is.tellg() == totalsize );
  return true;
}
static bool ProcessNone( const char *outfilename, const int nslices, const
  int buf_size, const char *buf, const std::streampos len,
  const char *crcbuf, const size_t crclen )
{
  std::vector< hframe > crcheaders;
  crcheaders.reserve( nslices );
    {
    std::istringstream is;
    is.str( std::string( crcbuf, crclen ) );
    hframe header;
    for( int r = 0; r < nslices; ++r )
      {
      is.read( (char*)&header, sizeof( header ));
#if 0
      std::cout « header.val0
         « " " « header.val1[0]
         « " " « header.val1[1]
         « " " « header.val2[0]
         « " " « header.val2[1]
         « " " « header.imgsize « std::endl;
#endif
      crcheaders.push_back( header );
      }
    }
  std::istringstream is;
  is.str( std::string( buf, (size_t)len ) );
  std::streampos totalsize;
  is.read( (char*)&totalsize, sizeof( totalsize ));
  assert( totalsize == len );
  uint32_t nframes;
  is.read( (char*)&nframes, sizeof( nframes ));
  assert( nframes == (uint32_t)nslices );
  std::vector< uint32_t > offsets;
  offsets.reserve( nframes );
  for( uint32_t frame = 0; frame < nframes ; ++frame )
    {
    uint32_t offset;
    is.read( (char*)&offset, sizeof( offset ));
    offsets.push_back( offset );
    //std::cout « offset « std::endl;
    }
  std::vector<char> outbuf;
  // No idea how to present the data, I'll just append everything, and present it as 2D
  std::stringstream ss;
  ss « outfilename;
  ss « '_';
  ss « crcheaders[0].imgsize; // FIXME: Assume all header are identical !
  ss « '_';
  ss « nframes;
  ss « ".raw";
  std::ofstream os( ss.str().c_str(), std::ios::binary );
  outbuf.resize( buf_size ); // overallocated + 16
  char *buffer = &outbuf[0];
  hframe header;
  for( unsigned int r = 0; r < nframes; ++r )
    {
    is.read( (char*)&header, sizeof( header ));
#if 0
      std::cout « header.val0
         « " " « header.val1[0]
         « " " « header.val1[1]
         « " " « header.val2[0]
         « " " « header.val2[1]
         « " " « header.imgsize « std::endl;
#endif
```

```
    assert( header == crcheaders[r] );
    is.read( buffer, buf_size - 16 );
    os.write( buffer, header.imgsize );
    }
  assert( is.tellg() == totalsize );
  os.close();
  return true;
}
#ifndef NDEBUG
static const char * const UDM_USD_DATATYPE_STRINGS[] = {
  "UDM_USD_DATATYPE_DIN_2D_ECHO",
  "UDM_USD_DATATYPE_DIN_2D_ECHO_CONTRAST",
  "UDM_USD_DATATYPE_DIN_DOPPLER_CW",
  "UDM_USD_DATATYPE_DIN_DOPPLER_PW",
  "UDM_USD_DATATYPE_DIN_DOPPLER_PW_TDI",
  "UDM_USD_DATATYPE_DIN_2D_COLOR_FLOW",
  "UDM_USD_DATATYPE_DIN_2D_COLOR_PMI",
  "UDM_USD_DATATYPE_DIN_2D_COLOR_CPA",
  "UDM_USD_DATATYPE_DIN_2D_COLOR_TDI",
  "UDM_USD_DATATYPE_DIN_MMODE_ECHO",
  "UDM_USD_DATATYPE_DIN_MMODE_COLOR",
  "UDM_USD_DATATYPE_DIN_MMODE_COLOR_TDI",
  "UDM_USD_DATATYPE_DIN_PARAM_BLOCK",
  "UDM_USD_DATATYPE_DIN_2D_COLOR_VELOCITY",
  "UDM_USD_DATATYPE_DIN_2D_COLOR_POWER",
  "UDM_USD_DATATYPE_DIN_2D_COLOR_VARIANCE",
  "UDM_USD_DATATYPE_DIN_DOPPLER_AUDIO",
  "UDM_USD_DATATYPE_DIN_DOPPLER_HIGHQ",
  "UDM_USD_DATATYPE_DIN_PHYSIO",
  "UDM_USD_DATATYPE_DIN_2D_COLOR_STRAIN",
  "UDM_USD_DATATYPE_DIN_COMPOSITE_RGB",
  "UDM_USD_DATATYPE_DIN_XFOV_REALTIME_GRAPHICS",
  "UDM_USD_DATATYPE_DIN_XFOV_MOSAIC",
  "UDM_USD_DATATYPE_DIN_COMPOSITE_R",
  "UDM_USD_DATATYPE_DIN_COMPOSITE_G",
  "UDM_USD_DATATYPE_DIN_COMPOSITE_B",
  "UDM_USD_DATATYPE_DIN_MMODE_COLOR_VELOCITY",
  "UDM_USD_DATATYPE_DIN_MMODE_COLOR_POWER",
  "UDM_USD_DATATYPE_DIN_MMODE_COLOR_VARIANCE",
  "UDM_USD_DATATYPE_DIN_2D_ELASTO",
};
static inline bool is_valid( const char * datatype_str )
{
  static const int n = sizeof( UDM_USD_DATATYPE_STRINGS ) / sizeof( *UDM_USD_DATATYPE_STRINGS );
  bool found = false;
  if( datatype_str )
    {
    for( int i = 0; !found && i < n; ++i )
      {
      found = strcmp( datatype_str, UDM_USD_DATATYPE_STRINGS[i] ) == 0;
      }
    }
  return found;
}
#endif
int main(int argc, char *argv[])
{
  if( argc < 2 ) return 1;
  using namespace gdcm;
  const char *filename = argv[1];
  gdcm::Reader reader;
  reader.SetFileName( filename );
  if( !reader.Read() ) return 1;
  gdcm::File &file = reader.GetFile();
  gdcm::DataSet &ds1 = file.GetDataSet();
  const PrivateTag tseq1(0x200d,0x3cf8,"Philips US Imaging DD 045");
  if( !ds1.FindDataElement( tseq1 ) ) return 1;
  const DataElement& seq1 = ds1.GetDataElement( tseq1 );
  SmartPointer<SequenceOfItems> sqi1 = seq1.GetValueAsSQ();
  assert( sqi1->GetNumberOfItems() >= 1 );
  const size_t nitems = sqi1->GetNumberOfItems();
  for( size_t item = 1; item < nitems; ++item )
    {
    Item &item1 = sqi1->GetItem(item);
    DataSet &ds2 = item1.GetNestedDataSet();
    // (200d,300d)  LO  28  UDM_USD_DATATYPE_DIN_2D_ECHO
    const PrivateTag tdatatype(0x200d,0x300d,"Philips US Imaging DD 033");
    if( !ds2.FindDataElement( tdatatype ) ) return 1;
    const DataElement& datatype = ds2.GetDataElement( tdatatype );
    const ByteValue *bvdatatype = datatype.GetByteValue();
    if( !bvdatatype ) return 1;
```

```
const PrivateTag tseq2(0x200d,0x3cf1,"Philips US Imaging DD 045");
if( !ds2.FindDataElement( tseq2 ) ) return 1;
const DataElement& seq2 = ds2.GetDataElement( tseq2 );
SmartPointer<SequenceOfItems> sqi2 = seq2.GetValueAsSQ();
assert( sqi2->GetNumberOfItems() >= 1 );
// FIXME: what if not in first Item ?
assert( sqi2->GetNumberOfItems() == 1 );
Item &item2 = sqi2->GetItem(1);
DataSet &ds3 = item2.GetNestedDataSet();
const PrivateTag tzlib(0x200d,0x3cfa,"Philips US Imaging DD 045");
if( !ds3.FindDataElement( tzlib ) ) return 1;
const DataElement& zlib = ds3.GetDataElement( tzlib );
const ByteValue *bv = zlib.GetByteValue();
if( !bv ) return 1;
if( bv->GetLength() != 4 ) return 1;
// (200d,3010)  IS  2  88
const PrivateTag tnslices(0x200d,0x3010,"Philips US Imaging DD 033");
if( !ds3.FindDataElement( tnslices ) ) return 1;
const DataElement& nslices = ds3.GetDataElement( tnslices );
Element<VR::IS,VM::VM1> elnslices;
elnslices.SetFromDataElement( nslices );
const int nslicesref = elnslices.GetValue();
assert( nslicesref >= 0 );
// (200d,3011)  IS  6  259648
const PrivateTag tzalloc(0x200d,0x3011,"Philips US Imaging DD 033");
if( !ds3.FindDataElement( tzalloc ) ) return 1;
const DataElement& zalloc = ds3.GetDataElement( tzalloc );
Element<VR::IS,VM::VM1> elzalloc;
elzalloc.SetFromDataElement( zalloc );
const int zallocref = elzalloc.GetValue();
assert( zallocref >= 0 );
// (200d,3021)  IS  2  0
const PrivateTag tzero(0x200d,0x3021,"Philips US Imaging DD 033");
if( !ds3.FindDataElement( tzero ) ) return 1;
const DataElement& zero = ds3.GetDataElement( tzero );
Element<VR::IS,VM::VM1> elzero;
elzero.SetFromDataElement( zero );
const int zerocref = elzero.GetValue();
assert( zerocref == 0 ); (void)zerocref;
// (200d,3cf3) OB
const PrivateTag tdeflate(0x200d,0x3cf3,"Philips US Imaging DD 045");
if( !ds3.FindDataElement( tdeflate) ) return 1;
const DataElement& deflate = ds3.GetDataElement( tdeflate );
const ByteValue *bv2 = deflate.GetByteValue();
// (200d,3cfb) OB
const PrivateTag tcrc(0x200d,0x3cfb,"Philips US Imaging DD 045");
if( !ds3.FindDataElement( tcrc ) ) return 1;
const DataElement& crc = ds3.GetDataElement( tcrc );
const ByteValue *bv3 = crc.GetByteValue();
std::string outfile = std::string( bvdatatype->GetPointer(), bvdatatype->GetLength() );
outfile = LOComp::Trim( outfile.c_str() );
const char *outfilename = outfile.c_str();
assert( is_valid(outfilename) );
if( bv2 )
  {
  assert( bv3 );
  assert( zallocref > 0 );
  assert( nslicesref > 0 );
  std::cout « ds2 « std::endl;
  if( strncmp(bv->GetPointer(), "ZLib", 4) == 0 )
    {
    if( !ProcessDeflate( outfilename, nslicesref, zallocref, bv2->GetPointer(),
        std::streampos(bv2->GetLength()), bv3->GetPointer(), bv3->GetLength() ) )
      {
      return 1;
      }
    }
  else if( strncmp(bv->GetPointer(), "None", 4) == 0 )
    {
    if( !ProcessNone( outfilename, nslicesref, zallocref, bv2->GetPointer(),
        std::streampos(bv2->GetLength()), bv3->GetPointer(), bv3->GetLength() ) )
      {
      return 1;
      }
    }
  else
    {
    std::string str( bv->GetPointer(), bv->GetLength() );
    std::cerr « "Unhandled:  " « str « std::endl;
    return 1;
    }
```

```
      }
    }
  return 0;
}
```

12.53 DumpSiemensBase64.cxx

```
/*=========================================================================

Program:  GDCM (Grassroots DICOM). A DICOM library

Copyright (c) 2006-2011 Mathieu Malaterre
All rights reserved.
See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.

This software is distributed WITHOUT ANY WARRANTY; without even
the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
PURPOSE.  See the above copyright notice for more information.

=========================================================================*/
/*
* https://groups.google.com/forum/#!msg/comp.protocols.dicom/2kZ2lLP8EcM/WzjFrtjnAgAJ
*/
#include "gdcmReader.h"
#include "gdcmPrivateTag.h"
#include "gdcmPrinter.h"
#include "gdcmDictPrinter.h"
#include "gdcmCSAHeader.h"
#include "gdcmBase64.h"
#include "gdcmExplicitDataElement.h"
#include "gdcmSwapper.h"
#include "gdcmPrinter.h"
#include <iostream>
#include <fstream>
#include <vector>
#include <assert.h>
int main(int argc, char *argv[])
{
  if( argc < 2 ) return 1;
  const char *filename = argv[1];
  gdcm::Reader reader;
  reader.SetFileName( filename );
  if( !reader.Read() )
  {
    std::cerr « "Failed to read:  " « filename « std::endl;
    return 1;
  }
  const gdcm::DataSet& ds = reader.GetFile().GetDataSet();
  gdcm::CSAHeader csa;
  const gdcm::PrivateTag &t1 = csa.GetCSAImageHeaderInfoTag();
  if( !ds.FindDataElement( t1 ) ) return 1;
  csa.LoadFromDataElement( ds.GetDataElement( t1 ) );
  //const char name[] = "MRDiffusion";
  const char name[] = "MR_ASL";
  if( csa.FindCSAElementByName(name) )
  {
    const gdcm::CSAElement & el = csa.GetCSAElementByName(name);
    const gdcm::ByteValue* bv = el.GetByteValue();
    std::string str( bv->GetPointer(), bv->GetLength() );
    str.erase(std::remove(str.begin(), str.end(), '\n'), str.end());
    size_t dl = gdcm::Base64::GetDecodeLength( str.c_str(), str.size() );
    std::vector<char> buf;
    buf.resize( dl );
    size_t dl2 = gdcm::Base64::Decode( &buf[0], buf.size(), str.c_str(), str.size() );
    (void)dl2;
    std::stringstream ss;
    ss.str( std::string(&buf[0], buf.size()) );
    gdcm::File file;
    gdcm::DataSet &ds2 = file.GetDataSet();
    gdcm::DataElement xde;
    try
    {
      while( xde.Read<gdcm::ExplicitDataElement,gdcm::SwapperNoOp>( ss ) )
      {
        ds2.Insert( xde );
      }
      assert( ss.eof() );
```

```
    }
    catch(std::exception &)
    {
      return 1;
    }
    gdcm::Printer p;
    p.SetFile( file );
    p.Print(std::cout);
  }
  return 0;
}
```

12.54 DumpToSQLITE3.cxx

```
/*=========================================================================

Program:  GDCM (Grassroots DICOM). A DICOM library

Copyright (c) 2006-2011 Mathieu Malaterre
All rights reserved.
See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.

This software is distributed WITHOUT ANY WARRANTY; without even
the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
PURPOSE.  See the above copyright notice for more information.

=========================================================================*/
/*
* Ref:
* http://massmail.spl.harvard.edu/public-archives/slicer-devel/2010/004408.html
*
* Implementation details:
* http://www.sqlite.org/c3ref/bind_blob.html
* http://www.adp-gmbh.ch/sqlite/bind_insert.html
*/
#include "gdcmScanner.h"
#include "gdcmDirectory.h"
#include "gdcmTag.h"
#include "gdcmTrace.h"
#include "sqlite3.h"
#include <stdio.h>
#include <time.h>
int main(int argc, char *argv[])
{
  if( argc < 2 )
    {
    return 1;
    }
  time_t time_start = time(nullptr);
  gdcm::Trace::SetDebug( false );
  gdcm::Trace::SetWarning( false );
  const char *inputdirectory = argv[1];
  gdcm::Directory d;
  unsigned int nfiles = d.Load( inputdirectory, true);
  gdcm::Scanner s;
  using gdcm::Tag;
  s.AddTag( Tag(0x20,0xd) ); // Study Instance UID
  s.AddTag( Tag(0x20,0xe) ); // Series Instance UID
  bool b0 = s.Scan( d.GetFilenames() );
  if( !b0 ) return 1;
  time_t time_scanner = time(nullptr);
  std::cout « "Finished loading data from :  " « nfiles « " files" « std::endl;
//  MappingType const &mappings = s.GetMappings();
  sqlite3* db;
  sqlite3_open("./dicom.db", &db);
  if(db == nullptr)
    {
    std::cerr « "Could not open database." « std::endl;
    return 1;
    }
  const char sql_stmt[] = "create table browser (seriesuid, studyuid)";
  int   ret;
  char *errmsg;
  ret = sqlite3_exec(db, sql_stmt, nullptr, nullptr, &errmsg);
  if(ret != SQLITE_OK)
    {
    printf("Error in statement:  %s [%s].\n", sql_stmt, errmsg);
```

```
    return 1;
    }
  using gdcm::Directory;
  using gdcm::Scanner;
  const Directory::FilenamesType& files = d.GetFilenames();
  Directory::FilenamesType::const_iterator file = files.begin();
  sqlite3_stmt *stmt;
  if ( sqlite3_prepare(
      db,
      "insert into browser values (?,?)",  // stmt
      -1, // If than zero, then stmt is read up to the first nul terminator
      &stmt,
      nullptr  // Pointer to unused portion of stmt
  )
    != SQLITE_OK)
    {
    printf("\nCould not prepare statement.");
    return 1;
    }
  //printf("\nThe statement has %d wildcards\n", sqlite3_bind_parameter_count(stmt));
  for(; file != files.end(); ++file)
    {
    const char *filename = file->c_str();
    bool b = s.IsKey(filename);
    if( b )
      {
      const Scanner::TagToValue &mapping = s.GetMapping(filename);
      Scanner::TagToValue::const_iterator it = mapping.begin();
      sqlite3_reset(stmt);
      for( int index = 1; it != mapping.end(); ++it, ++index)
        {
        //const Tag & tag = it->first;
        const char *value = it->second;
        if (sqlite3_bind_text (
            stmt,
            index,  // Index of wildcard
            value,
            (int)strlen(value),  // length of text
            SQLITE_STATIC // SQLite assumes that the information is in static
        )
          != SQLITE_OK)
          {
          printf("\nCould not bind int.\n");
          return 1;
          }
        }
      if (sqlite3_step(stmt) != SQLITE_DONE)
        {
        printf("\nCould not step (execute) stmt.\n");
        return 1;
        }
      }
    }
  sqlite3_close(db);
  time_t time_sqlite = time(nullptr);
  std::cout « "Time to scan DICOM files:  " « (time_scanner - time_start) « std::endl;
  std::cout « "Time to build SQLITE3:  " « (time_sqlite - time_scanner) « std::endl;
  return 0;
}
```

12.55 DumpToshibaDTI.cxx

```
/*=========================================================================

Program:  GDCM (Grassroots DICOM). A DICOM library

Copyright (c) 2006-2011 Mathieu Malaterre
All rights reserved.
See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.

This software is distributed WITHOUT ANY WARRANTY; without even
the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
PURPOSE.  See the above copyright notice for more information.

=========================================================================*/
/*
* https://groups.google.com/d/msg/comp.protocols.dicom/7IaIkT0ZG5U/k7LPu81VvAMJ
```

```
*/
#include "gdcmReader.h"
#include "gdcmPrivateTag.h"
#include "gdcmPrinter.h"
#include "gdcmDictPrinter.h"
#include <iostream>
#include <fstream>
#include <vector>
#include <assert.h>
static bool DumpToshibaDTI( const char * input, size_t len )
{
  static int i = 0;
  ++i;
  if( len % 2 ) return false;
  std::vector<char> copy( input, input + len );
  std::reverse( copy.begin(), copy.end() );
#if 0
  std::ostringstream f;
  f « "debug" « i;
  std::ofstream of( f.str().c_str(), std::ios::binary );
  of.write( &copy[0], copy.size() );
  of.close();
#else
  std::istringstream is;
  std::string dup( &copy[0], copy.size() );
  is.str( dup );
  gdcm::File file;
  gdcm::FileMetaInformation & fmi = file.GetHeader();
  fmi.SetDataSetTransferSyntax( gdcm::TransferSyntax::ExplicitVRLittleEndian );
  gdcm::DataSet & ds = file.GetDataSet();
  ds.Read<gdcm::ExplicitDataElement,gdcm::SwapperNoOp>( is );
  //gdcm::DictPrinter p;
  gdcm::Printer p;
  p.SetFile( file );
  p.SetColor( true );
  p.Print( std::cout );
#endif
  return true;
}
int main(int argc, char *argv[])
{
  if( argc < 2 ) return 1;
  const char *filename = argv[1];
  gdcm::Reader reader;
  reader.SetFileName( filename );
  if( !reader.Read() )
    {
    std::cerr « "Failed to read:  " « filename « std::endl;
    return 1;
    }
  const gdcm::DataSet& ds = reader.GetFile().GetDataSet();
  // (0029,0010) ??  (LO) [PMTF INFORMATION DATA ]                      # 22,1 Private Creator
  // (0029,1001) ??  (SQ) (Sequence with undefined length)              # u/l,1 ?
  const gdcm::PrivateTag tpmtf(0x0029,0x1,"PMTF INFORMATION DATA");
  if( !ds.FindDataElement( tpmtf) ) return 1;
  const gdcm::DataElement& pmtf = ds.GetDataElement( tpmtf );
  if ( pmtf.IsEmpty() ) return 1;
  gdcm::SmartPointer<gdcm::SequenceOfItems> seq = pmtf.GetValueAsSQ();
  if ( !seq || !seq->GetNumberOfItems() ) return 1;
  size_t n = seq->GetNumberOfItems();
  for( size_t i = 1; i <= n; ++i )
    {
    gdcm::Item &item = seq->GetItem(i);
    gdcm::DataSet &subds = item.GetNestedDataSet();
    // (0029,0010) ??  (LO) [PMTF INFORMATION DATA ]                  # 22,1 Private Creator
    // (0029,1090) ??  (OB) 00\05\00\13\00\12\00\22\                  # 202,1 ?
    const gdcm::PrivateTag tseq(0x0029,0x90,"PMTF INFORMATION DATA");
    if( subds.FindDataElement( tseq ) )
      {
      const gdcm::DataElement & de = subds.GetDataElement( tseq );
      const gdcm::ByteValue * bv = de.GetByteValue();
      if( !bv ) return 1;
      bool b = DumpToshibaDTI( bv->GetPointer(), bv->GetLength() );
      if( !b ) return 1;
      }
    }
  return 0;
}
```

12.56 DumpToshibaDTI2.cxx

```
/*=========================================================================

Program:  GDCM (Grassroots DICOM). A DICOM library

Copyright (c) 2006-2011 Mathieu Malaterre
All rights reserved.
See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.

This software is distributed WITHOUT ANY WARRANTY; without even
the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
PURPOSE.  See the above copyright notice for more information.

=========================================================================*/
/*
*
      https://gazelle.ihe.net/EVSClient/dicomResult.seam;jsessionid=x+Rf9Zs+ip49P+jC3L8SLZb8?&oid=1.3.6.1.4.1.12559.11.1.2.1.4.162
*/
#include "gdcmReader.h"
#include "gdcmPrivateTag.h"
#include "gdcmPrinter.h"
#include "gdcmDictPrinter.h"
#include <iostream>
#include <fstream>
#include <vector>
#include <assert.h>
static bool DumpToshibaDTI2( const char * input, size_t len )
{
  static int i = 0;
  ++i;
  if( len % 2 ) return false;
  std::vector<char> copy( input, input + len );
  std::reverse( copy.begin(), copy.end() );
#if 0
  std::ostringstream f;
  f « "debug" « i;
  std::ofstream of( f.str().c_str(), std::ios::binary );
  of.write( &copy[0], copy.size() );
  of.close();
#else
  std::istringstream is;
  std::string dup( &copy[0], copy.size() );
  is.str( dup );
  gdcm::File file;
  gdcm::FileMetaInformation & fmi = file.GetHeader();
  fmi.SetDataSetTransferSyntax( gdcm::TransferSyntax::ExplicitVRLittleEndian );
  gdcm::DataSet & ds = file.GetDataSet();
  ds.Read<gdcm::ExplicitDataElement,gdcm::SwapperNoOp>( is );
  //gdcm::DictPrinter p;
  gdcm::Printer p;
  p.SetFile( file );
  p.SetColor( true );
  p.Print( std::cout );
#endif
  return true;
}
int main(int argc, char *argv[])
{
  if( argc < 2 ) return 1;
  const char *filename = argv[1];
  gdcm::Reader reader;
  reader.SetFileName( filename );
  if( !reader.Read() )
    {
    std::cerr « "Failed to read:  " « filename « std::endl;
    return 1;
    }
  const gdcm::DataSet& ds = reader.GetFile().GetDataSet();
  /*
(0029,1001) SQ (Sequence with explicit length #=6)      # 18746, 1 Unknown Tag & Data
(fffe,e000) na (Item with explicit length #=2)          # 206, 1 Item
(0029,0010) LO [TOSHIBA_MEC_MR3]                        #  16, 1 PrivateCreator
(0029,1090) OB 00\07\00\06\00\05\00\04\00\03\00\02\00\0c\00\01\00\00\00\00\00\12...  # 170, 1 Unknown Tag & Data
(fffe,e00d) na (ItemDelimitationItem for re-encoding)   #   0, 0 ItemDelimitationItem
(fffe,e000) na (Item with explicit length #=2)          # 866, 1 Item
(0029,0010) LO [TOSHIBA_MEC_MR3]                        #  16, 1 PrivateCreator
(0029,1090) OB 45\4e\49\50\53\4c\20\52\41\5c\45\4e\49\50\53\4c\54\5c\52\45\53\55...  # 830, 1 Unknown Tag & Data
[...]
(0029,1002) SQ (Sequence with explicit length #=1)      # 120, 1 Unknown Tag & Data
```

```
(fffe,e000) na (Item with explicit length #=2)          # 112, 1 Item
(0029,0010) LO [TOSHIBA_MEC_MR3]                        #  16, 1 PrivateCreator
(0029,1090) OB 00\10\00\02\53\55\10\80\70\0d\30\31\5e\33\52\4d\5f\43\45\4d\5f\41...  #  76, 1 Unknown Tag & Data
(fffe,e00d) na (ItemDelimitationItem for re-encoding)   #   0, 0 ItemDelimitationItem
*/
  const gdcm::PrivateTag tmecmr3(0x0029,0x1,"TOSHIBA_MEC_MR3");
  if( !ds.FindDataElement( tmecmr3) ) return 1;
  const gdcm::DataElement& mecmr3 = ds.GetDataElement( tmecmr3 );
  if ( mecmr3.IsEmpty() ) return 1;
  gdcm::SmartPointer<gdcm::SequenceOfItems> seq = mecmr3.GetValueAsSQ();
  if ( !seq || !seq->GetNumberOfItems() ) return 1;
  size_t n = seq->GetNumberOfItems();
  for( size_t i = 1; i <= n; ++i )
    {
    gdcm::Item &item = seq->GetItem(i);
    gdcm::DataSet &subds = item.GetNestedDataSet();
    const gdcm::PrivateTag tseq(0x0029,0x90,"TOSHIBA_MEC_MR3");
    if( subds.FindDataElement( tseq ) )
      {
      const gdcm::DataElement & de = subds.GetDataElement( tseq );
      const gdcm::ByteValue * bv = de.GetByteValue();
      if( !bv ) return 1;
      bool b = DumpToshibaDTI2( bv->GetPointer(), bv->GetLength() );
      if( !b ) return 1;
      }
    }
  return 0;
}
```

12.57 DumpVisusChange.cxx

```
/*=========================================================================

Program:  GDCM (Grassroots DICOM). A DICOM library

Copyright (c) 2006-2011 Mathieu Malaterre
All rights reserved.
See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.

This software is distributed WITHOUT ANY WARRANTY; without even
the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
PURPOSE.  See the above copyright notice for more information.

=========================================================================*/
#include "gdcmReader.h"
#include "gdcmDirectory.h"
#include "gdcmStringFilter.h"
#include <vector>
#include <algorithm>
/*
*/
static bool process( std::vector<gdcm::DataElement> & ms, const char * filename)
{
  using namespace gdcm;
  Tag pd(0x7fe0,0x0000);
  std::set<gdcm::Tag> skiptags;
  skiptags.insert( pd );
  gdcm::Reader reader;
  reader.SetFileName( filename );
  if( !reader.ReadUpToTag( pd, skiptags ) )
  {
    std::cerr « "Failure to read:  " « filename « std::endl;
    return false;
  }
  gdcm::File &file = reader.GetFile();
  gdcm::DataSet &ds1 = file.GetDataSet();
  const gdcm::PrivateTag tseq1(0x5533,0x33,"Visus Change");
  if( !ds1.FindDataElement( tseq1 ) ) return true;
  const gdcm::DataElement& seq1 = ds1.GetDataElement( tseq1 );
  SmartPointer<SequenceOfItems> sqi1 = seq1.GetValueAsSQ();
  const size_t nitems = sqi1->GetNumberOfItems();
  for( size_t item = 1; item < nitems; ++item )
    {
    Item &item1 = sqi1->GetItem(item);
    DataSet &ds2 = item1.GetNestedDataSet();
    for(DataSet::ConstIterator it = ds2.Begin(); it != ds2.End(); ++it )
      {
```

```
      DataElement const & de = *it;
      // cannot simply use std::set here, see there is a discrepancy in between
      // operator== and operator<.
      // So only use operator== here:
      std::vector<DataElement>::iterator vit = std::find(ms.begin(), ms.end(), de);
      if( vit == ms.end() )
        ms.push_back(de);
      }
    }
  return true;
}
int main(int argc, char *argv[])
{
  bool usefastpath = true;
  if( argc < 2 ) return 1;
  using namespace gdcm;
  const char *filename = argv[1];
  gdcm::Directory::FilenamesType filenames;
  if( !gdcm::System::FileExists(filename) )
    {
    std::cerr « "Could not find file:  " « filename « std::endl;
    return 1;
    }
  gdcm::Directory dir;
  if( gdcm::System::FileIsDirectory(filename) )
  {
    unsigned int nfiles = dir.Load(filename, false);
    if( nfiles == 0 )
      {
      std::cerr « "Could not find files:  " « filename « std::endl;
      return 1;
      }
    filenames = dir.GetFilenames();
  }
  else
  {
    filenames.push_back( filename );
  }
  gdcm::StringFilter sf;
  Tag pd(0x7fe0,0x0000);
  std::set<gdcm::Tag> skiptags;
  skiptags.insert( pd );
  gdcm::Reader reader;
  reader.SetFileName( filenames[0].c_str() );
  if( !reader.ReadUpToTag( pd, skiptags ) )
  {
    std::cerr « "Could not read file:  " « filename « std::endl;
    return 1;
  }
  gdcm::File &file = reader.GetFile();
  sf.SetFile(file);
  if( usefastpath ) {
    // Heuristic, assume if private tag cannot be found in first file, skip the directory
    gdcm::DataSet &ds1 = file.GetDataSet();
    const gdcm::PrivateTag tseq1(0x5533,0x33,"Visus Change");
    if( !ds1.FindDataElement( tseq1 ) ){
      std::cerr « "Could not find private tag in first file skipping whole directory:  " « filename « std::endl;
      return 0;
    }
  }
  std::vector<DataElement> ms;
  for(gdcm::Directory::FilenamesType::const_iterator cit = filenames.begin(); cit != filenames.end(); ++cit )
  {
    if( !process(ms, cit->c_str()) ) {
      return 1;
    }
  }
  if( !ms.empty() ) {
    std::sort(ms.begin(), ms.end());
    std::cout « filename « ",\"";
    for(std::vector<DataElement>::const_iterator it = ms.begin(); it != ms.end(); ++it )
    {
      DataElement const & de = *it;
      std::string const & s = sf.ToString( de );
      std::cout « de.GetTag() « " " « s « std::endl;
    }
    std::cout « "\"" « std::endl;
  }
  return 0;
}
```

12.58 DuplicatePCDE.cxx

```
/*=========================================================================

Program:  GDCM (Grassroots DICOM). A DICOM library

Copyright (c) 2006-2011 Mathieu Malaterre
All rights reserved.
See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.

This software is distributed WITHOUT ANY WARRANTY; without even
the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
PURPOSE.  See the above copyright notice for more information.

=========================================================================*/
#include "gdcmReader.h"
#include "gdcmWriter.h"
#include "gdcmItem.h"
#include "gdcmImageReader.h"
#include "gdcmSequenceOfItems.h"
#include "gdcmFile.h"
#include "gdcmTag.h"
/*
Usage:
DuplicatePCDE gdcmData/D_CLUNIE_CT1_J2KI.dcm out.dcm

aka:
medical.nema.org/medical/dicom/DataSets/WG04/IMAGES/J2KI/CT1_J2KI

See:
gdcmConformanceTests/CT1_J2KI_DuplicatePCDE.dcm

Original thread can be found at:

http://groups.google.com/group/comp.protocols.dicom/browse_thread/thread/82f28c4db28963af


Question:
1.
There is no restriction for a specific Private Creator Data Element
(PCDE) to be unique within the same group, right ?
Decoders of Private Data would have to handle the case where a PCDE
would be repeated and should NOT stop on the first instance of a
particular PCDE, right ?

Eg.  when searching for the tag associated with
(0x0029,0x0010,"SIEMENS CSA HEADER") in the following (pseudo)
dataset:

(0029,0010) LO [SIEMENS CSA HEADER]                     #  18, 1
PrivateCreator
(0029,0011) LO [SIEMENS MEDCOM HEADER]                  #  22, 1
PrivateCreator
(0029,0012) LO [SIEMENS MEDCOM HEADER2]                 #  22, 1
PrivateCreator
(0029,0013) LO [SIEMENS CSA HEADER]                     #  18, 1
PrivateCreator
(0029,1008) CS [IMAGE NUM 4]                            #  12, 1
CSAImageHeaderType
(0029,1009) LO [20050723]                               #   8, 1
CSAImageHeaderVersion
(0029,1010) OB 53\56\31\30\04\03\02\01\38\00\00\00\4d
\00\00\00\45\63\68\6f\4c\69...  # 6788, 1 CSAImageHeaderInfo
(0029,1018) CS [MR]                                     #   2, 1
CSASeriesHeaderType
(0029,1019) LO [20050723]                               #   8, 1
CSASeriesHeaderVersion
(0029,1020) OB 53\56\31\30\04\03\02\01\2c\00\00\00\4d
\00\00\00\55\73\65\64\50\61...  # 51520, 1 CSASeriesHeaderInfo
(0029,1131) LO [4.0.163088300]                          #  14, 1
PMTFInformation1
(0029,1132) UL 32768                                    #   4, 1
PMTFInformation2
(0029,1133) UL 0                                        #   4, 1
PMTFInformation3
(0029,1134) CS [DB TO DICOM]                            #  12, 1
PMTFInformation4
(0029,1260) ??  63\6f\6d\20                              #   4, 1
Unknown Tag & Data
(0029,1310) OB 53\56\31\30\04\03\02\01\38\00\00\00\4d
```

```
\00\00\00\45\63\68\6f\4c\69...  # 6788, 1 CSAImageHeaderInfo

one should return two instances, correct ?

Answer:
I would say that this is covered in principle by the PS 3.5 7.1
"The Data Elements ...  shall occur at most once in a Data Set"
rule, since the data element is defined by the tuple
(private creator,gggg,ee) where xxee is the element
number and xx is arbitrary and has no inherent meaning and
does not serve to disambiguate the data element.

E.g.:

(0019,0030) Private Creator ID = "Smith"
...
(0019,0032) Private Creator ID = "Smith"
...
(0019,3015) Fractal Index = "32"
...
(0019,3215) Fractal Index = "32"

would be illegal because even though they are assigned different
(completely arbitrary) blocks, with the same group, element
number and private creator, (0019,3015) and (0019,3215) are the
"same" data element.

*/
int main(int argc, char *argv[])
{
  if( argc < 3 )
    {
    std::cerr « argv[0] « " input.dcm output.dcm" « std::endl;
    return 1;
    }
  const char *filename = argv[1];
  const char *outfilename = argv[2];
  gdcm::Reader reader;
  reader.SetFileName( filename );
  if( !reader.Read() )
    {
    return 1;
    }
  gdcm::File &file = reader.GetFile();
  gdcm::DataSet &ds = file.GetDataSet();
  // Let's get all private element from group 0x9:
/*
(0009,0010) LO [GEMS_IDEN_01]                                    # 12,1 Private Creator
(0009,1001) LO [GE_GENESIS_FF ]                                  # 14,1 Full fidelity
(0009,1002) SH [CT01]                                            # 4,1 Suite id
(0009,1004) SH [HiSpeed CT/i]                                    # 12,1 Product id
(0009,1027) SL 862399669                                         # 4,1 Image actual date
(0009,1030) SH (no value)                                        # 0,1 Service id
(0009,1031) SH (no value)                                        # 0,1 Mobile location number
(0009,10e6) SH [05]                                              # 2,1 Genesis Version - now
(0009,10e7) UL 973283917                                         # 4,1 Exam Record checksum
(0009,10e9) SL 862399669                                         # 4,1 Actual series data time stamp
*/
  gdcm::Tag start(0x0009,0x0);
  // Create a temporary duplicate dataset, since we cannot insert data element as we go over them (std::set
  // would reorganize itself as we go over it ...)
  gdcm::DataSet dup;
  gdcm::Tag new_private(0x0009,0x0);
  while (start.GetGroup() == 0x9 )
    {
    const gdcm::DataElement& de = ds.FindNextDataElement(start);
    const gdcm::Tag &t = de.GetTag();
    if( t.IsPrivateCreator() )
      {
      std::cout « t « std::endl;
      // Ok let's duplicate into the next available attribute:
      gdcm::DataElement duplicate = de;
      duplicate.GetTag().SetElement( (uint16_t)(t.GetElement() + 1) );
      dup.Insert( duplicate );
      new_private = duplicate.GetTag();
      }
    else if( t.IsPrivate() && !t.IsPrivateCreator() )
      {
      //std::cout « de « std::endl;
      std::string owner = ds.GetPrivateCreator( de.GetTag() );
      //std::cout « owner « std::endl;
```

```
      gdcm::DataElement duplicate = de;
      duplicate.GetTag().SetPrivateCreator( new_private );
      if( const gdcm::ByteValue *bv = duplicate.GetByteValue() )
        {
        // Warning:  when doing :  duplicate = de, only the pointer to the ByteValue is passed
        // (to avoid large memory duplicate).  We need to explicitly duplicate the bytevalue ourselves:
        gdcm::ByteValue *dupbv = new gdcm::ByteValue( bv->GetPointer(),
          bv->GetLength() );
        // Let's recognize the duplicated ASCII-type elements:
        if( duplicate.GetVR() & gdcm::VR::VRASCII )
          dupbv->Fill( 'X' );
        duplicate.SetValue( *dupbv );
        }
      dup.Insert( duplicate );
      }
    start = t;
    // move to next possible 'public' element
    start.SetElement( (uint16_t)(start.GetElement() + 1) );
    }
  gdcm::DataSet::ConstIterator it = dup.Begin();
  for( ; it != dup.End(); ++it )
    {
    ds.Insert( *it );
    }
  gdcm::Writer w;
  w.SetFile( file );
  w.SetFileName( outfilename );
  if (!w.Write() )
    {
    return 1;
    }
  return 0;
}
```

12.59 ELSCINT1WaveToText.cxx

```
/*=========================================================================

Program:  GDCM (Grassroots DICOM). A DICOM library

Copyright (c) 2006-2011 Mathieu Malaterre
All rights reserved.
See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.

This software is distributed WITHOUT ANY WARRANTY; without even
the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
PURPOSE.  See the above copyright notice for more information.

=========================================================================*/
#include "gdcmReader.h"
#include "gdcmPrivateTag.h"
/*
* This example shows how to read a Wave Information tag from ELSCINT1
* The wave information is stored in Tag (01e1,18,ELSCINT1) hidden in a
* Secondary Capture Image Storage (usually a 'N' Symbol is shown)
*
* Everything done in this code is for the sole purpose of writing interoperable
* software under Sect.  1201 (f) Reverse Engineering exception of the DMCA.
* If you believe anything in this code violates any law or any of your rights,
* please contact us (gdcm-developers@lists.sourceforge.net) so that we can
* find a solution.
*
* Everything you do with this code is at your own risk, since decompression
* algorithm was not written from specification documents.
*
* Special thanks to:
* Gauthier Bouilhol
*/
template <typename T>
bool dumpargs(std::ostream & os, T c1, T c2, T c3, T c4, T c5, T c6, T c7, T c8)
{
  static const char sep = '\t';
  os « c1 « sep « c2 « sep « c3 « sep « c4 « sep « c5 « sep « c6 « sep « c7 « sep « c8;
  os « std::endl;
  return true;
}
bool wave2stream( std::ostream &text_file, const char *in, size_t len )
```

```
{
  const short * buffer = (const short*)in;
  size_t length = len / sizeof( short );
  text_file « "COMPLETE_WAVE" « '\t' « "MASK"       « '\t' « "AQUISITION_PROFIL" « '\t' « "END-INHALE" « '\t' «
      "END-EXHALE" « '\t' « "AQUISITION_WAVE" « '\t' « "WAVE_STATISTICS" « '\t' « "MASK"        « std::endl;
  for (size_t i=0;i<length-76;i+=2)
    {
    if ( i < 74 )
      {
      if (buffer[i+75] == 0)
        text_file « buffer[i+74]    « '\t' « buffer[i+75] « '\t' « 0                « '\t' « "  "         «
      '\t' « "  "         « '\t' « "  "              « '\t' « buffer[i]         « '\t' « buffer[i+1] «
      std::endl;
      if (buffer[i+75] == 16384)
        text_file « buffer[i+74]    « '\t' « buffer[i+75] « '\t' « 0                « '\t' « buffer[i+74] «
      '\t' « "  "         « '\t' « "  "              « '\t' « buffer[i]         « '\t' « buffer[i+1] «
      std::endl;
      if (buffer[i+75] == 256)
        text_file « buffer[i+74]    « '\t' « buffer[i+75] « '\t' « 0                « '\t' « "  "         «
      '\t' « buffer[i+74] « '\t' « "  "              « '\t' « buffer[i]         « '\t' « buffer[i+1] «
      std::endl;
      if (buffer[i+75] == -32768)
        text_file « buffer[i+74]    « '\t' « buffer[i+75] « '\t' « 1                « '\t' « "  "         «
      '\t' « "  "         « '\t' « buffer[i+74]      « '\t' « buffer[i]         « '\t' « buffer[i+1] «
      std::endl;
      if (buffer[i+75] == -16384)
        text_file « buffer[i+74]    « '\t' « buffer[i+75] « '\t' « 1                « '\t' « buffer[i+74] «
      '\t' « "  "         « '\t' « buffer[i+74]      « '\t' « buffer[i]         « '\t' « buffer[i+1] «
      std::endl;
      if (buffer[i+75] == -32512)
        text_file « buffer[i+74]    « '\t' « buffer[i+75] « '\t' « 1                « '\t' « "  "         «
      '\t' « buffer[i+74] « '\t' « buffer[i+74]      « '\t' « buffer[i]         « '\t' « buffer[i+1] «
      std::endl;
      }
    else
      {
      if (buffer[i+75] == 0)
        text_file « buffer[i+74]    « '\t' « buffer[i+75] « '\t' « 0                « '\t' « "  "         «
      '\t' « "  "         « '\t' « "  "              « '\t' « "  "              « '\t' « "  "         «
      std::endl;
      if (buffer[i+75] == 16384)
        text_file « buffer[i+74]    « '\t' « buffer[i+75] « '\t' « 0                « '\t' « buffer[i+74] «
      '\t' « "  "         « '\t' « "  "              « '\t' « "  "              « '\t' « "  "         «
      std::endl;
      if (buffer[i+75] == 256)
        text_file « buffer[i+74]    « '\t' « buffer[i+75] « '\t' « 0                « '\t' « "  "         «
      '\t' « buffer[i+74] « '\t' « "  "              « '\t' « "  "              « '\t' « "  "         «
      std::endl;
      if (buffer[i+75] == -32768)
        text_file « buffer[i+74]    « '\t' « buffer[i+75] « '\t' « 1                « '\t' « "  "         «
      '\t' « "  "         « '\t' « buffer[i+74]      « '\t' « "  "              « '\t' « "  "         «
      std::endl;
      if (buffer[i+75] == -16384)
        text_file « buffer[i+74]    « '\t' « buffer[i+75] « '\t' « 1                « '\t' « buffer[i+74] «
      '\t' « "  "         « '\t' « buffer[i+74]      « '\t' « "  "              « '\t' « "  "         «
      std::endl;
      if (buffer[i+75] == -32512)
        text_file « buffer[i+74]    « '\t' « buffer[i+75] « '\t' « 1                « '\t' « "  "         «
      '\t' « buffer[i+74] « '\t' « buffer[i+74]      « '\t' « "  "              « '\t' « "  "         «
      std::endl;
      }
    }
  return true;
}
int main(int argc, char *argv [])
{
  if( argc < 3 ) return 1;
  const char *filename = argv[1];
  const char *outfilename = argv[2];
  gdcm::Reader reader;
  reader.SetFileName( filename );
  if( !reader.Read() )
    {
    std::cerr « "Failed to read:  " « filename « std::endl;
    return 1;
    }
  const gdcm::DataSet& ds = reader.GetFile().GetDataSet();
  const gdcm::PrivateTag twave(0x01e1,0x18,"ELSCINT1");
  if( !ds.FindDataElement( twave ) ) return 1;
  const gdcm::DataElement& wave = ds.GetDataElement( twave );
  if ( wave.IsEmpty() ) return 1;
```

```
  const gdcm::ByteValue * bv = wave.GetByteValue();
  assert( bv );
  std::ofstream os( outfilename, std::ios::binary );
  // Dump that to a CSV file:
  wave2stream( os, bv->GetPointer(), bv->GetLength() );
  os.close();
  return 0;
}
```

12.60 EmptyMask.cxx

```
/*=========================================================================

Program:  GDCM (Grassroots DICOM). A DICOM library

Copyright (c) 2006-2011 Mathieu Malaterre
All rights reserved.
See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.

This software is distributed WITHOUT ANY WARRANTY; without even
the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
PURPOSE.  See the above copyright notice for more information.

=========================================================================*/
#include "gdcmEmptyMaskGenerator.h"
#include <string>
#include <cstring>
int main( int argc, char *argv[] )
{
  std::string inputdir;
  std::string outputdir;
  bool input_sopclassuid = true;
  bool grayscale_secondary_sopclassuid = false;
  if( argc < 3 ) return 1;
  inputdir = argv[1];
  outputdir = argv[2];
  // input_sopclassuid -> Use original SOP Class UID from input DICOM (Default).
  // grayscale_secondary_sopclassuid -> Use Grayscale Secondary Image Storage SOP Class UID.
  if( argc >= 3 )
  {
    input_sopclassuid = false;
    if( strcmp("input_sopclassuid", argv[3]) == 0 )
      input_sopclassuid = true;
    else if (strcmp("grayscale_secondary_sopclassuid", argv[3]) == 0 ) {
      grayscale_secondary_sopclassuid = true;
    }
  }
  //
  gdcm::EmptyMaskGenerator emg;
  if( input_sopclassuid )
    emg.SetSOPClassUIDMode( gdcm::EmptyMaskGenerator::UseOriginalSOPClassUID );
  else if( grayscale_secondary_sopclassuid )
    emg.SetSOPClassUIDMode( gdcm::EmptyMaskGenerator::UseGrayscaleSecondaryImageStorage );
  emg.SetInputDirectory( inputdir.c_str() );
  emg.SetOutputDirectory( outputdir.c_str() );
  if( !emg.Execute() )
  {
    return 1;
  }
  return 0;
}
```

12.61 EncapsulateFileInRawData.cxx

```
/*=========================================================================

Program:  GDCM (Grassroots DICOM). A DICOM library

Copyright (c) 2006-2011 Mathieu Malaterre
All rights reserved.
See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.

This software is distributed WITHOUT ANY WARRANTY; without even
```

```
the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
PURPOSE.  See the above copyright notice for more information.

=========================================================================*/
#include "gdcmAnonymizer.h"
#include "gdcmWriter.h"
#include "gdcmUIDGenerator.h"
#include "gdcmFile.h"
#include "gdcmTag.h"
#include "gdcmSystem.h"
#include "magic.h" // libmagic, API to file command line tool
/*
* Let say you want to encapsulate a file type that is not defined in DICOM (exe, zip, png)
* PNG is a bad example, unless it contains transparency (which has been deprecated).
* It will take care of dispatching each chunk to an appropriate data item (pretty much like
* WaveformData)
*
* Usage:
* ./EncapsulateFileInRawData large_input_file.exe large_input_file.dcm
*/
// TODO:
// $ file -bi /tmp/gdcm-2.1.0.pdf
int main(int argc, char *argv[])
{
  if( argc < 3 )
    {
    std::cerr « argv[0] « " inputfile output.dcm" « std::endl;
    return 1;
    }
  const char *filename = argv[1];
  const char *outfilename = argv[2];
  if( !gdcm::System::FileExists( filename ) ) return 1;
  size_t s = gdcm::System::FileSize(filename);
  if( !s ) return 1;
  magic_t cookie = magic_open(MAGIC_NONE);
  const char * file_type = magic_file(cookie, filename);
  if( !file_type ) return 1;
  magic_close(cookie);
  gdcm::Writer w;
  gdcm::File &file = w.GetFile();
  //gdcm::DataSet &ds = file.GetDataSet();
  //w.SetCheckFileMetaInformation( true );
  w.SetFileName( outfilename );
  file.GetHeader().SetDataSetTransferSyntax( gdcm::TransferSyntax::ImplicitVRLittleEndian );
  gdcm::Anonymizer anon;
  anon.SetFile( file );
  gdcm::MediaStorage ms = gdcm::MediaStorage::RawDataStorage;
  gdcm::UIDGenerator gen;
  anon.Replace( gdcm::Tag(0x0008,0x16), ms.GetString() );
  std::cout « ms.GetString() « std::endl;
  anon.Replace( gdcm::Tag(0x0008,0x18), gen.Generate() );
  if (!w.Write() )
    {
    std::cerr « "Could not write:  " « outfilename « std::endl;
    return 1;
    }
  return 0;
}
```

12.62 ExtractEncryptedContent.cxx

```
/*=========================================================================

Program:  GDCM (Grassroots DICOM). A DICOM library

Copyright (c) 2006-2011 Mathieu Malaterre
All rights reserved.
See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.

This software is distributed WITHOUT ANY WARRANTY; without even
the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
PURPOSE.  See the above copyright notice for more information.

=========================================================================*/
#include "gdcmReader.h"
#include <fstream>
/*
```

```
openssl smime -encrypt -binary -aes256 -in outputfile.dcm -inform DER -out outputfile.der -outform DER
      ../trunk/Testing/Source/Data/certificate.pem

openssl smime -decrypt -binary -in out.der -inform DER -out outputfile.dcm -outform DER -inkey
      ../trunk/Testing/Source/Data/privatekey.pem ../trunk/Testing/Source/Data/certificate.pem

*/
int main(int argc, char *argv[])
{
  if( argc < 3 )
    {
    std::cerr « argv[0] « " input.dcm output.der" « std::endl;
    return 1;
    }
  const char *filename = argv[1];
  const char *outfilename = argv[2];
  gdcm::Reader reader;
  reader.SetFileName( filename );
  if( !reader.Read() )
    {
    return 1;
    }
  gdcm::File &file = reader.GetFile();
  gdcm::DataSet &ds = file.GetDataSet();
  const gdcm::DataElement &EncryptedAttributesSequence = ds.GetDataElement( gdcm::Tag( 0x0400,0x0500 ) );
  gdcm::SequenceOfItems *sqi = EncryptedAttributesSequence.GetValueAsSQ();
  if ( !sqi || sqi->GetNumberOfItems() != 1 ) return 1;
  gdcm::Item &item = sqi->GetItem(1);
  gdcm::DataSet &nestedds = item.GetNestedDataSet();
  if( !  nestedds.FindDataElement( gdcm::Tag( 0x0400,0x0520) ) ) return 1;
  const gdcm::DataElement &EncryptedContent = nestedds.GetDataElement( gdcm::Tag( 0x0400,0x0520) );
  const gdcm::ByteValue *bv = EncryptedContent.GetByteValue();
  std::ofstream of( outfilename, std::ios::binary );
  of.write( bv->GetPointer(), bv->GetLength() );
  of.close();
  return 0;
}
```

12.63 ExtractIconFromFile.cxx

```
/*=========================================================================

Program:  GDCM (Grassroots DICOM). A DICOM library

Copyright (c) 2006-2011 Mathieu Malaterre
All rights reserved.
See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.

This software is distributed WITHOUT ANY WARRANTY; without even
the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
PURPOSE.  See the above copyright notice for more information.

=========================================================================*/
/*
* This example shows how to either retrieve an Icon if present somewhere
* in the file, or else generate one.
*/
#include "gdcmImageReader.h"
#include "gdcmPNMCodec.h"
#include "gdcmIconImageFilter.h"
#include "gdcmIconImageGenerator.h"
bool WriteIconAsPNM(const char* filename, const gdcm::IconImage& icon)
{
  gdcm::PNMCodec pnm;
  pnm.SetDimensions( icon.GetDimensions() );
  pnm.SetPixelFormat( icon.GetPixelFormat() );
  pnm.SetPhotometricInterpretation( icon.GetPhotometricInterpretation() );
  pnm.SetLUT( icon.GetLUT() );
  const gdcm::DataElement& in = icon.GetDataElement();
  bool b = pnm.Write( filename, in );
  assert( b );
  return b;
}
int main(int argc, char *argv [])
{
  if( argc < 2 ) return 1;
```

```
  const char *filename = argv[1];
  gdcm::ImageReader reader;
  reader.SetFileName( filename );
  if( !reader.Read() )
    {
    std::cerr « "Failed to read (or not image):  " « filename « std::endl;
    return 1;
    }
  gdcm::IconImageFilter iif;
  iif.SetFile( reader.GetFile() );
  bool b = iif.Extract();
  if( b )
    {
    const gdcm::IconImage &icon = iif.GetIconImage(0);
    icon.Print( std::cout );
    if( !icon.GetTransferSyntax().IsEncapsulated() )
      {
      // Let's write out this icon as PNM file
      WriteIconAsPNM("icon.ppm", icon);
      }
    else if( icon.GetTransferSyntax() == gdcm::TransferSyntax::JPEGBaselineProcess1
      || icon.GetTransferSyntax() == gdcm::TransferSyntax::JPEGExtendedProcess2_4
    )
      {
      const gdcm::DataElement& in = icon.GetDataElement();
      const gdcm::ByteValue *bv = in.GetByteValue();
      assert( bv );
      std::ofstream out( "icon.jpg", std::ios::binary );
      out.write( bv->GetPointer(), bv->GetLength() );
      out.close();
      }
    }
  else
    {
    assert( iif.GetNumberOfIconImages() == 0 );
    std::cerr « "No Icon Found anywhere in file" « std::endl;
    const gdcm::Image &img = reader.GetImage();
    gdcm::IconImageGenerator iig;
    iig.AutoPixelMinMax(true);
    iig.SetPixmap( img );
    const unsigned int idims[2] = { 64, 64 };
    iig.SetOutputDimensions( idims );
    //iig.SetPixelMinMax(60, 868);
    if( !iig.Generate() ) return 1;
    const gdcm::IconImage & icon = iig.GetIconImage();
    WriteIconAsPNM("icon.ppm", icon);
    }
  return 0;
}
```

12.64 Extracting_All_Resolution.cxx

```
/*=========================================================================

Program:  GDCM (Grassroots DICOM). A DICOM library

Copyright (c) 2006-2011 Mathieu Malaterre
All rights reserved.
See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.

This software is distributed WITHOUT ANY WARRANTY; without even
the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
PURPOSE.  See the above copyright notice for more information.

=========================================================================*/
// This work was realised during the GSOC 2011 by Manoj Alwani
#include <fstream>
#include <stdint.h>
#include <string.h>
#include <assert.h>
#include <gdcm_j2k.h>
#include <gdcm_jp2.h>
#include<iostream>
#include <cstring>
#include <stdio.h>
#include <string.h>
#include <stdlib.h>
```

```
#include <math.h>
#include "gdcmImageReader.h"
#include "gdcmSequenceOfItems.h"
#include "gdcmSystem.h"
#include <fstream>
#include "gdcm_openjpeg.h"
#include "gdcmMediaStorage.h"
#include "gdcmWriter.h"
#include "gdcmItem.h"
#include "gdcmImageReader.h"
#include "gdcmAttribute.h"
#include "gdcmFile.h"
#include "gdcmTag.h"
#include "gdcmTransferSyntax.h"
#include "gdcmUIDGenerator.h"
#include "gdcmAnonymizer.h"
#include "gdcmStreamImageWriter.h"
#include "gdcmImageHelper.h"
#include "gdcmTrace.h"
void error_callback(const char *msg, void *) {
  (void)msg;
}
void warning_callback(const char *msg, void *) {
  (void)msg;
}
void info_callback(const char *msg, void *) {
  (void)msg;
}
bool Write_Resolution(gdcm::StreamImageWriter & theStreamWriter, const char *filename, int res, std::ostream&
      of, int flag,  gdcm::SequenceOfItems *sq, int No_Of_Resolutions)
{
  std::ifstream is;
  is.open( filename, std::ios::binary );
  opj_dparameters_t parameters;  /* decompression parameters */
  opj_event_mgr_t event_mgr;    /* event manager */
  opj_dinfo_t* dinfo;  /* handle to a decompressor */
  opj_cio_t *cio;
  opj_image_t *image = NULL;
  // FIXME: Do some stupid work:
  is.seekg( 0, std::ios::end);
  std::streampos buf_size = is.tellg();
  char *dummy_buffer = new char[(unsigned int)buf_size];
  is.seekg(0, std::ios::beg);
  is.read( dummy_buffer, buf_size);
  unsigned char *src = (unsigned char*)dummy_buffer;
  uint32_t file_length = (uint32_t)buf_size; // 32bits truncation should be ok since DICOM cannot have larger
      than 2Gb image
  /* configure the event callbacks (not required) */
  memset(&event_mgr, 0, sizeof(opj_event_mgr_t));
  event_mgr.error_handler = error_callback;
  event_mgr.warning_handler = warning_callback;
  event_mgr.info_handler = info_callback;
  /* set decoding parameters to default values */
  opj_set_default_decoder_parameters(&parameters);
  // default blindly copied
  parameters.cp_layer=0;
  parameters.cp_reduce= res;
  //   parameters.decod_format=-1;
  //   parameters.cod_format=-1;
  const char jp2magic[] = "\x00\x00\x00\x0C\x6A\x50\x20\x20\x0D\x0A\x87\x0A";
  if( memcmp( src, jp2magic, sizeof(jp2magic) ) == 0 )
    {
    /* JPEG-2000 compressed image data ...  sigh */
    // gdcmData/ELSCINT1_JP2vsJ2K.dcm
    // gdcmData/MAROTECH_CT_JP2Lossy.dcm
    //gdcmWarningMacro( "J2K start like JPEG-2000 compressed image data instead of codestream" );
    parameters.decod_format = 1; //JP2_CFMT;
    //assert(parameters.decod_format == JP2_CFMT);
    }
  else
    {
    /* JPEG-2000 codestream */
    //parameters.decod_format = J2K_CFMT;
    //assert(parameters.decod_format == J2K_CFMT);
    assert( 0 );
    }
  parameters.cod_format = 11; // PGX_DFMT;
  //assert(parameters.cod_format == PGX_DFMT);
  /* get a decoder handle */
    dinfo = opj_create_decompress(CODEC_JP2);
  /* catch events using our callbacks and give a local context */
```

```
  opj_set_event_mgr((opj_common_ptr)dinfo, &event_mgr, NULL);
  /* setup the decoder decoding parameters using user parameters */
  opj_setup_decoder(dinfo, &parameters);
  /* open a byte stream */
  cio = opj_cio_open((opj_common_ptr)dinfo, src, file_length);
  /* decode the stream and fill the image structure */
  image = opj_decode(dinfo, cio);
  if(!image) {
    opj_destroy_decompress(dinfo);
    opj_cio_close(cio);
    //gdcmErrorMacro( "opj_decode failed" );
    return 1;
  }
       opj_cp_t * cp = ((opj_jp2_t*)dinfo->jp2_handle)->j2k->cp;
       opj_tcp_t *tcp = &cp->tcps[0];
       opj_tccp_t *tccp = &tcp->tccps[0];
    /*   std::cout « "\n No of Cols In Image" « image->x1;
std::cout « "\n No of Rows In Image" « image->y1;
std::cout « "\n No of Components in Image" « image->numcomps;
std::cout « "\n No of Resolutions"« tccp->numresolutions « "\n";
*/
        opj_j2k_t* j2k = NULL;
        opj_jp2_t* jp2 = NULL;
        jp2 = (opj_jp2_t*)dinfo->jp2_handle;
        int reversible = jp2->j2k->cp->tcps->tccps->qmfbid;
        //std::  cout « reversible;
       int compno = 0;
       opj_image_comp_t *comp = &image->comps[compno];
       int Dimensions[2];
       Dimensions[0]= comp->w;
       Dimensions[1] = comp->h;
       opj_cio_close(cio);
       unsigned long len = Dimensions[0]*Dimensions[1] * image->numcomps;
       //std::cout « "\nTest" «image->comps[0].factor;
       char *raw = new char[len];
  for (unsigned int compno = 0; compno < (unsigned int)image->numcomps; compno++)
    {
    opj_image_comp_t *comp = &image->comps[compno];
    int w = image->comps[compno].w;
    int h = image->comps[compno].h;
      uint8_t *data8 = (uint8_t*)raw + compno;
      for (int i = 0; i < w * h ; i++)
        {
        int v = image->comps[compno].data[i];
        *data8 = (uint8_t)v;
        data8 += image->numcomps;
        }
    }
  gdcm::Writer w;
  gdcm::File &file = w.GetFile();
  gdcm::DataSet &ds = file.GetDataSet();
  file.GetHeader().SetDataSetTransferSyntax( gdcm::TransferSyntax::ExplicitVRLittleEndian );
  gdcm::UIDGenerator uid;
  gdcm::DataElement de( gdcm::Tag(0x8,0x18) ); // SOP Instance UID
  de.SetVR( gdcm::VR::UI );
  const char *u = uid.Generate();
  de.SetByteValue( u, strlen(u) );
  ds.Insert( de );
  gdcm::DataElement de1( gdcm::Tag(0x8,0x16) );
  de1.SetVR( gdcm::VR::UI );
  gdcm::MediaStorage ms( gdcm::MediaStorage::CTImageStorage );
  de1.SetByteValue( ms.GetString(), strlen(ms.GetString()));
  ds.Insert( de1 );
  const char mystr[] = "MONOCHROME2 ";
  gdcm::DataElement de2( gdcm::Tag(0x28,0x04) );
  //de.SetTag(gdcm::Tag(0x28,0x04));
  de2.SetVR( gdcm::VR::CS );
  de2.SetByteValue(mystr, strlen(mystr));
  ds.Insert( de2 );
   gdcm::Attribute<0x0028,0x0010> row = {image->comps[0].w};
   //row.SetValue(512);
   ds.Insert( row.GetAsDataElement() );
 //  w.SetCheckFileMetaInformation( true );
   gdcm::Attribute<0x0028,0x0011> col = {image->comps[0].h};
   ds.Insert( col.GetAsDataElement() );
   gdcm::Attribute<0x0028,0x0008> Number_Of_Frames = {1};
   ds.Insert( Number_Of_Frames.GetAsDataElement() );
   gdcm::Attribute<0x0028,0x0100> at = {8};
   ds.Insert( at.GetAsDataElement() );
   gdcm::Attribute<0x0028,0x0002> at1 = {image->numcomps};
   ds.Insert( at1.GetAsDataElement() );
```

```
  gdcm::Attribute<0x0028,0x0101> at2 = {8};
   ds.Insert( at2.GetAsDataElement() );
  gdcm::Attribute<0x0028,0x0102> at3 = {7};
  ds.Insert( at3.GetAsDataElement() );
  if (flag == 1)
  {
    for (int i=0; i < No_Of_Resolutions; i++)
    {
     int a = 1;
     int b =1;
     while(a!=((No_Of_Resolutions)-i))
        {
          b = b*2;
          a = a+1;
        }
     uint16_t row = (image->y1)/b;
     uint16_t col = (image->x1)/b;
     //std::cout « row;
     gdcm::Element<gdcm::VR::IS,gdcm::VM::VM1> el2;
     el2.SetValue(i+1);
     gdcm::DataElement rfn = el2.GetAsDataElement();      //ulr --> upper left row
     rfn.SetTag( gdcm::Tag(0x0008,0x1160) );
     gdcm::Element<gdcm::VR::US,gdcm::VM::VM2> el;
     el.SetValue(1,0);
     el.SetValue(1,1);
     gdcm::DataElement ulr = el.GetAsDataElement();      //ulr --> upper left col/row
     ulr.SetTag( gdcm::Tag(0x0048,0x0201) );
     gdcm::Element<gdcm::VR::US,gdcm::VM::VM2> el1;
     el1.SetValue(col,0);
     el1.SetValue(row,1);
     gdcm::DataElement brr = el1.GetAsDataElement();
     brr.SetTag( gdcm::Tag(0x0048,0x0202) );             //brr --> bottom right col/row
    gdcm::Item it;
    gdcm::DataSet &nds = it.GetNestedDataSet();
    nds.Insert( rfn );
    nds.Insert(ulr);
    nds.Insert(brr);
    sq->AddItem(it);
  }
  gdcm::Writer w1;
  gdcm::File &file1 = w1.GetFile();
  gdcm::DataSet &ds1 = file1.GetDataSet();
  file1.GetHeader().SetDataSetTransferSyntax( gdcm::TransferSyntax::ExplicitVRLittleEndian );
  gdcm::UIDGenerator uid1;
  gdcm::DataElement dea( gdcm::Tag(0x8,0x18) ); // SOP Instance UID
  dea.SetVR( gdcm::VR::UI );
  const char *u1 = uid1.Generate();
  dea.SetByteValue( u1, strlen(u1) );
  ds1.Insert( dea );
  gdcm::DataElement deb( gdcm::Tag(0x8,0x16) );
  deb.SetVR( gdcm::VR::UI );
  gdcm::MediaStorage ms1( gdcm::MediaStorage::VLWholeSlideMicroscopyImageStorage );
  deb.SetByteValue( ms1.GetString(), strlen(ms1.GetString()));
  ds1.Insert( deb );
  const char mystr1[] = "MONOCHROME2 ";
  gdcm::DataElement dec( gdcm::Tag(0x28,0x04) );
  //de.SetTag(gdcm::Tag(0x28,0x04));
  dec.SetVR( gdcm::VR::CS );
  dec.SetByteValue(mystr, strlen(mystr1));
  ds1.Insert( dec );
   gdcm::Attribute<0x0028,0x0010> row1 = {image->y1};
   //row.SetValue(512);
   ds1.Insert( row1.GetAsDataElement() );
//  w.SetCheckFileMetaInformation( true );
   gdcm::Attribute<0x0028,0x0011> col1 = {image->x1};
   ds1.Insert( col1.GetAsDataElement() );
   gdcm::Attribute<0x0028,0x0008> Number_Of_Frames1 = {tccp->numresolutions};
   ds1.Insert( Number_Of_Frames1.GetAsDataElement() );
   gdcm::Attribute<0x0028,0x0100> ata = {8};
   ds1.Insert( ata.GetAsDataElement() );
   gdcm::Attribute<0x0028,0x0002> atb = {image->numcomps};
   ds1.Insert( atb.GetAsDataElement() );
  gdcm::Attribute<0x0028,0x0101> atc = {8};
   ds1.Insert( atc.GetAsDataElement() );
  gdcm::Attribute<0x0028,0x0102> atd = {7};
  ds1.Insert( atd.GetAsDataElement() );
   theStreamWriter.SetFile(file1);
  gdcm::DataElement des( gdcm::Tag(0x0048,0x0200) );
  des.SetVR(gdcm::VR::SQ);
  //des.SetVR(gdcm::VM::VM1);
  des.SetValue(*sq);
```

```
  des.SetVLToUndefined();
  ds1.Insert(des);
  if (!theStreamWriter.WriteImageInformation()){
      std::cerr « "unable to write image information" « std::endl;
      return 1; //the CanWrite function should prevent getting here, else,
      //that's a test failure
    }
 }
   theStreamWriter.SetFile(file);
  if (!theStreamWriter.CanWriteFile()){
      delete [] raw;
      std::cout « "Not able to write";
      return 0;//this means that the file was unwritable, period.
      //very similar to a ReadImageInformation failure
    }
else
   std::cout«"\nabletoread";
 // Important to write here
 std::vector<unsigned int> extent = gdcm::ImageHelper::GetDimensionsValue(file);
    unsigned short xmax = extent[0];
    unsigned short ymax = extent[1];
    unsigned short theChunkSize = 4;
    unsigned short ychunk = extent[1]/theChunkSize; //go in chunk sizes of theChunkSize
    unsigned short zmax = extent[2];
    std::cout « "\n"«xmax « "\n" « ymax«"\n"«zmax«"\n" « image->numcomps«"\n";
    if (xmax == 0 || ymax == 0)
      {
      std::cerr « "Image has no size, unable to write zero-sized image." « std::endl;
      return 0;
      }
    int z, y, nexty;
    unsigned long prevLen = 0; //when going through the char buffer, make sure to grab
    //the bytes sequentially.    So, store how far you got in the buffer with each iteration.
    for (z = 0; z < zmax; ++z){
      for (y = 0; y < ymax; y += ychunk){
        nexty = y + ychunk;
        if (nexty > ymax) nexty = ymax;
        theStreamWriter.DefinePixelExtent(0, xmax, y, nexty, z, z+1);
        unsigned long len = theStreamWriter.DefineProperBufferLength();
        std::cout « "\n" «len;
        char* finalBuffer = new char[len];
        memcpy(finalBuffer, &(raw[prevLen]), len);
        std::cout « "\nable to write";
        if (!theStreamWriter.Write(finalBuffer, len)){
          std::cerr « "writing failure:" « "output.dcm" « " at y = " « y « " and z= " « z « std::endl;
          delete [] raw;
          delete [] finalBuffer;
          return 1;
        }
        delete [] finalBuffer;
        prevLen += len;
      }
    }
    delete raw;
  delete[] src;  //FIXME
if(dinfo) {
    opj_destroy_decompress(dinfo);
  }
 opj_image_destroy(image);
 return true;
}
bool Different_Resolution( gdcm::StreamImageWriter & theStreamWriter, const char *filename, int res,
      std::ostream& of)
{
  //std::vector<std::string>::const_iterator it = filenames.begin();
  bool b = true;
  int flag = 1;
 gdcm::SmartPointer<gdcm::SequenceOfItems> sq = new gdcm::SequenceOfItems();
 sq->SetLengthToUndefined();
  for(int i = res-1 ; i>=0; --i)
  {
    b = b && Write_Resolution( theStreamWriter, filename, i, of ,flag,sq,res);
   // b = b && Get_Resolution( theStreamWriter, filename, i, of ,0);
    flag = 0;
  }
  //b = b && Get_Lowest_Resolution( writer, sq, filename, res-1 );
  //b = b && PopulateSingeFile( writer, sq, jpeg, filename2 );
  //image.SetDimension(2,  res )
  return b;
}
int main(int argc, char *argv[])
```

```
{
   if( argc < 4 )
    {
    std::cerr « argv[0] « " input.jp2 output.dcm No.  Of Resolutions " « std::endl;
    return 1;
    }
   const char *filename = argv[1];
  const char *outfilename = argv[2];
  char *resolutions = argv[3];
   int res = int((*resolutions)-48);
    //std::  cout « "\nres"« res;
   gdcm::StreamImageWriter theStreamWriter;
   std::ofstream of;
   of.open( outfilename, std::ios::out | std::ios::binary );
   theStreamWriter.SetStream(of);
   if( !Different_Resolution( theStreamWriter, filename,res,of ) ) return 1;
 uint16_t firstTag1 =  0xfffe;
 uint16_t secondTag1 = 0xe0dd;
 uint32_t thirdTag1 =  0x00000000;
 //uint16_t fourthTag1 = 0xffff;
 const int theBufferSize1 = 2*sizeof(uint16_t)+sizeof(uint32_t);
 char* tmpBuffer2 = new char[theBufferSize1];
 memcpy(&(tmpBuffer2[0]), &firstTag1, sizeof(uint16_t));
 memcpy(&(tmpBuffer2[sizeof(uint16_t)]), &secondTag1, sizeof(uint16_t));
 memcpy(&(tmpBuffer2[2*sizeof(uint16_t)]), &thirdTag1, sizeof(uint32_t));
 //memcpy(&(tmpBuffer2[3*sizeof(uint16_t)]), &fourthTag1, sizeof(uint16_t));
 assert( of && !of.eof() && of.good() );
 of.write(tmpBuffer2, theBufferSize1);
 of.flush();
 assert( of );
return 0;
}
```

12.65 Fake_Image_Using_Stream_Image_Writer.cxx

```
/*=========================================================================

Program:  GDCM (Grassroots DICOM). A DICOM library

Copyright (c) 2006-2011 Mathieu Malaterre
All rights reserved.
See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.

This software is distributed WITHOUT ANY WARRANTY; without even
the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
PURPOSE.  See the above copyright notice for more information.

=========================================================================*/
// This work was realised during the GSOC 2011 by Manoj Alwani
#include "gdcmReader.h"
#include "gdcmMediaStorage.h"
#include "gdcmWriter.h"
#include "gdcmItem.h"
#include "gdcmImageReader.h"
#include "gdcmAttribute.h"
#include "gdcmFile.h"
#include "gdcmTag.h"
#include "gdcmTransferSyntax.h"
#include "gdcmUIDGenerator.h"
#include "gdcmAnonymizer.h"
#include "gdcmStreamImageWriter.h"
#include "gdcmImageHelper.h"
#include "gdcmTrace.h"
int main(int, char *[])
{
  char * buffer = new char[ 256 * 256 *3 ];
  // *p = (uint8_t*)buffer;
  char * p = buffer;
  gdcm::Trace::DebugOn();
  gdcm::Trace::WarningOn();
  for(int row = 0; row < 256; ++row)
  {
    for(int col = 0; col < 256; ++col)
      //for(int b = 0; b < 256; ++b)
      {
        *p++ = 255;
        *p++ = 0;
```

```
        *p++ = 0;
       }
  }
  gdcm::Writer w;
  gdcm::File &file = w.GetFile();
  gdcm::DataSet &ds = file.GetDataSet();
  file.GetHeader().SetDataSetTransferSyntax( gdcm::TransferSyntax::ExplicitVRLittleEndian );
  gdcm::UIDGenerator uid;
  gdcm::DataElement de( gdcm::Tag(0x8,0x18) ); // SOP Instance UID
  de.SetVR( gdcm::VR::UI );
  const char *u = uid.Generate();
  de.SetByteValue( u, strlen(u) );
  ds.Insert( de );
  gdcm::DataElement de1( gdcm::Tag(0x8,0x16) );
  de1.SetVR( gdcm::VR::UI );
  gdcm::MediaStorage ms( gdcm::MediaStorage::VLWholeSlideMicroscopyImageStorage );
  de1.SetByteValue( ms.GetString(), strlen(ms.GetString()));
  ds.Insert( de1 );
  const char mystr[] = "RGB";
  gdcm::DataElement de2( gdcm::Tag(0x28,0x04) );
  //de.SetTag(gdcm::Tag(0x28,0x04));
  de2.SetVR( gdcm::VR::CS );
  de2.SetByteValue(mystr, strlen(mystr));
  ds.Insert( de2 );
   gdcm::Attribute<0x0028,0x0010> row = {256};
   //row.SetValue(512);
   ds.Insert( row.GetAsDataElement() );
 //  w.SetCheckFileMetaInformation( true );
   gdcm::Attribute<0x0028,0x0011> col = {256};
   ds.Insert( col.GetAsDataElement() );
   gdcm::Attribute<0x0028,0x0008> Number_Of_Frames = {1};
   ds.Insert( Number_Of_Frames.GetAsDataElement() );
   gdcm::Attribute<0x0028,0x0100> at = {8};
   ds.Insert( at.GetAsDataElement() );
   gdcm::Attribute<0x0028,0x0002> at1 = {3}; //bits per pixel
   ds.Insert( at1.GetAsDataElement() );
  gdcm::Attribute<0x0028,0x0101> at2 = {8};
   ds.Insert( at2.GetAsDataElement() );
  gdcm::Attribute<0x0028,0x0102> at3 = {7};
   ds.Insert( at3.GetAsDataElement() );
 gdcm::Attribute<0x0028,0x006> at4 = {0};
 ds.Insert( at4.GetAsDataElement() );
 gdcm::Attribute<0x0028,0x0103> at5 = {0};
 ds.Insert( at5.GetAsDataElement() );
 //de.SetTag(gdcm::Tag(0x7fe0,0x0010));
 //ds.Insert(de);
  gdcm::StreamImageWriter theStreamWriter;
gdcm::SmartPointer<gdcm::SequenceOfItems> sq = new gdcm::SequenceOfItems();
 sq->SetLengthToUndefined();
     uint16_t row1 = 256;
     uint16_t col1 = 256;
     //std::cout « row;
     gdcm::Element<gdcm::VR::IS,gdcm::VM::VM1> el2;
     el2.SetValue(1);
     gdcm::DataElement rfn = el2.GetAsDataElement();     //rfn ---> reference frame number
     rfn.SetTag( gdcm::Tag(0x0008,0x1160) );
     gdcm::Element<gdcm::VR::US,gdcm::VM::VM2> el;
     el.SetValue(1,0);
     el.SetValue(1,1);
     gdcm::DataElement ulr = el.GetAsDataElement();     //ulr --> upper left col/row
     ulr.SetTag( gdcm::Tag(0x0048,0x0201) );
     gdcm::Element<gdcm::VR::US,gdcm::VM::VM2> el1;
     el1.SetValue(col1,0);
     el1.SetValue(row1,1);
     gdcm::DataElement brr = el1.GetAsDataElement();
     brr.SetTag( gdcm::Tag(0x0048,0x0202) );            //brr --> bottom right col/row
    gdcm::Item it;
    gdcm::DataSet &nds = it.GetNestedDataSet();
    nds.Insert( rfn );
    nds.Insert(ulr);
    nds.Insert(brr);
    sq->AddItem(it);
  gdcm::DataElement des( gdcm::Tag(0x0048,0x0200) );
  des.SetVR(gdcm::VR::SQ);
  des.SetValue(*sq);
  des.SetVLToUndefined();
  ds.Insert(des);
 theStreamWriter.SetFile(file);
  std::ofstream of;
  of.open( "output.dcm", std::ios::out | std::ios::binary );
  theStreamWriter.SetStream(of);
```

```
if (!theStreamWriter.CanWriteFile()){
      delete [] buffer;
      std::cout « "Not able to write";
      return 0;//this means that the file was unwritable, period.
      //very similar to a ReadImageInformation failure
    }
else
   std::cout«"\nabletoread";
if (!theStreamWriter.WriteImageInformation()){
      std::cerr « "unable to write image information" « std::endl;
      delete [] buffer;
      return 1; //the CanWrite function should prevent getting here, else,
      //that's a test failure
    }
 std::vector<unsigned int> extent =
      gdcm::ImageHelper::GetDimensionsValue(file);
    unsigned short xmax = extent[0];
    unsigned short ymax = extent[1];
    unsigned short theChunkSize = 1;
    unsigned short ychunk = extent[1]/theChunkSize; //go in chunk sizes of theChunkSize
    unsigned short zmax = extent[2];
    std::cout « xmax « ymax « zmax;
    if (xmax == 0 || ymax == 0)
      {
      std::cerr « "Image has no size, unable to write zero-sized image." « std::endl;
      return 0;
      }
    int z, y, nexty;
    unsigned long prevLen = 0; //when going through the char buffer, make sure to grab
    //the bytes sequentially.    So, store how far you got in the buffer with each iteration.
    for (z = 0; z < zmax; ++z){
      for (y = 0; y < ymax; y += ychunk){
        nexty = y + ychunk;
        if (nexty > ymax) nexty = ymax;
        theStreamWriter.DefinePixelExtent(0, xmax, y, nexty, z, z+1);
        unsigned long len = theStreamWriter.DefineProperBufferLength();
        std::cout « "\n" «len;
        char* finalBuffer = new char[len];
        memcpy(finalBuffer, &(buffer[prevLen]), len);
        std::cout « "\nable to write";
        if (!theStreamWriter.Write(finalBuffer, len)){
          std::cerr « "writing failure:" « "output.dcm" « " at y = " « y « " and z= " « z « std::endl;
          delete [] buffer;
          delete [] finalBuffer;
          return 1;
        }
        delete [] finalBuffer;
        prevLen += len;
      }
    }
    delete buffer;
 uint16_t firstTag1 =  0xfffe;
 uint16_t secondTag1 = 0xe0dd;
 uint32_t thirdTag1 =  0x00000000;
 //uint16_t fourthTag1 = 0xffff;
 const int theBufferSize1 = 2*sizeof(uint16_t)+sizeof(uint32_t);
 char* tmpBuffer2 = new char[theBufferSize1];
 memcpy(&(tmpBuffer2[0]), &firstTag1, sizeof(uint16_t));
 memcpy(&(tmpBuffer2[sizeof(uint16_t)]), &secondTag1, sizeof(uint16_t));
 memcpy(&(tmpBuffer2[2*sizeof(uint16_t)]), &thirdTag1, sizeof(uint32_t));
 //memcpy(&(tmpBuffer2[3*sizeof(uint16_t)]), &fourthTag1, sizeof(uint16_t));
 assert( of && !of.eof() && of.good() );
 of.write(tmpBuffer2, theBufferSize1);
 of.flush();
 assert( of );
  return 0;
}
```

12.66 FixBrokenJ2K.cxx

```
/*=========================================================================

Program:  GDCM (Grassroots DICOM). A DICOM library

Copyright (c) 2006-2011 Mathieu Malaterre
All rights reserved.
See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.
```

```
This software is distributed WITHOUT ANY WARRANTY; without even
the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
PURPOSE.  See the above copyright notice for more information.

=========================================================================*/
#include "gdcmReader.h"
#include "gdcmWriter.h"
#include "gdcmImageReader.h"
#include "gdcmSequenceOfFragments.h"
#include "gdcmFile.h"
// http://www.lost.in.ua/dicom/c.dcm
//
// -> BuggyJ2Kvvvua-fixed2-j2k.dcm
/*
* This program attemps to fix a broken J2K/DICOM:
* It contains 2 bugs:
* 1.  The first 8 bytes seems to be random bytes:  remove them
* 2.  YCC is set to 1, while image is grayscale need to set it back to 0
*
* Ref:
* It's a software from http://rentgenprom.ru/ , shipped with universal digital radiographic units
* "ProScan-2000".  The Ukrainian manufacturer developed own digital radiographic unit and it is
* compatible with software from "ProScan-2000".
* Information found in DICOM file is:
*
* (0008,0070) LO [ZAO "Renthenprom" (JSC Rentgenprom) ]             # 36,1 Manufacturer
* (0018,1020) LO [2.13.1.7]                                         # 8,1-n Software Version(s)
*
*/
int main(int argc, char *argv[])
{
  if( argc < 3 )
    {
    std::cerr « argv[0] « " input.dcm output.dcm" « std::endl;
    return 1;
    }
  const char *filename = argv[1];
  const char *outfilename = argv[2];
  gdcm::Reader reader;
  reader.SetFileName( filename );
  if( !reader.Read() )
    {
    return 1;
    }
  gdcm::File &file = reader.GetFile();
  const gdcm::DataElement &pixeldata0 = file.GetDataSet().GetDataElement( gdcm::Tag(0x7fe0,0x0010) );
  const gdcm::SequenceOfFragments *sqf = pixeldata0.GetSequenceOfFragments();
  if( !sqf )
    {
    return 1;
    }
  const gdcm::Fragment &frag0 = sqf->GetFragment(0);
  gdcm::ByteValue *bv = const_cast<gdcm::ByteValue*>(frag0.GetByteValue());
  char *ptr = (char*)bv->GetVoidPointer();
  size_t len = bv->GetLength();
  static const unsigned char sig[] = {0,0,0,0,0x6A,0x70,0x32,0x63};
  if( memcmp(ptr, sig, sizeof(sig)) != 0 )
    {
    std::cerr « "magic random signature not found" « std::endl;
    return 1;
    }
  // Apparently the flag to enable a color transform on 3 color components is set in
  // the COD marker.  (YCC is byte[6] in the COD marker)
  // we need to disable this flag;
  char *cod_marker = ptr + 0x35; /* 0x2d + 0x8 */ // FIXME
  if( cod_marker[0] == (char)0xff && cod_marker[1] == 0x52 )
    {
    // found start of COD
    if( cod_marker[6+2] == 1 )
      {
      // Change in place:
      *((char*)cod_marker + 6+2) = 0;
      // Prepare a new DataElement:
      gdcm::DataElement pixeldata( gdcm::Tag(0x7fe0,0x0010) );
      pixeldata.SetVR( gdcm::VR::OB );
      gdcm::SmartPointer<gdcm::SequenceOfFragments> sq = new gdcm::SequenceOfFragments;
      gdcm::Fragment frag;
      // remove 8 first bytes:
      frag.SetByteValue( ptr + 8, (uint32_t)(len - 8) );
      sq->AddFragment( frag );
```

```
      pixeldata.SetValue( *sq );
      file.GetDataSet().Replace( pixeldata );
      }
    else
      {
      return 1;
      }
    }
  else
    {
    std::cerr « "COD not found" « (int)cod_marker[0] « std::endl;
    return 1;
    }
  gdcm::Writer writer;
  writer.SetFile( reader.GetFile() );
  writer.SetFileName( outfilename );
  writer.CheckFileMetaInformationOff();
  if( !writer.Write() )
    {
    std::cerr « "Could not write" « std::endl;
    }
  // paranoid check:
  gdcm::ImageReader ireader;
  ireader.SetFileName( outfilename );
  if( !ireader.Read() )
    {
    std::cerr « "file written is still not valid, please report" « std::endl;
    return 1;
    }
  return 0;
}
```

12.67 FixJAIBugJPEGLS.cxx

```
/*=========================================================================

Program:  GDCM (Grassroots DICOM). A DICOM library

Copyright (c) 2006-2011 Mathieu Malaterre
All rights reserved.
See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.

This software is distributed WITHOUT ANY WARRANTY; without even
the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
PURPOSE.  See the above copyright notice for more information.

=========================================================================*/
#include "gdcmReader.h"
#include "gdcmWriter.h"
#include "gdcmImageReader.h"
#include <fstream>
#include "gdcm_charls.h"
/*
* This small example should show how one can handle the famous JAI-JPEGLS bug
* It will take in as invalid DICOM/JAI-JPEG-LS and write out as Explicit Little
* Endian.  One can use `gdcmconv --jpegls` to recompress properly
*
* References:
* http://charls.codeplex.com/discussions/230307?ProjectName=charls
* http://charls.codeplex.com/workitem/7297
* http://www.dcm4che.org/jira/browse/DCM-442
* http://www.dcm4che.org/jira/browse/DCMEE-1144
* http://java.net/jira/browse/JAI_IMAGEIO_CORE-183
*
* Explanation of the issue:
*
* Seems, the error is in the calculation of the default values for thresholds T1,
* T2, T3, in particular min(MAXVAL, 4095) is not applied in
*
* FACTOR = (min(MAXVAL, 4095) + 128)/256
*
* as specified in http://www.itu.int/rec/T-REC-T.87-199806-I/en .
*
*/
int main(int argc, char *argv[])
{
  if( argc < 3 )
```

```
    {
    std::cerr « argv[0] « " input.dcm output.dcm" « std::endl;
    return 1;
    }
  const char *filename = argv[1];
  const char *outfilename = argv[2];
  gdcm::FileMetaInformation::SetSourceApplicationEntityTitle( "FixJAIBugJPEGLS" );
  gdcm::ImageReader reader;
  reader.SetFileName( filename );
  if( !reader.Read() )
    {
    return 1;
    }
  gdcm::Image &image = reader.GetImage();
  //unsigned long len = image.GetBufferLength();
  const gdcm::DataElement & in =
    reader.GetFile().GetDataSet().GetDataElement( gdcm::Tag(0x7fe0,0x0010) );
  const gdcm::SequenceOfFragments *sf = in.GetSequenceOfFragments();
  if( !sf )
    {
    std::cerr « "No pixel data (or not encapsulated)" « std::endl;
    return 1;
    }
  const unsigned int *dims = image.GetDimensions();
  if ( sf->GetNumberOfFragments() != dims[2] )
    {
    std::cerr « "Unsupported" « std::endl;
    return 1;
    }
//  unsigned long totalLen = sf->ComputeByteLength();
  std::vector<unsigned char> rgbyteOutall;
  for(unsigned int i = 0; i < sf->GetNumberOfFragments(); ++i)
    {
    const gdcm::Fragment &frag = sf->GetFragment(i);
    if( frag.IsEmpty() ) return 1;
    const gdcm::ByteValue *bv = frag.GetByteValue();
    if( !bv ) return 1;
    unsigned long totalLen = bv->GetLength();
    std::vector<char> vbuffer;
    vbuffer.resize( totalLen );
    char *buffer = &vbuffer[0];
    bv->GetBuffer(buffer, totalLen);
    const unsigned char* pbyteCompressed0 = (const unsigned char*)buffer;
    while( totalLen > 0 && pbyteCompressed0[totalLen-1] != 0xd9 )
      {
      totalLen--;
      }
    JlsParameters metadata;
    char errorMsg[256+1]={'\0'};
    if (JpegLsReadHeader(buffer, totalLen, &metadata, errorMsg) != charls::ApiResult::OK)
      {
      std::cerr « "Cant parse jpegls:  " « errorMsg « std::endl;
      return 1;
      }
    std::cout « metadata.width « std::endl;
    std::cout « metadata.height « std::endl;
    std::cout « metadata.bitsPerSample « std::endl;
    gdcm::PixelFormat const & pf = image.GetPixelFormat();
    std::cout « pf « std::endl;
    // http://charls.codeplex.com/discussions/230307?ProjectName=charls
    unsigned char marker_lse_13[] = {
      0xFF, 0xF8, 0x00, 0x0D,
      0x01,
      0x1F, 0xFF,
      0x00, 0x22,  // T1 = 34
      0x00, 0x83,  // T2 = 131
      0x02, 0x24,  // T3 = 548
      0x00, 0x40
    };
    unsigned char marker_lse_14[] = {
      0xFF, 0xF8, 0x00, 0x0D,
      0x01,
      0x3F, 0xFF,
      0x00, 0x42, // T1 = 66
      0x01, 0x03, // T2 = 259
      0x04, 0x44, // T3 = 1092
      0x00, 0x40
    };
    unsigned char marker_lse_15[] = {
      0xFF, 0xF8, 0x00, 0x0D,
      0x01,
```

```
      0x7F, 0xFF,
      0x00, 0x82, // T1 = 130
      0x02, 0x03, // T2 = 515
      0x08, 0x84, // T3 = 2180
      0x00, 0x40
    };
    unsigned char marker_lse_16[] = {
      0xFF, 0xF8, 0x00, 0x0D,
      0x01,
      0xFF, 0xFF,
      0x01, 0x02, // T1 = 258
      0x04, 0x03, // T2 = 1027
      0x11, 0x04, // T3 = 4356
      0x00, 0x40
    };
    const unsigned char *marker_lse = nullptr;
    switch( metadata.bitsPerSample )
      {
    case 13:
      marker_lse = marker_lse_13;
      break;
    case 14:
      marker_lse = marker_lse_14;
      break;
    case 15:
      marker_lse = marker_lse_15;
      break;
    case 16:
      marker_lse = marker_lse_16;
      break;
      }
    if( !marker_lse )
      {
      std::cerr « "Cant handle:  " « metadata.bitsPerSample « std::endl;
      return 1;
      }
    // FIXME: One should recompute the value for 0x0F
    vbuffer.insert (vbuffer.begin() + 0x0F, marker_lse, marker_lse+15);
#if 0
    std::ofstream of( "/tmp/d.jls", std::ios::binary );
    of.write( &vbuffer[0], vbuffer.size() );
    of.close();
#endif
    const char *pbyteCompressed = &vbuffer[0];
    size_t cbyteCompressed = vbuffer.size(); // updated legnth
    JlsParameters params;
    JpegLsReadHeader(pbyteCompressed, cbyteCompressed, &params, nullptr);
    std::vector<unsigned char> rgbyteOut;
    //rgbyteOut.resize( image.GetBufferLength() );
    rgbyteOut.resize(params.height *params.width * ((params.bitsPerSample + 7)
        / 8) * params.components);
    CharlsApiResultType result =
      JpegLsDecode(&rgbyteOut[0], rgbyteOut.size(), pbyteCompressed, cbyteCompressed, &params, errorMsg );
    if (result != charls::ApiResult::OK)
      {
      std::cerr « "Could not patch JAI-JPEGLS: " « errorMsg « std::endl;
      return 1;
      }
    rgbyteOutall.insert( rgbyteOutall.end(), rgbyteOut.begin(), rgbyteOut.end() );
    }
  gdcm::DataElement pixeldata( gdcm::Tag(0x7fe0,0x0010) );
  pixeldata.SetVR( gdcm::VR::OW );
  pixeldata.SetByteValue( (char*)&rgbyteOutall[0], (uint32_t)rgbyteOutall.size() );
  // Add the pixel data element
  reader.GetFile().GetDataSet().Replace( pixeldata );
  reader.GetFile().GetHeader().SetDataSetTransferSyntax(
    gdcm::TransferSyntax::ExplicitVRLittleEndian);
  gdcm::Writer writer;
  writer.SetFileName( outfilename );
  writer.SetFile( reader.GetFile() );
  writer.Write();
  std::cout « "Success !" « std::endl;
  return 0;
}
```

12.68 FixOrientation.cxx

```
/*=========================================================================
```

```
Program:  GDCM (Grassroots DICOM). A DICOM library

Copyright (c) 2006-2011 Mathieu Malaterre
All rights reserved.
See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.

This software is distributed WITHOUT ANY WARRANTY; without even
the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
PURPOSE.  See the above copyright notice for more information.

=========================================================================*/
#include "gdcmReader.h"
#include "gdcmWriter.h"
#include "gdcmFile.h"
#include "gdcmOrientation.h"
#include "gdcmAttribute.h"
// Very simple orientation changer, fix invalid dataset
int main(int argc, char* argv[] )
{
  // assume AXIAL input for now
  if( argc < 3 )
    {
    std::cerr « argv[0] « " input.dcm output.dcm" « std::endl;
    return 1;
    }
  const char *filename = argv[1];
  const char *outfilename = argv[2];
  gdcm::Reader reader;
  reader.SetFileName( filename );
  if (!  reader.Read() )
    {
    return 1;
    }
  const double axial[] = { 1,0,0, 0,1,0 };
  (void)axial;
  const double coronal[] = { 0,0,1, 1,0,0 };
  (void)coronal;
  const double sagittal[] = { 0,1,0, 0,0,1 };
  (void)sagittal;
  gdcm::Attribute<0x0020,0x0032> at1; // IPP
  (void)at1;
  gdcm::Attribute<0x0020,0x0037> at2; // IOP
  (void)at2;
  gdcm::File & f = reader.GetFile();
  gdcm::DataSet & ds = f.GetDataSet();
  at1.SetFromDataSet( ds );
#if 0
  at2.SetFromDataSet( ds );
  const double * iop = at2.GetValues();
  if( !std::equal(iop, iop + 6, axial ) )
  {
    gdcm::Orientation::OrientationType type = gdcm::Orientation::GetType ( iop );
    std::cerr « "Wrong orientation:  " « gdcm::Orientation::GetLabel( type ) « std::endl;
    return 1;
  }
  at2.SetValues( sagittal );
  ds.Replace( at2.GetAsDataElement() );
#endif
  // for sagittal:  swap element 0 & 2
  const double tmp0 = at1.GetValue(0);
  const double tmp2 = at1.GetValue(2);
  (void)tmp2;
  //at1.SetValue(tmp2, 0);
  //at1.SetValue(tmp0, 2);
  at1.SetValue( - tmp0 );
  ds.Replace( at1.GetAsDataElement() );
  gdcm::Writer writer;
  writer.SetFile( f );
  writer.SetFileName( outfilename );
  if ( !writer.Write() )
    {
    return 1;
    }
  return 0;
}
```

12.69 GenAllVR.cxx

```
/*=========================================================================

Program:  GDCM (Grassroots DICOM). A DICOM library

Copyright (c) 2006-2011 Mathieu Malaterre
All rights reserved.
See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.

This software is distributed WITHOUT ANY WARRANTY; without even
the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
PURPOSE.  See the above copyright notice for more information.

=========================================================================*/
#include "gdcmReader.h"
#include "gdcmGlobal.h"
#include "gdcmDummyValueGenerator.h"
#include "gdcmMediaStorage.h"
#include "gdcmWriter.h"
#include "gdcmItem.h"
#include "gdcmImageReader.h"
#include "gdcmSequenceOfItems.h"
#include "gdcmFile.h"
#include "gdcmTag.h"
#include "gdcmDict.h"
#include "gdcmDictEntry.h"
#include "gdcmDicts.h"
#include "gdcmTransferSyntax.h"
#include "gdcmUIDGenerator.h"
#include "gdcmFileExplicitFilter.h"
#include <cstdlib>
#include <cstring>
gdcm::Tag FindTagFromVR(gdcm::Dict const &dict, gdcm::VR const &vr)
{
  using gdcm::Dict;
  Dict::ConstIterator beg = dict.Begin();
  Dict::ConstIterator end = dict.End();
  Dict::ConstIterator it;
  for( it = beg; it != end; ++it)
    {
    const gdcm::Tag &t = it->first;
    const gdcm::DictEntry &de = it->second;
    const gdcm::VR &vr_de = de.GetVR();
    if( vr == vr_de && !de.GetRetired() && t.GetGroup() >= 0x8 )
      {
      return t;
      }
    }
  return gdcm::Tag(0xffff,0xffff);
}
struct rnd_gen {
  rnd_gen(char const* r = "abcdefghijklmnopqrstuvwxyz0123456789")
    :  range(r), len(std::strlen(r)) { }
  char operator ()()const {
    return range[static_cast<std::size_t>(std::rand() * (1.0 / ((double)RAND_MAX + 1.0 )) * (double)len)];
  }
private:
  char const* range;
  std::size_t len;
};
/*
*/
int main(int argc, char *argv[])
{
  if( argc < 2 )
    {
    std::cerr « argv[0] « " output.dcm" « std::endl;
    return 1;
    }
  const char *outfilename = argv[1];
  static const gdcm::Global &g = gdcm::Global::GetInstance();
  static const gdcm::Dicts &dicts = g.GetDicts();
  static const gdcm::Dict &pubdict = dicts.GetPublicDict();
  using gdcm::VR;
  using gdcm::Tag;
  gdcm::Writer w;
  gdcm::File &f = w.GetFile();
  gdcm::DataSet &ds = f.GetDataSet();
  gdcm::FileExplicitFilter fef;
```

```
  //fef.SetChangePrivateTags( true );
  fef.SetFile( w.GetFile() );
  if( !fef.Change() )
    {
    std::cerr « "Failed to change" « std::endl;
    return 1;
    }
  gdcm::SmartPointer<gdcm::SequenceOfItems> sq = new gdcm::SequenceOfItems();
  sq->SetLengthToUndefined();
//  gdcm::DummyValueGenerator dvg;
  const std::size_t len = 10;
  char ss[len+1];
  ss[len] = '\0';
  const char owner_str[] = "GDCM CONFORMANCE TESTS";
  gdcm::DataElement owner( gdcm::Tag(0x4d4d, 0x10) );
  owner.SetByteValue(owner_str, (uint32_t)strlen(owner_str));
  owner.SetVR( gdcm::VR::LO );
  // Create an item
  gdcm::Item it;
  it.SetVLToUndefined();
  gdcm::DataSet &nds = it.GetNestedDataSet();
  //    nds.Insert(owner);
  //    nds.Insert(de);
  // Insert sequence into data set
  gdcm::DataElement des( gdcm::Tag(0x4d4d,0x1001) );
  des.SetVR(gdcm::VR::SQ);
  des.SetValue(*sq);
  des.SetVLToUndefined();
  ds.Insert(owner);
  ds.Insert(des);
  // avoid INVALID = 0
  for(int i = 1; i < 27; ++i)
    {
    VR vr = (VR::VRType)(1LL « i);
    Tag t = FindTagFromVR( pubdict, vr );
    if( vr != VR::UN && vr != VR::SQ )
      {
      assert( t != Tag(0xffff,0xffff) );
      gdcm::DataElement de( t );
      std::generate_n(ss, len, rnd_gen());
      de.SetVR( vr );
      de.SetByteValue( ss, (uint32_t)std::strlen( ss ) );
      nds.Insert( de );
      }
    }
  sq->AddItem(it);
  // Make sure to override any UID stuff
  gdcm::UIDGenerator uid;
  gdcm::DataElement de( Tag(0x8,0x18) ); // SOP Instance UID
  de.SetVR( VR::UI );
  const char *u = uid.Generate();
  de.SetByteValue( u, (uint32_t)strlen(u) );
  ds.Insert( de );
  de.SetTag( Tag(0x8,0x16) ); // SOP Class UID
  de.SetVR( VR::UI );
  gdcm::MediaStorage ms( gdcm::MediaStorage::RawDataStorage );
  de.SetByteValue( ms.GetString(), (uint32_t)strlen(ms.GetString()));
  ds.Insert( de );
  gdcm::FileMetaInformation &fmi = f.GetHeader();
  //fmi.SetDataSetTransferSyntax( gdcm::TransferSyntax::ImplicitVRLittleEndian );
  fmi.SetDataSetTransferSyntax( gdcm::TransferSyntax::ExplicitVRLittleEndian );
  w.SetCheckFileMetaInformation( true );
  w.SetFileName( outfilename );
  if (!w.Write() )
    {
    return 1;
    }
  return 0;
}
```

12.70 GenFakeIdentifyFile.cxx

```
/*=========================================================================

Program:  GDCM (Grassroots DICOM). A DICOM library

Copyright (c) 2006-2011 Mathieu Malaterre
```

```
All rights reserved.
See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.

This software is distributed WITHOUT ANY WARRANTY; without even
the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
PURPOSE.  See the above copyright notice for more information.

=========================================================================*/
#include "gdcmReader.h"
#include "gdcmGlobal.h"
#include "gdcmDummyValueGenerator.h"
#include "gdcmMediaStorage.h"
#include "gdcmWriter.h"
#include "gdcmItem.h"
#include "gdcmImageReader.h"
#include "gdcmSequenceOfItems.h"
#include "gdcmAttribute.h"
#include "gdcmFile.h"
#include "gdcmTag.h"
#include "gdcmDict.h"
#include "gdcmDictEntry.h"
#include "gdcmDicts.h"
#include "gdcmTransferSyntax.h"
#include "gdcmUIDGenerator.h"
#include "gdcmAnonymizer.h"
#include <cstdlib>
#include <cstring>
gdcm::DataElement CreateFakeElement(gdcm::Tag const &tag, bool toremove)
{
  static const gdcm::Global &g = gdcm::Global::GetInstance();
  static const gdcm::Dicts &dicts = g.GetDicts();
  static const gdcm::Dict &pubdict = dicts.GetPublicDict();
  static size_t countglobal = 0;
  static std::vector<gdcm::Tag> balcptags =
    gdcm::Anonymizer::GetBasicApplicationLevelConfidentialityProfileAttributes();
  size_t count = countglobal % balcptags.size();
  const gdcm::DictEntry &dictentry = pubdict.GetDictEntry(tag);
  gdcm::DataElement de;
  de.SetTag( tag );
  using gdcm::VR;
  const VR &vr = dictentry.GetVR();
  //if( vr != VR::INVALID )
  if( vr.IsDual() )
    {
    if( vr == VR::US_SS )
      {
      de.SetVR( VR::US );
      }
    else if( vr == VR::US_SS_OW )
      {
      de.SetVR( VR::OW );
      }
    else if( vr == VR::OB_OW )
      {
      de.SetVR( VR::OB );
      }
    }
  else
    {
    de.SetVR( vr );
    }
  const char str[] = "BasicApplicationLevelConfidentialityProfileAttributes";
  const char safe[] = "This is safe to keep";
  if( de.GetVR() != VR::SQ )
    {
    if( toremove )
      de.SetByteValue( str, (uint32_t)strlen(str) );
    else
      de.SetByteValue( safe, (uint32_t)strlen(safe) );
    }
  else
    {
    // Create an item
    gdcm::Item it;
    it.SetVLToUndefined();
    gdcm::DataSet &nds = it.GetNestedDataSet();
    // Insert sequence into data set
    assert(de.GetVR() == gdcm::VR::SQ );
    gdcm::SmartPointer<gdcm::SequenceOfItems> sq = new gdcm::SequenceOfItems();
    sq->SetLengthToUndefined();
    de.SetValue(*sq);
```

```
    de.SetVLToUndefined();
    //ds.Insert(de);
    if( !toremove )
      {
      nds.Insert( CreateFakeElement( balcptags[count], true ) );
      countglobal++;
      }
    else
      {
      gdcm::Attribute<0x0008,0x0000> at1 = { 0 }; // This element has no reason to be 'anonymized'...
      nds.Insert( at1.GetAsDataElement() );
      gdcm::Attribute<0x000a,0x0000> at2 = { 0 };
      nds.Insert( at2.GetAsDataElement() );
      }
    sq->AddItem(it);
    }
  return de;
}
/*
*/
int main(int argc, char *argv[])
{
  if( argc < 2 )
    {
    std::cerr « argv[0] « " output.dcm" « std::endl;
    return 1;
    }
  using gdcm::Tag;
  using gdcm::VR;
  const char *outfilename = argv[1];
  std::vector<gdcm::Tag> balcptags =
    gdcm::Anonymizer::GetBasicApplicationLevelConfidentialityProfileAttributes();
  gdcm::Writer w;
  gdcm::File &f = w.GetFile();
  gdcm::DataSet &ds = f.GetDataSet();
  // Add attribute that need to be anonymized:
  std::vector<gdcm::Tag>::const_iterator it = balcptags.begin();
  for(; it != balcptags.end(); ++it)
    {
    ds.Insert( CreateFakeElement( *it, true ) );
    }
  // Add attribute that do NOT need to be anonymized:
  static const gdcm::Global &g = gdcm::Global::GetInstance();
  static const gdcm::Dicts &dicts = g.GetDicts();
  static const gdcm::Dict &pubdict = dicts.GetPublicDict();
  using gdcm::Dict;
  Dict::ConstIterator dictit = pubdict.Begin();
  for(; dictit != pubdict.End(); ++dictit)
    {
    const gdcm::Tag &dicttag = dictit->first;
    if( dicttag == Tag(0x6e65,0x6146) ) break;
    //const gdcm::DictEntry &dictentry = dictit->second;
    ds.Insert( CreateFakeElement( dicttag, false ) );
    }
  ds.Remove( gdcm::Tag(0x400,0x500) );
  ds.Remove( gdcm::Tag(0x12,0x62) );
  ds.Remove( gdcm::Tag(0x12,0x63) );
  // Make sure to override any UID stuff
  gdcm::UIDGenerator uid;
  gdcm::DataElement de( Tag(0x8,0x18) ); // SOP Instance UID
  de.SetVR( VR::UI );
  const char *u = uid.Generate();
  de.SetByteValue( u, (uint32_t)strlen(u) );
  //ds.Insert( de );
  ds.Replace( de );
  de.SetTag( Tag(0x8,0x16) ); // SOP Class UID
  de.SetVR( VR::UI );
  gdcm::MediaStorage ms( gdcm::MediaStorage::RawDataStorage );
  de.SetByteValue( ms.GetString(), (uint32_t)strlen(ms.GetString()));
  ds.Replace( de ); // replace !
  gdcm::FileMetaInformation &fmi = f.GetHeader();
  //fmi.SetDataSetTransferSyntax( gdcm::TransferSyntax::ImplicitVRLittleEndian );
  fmi.SetDataSetTransferSyntax( gdcm::TransferSyntax::ExplicitVRLittleEndian );
  w.SetCheckFileMetaInformation( true );
  w.SetFileName( outfilename );
  if (!w.Write() )
    {
    return 1;
    }
  return 0;
}
```

12.71 GenLongSeqs.cxx

```
/*=========================================================================

Program:  GDCM (Grassroots DICOM). A DICOM library

Copyright (c) 2006-2011 Mathieu Malaterre
All rights reserved.
See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.

This software is distributed WITHOUT ANY WARRANTY; without even
the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
PURPOSE.  See the above copyright notice for more information.

=========================================================================*/
#include "gdcmReader.h"
#include "gdcmWriter.h"
#include "gdcmItem.h"
#include "gdcmImageReader.h"
#include "gdcmSequenceOfItems.h"
#include "gdcmFile.h"
#include "gdcmTag.h"
/*
* This example is used to generate the file:
*
*
* There is a flaw in the DICOM design were it is assumed that Sequence can be
* either represented as undefined length or defined length.  This should work
* in most case, but the undefined length is a little more general and can
* store sequence of items that a defined length cannot.
* We need to make sure that we can store numerous Item in a SQ
*
* Warning:  do not try to compute the group length elements !
* Warning:  You may need a 64bits machine for this example to work.
*/
int main(int argc, char *argv[])
{
  if( argc < 3 )
    {
    std::cerr « argv[0] « " input.dcm output.dcm" « std::endl;
    return 1;
    }
  const char *filename = argv[1];
  const char *outfilename = argv[2];
  gdcm::Reader reader;
  reader.SetFileName( filename );
  if( !reader.Read() )
    {
    return 1;
    }
  gdcm::File &file = reader.GetFile();
  gdcm::DataSet &ds = file.GetDataSet();
  // Create a Sequence
  gdcm::SmartPointer<gdcm::SequenceOfItems> sq = new gdcm::SequenceOfItems();
  sq->SetLengthToUndefined();
  const char owner_str[] = "GDCM CONFORMANCE TESTS";
  gdcm::DataElement owner( gdcm::Tag(0x4d4d, 0x10) );
  owner.SetByteValue(owner_str, (uint32_t)strlen(owner_str));
  owner.SetVR( gdcm::VR::LO );
  size_t nitems = 1000;
  nitems += std::numeric_limits<uint32_t>::max();
  for(unsigned int idx = 0; idx < nitems; ++idx)
    {
    // Create a dataelement
    //gdcm::DataElement de( gdcm::Tag(0x4d4d, 0x1002) );
    //de.SetByteValue(ptr, ptr_len);
    //de.SetVR( gdcm::VR::OB );
    // Create an item
    gdcm::Item it;
    it.SetVLToUndefined();
    //gdcm::DataSet &nds = it.GetNestedDataSet();
    //nds.Insert(owner);
    //nds.Insert(de);
    sq->AddItem(it);
    }
  // Insert sequence into data set
  gdcm::DataElement des( gdcm::Tag(0x4d4d,0x1001) );
  des.SetVR(gdcm::VR::SQ);
  des.SetValue(*sq);
  des.SetVLToUndefined();
```

```
  ds.Insert(owner);
  ds.Insert(des);
  gdcm::Writer w;
  w.SetFile( file );
  //w.SetCheckFileMetaInformation( true );
  w.SetFileName( outfilename );
  if (!w.Write() )
    {
    return 1;
    }
  return 0;
}
```

12.72 GenSeqs.cxx

```
/*=========================================================================

Program:  GDCM (Grassroots DICOM). A DICOM library

Copyright (c) 2006-2011 Mathieu Malaterre
All rights reserved.
See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.

This software is distributed WITHOUT ANY WARRANTY; without even
the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
PURPOSE.  See the above copyright notice for more information.

=========================================================================*/
#include "gdcmReader.h"
#include "gdcmWriter.h"
#include "gdcmItem.h"
#include "gdcmImageReader.h"
#include "gdcmSequenceOfItems.h"
#include "gdcmFile.h"
#include "gdcmTag.h"
/*
* This example is used to generate the file:
*
* gdcmConformanceTests/SequenceWithUndefinedLengthNotConvertibleToDefinedLength.dcm
*
* There is a flaw in the DICOM design were it is assumed that Sequence can be
* either represented as undefined length or defined length.  This should work
* in most case, but the undefined length is a little more general and can
* store sequence of items that a defined length cannot.
* Deflated syntax was used in this case since this synthetic example can be
* nicely compressed using this transfer syntax.
*
* Warning:  do not try to compute the group length elements !
* Warning:  You may need a 64bits machine for this example to work.
*/
int main(int argc, char *argv[])
{
  if( argc < 3 )
    {
    std::cerr « argv[0] « " input.dcm output.dcm" « std::endl;
    return 1;
    }
  const char *filename = argv[1];
  const char *outfilename = argv[2];
  gdcm::Reader reader;
  reader.SetFileName( filename );
  if( !reader.Read() )
    {
    return 1;
    }
  gdcm::File &file = reader.GetFile();
  gdcm::DataSet &ds = file.GetDataSet();
  //const unsigned int nitems = 1000;
  const unsigned int ptr_len = 42; /*94967296 / nitems; */
  //assert( ptr_len == 42949672 );
  char *ptr = new char[ptr_len];
  memset(ptr,0,ptr_len);
  // Create a Sequence
  gdcm::SmartPointer<gdcm::SequenceOfItems> sq = new gdcm::SequenceOfItems();
  sq->SetLengthToUndefined();
  const char owner_str[] = "GDCM CONFORMANCE TESTS";
  gdcm::DataElement owner( gdcm::Tag(0x4d4d, 0x10) );
```

```
  owner.SetByteValue(owner_str, (uint32_t)strlen(owner_str));
  owner.SetVR( gdcm::VR::LO );
  for(unsigned int idx = 0; idx < 10/* nitems*/; ++idx)
    {
    // Create a dataelement
    gdcm::DataElement de( gdcm::Tag(0x4d4d, 0x1002) );
    de.SetByteValue(ptr, ptr_len);
    de.SetVR( gdcm::VR::OB );
    // Create an item
    gdcm::Item it;
    it.SetVLToUndefined();
    gdcm::DataSet &nds = it.GetNestedDataSet();
    nds.Insert(owner);
    nds.Insert(de);
    sq->AddItem(it);
    }
  // Insert sequence into data set
  gdcm::DataElement des( gdcm::Tag(0x4d4d,0x1001) );
  des.SetVR(gdcm::VR::SQ);
  des.SetValue(*sq);
  des.SetVLToUndefined();
  ds.Insert(owner);
  ds.Insert(des);
  gdcm::Writer w;
  w.SetFile( file );
  //w.SetCheckFileMetaInformation( true );
  w.SetFileName( outfilename );
  if (!w.Write() )
    {
    return 1;
    }
  return 0;
}
```

12.73 GenerateStandardSOPClasses.cxx

```
/*=========================================================================

Program:  GDCM (Grassroots DICOM). A DICOM library

Copyright (c) 2006-2011 Mathieu Malaterre
All rights reserved.
See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.

This software is distributed WITHOUT ANY WARRANTY; without even
the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
PURPOSE.  See the above copyright notice for more information.

=========================================================================*/
/*
*/
#include "gdcmDefs.h"
#include "gdcmUIDs.h"
#include "gdcmGlobal.h"
#include "gdcmMediaStorage.h"
#include "gdcmSOPClassUIDToIOD.h"
int main(int , char *[])
{
  using gdcm::MediaStorage;
  gdcm::Global& g = gdcm::Global::GetInstance();
  if( !g.LoadResourcesFiles() )
    {
    std::cerr « "Could not LoadResourcesFiles" « std::endl;
    return 1;
    }
  const gdcm::Defs &defs = g.GetDefs();
  int ret = 0;
  //std::cout « "Table B.5-1 STANDARD SOP CLASSES" « std::endl;
  std::cout « "SOP Class Name,SOP Class UID,IOD Specification (defined in PS 3.3)" « std::endl;
  gdcm::MediaStorage::MSType mst;
  for ( mst = gdcm::MediaStorage::MediaStorageDirectoryStorage; mst < gdcm::MediaStorage::MS_END;
    mst = (gdcm::MediaStorage::MSType)(mst + 1) )
    {
    const char *iod = defs.GetIODNameFromMediaStorage(mst);
    gdcm::UIDs uid;
    uid.SetFromUID( gdcm::MediaStorage::GetMSString(mst) /*mst.GetString()*/ );
    if( iod )
```

```
      {
      const char *iod_ref = gdcm::SOPClassUIDToIOD::GetIOD(uid);
      if( iod_ref )
        {
        std::string iod_ref_str = iod_ref;
        //iod_ref_str += " IOD Modules";
        //if( iod_ref_str != iod )
          {
          //std::cout « "UID: " « uid « "    ";
          std::cout « '"' « uid.GetName() « '"' «"," « '"' «uid.GetString() « '"' «"," « '"' «iod « '"' «
      std::endl;
          //std::cout « "Incompatible IODs:  [" « iod « "] versus ref= [" « iod_ref_str « "]" « std::endl;
          ++ret;
          }
        }
      }
    }
  return 0;
}
```

12.74 GetJPEGSamplePrecision.cxx

```
/*=========================================================================

Program:  GDCM (Grassroots DICOM). A DICOM library

Copyright (c) 2006-2011 Mathieu Malaterre
All rights reserved.
See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.

This software is distributed WITHOUT ANY WARRANTY; without even
the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
PURPOSE.  See the above copyright notice for more information.

=========================================================================*/
/*
* This example is a little helper to detect the famous SIEMENS JPEG lossless compressed image
* where DICOM is declared as:
*
*  (0028,0100) US 16                                       # 2,1 Bits Allocated
*  (0028,0101) US 12                                       # 2,1 Bits Stored
*  (0028,0102) US 11                                       # 2,1 High Bit
*  (0028,0103) US 0                                        # 2,1 Pixel Representation
*
* But where JPEG is:
*
*        JPEG_SOF_Parameters:
*                 SamplePrecision = 16
*                 nLines = 192
*                 nSamplesPerLine = 192
*                 nComponentsInFrame = 1
*                 component 0
*                         ComponentIdentifier = 1
*                         HorizontalSamplingFactor = 1
*                         VerticalSamplingFactor = 1
*                         QuantizationTableDestinationSelector = 0
*
*
* This case is valid.  One simply has to use the 16bits jpeg decoder to decode the 12bits stored image.
* This used to be an issue in GDCM 1.2.x (fixed in GDCM 1.2.5)
*
* The main return 0 (no error) when the file read is actually a potential problem.  At the end of the main
* function, the jpeg stream is stored in the filename specified as second argument
*/
#include "gdcmImageReader.h"
#include "gdcmSequenceOfFragments.h"
#include "gdcmJPEGCodec.h"
#include <iostream>
#include <fstream>
int main(int argc, char *argv[])
{
  if( argc < 3 )
    {
    std::cerr « argv[0] « " input.dcm output.jpg" « std::endl;
    return 1;
    }
  const char *filename = argv[1];
```

```
  const char *outfilename = argv[2];
  gdcm::ImageReader reader;
  reader.SetFileName( filename );
  if( !reader.Read() )
    {
    std::cerr « "Could not read:  " « filename « std::endl;
    return 1;
    }
  // The output of gdcm::Reader is a gdcm::File
  const gdcm::File &file = reader.GetFile();
  const gdcm::Image &image = reader.GetImage();
  const gdcm::TransferSyntax &ts = file.GetHeader().GetDataSetTransferSyntax();
  if( ts != gdcm::TransferSyntax::JPEGLosslessProcess14 && ts != gdcm::TransferSyntax::JPEGLosslessProcess14_1 )
    {
    std::cerr « "Input is not a lossless JPEG" « std::endl;
    return 1;
    }
  // the dataset is the the set of element we are interested in:
  const gdcm::DataSet &ds = file.GetDataSet();
  const gdcm::Tag rawTag(0x7fe0, 0x0010); // Default to Pixel Data
  const gdcm::DataElement& pdde = ds.GetDataElement( rawTag );
  const gdcm::SequenceOfFragments *sf = pdde.GetSequenceOfFragments();
  if( sf )
    {
    std::ofstream output(outfilename, std::ios::binary);
    sf->WriteBuffer(output);
    }
  else
    {
    std::cerr « "Error" « std::endl;
    return 1;
    }
  gdcm::JPEGCodec jpeg;
  std::ifstream is(outfilename, std::ios::binary);
  gdcm::PixelFormat pf ( gdcm::PixelFormat::UINT8 ); // let's pretend it's a 8bits jpeg
  jpeg.SetPixelFormat( pf );
  gdcm::TransferSyntax ts_jpg;
  bool b = jpeg.GetHeaderInfo( is, ts_jpg );
  if( !b )
    {
    return 1;
    }
  //jpeg.Print( std::cout );
  if( jpeg.GetPixelFormat().GetBitsAllocated() != image.GetPixelFormat().GetBitsAllocated()
   || jpeg.GetPixelFormat().GetBitsStored() != image.GetPixelFormat().GetBitsStored() )
    {
    std::cerr « "There is a mismatch in between DICOM declared Pixel Format and Sample Precision used in the
      JPEG stream" « std::endl;
    return 0;
    }
  std::cout « jpeg.GetPixelFormat() « std::endl;
  std::cout « image.GetPixelFormat() « std::endl;
  return 1;
}
```

12.75 GetSequenceUltrasound.cxx

```
/*=========================================================================

Program:  GDCM (Grassroots DICOM). A DICOM library

Copyright (c) 2006-2011 Mathieu Malaterre
All rights reserved.
See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.

This software is distributed WITHOUT ANY WARRANTY; without even
the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
PURPOSE.  See the above copyright notice for more information.

=========================================================================*/
#include "gdcmReader.h"
#include "gdcmAttribute.h"
bool Region ( char* nomefile, unsigned int* X_min, unsigned int* Y_min, unsigned int* X_max, unsigned int* Y_max
      );
int main(int argc, char* argv[] )
{
  // Controllo del numero di argomenti introdotti da riga di comando
```

```
  if( argc < 2 )
    {
    std::cerr « "Usage:  " « std::endl;
    std::cerr « argv[0] « " inputImageFile  " « std::endl;
    return EXIT_FAILURE;
    }
  unsigned int x_min = 1;
  unsigned int y_min = 1;
  unsigned int x_max = 1;
  unsigned int y_max = 1;
  if( Region ( argv[1], &x_min, &y_min, &x_max, &y_max ) )
    {
    std::cout « "x_min = " « x_min « std::endl;
    std::cout « "y_min = " « y_min « std::endl;
    std::cout « "x_max = " « x_max « std::endl;
    std::cout « "y_max = " « y_max « std::endl;
    }
  else
    {
    std::cout « "no\n";
    }
}
bool Region ( char* nomefile, unsigned int* X_min, unsigned int* Y_min, unsigned int* X_max, unsigned int* Y_max
    )
{
  gdcm::Reader reader;
  reader.SetFileName( nomefile );
  if( !reader.Read() )
    {
    std::cerr « "Could not read:  " « nomefile « std::endl;
    return false;
    }
  gdcm::File &file = reader.GetFile();
  gdcm::DataSet &ds = file.GetDataSet();
  gdcm::Tag tsqur(0x0018,0x6011);
  if( !ds.FindDataElement( tsqur ) )
    {
    return false;
    }
  const gdcm::DataElement &squr= ds.GetDataElement( tsqur );
  //std::cout « squr « std::endl;
  const gdcm::SequenceOfItems *sqi = squr.GetValueAsSQ();
  if( !sqi || !sqi->GetNumberOfItems() )
    {
    return false;
    }
  //std::cout « sqi « std::endl;
  const gdcm::Item & item = sqi->GetItem(1);
  //std::cout « item « std::endl;
  const gdcm::DataSet& nestedds = item.GetNestedDataSet();
  //std::cout « nestedds « std::endl;
  gdcm::Tag tX0(0x0018,0x6018);
  gdcm::Tag tY0(0x0018,0x601a);
  gdcm::Tag tX1(0x0018,0x601c);
  gdcm::Tag tY1(0x0018,0x601e);
  if( (!nestedds.FindDataElement( tX0 ))||(!nestedds.FindDataElement( tY0 ))||(!nestedds.FindDataElement( tX1
      ))||(!nestedds.FindDataElement( tY1 )) )
    {
    return false;
    }
  const gdcm::DataElement& deX0 = nestedds.GetDataElement( tX0 );
  const gdcm::DataElement& deY0 = nestedds.GetDataElement( tY0 );
  const gdcm::DataElement& deX1 = nestedds.GetDataElement( tX1 );
  const gdcm::DataElement& deY1 = nestedds.GetDataElement( tY1 );
  //std::cout « deX0 « std::endl « deY0 « std::endl « deX1 « std::endl « deY1 « std::endl;
  //const gdcm::ByteValue *bvX0 = deX0.GetByteValue();
  //const gdcm::ByteValue *bvY0 = deY0.GetByteValue();
  //const gdcm::ByteValue *bvX1 = deX1.GetByteValue();
  //const gdcm::ByteValue *bvY1 = deY1.GetByteValue();
  //std::cout « bvX0 « std::endl « bvY0 « std::endl « bvX1 « std::endl « bvY1 « std::endl;
  gdcm::Attribute<0x0018,0x6018> atX0;
  gdcm::Attribute<0x0018,0x601a> atY0;
  gdcm::Attribute<0x0018,0x601c> atX1;
  gdcm::Attribute<0x0018,0x601e> atY1;
  atX0.SetFromDataElement( deX0 );
  atY0.SetFromDataElement( deY0 );
  atX1.SetFromDataElement( deX1 );
  atY1.SetFromDataElement( deY1 );
  uint32_t X0 = atX0.GetValue();
  uint32_t Y0 = atY0.GetValue();
  uint32_t X1 = atX1.GetValue();
```

```
  uint32_t Y1 = atY1.GetValue();
  std::cout « X0 « std::endl « Y0 « std::endl « X1 « std::endl « Y1 « std::endl;
  *X_min = static_cast<unsigned int>(X0);
  *Y_min = static_cast<unsigned int>(Y0);
  *X_max = static_cast<unsigned int>(X1);
  *Y_max = static_cast<unsigned int>(Y1);
  //std::cout « "X_min = " « *X_min « std::endl;
  //std::cout « "Y_min = " « *Y_min « std::endl;
  //std::cout « "X_max = " « *X_max « std::endl;
  //std::cout « "Y_max = " « *Y_max « std::endl;
  return true;
}
```

12.76 GetSubSequenceData.cxx

```
/*=========================================================================

Program:  GDCM (Grassroots DICOM). A DICOM library

Copyright (c) 2006-2011 Mathieu Malaterre
All rights reserved.
See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.

This software is distributed WITHOUT ANY WARRANTY; without even
the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
PURPOSE.  See the above copyright notice for more information.

=========================================================================*/
#include "gdcmReader.h"
#include "gdcmImage.h"
#include "gdcmImageWriter.h"
#include "gdcmDataElement.h"
#include "gdcmPrivateTag.h"
#include "gdcmUIDGenerator.h"
#include <iostream>
#include <string>
#include <map>
/*
* This example will extract the Movie from the private group of
* GEMS_Ultrasound_MovieGroup_001 See Attribute
* (7fe1,60,GEMS_Ultrasound_MovieGroup_001)
*
* The output file will be stored in `outvid.dcm` as
* MultiframeGrayscaleByteSecondaryCaptureImageStorage
*/
int main(int argc, char *argv[])
{
  if( argc < 2 ) return 1;
  using namespace gdcm;
  const char *filename = argv[1];
  gdcm::Reader reader;
  reader.SetFileName( filename );
  reader.Read();
  gdcm::File &file = reader.GetFile();
  gdcm::DataSet &ds = file.GetDataSet();
  const PrivateTag tseq(0x7fe1,0x1,"GEMS_Ultrasound_MovieGroup_001");
  if( !ds.FindDataElement( tseq ) ) return 1;
  const DataElement& seq = ds.GetDataElement( tseq );
  SmartPointer<SequenceOfItems> sqi = seq.GetValueAsSQ();
  assert( sqi->GetNumberOfItems() == 1 );
  Item &item = sqi->GetItem(1);
  DataSet &subds = item.GetNestedDataSet();
  const PrivateTag tseq1(0x7fe1,0x10,"GEMS_Ultrasound_MovieGroup_001");
  if( !subds.FindDataElement( tseq1 ) ) return 1;
  const DataElement& seq1 = subds.GetDataElement( tseq1 );
  SmartPointer<SequenceOfItems> sqi2 = seq1.GetValueAsSQ();
  //int n = sqi2->GetNumberOfItems();
  int index = 1;
  Item &item2 = sqi2->GetItem(index);
  DataSet &subds2 = item2.GetNestedDataSet();
  const PrivateTag tseq2(0x7fe1,0x20,"GEMS_Ultrasound_MovieGroup_001");
  if( !subds2.FindDataElement( tseq2 ) ) return 1;
  const DataElement& seq2 = subds2.GetDataElement( tseq2 );
//    std::cout « seq2 « std::endl;
  SmartPointer<SequenceOfItems> sqi3 = seq2.GetValueAsSQ();
  size_t ni3 = sqi3->GetNumberOfItems(); (void)ni3;
  assert( sqi3->GetNumberOfItems() >= 1 );
```

```
  Item &item3 = sqi3->GetItem(1);
  DataSet &subds3 = item3.GetNestedDataSet();
  const PrivateTag tseq6(0x7fe1,0x26,"GEMS_Ultrasound_MovieGroup_001");
  if( !subds3.FindDataElement( tseq6 ) ) return 1;
  const DataElement& seq6 = subds3.GetDataElement( tseq6 );
  SmartPointer<SequenceOfItems> sqi6 = seq6.GetValueAsSQ();
  size_t ni6= sqi6->GetNumberOfItems();
  assert( sqi6->GetNumberOfItems() >= 1 );
  const PrivateTag tseq7(0x7fe1,0x86,"GEMS_Ultrasound_MovieGroup_001");
  int dimx = 0, dimy = 0;
  for( size_t i6 = 1; i6 <= ni6; ++i6 )
    {
    Item &item6 = sqi6->GetItem(i6);
    DataSet &subds6 = item6.GetNestedDataSet();
    if( subds6.FindDataElement( tseq7 ) )
      {
      Element<VR::SL, VM::VM4> el;
      el.SetFromDataElement( subds6.GetDataElement( tseq7 ) );
      std::cout « "El= " « el.GetValue() « std::endl;
      dimx = el.GetValue(0);
      dimy = el.GetValue(1);
      }
    }
  const PrivateTag tseq3(0x7fe1,0x36,"GEMS_Ultrasound_MovieGroup_001");
  if( !subds3.FindDataElement( tseq3 ) ) return 1;
  const DataElement& seq3 = subds3.GetDataElement( tseq3 );
//    std::cout « seq3 « std::endl;
  SmartPointer<SequenceOfItems> sqi4 = seq3.GetValueAsSQ();
  size_t ni4= sqi4->GetNumberOfItems();
  assert( sqi4->GetNumberOfItems() >= 1 );
  const PrivateTag tseq8(0x7fe1,0x37,"GEMS_Ultrasound_MovieGroup_001");
  const PrivateTag tseq4(0x7fe1,0x43,"GEMS_Ultrasound_MovieGroup_001");
  const PrivateTag tseq5(0x7fe1,0x60,"GEMS_Ultrasound_MovieGroup_001");
  std::vector<char> imbuffer;
  int dimz = 0;
  for( size_t i4 = 1; i4 <= ni4; ++i4 )
    {
    Item &item4 = sqi4->GetItem(i4);
    DataSet &subds4 = item4.GetNestedDataSet();
    if( !subds4.FindDataElement( tseq8 ) ) return 1;
    const DataElement& de8 = subds4.GetDataElement( tseq8 );
    Element<VR::UL,VM::VM1> ldimz;
    ldimz.SetFromDataElement( de8 );
    dimz += ldimz.GetValue();
    if( !subds4.FindDataElement( tseq4 ) ) return 1;
    const DataElement& seq4 = subds4.GetDataElement( tseq4 );
    if( !subds4.FindDataElement( tseq5 ) ) return 1;
    const DataElement& seq5 = subds4.GetDataElement( tseq5 );
    //    std::cout « seq4 « std::endl;
    //    std::cout « seq5 « std::endl;
    const ByteValue *bv4 = seq4.GetByteValue();
    (void)bv4;
#if 0
      {
      std::ofstream out( "/tmp/mo4", std::ios::binary );
      out.write( bv4->GetPointer(), bv4->GetLength());
      out.close();
      }
#endif
  const ByteValue *bv5 = seq5.GetByteValue();
#if 0
    {
    std::ofstream out( "/tmp/mo5", std::ios::binary );
    out.write( bv5->GetPointer(), bv5->GetLength());
    out.close();
    }
#endif
    std::cout « bv5->GetLength() « std::endl;
    imbuffer.insert( imbuffer.begin(), bv5->GetPointer(), bv5->GetPointer() + bv5->GetLength() );
    }
  DataElement fakedata;
  fakedata.SetByteValue( &imbuffer[0], (uint32_t)imbuffer.size() );
  gdcm::SmartPointer<gdcm::Image> im = new gdcm::Image;
  im->SetNumberOfDimensions( 3 );
  im->SetDimension(0, dimx );
  im->SetDimension(1, dimy );
  im->SetDimension(2, dimz );
  size_t l1 = imbuffer.size();
  (void)l1;
  size_t l2 = im->GetBufferLength();
  (void)l2;
```

```
  assert( im->GetBufferLength() == imbuffer.size() );
  im->SetPhotometricInterpretation( gdcm::PhotometricInterpretation::MONOCHROME2 );
  im->SetDataElement( fakedata );
  gdcm::ImageWriter w;
  w.SetImage( *im );
  DataSet &dataset = w.GetFile().GetDataSet();
  gdcm::UIDGenerator uid;
  gdcm::DataElement de( Tag(0x8,0x18) ); // SOP Instance UID
  de.SetVR( VR::UI );
  const char *u = uid.Generate();
  de.SetByteValue( u, (uint32_t)strlen(u) );
  //ds.Insert( de );
  dataset.Replace( de );
  de.SetTag( Tag(0x8,0x16) ); // SOP Class UID
  de.SetVR( VR::UI );
  gdcm::MediaStorage ms(
    gdcm::MediaStorage::MultiframeGrayscaleByteSecondaryCaptureImageStorage );
  de.SetByteValue( ms.GetString(), (uint32_t)strlen(ms.GetString()));
  dataset.Replace( de ); // replace !
  w.SetFileName( "outvid.dcm" );
  if( !w.Write() )
    {
    return 1;
    }
  return 0;
}
```

12.77 HelloVizWorld.cxx

```
/*=========================================================================

Program:  GDCM (Grassroots DICOM). A DICOM library

Copyright (c) 2006-2011 Mathieu Malaterre
All rights reserved.
See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.

This software is distributed WITHOUT ANY WARRANTY; without even
the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
PURPOSE.  See the above copyright notice for more information.

=========================================================================*/
/*
* Basic example for dealing with a DICOM file that contains an Image
* (read:  Pixel Data element)
*/
#include "gdcmImageReader.h"
#include "gdcmImageWriter.h"
#include "gdcmImage.h"
#include "gdcmPhotometricInterpretation.h"
#include <iostream>
int main(int argc, char *argv[])
{
  if( argc < 3 )
    {
    std::cerr « argv[0] « " input.dcm output.dcm" « std::endl;
    return 1;
    }
  const char *filename = argv[1];
  const char *outfilename = argv[2];
  // Instanciate the image reader:
  gdcm::ImageReader reader;
  reader.SetFileName( filename );
  if( !reader.Read() )
    {
    std::cerr « "Could not read:  " « filename « std::endl;
    return 1;
    }
  // If we reach here, we know for sure 2 things:
  // 1.  It is a valid DICOM
  // 2.  And it contains an Image !
  // The output of superclass gdcm::Reader is a gdcm::File
  //gdcm::File &file = reader.GetFile();
  // The other output of gdcm::ImageReader is a gdcm::Image
  const gdcm::Image &image = reader.GetImage();
  // Let's get some property from the image:
  unsigned int ndim = image.GetNumberOfDimensions();
```

```
  // Dimensions of the image:
  const unsigned int *dims = image.GetDimensions();
  // Origin
  const double *origin = image.GetOrigin();
  const gdcm::PhotometricInterpretation &pi = image.GetPhotometricInterpretation();
  for(unsigned int i = 0; i < ndim; ++i)
    {
    std::cout « "Dim(" « i « "):  " « dims[i] « std::endl;
    }
  for(unsigned int i = 0; i < ndim; ++i)
    {
    std::cout « "Origin(" « i « "):  " « origin[i] « std::endl;
    }
  std::cout « "PhotometricInterpretation:  " « pi « std::endl;
  // Write the modified DataSet back to disk
  gdcm::ImageWriter writer;
  writer.SetImage( image );
  writer.SetFileName( outfilename );
  //writer.SetFile( file ); // We purposely NOT copy the meta information from the input
                            // file, and instead only pass the image
  if( !writer.Write() )
    {
    std::cerr « "Could not write:  " « outfilename « std::endl;
    return 1;
    }
  return 0;
}
```

12.78 HelloWorld.cxx

```
/*=========================================================================

Program:  GDCM (Grassroots DICOM). A DICOM library

Copyright (c) 2006-2011 Mathieu Malaterre
All rights reserved.
See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.

This software is distributed WITHOUT ANY WARRANTY; without even
the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
PURPOSE.  See the above copyright notice for more information.

=========================================================================*/
/*
* This example is ...  guess what this is for :)
*/
#include "gdcmReader.h"
#include "gdcmWriter.h"
#include "gdcmAttribute.h"
#include <iostream>
int main(int argc, char *argv[])
{
  if( argc < 3 )
    {
    std::cerr « argv[0] « " input.dcm output.dcm" « std::endl;
    return 1;
    }
  const char *filename = argv[1];
  const char *outfilename = argv[2];
  // Instanciate the reader:
  gdcm::Reader reader;
  reader.SetFileName( filename );
  if( !reader.Read() )
    {
    std::cerr « "Could not read:  " « filename « std::endl;
    return 1;
    }
  // If we reach here, we know for sure only 1 thing:
  // It is a valid DICOM file (potentially an old ACR-NEMA 1.0/2.0 file)
  // (Maybe, it's NOT a Dicom image -could be a DICOMDIR, a RTSTRUCT, etc-)
  // The output of gdcm::Reader is a gdcm::File
  gdcm::File &file = reader.GetFile();
  // the dataset is the the set of element we are interested in:
  gdcm::DataSet &ds = file.GetDataSet();
  // Contruct a static(*) type for Image Comments :
  gdcm::Attribute<0x0020,0x4000> imagecomments;
  imagecomments.SetValue( "Hello, World !" );
```

```
  // Now replace the Image Comments from the dataset with our:
  ds.Replace( imagecomments.GetAsDataElement() );
  // Write the modified DataSet back to disk
  gdcm::Writer writer;
  writer.CheckFileMetaInformationOff(); // Do not attempt to reconstruct the file meta to preserve the file
                                        // as close to the original as possible.
  writer.SetFileName( outfilename );
  writer.SetFile( file );
  if( !writer.Write() )
    {
    std::cerr « "Could not write:  " « outfilename « std::endl;
    return 1;
    }
  return 0;
}
/*
* (*) static type, means that extra DICOM information VR & VM are computed at compilation time.
* The compiler is deducing those values from the template arguments of the class.
*/
```

12.79 LargeVRDSExplicit.cxx

```
/*=========================================================================

Program:  GDCM (Grassroots DICOM). A DICOM library

Copyright (c) 2006-2011 Mathieu Malaterre
All rights reserved.
See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.

This software is distributed WITHOUT ANY WARRANTY; without even
the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
PURPOSE.  See the above copyright notice for more information.

=========================================================================*/
#include "gdcmReader.h"
#include "gdcmWriter.h"
#include "gdcmAttribute.h"
#include "gdcmFileExplicitFilter.h"
#include "gdcmSequenceOfItems.h"
bool interpolate(const double * pts, size_t npts, std::vector<double> &out )
{
  out.clear();
  for(size_t i = 0; i < 2*npts; ++i )
    {
    const size_t j = i / 2;
    if( i % 2 )
      {
      if( j != npts - 1 )
        {
        assert( 3*j+5 < 3*npts );
        const double midpointx = (pts[3*j+0] + pts[3*j+3]) / 2;
        const double midpointy = (pts[3*j+1] + pts[3*j+4]) / 2;
        const double midpointz = (pts[3*j+2] + pts[3*j+5]) / 2;
        out.push_back( midpointx );
        out.push_back( midpointy );
        out.push_back( midpointz );
        }
      }
    else
      {
      assert( j < npts );
      out.push_back( pts[3*j+0] );
      out.push_back( pts[3*j+1] );
      out.push_back( pts[3*j+2] );
      }
    }
  assert( out.size() == 2 * npts * 3 - 3 );
  return true;
}
int main(int argc, char *argv[])
{
  if( argc < 3 )
    {
    std::cerr « argv[0] « " input.dcm output.dcm" « std::endl;
    return 1;
    }
```

```
const char *filename = argv[1];
const char *outfilename = argv[2];
gdcm::Reader reader;
reader.SetFileName( filename );
if( !reader.Read() )
  {
  return 1;
  }
gdcm::File &file = reader.GetFile();
gdcm::DataSet &ds = file.GetDataSet();
gdcm::FileExplicitFilter fef;
//fef.SetChangePrivateTags( changeprivatetags );
fef.SetFile( reader.GetFile() );
if( !fef.Change() )
  {
  std::cerr « "Failed to change:  " « filename « std::endl;
  return 1;
  }
// (3006,0039) SQ (Sequence with undefined length #=4)     # u/l, 1 ROIContourSequence
gdcm::Tag tag(0x3006,0x0039);
const gdcm::DataElement &roicsq = ds.GetDataElement( tag );
gdcm::SmartPointer<gdcm::SequenceOfItems> sqi = roicsq.GetValueAsSQ();
//sqi->SetNumberOfItems( 1 );
const gdcm::Item & item = sqi->GetItem(1); // Item start at #1
const gdcm::DataSet& nestedds = item.GetNestedDataSet();
gdcm::Tag tcsq(0x3006,0x0040);
if( !nestedds.FindDataElement( tcsq ) )
  {
  return 0;
  }
const gdcm::DataElement& csq = nestedds.GetDataElement( tcsq );
gdcm::SmartPointer<gdcm::SequenceOfItems> sqi2 = csq.GetValueAsSQ();
if( !sqi2 || !sqi2->GetNumberOfItems() )
  {
  return 0;
  }
//unsigned int nitems = sqi2->GetNumberOfItems();
gdcm::Item & item2 = sqi2->GetItem(1); // Item start at #1
gdcm::DataSet& nestedds2 = item2.GetNestedDataSet();
//item2.SetVLToUndefined();
//std::cout « nestedds2 « std::endl;
// (3006,0050) DS [43.57636\65.52504\-10.0\46.043102\62.564945\-10.0\49.126537\60.714...  # 398,48 ContourData
gdcm::Tag tcontourdata(0x3006,0x0050);
const gdcm::DataElement & contourdata = nestedds2.GetDataElement( tcontourdata );
//std::cout « contourdata « std::endl;
//const gdcm::ByteValue *bv = contourdata.GetByteValue();
gdcm::Attribute<0x3006,0x0046> ncontourpoints;
ncontourpoints.Set( nestedds2 );
gdcm::Attribute<0x3006,0x0050> at;
at.SetFromDataElement( contourdata );
const double* pts = at.GetValues();
unsigned int npts = at.GetNumberOfValues() / 3;
std::vector<double> out( pts, pts + npts * 3 );
std::vector<double> out2;
//const unsigned int niter = 7;
const unsigned int niter = 8;
for( unsigned int i = 0; i < niter; ++i)
  {
  //bool b =
  interpolate(&out[0], out.size() / 3, out2);
  //const double *pout = &out[0];
  out = out2;
  out2.clear();
  }
assert( out.size() % 3 == 0 );
gdcm::Attribute<0x3006,0x0050> at_interpolate;
at_interpolate.SetNumberOfValues( (unsigned int)(out.size() / 3) );
at_interpolate.SetValues( &out[0], (uint32_t)out.size() );
ncontourpoints.SetValue( at_interpolate.GetNumberOfValues() / 3 );
nestedds2.Replace( at_interpolate.GetAsDataElement() );
nestedds2.Replace( ncontourpoints.GetAsDataElement() );
//assert(0);
// Let's take item one and subdivide it
gdcm::TransferSyntax ts = gdcm::TransferSyntax::ImplicitVRLittleEndian;
ts = gdcm::TransferSyntax::ExplicitVRLittleEndian;
gdcm::FileMetaInformation &fmi = file.GetHeader();
const char *tsuid = gdcm::TransferSyntax::GetTSString( ts );
// const char * is ok since padding is \0 anyway...
gdcm::DataElement de( gdcm::Tag(0x0002,0x0010) );
de.SetByteValue( tsuid, (uint32_t)strlen(tsuid) );
de.SetVR( gdcm::Attribute<0x0002, 0x0010>::GetVR() );
```

```
  fmi.Replace( de );
  fmi.Remove( gdcm::Tag(0x0002,0x0012) ); // will be regenerated
  fmi.Remove( gdcm::Tag(0x0002,0x0013) ); //  '   '    '
  fmi.SetDataSetTransferSyntax(ts);
  gdcm::Writer w;
  w.SetFile( file );
  w.SetFileName( outfilename );
  if (!w.Write() )
    {
    return 1;
    }
  return 0;
}
```

12.80 MakeTemplate.cxx

```
/*=========================================================================

Program:  GDCM (Grassroots DICOM). A DICOM library

Copyright (c) 2006-2011 Mathieu Malaterre
All rights reserved.
See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.

This software is distributed WITHOUT ANY WARRANTY; without even
the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
PURPOSE.  See the above copyright notice for more information.

=========================================================================*/
#include "gdcmFileAnonymizer.h"
#include "gdcmReader.h"
#include "gdcmWriter.h"
int main(int argc, char *argv[])
{
  if( argc < 3 ) return 1;
  const char* filename = argv[1];
  const char* outfilename = argv[2];
  //gdcm::Trace::DebugOn();
  // Remove Pixel Data element:
  gdcm::FileAnonymizer fa;
  fa.SetInputFileName( filename );
  fa.SetOutputFileName( outfilename );
  fa.Empty( gdcm::Tag(0x7fe0,0x10) );
  // cannot replace in-place DICOM header:
  //fa.Replace( gdcm::Tag(0x2,0x2), "1.2.840.10008.5.1.4.1.1.7" );
  if( !fa.Write() )
    {
    std::cerr « "impossible to remove Pixel Data attribute" « std::endl;
    return 1;
    }
  // Update the DICOM Header:
  gdcm::Reader reader;
  reader.SetFileName( outfilename );
  if( !reader.Read() )
    {
    std::cerr « "could not read back" « std::endl;
    return 1;
    }
  gdcm::File & file = reader.GetFile();
  gdcm::FileMetaInformation &fmi = file.GetHeader();
  gdcm::TransferSyntax ts = gdcm::TransferSyntax::ImplicitVRLittleEndian;
  ts = gdcm::TransferSyntax::ExplicitVRLittleEndian;
  fmi.SetDataSetTransferSyntax(ts);
  gdcm::Writer writer;
  writer.SetFile( file );
  writer.SetFileName( outfilename ); // warning overwrite file !
  if( !writer.Write() )
    {
    std::cerr « "could not write back" « std::endl;
    return 1;
    }
  return 0;
}
```

12.81 MergeTwoFiles.cxx

```
/*=========================================================================

Program:  GDCM (Grassroots DICOM). A DICOM library

Copyright (c) 2006-2011 Mathieu Malaterre
All rights reserved.
See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.

This software is distributed WITHOUT ANY WARRANTY; without even
the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
PURPOSE.  See the above copyright notice for more information.

=========================================================================*/
/*
* This example will show how one can read in two DICOM files, use the dataset
* from file1 and use image from file2  to save it in a 3rd file.
*
* Eg:
* MergeTwoFiles gdcmData/012345.002.050.dcm gdcmData/test.acr merge.dcm
*/
#include "gdcmReader.h"
#include "gdcmImageReader.h"
#include "gdcmImageWriter.h"
#include "gdcmWriter.h"
#include "gdcmDataSet.h"
#include "gdcmAttribute.h"
int main(int argc, char *argv[])
{
  if( argc < 3 )
    {
    return 1;
    }
  const char *file1 = argv[1];
  const char *file2 = argv[2];
  const char *file3 = argv[3];
  // Read file1
  gdcm::ImageReader reader1;
  reader1.SetFileName( file1 );
  if( !reader1.Read() )
    {
    return 1;
    }
  // Read file2
  gdcm::ImageReader reader2;
  reader2.SetFileName( file2 );
  if( !reader2.Read() )
    {
    return 1;
    }
  // Ok now let's take the DataSet from file1 and the Image from file2
  // Warning:  if file2 is -for example- a Secondary Capture Storage, then it has no
  // Image Orientation (Patient) thus any Image Orientation (Patient) from file1
  // will be discarded...
  // let's be fancy.  In case reader2 contains explicit, but reader1 is implicit
  // we would rather see an implicit output
  if( reader1.GetFile().GetHeader().GetDataSetTransferSyntax() == gdcm::TransferSyntax::ImplicitVRLittleEndian )
    {
    reader2.GetImage().SetTransferSyntax( gdcm::TransferSyntax::ImplicitVRLittleEndian );
    }
  gdcm::ImageWriter writer;
  writer.SetFileName( file3 );
  writer.SetFile( reader1.GetFile() );
  // ImageWriter will always use all of gdcm::Image information an override anything wrong from
  // reader1.GetFile(), including the Transfer Syntax
  writer.SetImage( reader2.GetImage() );
  gdcm::DataSet &ds = reader1.GetFile().GetDataSet();
  // Make sure that SOPInstanceUID are different
  // Simply removing it is sufficient as gdcm::ImageWriter will generate one by default
  // if not found.
  ds.Remove( gdcm::Tag(0x0008,0x0018) );
  if( !writer.Write() )
    {
    return 1;
    }
  return 0;
}
```

12.82 MrProtocol.cxx

```
/*=========================================================================

Program:  GDCM (Grassroots DICOM). A DICOM library

Copyright (c) 2006-2011 Mathieu Malaterre
All rights reserved.
See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.

This software is distributed WITHOUT ANY WARRANTY; without even
the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
PURPOSE.  See the above copyright notice for more information.

=========================================================================*/
/*
*
*/
/*
28 - 'MrProtocol' VM 1, VR UN, SyngoDT 0, NoOfItems 6, Data '### ASCCONV BEGIN ###
ulVersion                               = 0xbee332
tSequenceFileName                       = "%SiemensSeq%\fl_fq_shphs"
tProtocolName                           = "flash+AF8-100+AF8-through-plane+AF8-V"
tReferenceImage0                        = "1.3.12.2.1107.5.2.9.16041.30000007062106100181200004658"
tReferenceImage1                        = "1.3.12.2.1107.5.2.9.16041.30000007062106100181200004635"
tReferenceImage2                        = "1.3.12.2.1107.5.2.9.16041.30000007062106100181200004683"
ucScanRegionPosValid                    = 0x1
sProtConsistencyInfo.tBaselineString    = "N4_VB11A_LATEST_20031004"
sProtConsistencyInfo.flNominalB0        = 1.494
sProtConsistencyInfo.flGMax             = 22
sProtConsistencyInfo.flRiseTime         = 10
sGRADSPEC.sEddyCompensationX.aflAmplitude[0] = 0.0141111
sGRADSPEC.sEddyCompensationX.aflAmplitude[1] = 0.057038
sGRADSPEC.sEddyCompensationX.aflAmplitude[2] = -0.00986504
sGRADSPEC.sEddyCompensationX.aflAmplitude[3] = 0.00247627
sGRADSPEC.sEddyCompensationX.aflAmplitude[4] = 0.0026377
sGRADSPEC.sEddyCompensationX.aflTimeConstant[0] = 1.53826
sGRADSPEC.sEddyCompensationX.aflTimeConstant[1] = 0.746617
sGRADSPEC.sEddyCompensationX.aflTimeConstant[2] = 0.339236
sGRADSPEC.sEddyCompensationX.aflTimeConstant[3] = 0.0309809
sGRADSPEC.sEddyCompensationX.aflTimeConstant[4] = 0.00067694
sGRADSPEC.sEddyCompensationY.aflAmplitude[0] = 0.0156411
sGRADSPEC.sEddyCompensationY.aflAmplitude[1] = 0.0440623
sGRADSPEC.sEddyCompensationY.aflAmplitude[2] = -0.00782663
sGRADSPEC.sEddyCompensationY.aflAmplitude[3] = 0.00186828
sGRADSPEC.sEddyCompensationY.aflAmplitude[4] = 0.00154504
sGRADSPEC.sEddyCompensationY.aflTimeConstant[0] = 1.47145
sGRADSPEC.sEddyCompensationY.aflTimeConstant[1] = 0.750538
sGRADSPEC.sEddyCompensationY.aflTimeConstant[2] = 0.339397
sGRADSPEC.sEddyCompensationY.aflTimeConstant[3] = 0.0312962
sGRADSPEC.sEddyCompensationY.aflTimeConstant[4] = 0.000895133
sGRADSPEC.sEddyCompensationZ.aflAmplitude[0] = 0.00618504
sGRADSPEC.sEddyCompensationZ.aflAmplitude[1] = 0.00313121
sGRADSPEC.sEddyCompensationZ.aflAmplitude[2] = 0.000289346
sGRADSPEC.sEddyCompensationZ.aflAmplitude[3] = -0.00019677
sGRADSPEC.sEddyCompensationZ.aflAmplitude[4] = 7.66445e-005
sGRADSPEC.sEddyCompensationZ.aflTimeConstant[0] = 3.37462
sGRADSPEC.sEddyCompensationZ.aflTimeConstant[1] = 0.999351
sGRADSPEC.sEddyCompensationZ.aflTimeConstant[2] = 0.0174646
sGRADSPEC.sEddyCompensationZ.aflTimeConstant[3] = 0.0110094
sGRADSPEC.sEddyCompensationZ.aflTimeConstant[4] = 0.00199922
sGRADSPEC.bEddyCompensationValid        = 1
sGRADSPEC.sB0CompensationX.aflAmplitude[0] = 0.307474
sGRADSPEC.sB0CompensationX.aflAmplitude[1] = 0.029337
sGRADSPEC.sB0CompensationX.aflAmplitude[2] = -0.187118
sGRADSPEC.sB0CompensationX.aflTimeConstant[0] = 0.98583
sGRADSPEC.sB0CompensationX.aflTimeConstant[1] = 0.0308443
sGRADSPEC.sB0CompensationX.aflTimeConstant[2] = 0.000466792
sGRADSPEC.sB0CompensationY.aflAmplitude[0] = 0.365257
sGRADSPEC.sB0CompensationY.aflAmplitude[1] = -0.318647
sGRADSPEC.sB0CompensationY.aflAmplitude[2] = -0.0118978
sGRADSPEC.sB0CompensationY.aflTimeConstant[0] = 0.61535
sGRADSPEC.sB0CompensationY.aflTimeConstant[1] = 0.488831
sGRADSPEC.sB0CompensationY.aflTimeConstant[2] = 0.00199991
sGRADSPEC.sB0CompensationZ.aflAmplitude[0] = -0.44647
sGRADSPEC.sB0CompensationZ.aflAmplitude[1] = -0.0455154
sGRADSPEC.sB0CompensationZ.aflAmplitude[2] = -0.0304901
sGRADSPEC.sB0CompensationZ.aflTimeConstant[0] = 0.959231
sGRADSPEC.sB0CompensationZ.aflTimeConstant[1] = 0.0720189
sGRADSPEC.sB0CompensationZ.aflTimeConstant[2] = 0.00190141
```

```
sGRADSPEC.bB0CompensationValid          = 1
sGRADSPEC.sCrossTermCompensationXY.aflAmplitude[0] = 0.00105046
sGRADSPEC.sCrossTermCompensationXY.aflTimeConstant[0] = 0.842014
sGRADSPEC.sCrossTermCompensationXZ.aflAmplitude[0] = -0.00150189
sGRADSPEC.sCrossTermCompensationXZ.aflTimeConstant[0] = 0.736169
sGRADSPEC.sCrossTermCompensationYX.aflAmplitude[0] = -5.5278e-005
sGRADSPEC.sCrossTermCompensationYX.aflTimeConstant[0] = 0.228697
sGRADSPEC.sCrossTermCompensationYZ.aflAmplitude[0] = 0.000307999
sGRADSPEC.sCrossTermCompensationYZ.aflTimeConstant[0] = 1.19431
sGRADSPEC.sCrossTermCompensationZX.aflAmplitude[0] = -0.000286868
sGRADSPEC.sCrossTermCompensationZX.aflTimeConstant[0] = 0.665979
sGRADSPEC.sCrossTermCompensationZY.aflAmplitude[0] = 0.000355175
sGRADSPEC.sCrossTermCompensationZY.aflTimeConstant[0] = 0.844189
sGRADSPEC.bCrossTermCompensationValid    = 1
sGRADSPEC.lOffsetX                      = 25
sGRADSPEC.lOffsetY                      = 84
sGRADSPEC.lOffsetZ                      = 47
sGRADSPEC.bOffsetValid                  = 1
sGRADSPEC.lDelayX                       = 12
sGRADSPEC.lDelayY                       = 11
sGRADSPEC.lDelayZ                       = 9
sGRADSPEC.bDelayValid                   = 1
sGRADSPEC.flSensitivityX                = 0.000264087
sGRADSPEC.flSensitivityY                = 0.000272009
sGRADSPEC.flSensitivityZ                = 0.000272677
sGRADSPEC.bSensitivityValid             = 1
sGRADSPEC.alShimCurrent[0]              = 183
sGRADSPEC.alShimCurrent[1]              = -25
sGRADSPEC.alShimCurrent[2]              = -85
sGRADSPEC.alShimCurrent[3]              = 378
sGRADSPEC.alShimCurrent[4]              = 82
sGRADSPEC.bShimCurrentValid             = 1
sGRADSPEC.ucMode                        = 0x2
sTXSPEC.asNucleusInfo[0].tNucleus       = "1H"
sTXSPEC.asNucleusInfo[0].lFrequency     = 63684693
sTXSPEC.asNucleusInfo[0].bFrequencyValid = 1
sTXSPEC.asNucleusInfo[0].flReferenceAmplitude = 359.734
sTXSPEC.asNucleusInfo[0].bReferenceAmplitudeValid = 1
sTXSPEC.asNucleusInfo[0].flAmplitudeCorrection = 1
sTXSPEC.asNucleusInfo[0].bAmplitudeCorrectionValid = 1
sTXSPEC.asNucleusInfo[1].bFrequencyValid = 1
sTXSPEC.asNucleusInfo[1].bReferenceAmplitudeValid = 1
sTXSPEC.asNucleusInfo[1].bAmplitudeCorrectionValid = 1
sTXSPEC.aRFPULSE[0].tName               = "03GreFCE"
sTXSPEC.aRFPULSE[0].bAmplitudeValid     = 0x1
sTXSPEC.aRFPULSE[0].flAmplitude         = 147.095
sTXSPEC.aRFPULSE[1].tName               = "02GreFCE"
sTXSPEC.aRFPULSE[1].bAmplitudeValid     = 0x1
sTXSPEC.aRFPULSE[1].flAmplitude         = 147.095
sTXSPEC.aRFPULSE[2].tName               = "01GreFCE"
sTXSPEC.aRFPULSE[2].bAmplitudeValid     = 0x1
sTXSPEC.aRFPULSE[2].flAmplitude         = 147.095
sTXSPEC.lNoOfTraPulses                  = 3
sTXSPEC.lBTB1ParallelCapacity           = 2
sTXSPEC.lBTB1SerialCapacity             = 24
sTXSPEC.lBTB2ParallelCapacity           = 2
sTXSPEC.lBTB2SerialCapacity             = 26
sTXSPEC.bBTBValid                       = 1
sTXSPEC.flKDynMagnitudeMin              = 0.5
sTXSPEC.flKDynMagnitudeMax              = 1.5
sTXSPEC.flKDynMagnitudeClipLow          = 0.96
sTXSPEC.flKDynMagnitudeClipHigh         = 1.04
sTXSPEC.flKDynPhaseMax                  = 0.698132
sTXSPEC.flKDynPhaseClip                 = 0.174533
sTXSPEC.bKDynValid                      = 1
sTXSPEC.ucRFPulseType                   = 0x1
sTXSPEC.ucExcitMode                     = 0x1
sTXSPEC.ucSimultaneousExcitation        = 0x1
sRXSPEC.lGain                           = 1
sRXSPEC.bGainValid                      = 1
sRXSPEC.aFFT_SCALE[0].lRxChannel        = 1
sRXSPEC.aFFT_SCALE[0].flFactor          = 1.06857
sRXSPEC.aFFT_SCALE[0].bValid            = 1
sRXSPEC.aFFT_SCALE[1].lRxChannel        = 2
sRXSPEC.aFFT_SCALE[1].flFactor          = 1.07454
sRXSPEC.aFFT_SCALE[1].bValid            = 1
sRXSPEC.aFFT_SCALE[2].lRxChannel        = 3
sRXSPEC.aFFT_SCALE[2].flFactor          = 1.06622
sRXSPEC.aFFT_SCALE[2].bValid            = 1
sRXSPEC.aFFT_SCALE[3].lRxChannel        = 4
sRXSPEC.aFFT_SCALE[3].flFactor          = 1.06524
```

```
sRXSPEC.aFFT_SCALE[3].bValid             = 1
sRXSPEC.aFFT_SCALE[4].lRxChannel         = 5
sRXSPEC.aFFT_SCALE[4].flFactor           = 0.982692
sRXSPEC.aFFT_SCALE[4].bValid             = 1
sRXSPEC.aFFT_SCALE[5].lRxChannel         = 6
sRXSPEC.aFFT_SCALE[5].flFactor           = 0.988603
sRXSPEC.aFFT_SCALE[5].bValid             = 1
sRXSPEC.aFFT_SCALE[6].lRxChannel         = 7
sRXSPEC.aFFT_SCALE[6].flFactor           = 0.981538
sRXSPEC.aFFT_SCALE[6].bValid             = 1
sRXSPEC.aFFT_SCALE[7].lRxChannel         = 8
sRXSPEC.aFFT_SCALE[7].flFactor           = 1.00856
sRXSPEC.aFFT_SCALE[7].bValid             = 1
sRXSPEC.bVariCapVoltagesValid            = 1
sRXSPEC.alDwellTime[0]                   = 8500
sAdjFreSpec.ulMode                       = 0x1
sAdjFreSpec.ucAdjWithBC                  = 0x1
sAdjTraSpec.ucAdjWithBC                  = 0x1
sAdjShimSpec.ulMode                      = 0x1
sAdjShimSpec.ucAdjWithBC                 = 0x1
sAdjWatSupSpec.ulMode                    = 0x1
sAdjWatSupSpec.ucAdjWithBC               = 0x1
alTR[0]                                  = 37000
lContrasts                               = 1
alTE[0]                                  = 4000
acFlowComp[0]                            = 1
lCombinedEchoes                          = 1
sSliceArray.asSlice[0].sPosition.dSag    = 35.31199581
sSliceArray.asSlice[0].sPosition.dCor    = -8.387765754
sSliceArray.asSlice[0].sPosition.dTra    = -23.13178296
sSliceArray.asSlice[0].sNormal.dSag      = 0.771051253
sSliceArray.asSlice[0].sNormal.dCor      = 0.5863890019
sSliceArray.asSlice[0].sNormal.dTra      = -0.2482496801
sSliceArray.asSlice[0].dThickness        = 6
sSliceArray.asSlice[0].dPhaseFOV         = 187.5
sSliceArray.asSlice[0].dReadoutFOV       = 250
sSliceArray.lSize                        = 1
sSliceArray.lSag                         = 1
sSliceArray.lConc                        = 1
sSliceArray.ucMode                       = 0x1
sSliceArray.sTSat.dThickness             = 40
sSliceArray.sTSat.dGap                   = 10
sGroupArray.asGroup[0].nSize             = 1
sGroupArray.asGroup[0].dDistFact         = 0.2
sGroupArray.anMember[1]                  = -1
sGroupArray.lSize                        = 1
sGroupArray.sPSat.dThickness             = 50
sGroupArray.sPSat.dGap                   = 10
sAutoAlign.dAAMatrix[0]                  = 1
sAutoAlign.dAAMatrix[5]                  = 1
sAutoAlign.dAAMatrix[10]                 = 1
sAutoAlign.dAAMatrix[15]                 = 1
sNavigatorPara.ucRespComp                = 0x4
sPrepPulses.ucFatSat                     = 0x4
sPrepPulses.ucWaterSat                   = 0x4
sPrepPulses.ucInversion                  = 0x4
sPrepPulses.ucSatRecovery                = 0x1
sPrepPulses.ucFatSatMode                 = 0x2
sKSpace.lBaseResolution                  = 256
sKSpace.lPhaseEncodingLines              = 192
sKSpace.dPhaseResolution                 = 1
sKSpace.lPartitions                      = 32
sKSpace.lImagesPerSlab                   = 32
sKSpace.dSliceResolution                 = 1
sKSpace.ucPhasePartialFourier            = 0x10
sKSpace.ucSlicePartialFourier            = 0x10
sKSpace.ucAveragingMode                  = 0x2
sKSpace.ucMultiSliceMode                 = 0x1
sKSpace.ucDimension                      = 0x2
sKSpace.ucAsymmetricEchoAllowed          = 0x1
sKSpace.unReordering                     = 0x1
sFastImaging.lEPIFactor                  = 1
sFastImaging.lTurboFactor                = 1
sFastImaging.lSegments                   = 3
sFastImaging.ulEnableRFSpoiling          = 0x1
sPhysioImaging.lSignal1                  = 2
sPhysioImaging.lMethod1                  = 2
sPhysioImaging.lSignal2                  = 1
sPhysioImaging.lMethod2                  = 1
sPhysioImaging.lPhases                   = 21
sPhysioImaging.lRetroGatedImages         = 16
```

```
sPhysioImaging.sPhysioECG.lScanWindow    = 805
sPhysioImaging.sPhysioECG.lTriggerPulses = 1
sPhysioImaging.sPhysioECG.lTriggerWindow = 5
sPhysioImaging.sPhysioECG.lArrhythmiaDetection = 1
sPhysioImaging.sPhysioECG.lCardiacGateOnThreshold = 100000
sPhysioImaging.sPhysioECG.lCardiacGateOffThreshold = 700000
sPhysioImaging.sPhysioPulse.lTriggerPulses = 1
sPhysioImaging.sPhysioPulse.lTriggerWindow = 5
sPhysioImaging.sPhysioPulse.lCardiacGateOnThreshold = 100000
sPhysioImaging.sPhysioPulse.lCardiacGateOffThreshold = 700000
sPhysioImaging.sPhysioExt.lTriggerPulses = 1
sPhysioImaging.sPhysioExt.lTriggerWindow = 5
sPhysioImaging.sPhysioExt.lCardiacGateOnThreshold = 100000
sPhysioImaging.sPhysioExt.lCardiacGateOffThreshold = 700000
sPhysioImaging.sPhysioResp.lRespGateThreshold = 20
sPhysioImaging.sPhysioResp.lRespGatePhase = 2
sPhysioImaging.sPhysioResp.dGatingRatio  = 0.3
sSpecPara.lPhaseCyclingType              = 1
sSpecPara.lPhaseEncodingType             = 1
sSpecPara.lRFExcitationBandwidth         = 1
sSpecPara.ucRemoveOversampling           = 0x1
sSpecPara.lDecouplingType                = 1
sSpecPara.lNOEType                       = 1
sSpecPara.lExcitationType                = 1
sSpecPara.lSpectralSuppression           = 1
sDiffusion.ulMode                        = 0x1
sAngio.sFlowArray.asElm[0].nVelocity     = 100
sAngio.sFlowArray.asElm[0].nDir          = 0x4
sAngio.sFlowArray.lSize                  = 1
sAngio.ucPCFlowMode                      = 0x2
sAngio.ucTOFInflow                       = 0x4
sAngio.ucRephasedImage                   = 0x1
sAngio.ucPhaseImage                      = 0x1
sEllipticalFilter.ucMode                 = 0x1
sPat.lAccelFactPE                        = 1
sPat.lAccelFact3D                        = 1
sPat.ucPATMode                           = 0x1
sPat.ucRefScanMode                       = 0x1
ucAutoMovie                              = 0x1
ucDisableChangeStoreImages               = 0x1
ucReconstructionMode                     = 0x1
ucPHAPSMode                              = 0x1
ucDixon                                  = 0x1
lAverages                                = 2
adFlipAngleDegree[0]                     = 30
lScanTimeSec                             = 103
lTotalScanTimeSec                        = 112
dRefSNR                                  = 165404.1473
dRefSNR_VOI                              = 165404.1473
tdefaultEVAProt                          = "%SiemensEvaDefProt%\Inline\Inline.evp"
tcurrentEVAProt                          = "%CURRENTEVAPROT%\EVA2A5.tmp"
sCOIL_SELECT_MEAS.asList[0].sCoilElementID.tCoilID = "6_Ch_Body_P"
sCOIL_SELECT_MEAS.asList[0].sCoilElementID.lCoilCopy = 1
sCOIL_SELECT_MEAS.asList[0].sCoilElementID.tElement = "PP6"
sCOIL_SELECT_MEAS.asList[0].lElementSelected = 1
sCOIL_SELECT_MEAS.asList[0].lRxChannelConnected = 1
sCOIL_SELECT_MEAS.asList[1].sCoilElementID.tCoilID = "6_Ch_Body_P"
sCOIL_SELECT_MEAS.asList[1].sCoilElementID.lCoilCopy = 1
sCOIL_SELECT_MEAS.asList[1].sCoilElementID.tElement = "PP5"
sCOIL_SELECT_MEAS.asList[1].lElementSelected = 1
sCOIL_SELECT_MEAS.asList[1].lRxChannelConnected = 1
sCOIL_SELECT_MEAS.asList[2].sCoilElementID.tCoilID = "6_Ch_Body_P"
sCOIL_SELECT_MEAS.asList[2].sCoilElementID.lCoilCopy = 1
sCOIL_SELECT_MEAS.asList[2].sCoilElementID.tElement = "PP3"
sCOIL_SELECT_MEAS.asList[2].lElementSelected = 1
sCOIL_SELECT_MEAS.asList[2].lRxChannelConnected = 2
sCOIL_SELECT_MEAS.asList[3].sCoilElementID.tCoilID = "6_Ch_Body_P"
sCOIL_SELECT_MEAS.asList[3].sCoilElementID.lCoilCopy = 1
sCOIL_SELECT_MEAS.asList[3].sCoilElementID.tElement = "PP4"
sCOIL_SELECT_MEAS.asList[3].lElementSelected = 1
sCOIL_SELECT_MEAS.asList[3].lRxChannelConnected = 3
sCOIL_SELECT_MEAS.asList[4].sCoilElementID.tCoilID = "6_Ch_Body_P"
sCOIL_SELECT_MEAS.asList[4].sCoilElementID.lCoilCopy = 1
sCOIL_SELECT_MEAS.asList[4].sCoilElementID.tElement = "PP2"
sCOIL_SELECT_MEAS.asList[4].lElementSelected = 1
sCOIL_SELECT_MEAS.asList[4].lRxChannelConnected = 4
sCOIL_SELECT_MEAS.asList[5].sCoilElementID.tCoilID = "6_Ch_Body_P"
sCOIL_SELECT_MEAS.asList[5].sCoilElementID.lCoilCopy = 1
sCOIL_SELECT_MEAS.asList[5].sCoilElementID.tElement = "PP1"
sCOIL_SELECT_MEAS.asList[5].lElementSelected = 1
sCOIL_SELECT_MEAS.asList[5].lRxChannelConnected = 4
```

```
sCOIL_SELECT_MEAS.asList[6].sCoilElementID.tCoilID = "6_Ch_Body_A"
sCOIL_SELECT_MEAS.asList[6].sCoilElementID.lCoilCopy = 1
sCOIL_SELECT_MEAS.asList[6].sCoilElementID.tElement = "PA6"
sCOIL_SELECT_MEAS.asList[6].lElementSelected = 1
sCOIL_SELECT_MEAS.asList[6].lRxChannelConnected = 5
sCOIL_SELECT_MEAS.asList[7].sCoilElementID.tCoilID = "6_Ch_Body_A"
sCOIL_SELECT_MEAS.asList[7].sCoilElementID.lCoilCopy = 1
sCOIL_SELECT_MEAS.asList[7].sCoilElementID.tElement = "PA5"
sCOIL_SELECT_MEAS.asList[7].lElementSelected = 1
sCOIL_SELECT_MEAS.asList[7].lRxChannelConnected = 5
sCOIL_SELECT_MEAS.asList[8].sCoilElementID.tCoilID = "6_Ch_Body_A"
sCOIL_SELECT_MEAS.asList[8].sCoilElementID.lCoilCopy = 1
sCOIL_SELECT_MEAS.asList[8].sCoilElementID.tElement = "PA3"
sCOIL_SELECT_MEAS.asList[8].lElementSelected = 1
sCOIL_SELECT_MEAS.asList[8].lRxChannelConnected = 6
sCOIL_SELECT_MEAS.asList[9].sCoilElementID.tCoilID = "6_Ch_Body_A"
sCOIL_SELECT_MEAS.asList[9].sCoilElementID.lCoilCopy = 1
sCOIL_SELECT_MEAS.asList[9].sCoilElementID.tElement = "PA4"
sCOIL_SELECT_MEAS.asList[9].lElementSelected = 1
sCOIL_SELECT_MEAS.asList[9].lRxChannelConnected = 7
sCOIL_SELECT_MEAS.asList[10].sCoilElementID.tCoilID = "6_Ch_Body_A"
sCOIL_SELECT_MEAS.asList[10].sCoilElementID.lCoilCopy = 1
sCOIL_SELECT_MEAS.asList[10].sCoilElementID.tElement = "PA2"
sCOIL_SELECT_MEAS.asList[10].lElementSelected = 1
sCOIL_SELECT_MEAS.asList[10].lRxChannelConnected = 8
sCOIL_SELECT_MEAS.asList[11].sCoilElementID.tCoilID = "6_Ch_Body_A"
sCOIL_SELECT_MEAS.asList[11].sCoilElementID.lCoilCopy = 1
sCOIL_SELECT_MEAS.asList[11].sCoilElementID.tElement = "PA1"
sCOIL_SELECT_MEAS.asList[11].lElementSelected = 1
sCOIL_SELECT_MEAS.asList[11].lRxChannelConnected = 8
sCOIL_SELECT_MEAS.sCOILPLUGS.aulPlugId[0] = 0xff
sCOIL_SELECT_MEAS.sCOILPLUGS.aulPlugId[1] = 0x76
sCOIL_SELECT_MEAS.sCOILPLUGS.aulPlugId[2] = 0x78
sCOIL_SELECT_MEAS.sCOILPLUGS.aulPlugId[3] = 0x87
sCOIL_SELECT_MEAS.sCOILPLUGS.aulPlugId[4] = 0x67
sCOIL_SELECT_MEAS.sCOILPLUGS.auiNmbrOfNibbles[0] = 0x2
sCOIL_SELECT_MEAS.sCOILPLUGS.auiNmbrOfNibbles[1] = 0x2
sCOIL_SELECT_MEAS.sCOILPLUGS.auiNmbrOfNibbles[2] = 0x2
sCOIL_SELECT_MEAS.sCOILPLUGS.auiNmbrOfNibbles[3] = 0x2
sCOIL_SELECT_MEAS.sCOILPLUGS.auiNmbrOfNibbles[4] = 0x2
sEFISPEC.bEFIDataValid                   = 1
### ASCCONV END ###
'
*/
/*
* Table of equivalence:
*
ulVersion                                = 0xbee332
<=>
27 - 'MrProtocolVersion' VM 1, VR IS, SyngoDT 6, NoOfItems 6, Data '12510002'
*/
#include "gdcmReader.h"
#include "gdcmImageReader.h"
#include "gdcmImageWriter.h"
#include "gdcmCSAHeader.h"
#include "gdcmAttribute.h"
#include "gdcmGlobal.h"
#include "gdcmDicts.h"
#include <map>
#include <math.h>
int main(int argc, char *argv [])
{
  if( argc < 2 ) return 1;
  const char *filename = argv[1];
  gdcm::ImageReader reader;
  reader.SetFileName( filename );
  if( !reader.Read() )
    {
    std::cerr « "Failed to read:  " « filename « std::endl;
    return 1;
    }
  gdcm::CSAHeader csa;
  const gdcm::DataSet& ds = reader.GetFile().GetDataSet();
  //const gdcm::PrivateTag &t1 = csa.GetCSAImageHeaderInfoTag();
  const gdcm::PrivateTag &t2 = csa.GetCSASeriesHeaderInfoTag();
  if( ds.FindDataElement( t2 ) )
    {
    csa.LoadFromDataElement( ds.GetDataElement( t2 ) );
    //csa.Print( std::cout );
    }
  if( !csa.FindCSAElementByName( "MrProtocol" ) )
```

```
    {
    return 1;
    }
  const gdcm::CSAElement &csael = csa.GetCSAElementByName( "MrProtocol" );
  //std::cout « csael « std::endl;
  const gdcm::ByteValue *bv = csael.GetByteValue();
  if( !bv )
    {
    return 1;
    }
  std::string str(bv->GetPointer(), bv->GetLength());
  std::istringstream is(str);
  std::string s;
  typedef std::map< std::string, std::string > MyMapType;
  MyMapType mymap;
  while( std::getline(is, s ) )
    {
    std::string::size_type pos = s.find( '=' );
    if( pos != std::string::npos )
      {
      std::string sub1 = s.substr(0, pos);
      sub1.erase( sub1.find_last_not_of(' ') + 1);
      std::string sub2 = s.substr(pos+1); // skip the '=' char
      sub2.erase( 0, sub2.find_first_not_of(' '));
      //std::cout « sub1 « std::endl;
      mymap.insert( MyMapType::value_type(sub1, sub2) );
      }
    else
      {
      // ### ASCCONV BEGIN ###
      // ### ASCCONV END ###
      }
    }
  const char fourierstr[] = "sKSpace.ucSlicePartialFourier";
  const gdcm::CSAHeaderDict &csadict = gdcm::Global::GetInstance().GetDicts().GetCSAHeaderDict();
  const gdcm::CSAHeaderDictEntry &fourier = csadict.GetCSAHeaderDictEntry( fourierstr );
  std::cout « fourier « std::endl;
  MyMapType::const_iterator it = mymap.find ( fourierstr );
  if( it == mymap.end() ) return 1;
  //std::cout « it->second « std::endl;
  const std::string &partial_fourier = it->second;
  if( partial_fourier == "0x1" )
    {
    std::cout « "partial fourier is 4/8" « std::endl;
    }
  else if( partial_fourier == "0x2" )
    {
    std::cout « "partial fourier is 5/8" « std::endl;
    }
  else if( partial_fourier == "0x4" )
    {
    std::cout « "partial fourier is 6/8" « std::endl;
    }
  else if( partial_fourier == "0x8" )
    {
    std::cout « "partial fourier is 7/8" « std::endl;
    }
  else if( partial_fourier == "0x10" )
    {
    std::cout « "partial fourier is 8/8" « std::endl;
    }
  else
    {
    std::cerr « "Impossible:  " « partial_fourier « std::endl;
    return 1;
    }
/*
This is the Flip Angle:
adFlipAngleDegree[0]                    = 30

One can find it also in the protocol:

...
<ParamFunctor.""T1mapFunctor"">
{
<Class> ""T1mapFunctor@IceImagePostProcFunctors""

<ParamBool.""EXECUTE"">  { }
<ParamDouble.""Flip1_deg"">  { <Precision> 16  14.7378520000000000  }
...
```

```
*/
  // Below is an attemp to play with the CSAHeader dict:
#if 0
  const char gspec[] = "sGRADSPEC.flSensitivityX";
  it = mymap.find( gspec );
  if( it == mymap.end() ) return 1;
  const std::string &dummy = it->second;
  std::cout « dummy « std::endl;
  const gdcm::CSAHeaderDictEntry &csaentry = csadict.GetCSAHeaderDictEntry( gspec );
  std::cout « csaentry  « std::endl;
#endif
/*
sSliceArray.ucMode -- should be in (1, 2, 4)
enum SeriesMode
{
ASCENDING   = 0x01,
DESCENDING  = 0x02,
INTERLEAVED = 0x04
};
*/
  const char sliceorderstr[] = "sSliceArray.ucMode";
  const gdcm::CSAHeaderDictEntry &sliceorder = csadict.GetCSAHeaderDictEntry( sliceorderstr );
  std::cout « sliceorder « std::endl;
  it = mymap.find ( sliceorderstr );
  if( it == mymap.end() ) return 1;
  const std::string &slice_order = it->second;
  if( slice_order == "0x1" )
    {
    std::cout « "slice_order:  ASCENDING" « std::endl;
    }
  else if( slice_order == "0x2" )
    {
    std::cout « "slice_order:  DESCENDING" « std::endl;
    }
  else if( slice_order == "0x4" )
    {
    std::cout « "slice_order:  INTERLEAVED" « std::endl;
    }
  else
    {
    std::cerr « "Impossible:  " « slice_order « std::endl;
    return 1;
    }
  gdcm::MrProtocol mrprot;
  if(  csa.GetMrProtocol(ds, mrprot) )
  {
    std::cout « mrprot « std::endl;
  }
  return 0;
}
```

12.83 PrintLUT.cxx

```
/*=========================================================================

Program:  GDCM (Grassroots DICOM). A DICOM library

Copyright (c) 2006-2011 Mathieu Malaterre
All rights reserved.
See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.

This software is distributed WITHOUT ANY WARRANTY; without even
the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
PURPOSE.  See the above copyright notice for more information.

=========================================================================*/
/*
*/
#include "gdcmImageReader.h"
#include "gdcmImageWriter.h"
#include "gdcmImage.h"
#include "gdcmPhotometricInterpretation.h"
#include <iostream>
int main(int argc, char *argv[])
{
  if( argc < 2 )
    {
```

```
    std::cerr « argv[0] « " input.dcm" « std::endl;
    return 1;
    }
  const char *filename = argv[1];
  // Instanciate the image reader:
  gdcm::ImageReader reader;
  reader.SetFileName( filename );
  if( !reader.Read() )
    {
    std::cerr « "Could not read:  " « filename « std::endl;
    return 1;
    }
  const gdcm::Image &image = reader.GetImage();
  const gdcm::LookupTable & lut = image.GetLUT();
  lut.Print( std::cout );
  return 0;
}
```

12.84 PublicDict.cxx

```
/*=========================================================================

Program:  GDCM (Grassroots DICOM). A DICOM library

Copyright (c) 2006-2011 Mathieu Malaterre
All rights reserved.
See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.

This software is distributed WITHOUT ANY WARRANTY; without even
the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
PURPOSE.  See the above copyright notice for more information.

=========================================================================*/
/*
* Dummy example to show GDCM Dict(s) API (Part 6) + Collected Private Attributes:
*/
#include "gdcmGlobal.h"
#include "gdcmDicts.h"
#include "gdcmDict.h"
#include "gdcmCSAHeader.h"
#include "gdcmPrivateTag.h"
int main(int , char *[])
{
  const gdcm::Global& g = gdcm::Global::GetInstance(); // sum of all knowledge !
  const gdcm::Dicts &dicts = g.GetDicts();
  const gdcm::Dict &pub = dicts.GetPublicDict(); // Part 6
  //std::cout « pub « std::endl;
  // 3 differents way to access the same information
  // 1.  From the public dict only:
  gdcm::Tag patient_name(0x10,0x10);
  const gdcm::DictEntry &entry1 = pub.GetDictEntry(patient_name);
  std::cout « entry1 « std::endl;
  // 2.  From all dicts:
  const gdcm::DictEntry &entry2 = dicts.GetDictEntry(patient_name);
  std::cout « entry2 « std::endl;
  // 3.  This solution is the most flexible solution as you can request using the same
  // API either a public tag or a private tag
  const char *strowner = nullptr;
  const gdcm::DictEntry &entry3 = dicts.GetDictEntry(patient_name,strowner);
  std::cout « entry3 « std::endl;
  // Private attributes:
  // try with a private tag now:
  const gdcm::PrivateTag &private_tag = gdcm::CSAHeader::GetCSAImageHeaderInfoTag();
  //std::cout « private_tag « std::endl;
  const gdcm::DictEntry &entry4 = dicts.GetDictEntry(private_tag,private_tag.GetOwner());
  std::cout « entry4 « std::endl;
  // Let's pretend that private lookup is on 0x10xx elements:
  gdcm::PrivateTag dummy = private_tag;
  dummy.SetElement( (uint16_t)(0x1000 + dummy.GetElement()) );
  const gdcm::DictEntry &entry5 = dicts.GetDictEntry(dummy,dummy.GetOwner());
  std::cout « entry5 « std::endl;
  return 0;
}
```

12.85 QIDO-RS.cxx

```
/*=========================================================================

Program:  GDCM (Grassroots DICOM). A DICOM library

Copyright (c) 2006-2011 Mathieu Malaterre
All rights reserved.
See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.

This software is distributed WITHOUT ANY WARRANTY; without even
the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
PURPOSE.  See the above copyright notice for more information.

=========================================================================*/
#include "gdcmReader.h"
#include "gdcmWriter.h"
#include "gdcmJSON.h"
/*
* Simple QIDO-RS round-trip to test implementation of gdcm::JSON
* See Sup166 for details
*/
int main(int argc, char *argv[])
{
  if( argc < 2 ) return 1;
  using namespace gdcm;
  const char *filename = argv[1];
  gdcm::Reader reader;
  reader.SetFileName( filename );
  if( !reader.Read() ) return 1;
  gdcm::JSON json;
  json.PrettyPrintOn();
  std::stringstream ss;
  const gdcm::File & f = reader.GetFile();
  json.Code( f.GetDataSet(), ss);
  std::cout « ss.str() « std::endl;
  gdcm::Writer w;
  gdcm::File & ff = w.GetFile();
  ff.GetHeader().SetDataSetTransferSyntax( gdcm::TransferSyntax::ExplicitVRLittleEndian );
  if( !json.Decode(ss, ff.GetDataSet() ) )
    {
    std::cerr « "Could not decode" « std::endl;
    return 1;
    }
  w.SetFileName( "/tmp/debug.dcm" );
  if( !w.Write() ) return 1;
  return 0;
}
```

12.86 ReadAndDumpDICOMDIR.cxx

```
/*=========================================================================

Program:  GDCM (Grassroots DICOM). A DICOM library

Copyright (c) 2006-2011 Mathieu Malaterre
All rights reserved.
See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.

This software is distributed WITHOUT ANY WARRANTY; without even
the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
PURPOSE.  See the above copyright notice for more information.

=========================================================================*/
/*
* This example shows how to read and dump a DICOMDIR File
*
* Thanks:
*   Tom Marynowski (lordglub gmail) for contributing this example
*/
#include "gdcmReader.h"
#include "gdcmMediaStorage.h"
typedef std::set<gdcm::DataElement> DataElementSet;
typedef DataElementSet::const_iterator ConstIterator;
int main(int argc, char *argv [])
{
  if( argc < 2 ) return 1;
```

```
const char *filename = argv[1];
gdcm::Reader reader;
reader.SetFileName( filename);
if( !reader.Read() )
  {
  std::cerr « "Could not read:  " « filename « std::endl;
  return 1;
  }
std::stringstream strm;
gdcm::File &file = reader.GetFile();
gdcm::DataSet &ds = file.GetDataSet();
gdcm::FileMetaInformation &fmi = file.GetHeader();
gdcm::MediaStorage ms;
ms.SetFromFile(file);
if( ms != gdcm::MediaStorage::MediaStorageDirectoryStorage )
  {
  std::cout « "This file is not a DICOMDIR" « std::endl;
  return 1;
  }
if (fmi.FindDataElement( gdcm::Tag (0x0002, 0x0002)))
  {   strm.str("");
  fmi.GetDataElement( gdcm::Tag (0x0002, 0x0002) ).GetValue().Print(strm);
  }
else
  {
  std::cerr « " Media Storage Sop Class UID not present" « std::endl;
  }
//TODO il faut trimer strm.str() avant la comparaison au cas ou...
if ("1.2.840.10008.1.3.10"!=strm.str())
  {
  std::cout « "This file is not a DICOMDIR" « std::endl;
  return 1;
  }
ConstIterator it = ds.GetDES().begin();
for( ; it != ds.GetDES().end(); ++it)
  {
  if (it->GetTag()==gdcm::Tag (0x0004, 0x1220))
    {
    const gdcm::DataElement &de = (*it);
    // ne pas utiliser GetSequenceOfItems pour extraire les items
    gdcm::SmartPointer<gdcm::SequenceOfItems> sqi =de.GetValueAsSQ();
    unsigned int itemused = 1;
    while (itemused<=sqi->GetNumberOfItems())
      {
      strm.str("");
      if (sqi->GetItem(itemused).FindDataElement(gdcm::Tag (0x0004, 0x1430)))
        sqi->GetItem(itemused).GetDataElement(gdcm::Tag (0x0004, 0x1430)).GetValue().Print(strm);
      //TODO il faut trimer strm.str() avant la comparaison
      while((strm.str()=="PATIENT")||((strm.str()=="PATIENT ")))
        {
        std::cout « strm.str() « std::endl;
        strm.str("");
        if (sqi->GetItem(itemused).FindDataElement(gdcm::Tag (0x0010, 0x0010)))
          sqi->GetItem(itemused).GetDataElement(gdcm::Tag (0x0010, 0x0010)).GetValue().Print(strm);
        std::cout « "PATIENT NAME : " « strm.str() « std::endl;
        //PATIENT ID
        strm.str("");
        if (sqi->GetItem(itemused).FindDataElement(gdcm::Tag (0x0010, 0x0020)))
          sqi->GetItem(itemused).GetDataElement(gdcm::Tag (0x0010, 0x0020)).GetValue().Print(strm);
        std::cout « "PATIENT ID : " « strm.str() « std::endl;
        /*ADD TAG TO READ HERE*/
        std::cout « "============================ "  « std::endl;
        itemused++;
        strm.str("");
        if (sqi->GetItem(itemused).FindDataElement(gdcm::Tag (0x0004, 0x1430)))
          sqi->GetItem(itemused).GetDataElement(gdcm::Tag (0x0004, 0x1430)).GetValue().Print(strm);
        //TODO il faut trimer strm.str() avant la comparaison
        while((strm.str()=="STUDY")||((strm.str()=="STUDY ")))
          {
          std::cout « "  " « strm.str() « std::endl;
          //UID
          strm.str("");
          if (sqi->GetItem(itemused).FindDataElement(gdcm::Tag (0x0020, 0x000d)))
            sqi->GetItem(itemused).GetDataElement(gdcm::Tag (0x0020, 0x000d)).GetValue().Print(strm);
          std::cout « "      STUDY UID : " « strm.str() « std::endl;
          //STUDY DATE
          strm.str("");
          if (sqi->GetItem(itemused).FindDataElement(gdcm::Tag (0x0008, 0x0020)))
            sqi->GetItem(itemused).GetDataElement(gdcm::Tag (0x0008, 0x0020)).GetValue().Print(strm);
          std::cout « "      STUDY DATE : " « strm.str() « std::endl;
          //STUDY DESCRIPTION
```

```
      strm.str("");
      if (sqi->GetItem(itemused).FindDataElement(gdcm::Tag (0x0008, 0x1030)))
        sqi->GetItem(itemused).GetDataElement(gdcm::Tag (0x0008, 0x1030)).GetValue().Print(strm);
      std::cout « "      STUDY DESCRIPTION : " « strm.str() « std::endl;
      /*ADD TAG TO READ HERE*/
      std::cout « "      " « "========================== "  « std::endl;
      itemused++;
      strm.str("");
      if (sqi->GetItem(itemused).FindDataElement(gdcm::Tag (0x0004, 0x1430)))
        sqi->GetItem(itemused).GetDataElement(gdcm::Tag (0x0004, 0x1430)).GetValue().Print(strm);
      //TODO il faut trimer strm.str() avant la comparaison
      while((strm.str()=="SERIES")||((strm.str()=="SERIES ")))
        {
        std::cout « "      " « strm.str() « std::endl;
        strm.str("");
        if (sqi->GetItem(itemused).FindDataElement(gdcm::Tag (0x0020, 0x000e)))
          sqi->GetItem(itemused).GetDataElement(gdcm::Tag (0x0020, 0x000e)).GetValue().Print(strm);
        std::cout « "          SERIE UID" « strm.str() « std::endl;
        //SERIE MODALITY
        strm.str("");
        if (sqi->GetItem(itemused).FindDataElement(gdcm::Tag (0x0008, 0x0060)))
          sqi->GetItem(itemused).GetDataElement(gdcm::Tag (0x0008, 0x0060)).GetValue().Print(strm);
        std::cout « "          SERIE MODALITY" « strm.str() « std::endl;
        //SERIE DESCRIPTION
        strm.str("");
        if (sqi->GetItem(itemused).FindDataElement(gdcm::Tag (0x0008, 0x103e)))
          sqi->GetItem(itemused).GetDataElement(gdcm::Tag (0x0008, 0x103e)).GetValue().Print(strm);
        std::cout « "          SERIE DESCRIPTION" « strm.str() « std::endl;
        /*ADD TAG TO READ HERE*/
        std::cout « "          " « "========================== "  « std::endl;
        itemused++;
        strm.str("");
        if (sqi->GetItem(itemused).FindDataElement(gdcm::Tag (0x0004, 0x1430)))
          sqi->GetItem(itemused).GetDataElement(gdcm::Tag (0x0004, 0x1430)).GetValue().Print(strm);
        //TODO il faut trimer strm.str() avant la comparaison
        while ((strm.str()=="IMAGE")||((strm.str()=="IMAGE ")))
          // if(tmp=="IMAGE")
          {
          std::cout « "          " « strm.str() « std::endl;
          //UID
          strm.str("");
          if (sqi->GetItem(itemused).FindDataElement(gdcm::Tag (0x0004, 0x1511)))
            sqi->GetItem(itemused).GetDataElement(gdcm::Tag (0x0004, 0x1511)).GetValue().Print(strm);
          std::cout « "              IMAGE UID : " « strm.str() « std::endl;
          //PATH de l'image
          strm.str("");
          if (sqi->GetItem(itemused).FindDataElement(gdcm::Tag (0x0004, 0x1500)))
            sqi->GetItem(itemused).GetDataElement(gdcm::Tag (0x0004, 0x1500)).GetValue().Print(strm);
          std::cout « "              IMAGE PATH : " « strm.str() « std::endl;
          /*ADD TAG TO READ HERE*/
          if(itemused < sqi->GetNumberOfItems())
            {itemused++;
            }else{break;}
          strm.str("");
          if (sqi->GetItem(itemused).FindDataElement(gdcm::Tag (0x0004, 0x1430)))
            sqi->GetItem(itemused).GetDataElement(gdcm::Tag (0x0004, 0x1430)).GetValue().Print(strm);
          }
        }
      }
    }
  itemused++;
  }
 }
}
  return 0;
}
```

12.87 ReadAndDumpDICOMDIR2.cxx

```
/*=========================================================================

Program:  GDCM (Grassroots DICOM). A DICOM library

Copyright (c) 2006-2017 Mathieu Malaterre
All rights reserved.
See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.
```

```
This software is distributed WITHOUT ANY WARRANTY; without even
the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
PURPOSE.  See the above copyright notice for more information.

=========================================================================*/
/*
* This example shows how to read and dump a DICOMDIR File
*
* Thanks:
*   Tom Marynowski (lordglub gmail) for contributing the original
*    ReadAndDumpDICOMDIR.cxx example
*   Mihail Isakov for contributing offset calculation code here:
*    https://sourceforge.net/p/gdcm/mailman/gdcm-developers/?viewmonth=201707&viewday=15
*   Tod Baudais for combining the above and cleaning up this example
*/
#include <string>
#include <unordered_map>
#include <iostream>
#include <memory>
#include "gdcmReader.h"
#include "gdcmAttribute.h"
#include "gdcmDirectory.h"
//=============================================================================
//=============================================================================
#define TAG_MEDIA_STORAGE_SOP_CLASS_UID 0x0002,0x0002
#define TAG_DIRECTORY_RECORD_SEQUENCE 0x0004,0x1220
#define TAG_DIRECTORY_RECORD_TYPE 0x0004,0x1430
#define TAG_PATIENTS_NAME 0x0010,0x0010
#define TAG_PATIENT_ID 0x0010,0x0020
#define TAG_STUDY_DATE 0x0008,0x0020
#define TAG_STUDY_DESCRIPTION 0x0008,0x1030
#define TAG_MODALITY 0x0008,0x0060
#define TAG_SERIES_DESCRIPTION 0x0008,0x103E
#define TAG_REFERENCED_FILE_ID 0x0004,0x1500
#define TAG_REFERENCED_LOWER_LEVEL_DIRECTORY_ENTITY_OFFSET 0x0004,0x1420
#define TAG_NEXT_DIRECTORY_RECORD_OFFSET 0x0004,0x1400
//=============================================================================
// Some handy utility functions
//=============================================================================
std::string left_trim(const std::string &s) {
    std::string ss(s);
    ss.erase(ss.begin(), std::find_if(ss.begin(), ss.end(), std::not1(std::ptr_fun<int, int>(std::isspace))));
    return ss;
}
std::string right_trim(const std::string &s) {
    std::string ss(s);
    ss.erase(std::find_if(ss.rbegin(), ss.rend(), std::not1(std::ptr_fun<int, int>(std::isspace))).base(),
      ss.end());
    return ss;
}
std::string trim(const std::string &s) {
    return left_trim(right_trim(s));
}
//=============================================================================
// This code could be put in a header file somewhere
//=============================================================================
class DICOMDIRReader {
    public:
                                DICOMDIRReader      (void) {}
                                DICOMDIRReader      (const DICOMDIRReader &rhs) = delete;
                                DICOMDIRReader      (DICOMDIRReader &&rhs) = delete;
        DICOMDIRReader &        operator =          (const DICOMDIRReader &rhs) = delete;
        DICOMDIRReader &        operator =          (DICOMDIRReader &&rhs) = delete;
        virtual                 ~DICOMDIRReader     (void) {}
    public:
        struct Common {
            int64_t child_offset;
            int64_t sibling_offset;
        };
        struct Image:  public Common {
            std::string path;
        };
        struct Series:  public Common {
            std::string modality;
            std::string description;
            std::vector<std::shared_ptr<Image>> children;
        };
        struct Study:  public Common {
            std::string date;
            std::string description;
            std::vector<std::shared_ptr<Series>> children;
```

```
        };
        struct Patient:  public Common {
            std::string name;
            std::string id;
            std::vector<std::shared_ptr<Study» children;
        };
        struct Other:  public Common {
        };
        const std::vector<std::shared_ptr<Patient»&    load        (const std::string &path);
        const std::vector<std::shared_ptr<Patient»&    patients    (void)  {   return _patients;   }
    private:
        template <class T>
        std::string     get_string              (const T &ds, const gdcm::Tag &tag)
        {
            std::stringstream strm;
            if (ds.FindDataElement(tag)) {
                auto &de = ds.GetDataElement(tag);
                if (!de.IsEmpty() && !de.IsUndefinedLength())
                    de.GetValue().Print(strm);
            }
            return trim(strm.str());
        }
        template <class P, class C, class O>
        void            reassemble_hierarchy    (P &parent_offsets, C &child_offsets, O &other_offsets)
        {
            for (auto &parent :  parent_offsets) {
                int64_t sibling_offset;
                auto c = child_offsets[parent.second->child_offset];
                if (!c) {
                    auto o = other_offsets[parent.second->child_offset];
                    if (!o) {
                        continue;
                    } else {
                        sibling_offset = o->sibling_offset;
                    }
                } else {
                    parent.second->children.push_back(c);
                    sibling_offset = c->sibling_offset;
                }
                // Get all siblings
                while (sibling_offset) {
                    c = child_offsets[sibling_offset];
                    if (!c) {
                        auto o = other_offsets[sibling_offset];
                        if (!o) {
                            break;
                        } else {
                            sibling_offset = o->sibling_offset;
                        }
                    } else {
                        parent.second->children.push_back(c);
                        sibling_offset = c->sibling_offset;
                    }
                }
            }
        }
        std::vector<std::shared_ptr<Patient» _patients;
};
//================================================================================
// This code could be put in an implementation file somewhere
//================================================================================
const std::vector<std::shared_ptr<DICOMDIRReader::Patient»& DICOMDIRReader::load (const std::string &path)
{
    _patients.clear();
    //
    // Read the dataset from the DICOMDIR file
    //
    gdcm::Reader reader;
    reader.SetFileName(path.c_str());
    if(!reader.Read()) {
        throw std::runtime_error("Unable to read file");
    }
    // Retrieve information from file
    auto &file = reader.GetFile();
    auto &data_set = file.GetDataSet();
    auto &file_meta_information = file.GetHeader();
    // Retrieve and check the Media Storage class from file
    gdcm::MediaStorage media_storage;
    media_storage.SetFromFile(file);
    if(media_storage != gdcm::MediaStorage::MediaStorageDirectoryStorage) {
        throw std::runtime_error("This file is not a DICOMDIR");
```

```
}
auto media_storage_sop_class_uid = get_string(file_meta_information,
  gdcm::Tag(TAG_MEDIA_STORAGE_SOP_CLASS_UID));
// Make sure we have a DICOMDIR file
if (media_storage_sop_class_uid != "1.2.840.10008.1.3.10") {
    throw std::runtime_error("This file is not a DICOMDIR");
}
//
// Offset to first item courtesy of Mihail Isakov
//
gdcm::VL first_item_offset = 0;
auto it = data_set.Begin();
for(; it != data_set.End() && it->GetTag() != gdcm::Tag(TAG_DIRECTORY_RECORD_SEQUENCE); ++it) {
    first_item_offset += it->GetLength<gdcm::ExplicitDataElement>();
}
// Tag (4 bytes)
first_item_offset += it->GetTag().GetLength();
// VR field
first_item_offset += it->GetVR().GetLength();
// VL field
// For Explicit VR: adventitiously VL field lenght = VR field lenght,
// for SQ 4 bytes:
// http://dicom.nema.org/medical/dicom/current/output/html/part05.html#table_7.1-1
first_item_offset += it->GetVR().GetLength();
//
// Iterate all data elements
//
// For each item in data set
for(auto data_element :  data_set.GetDES()) {
    // Only look at Directory sequence
    if (data_element.GetTag() != gdcm::Tag(TAG_DIRECTORY_RECORD_SEQUENCE))
        continue;
    auto item_sequence = data_element.GetValueAsSQ();
    auto num_items = item_sequence->GetNumberOfItems();
    //
    // Compute an offset table
    //
    // Start calculation of offset to each item courtesy of Mihail Isakov
    std::vector<int64_t> item_offsets(num_items+1);
    item_offsets[0] = file_meta_information.GetFullLength() + static_cast<int64_t>(first_item_offset);
    //
    // Extract out all of the items
    //
    std::unordered_map<int64_t, std::shared_ptr<Patient» patient_offsets;
    std::unordered_map<int64_t, std::shared_ptr<Study» study_offsets;
    std::unordered_map<int64_t, std::shared_ptr<Series» series_offsets;
    std::unordered_map<int64_t, std::shared_ptr<Image» image_offsets;
    std::unordered_map<int64_t, std::shared_ptr<Other» other_offsets;
    for (uint32_t item_index = 1; item_index <= num_items; ++item_index) {
        auto &item = item_sequence->GetItem(item_index);
        // Add offset for item to offset table
        item_offsets[item_index] = item_offsets[item_index-1] + item.GetLength<gdcm::ExplicitDataElement>();
        // Child offset
        gdcm::Attribute<TAG_REFERENCED_LOWER_LEVEL_DIRECTORY_ENTITY_OFFSET> child_offset;
        child_offset.SetFromDataElement(item.GetDataElement(gdcm::Tag
  (TAG_REFERENCED_LOWER_LEVEL_DIRECTORY_ENTITY_OFFSET)));
        // Sibling offset
        gdcm::Attribute<TAG_NEXT_DIRECTORY_RECORD_OFFSET> sibling_offset;
        sibling_offset.SetFromDataElement(item.GetDataElement(gdcm::Tag
  (TAG_NEXT_DIRECTORY_RECORD_OFFSET)));
        // Record Type
        auto record_type = trim(get_string(item, gdcm::Tag (TAG_DIRECTORY_RECORD_TYPE)));
        // std::cout « "record_type " « record_type « " at " « item_offsets[item_index-1] « std::endl;
        // std::cout « " child_offset " « child_offset.GetValue() « std::endl;
        // std::cout « " sibling_offset " « sibling_offset.GetValue() « std::endl;
        // Extract patient information
        if (record_type == "PATIENT") {
            auto patient = std::make_shared<Patient>();
            patient->name = get_string(item, gdcm::Tag (TAG_PATIENTS_NAME));
            patient->id = get_string(item, gdcm::Tag (TAG_PATIENT_ID));
            patient->child_offset = child_offset.GetValue();
            patient->sibling_offset = sibling_offset.GetValue();
            patient_offsets[item_offsets[item_index-1]] = patient;
        // Extract study information
        } else if (record_type == "STUDY") {
            auto study = std::make_shared<Study>();
            study->date = get_string(item, gdcm::Tag (TAG_STUDY_DATE));
            study->description = get_string(item, gdcm::Tag (TAG_STUDY_DESCRIPTION));
            study->child_offset = child_offset.GetValue();
            study->sibling_offset = sibling_offset.GetValue();
            study_offsets[item_offsets[item_index-1]] = study;
```

```
            // Extract series information
            } else if (record_type == "SERIES") {
                auto series = std::make_shared<Series>();
                series->modality = get_string(item, gdcm::Tag (TAG_MODALITY));
                series->description = get_string(item, gdcm::Tag (TAG_SERIES_DESCRIPTION));
                series->child_offset = child_offset.GetValue();
                series->sibling_offset = sibling_offset.GetValue();
                series_offsets[item_offsets[item_index-1]] = series;
            // Extract image information
            } else if (record_type == "IMAGE") {
                auto image = std::make_shared<Image>();
                image->path = get_string(item, gdcm::Tag (TAG_REFERENCED_FILE_ID));
                image->child_offset = child_offset.GetValue();
                image->sibling_offset = sibling_offset.GetValue();
                image_offsets[item_offsets[item_index-1]] = image;
            } else {
                auto other = std::make_shared<Other>();
                other->child_offset = child_offset.GetValue();
                other->sibling_offset = sibling_offset.GetValue();
                other_offsets[item_offsets[item_index-1]] = other;
            }
        }
        // Check validity
        if (patient_offsets.size() == 0)
            throw std::runtime_error("Unable to find patient record");
        reassemble_hierarchy(series_offsets, image_offsets, other_offsets);
        reassemble_hierarchy(study_offsets, series_offsets, other_offsets);
        reassemble_hierarchy(patient_offsets, study_offsets, other_offsets);
        // Set the new root
        for (auto &patient :  patient_offsets) {
            _patients.push_back(patient.second);
        }
    }
    return _patients;
}
//=============================================================================
// Quick test
//=============================================================================
int main(int argc, const char *argv[]) {
    DICOMDIRReader reader;
    try {
        if (argc != 2)
            throw std::runtime_error("Wrong number of arguments");
        auto &patients = reader.load(argv[1]);
        for (auto &patient :  patients) {
            std::cout « "PATIENT" « std::endl;
            std::cout « "NAME: " « patient->name « std::endl;
            std::cout « "ID: " « patient->id « std::endl;
            int x = 0;
            for (auto &study :  patient->children) {
                std::cout « "    STUDY" « std::endl;
                std::cout « "    DESCRIPTION: " « study->description « std::endl;
                std::cout « "    DATE: " « study->date « std::endl;
                for (auto &series :  study->children) {
                    x+=1;
                    std::cout « "        SERIES " « x « std::endl;
                    std::cout « "        DESCRIPTION: " « series->description « std::endl;
                    std::cout « "        MODALITY: " « series->modality « std::endl;
                    for (auto &image :  series->children) {
                        std::cout « "            IMAGE PATH: " « image->path « std::endl;
                    }
                }
            }
        }
    }
    catch (...)  {
        // TODO handle this
        return EXIT_FAILURE;
    }
    return EXIT_SUCCESS;
}
```

12.88 ReadAndPrintAttributes.cxx

```
/*=========================================================================

Program:  GDCM (Grassroots DICOM). A DICOM library
```

```
Copyright (c) 2006-2011 Mathieu Malaterre
All rights reserved.
See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.

This software is distributed WITHOUT ANY WARRANTY; without even
the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
PURPOSE.  See the above copyright notice for more information.

=========================================================================*/
/*
* This small example will show how one can read and print
* a DICOM Attribute using different technique (by tag or by name)
*/
#include "gdcmReader.h"
#include "gdcmGlobal.h"
#include "gdcmDicts.h"
#include "gdcmDict.h"
#include "gdcmAttribute.h"
#include "gdcmStringFilter.h"
#include <iostream>
int main(int argc, char *argv[])
{
  if( argc < 2 )
    {
    std::cerr « argv[0] « " input.dcm" « std::endl;
    return 1;
    }
  const char *filename = argv[1];
  // Instanciate the reader:
  gdcm::Reader reader;
  reader.SetFileName( filename );
  if( !reader.Read() )
    {
    std::cerr « "Could not read:  " « filename « std::endl;
    return 1;
    }
  // The output of gdcm::Reader is a gdcm::File
  gdcm::File &file = reader.GetFile();
  // the dataset is the the set of element we are interested in:
  gdcm::DataSet &ds = file.GetDataSet();
  const gdcm::Global& g = gdcm::Global::GetInstance();
  const gdcm::Dicts &dicts = g.GetDicts();
  const gdcm::Dict &pubdict = dicts.GetPublicDict();
  using namespace gdcm;
  // In this example we will show why using name to lookup attribute can be
  // dangerous.
  Tag tPatientName(0x0,0x0);
  //const DictEntry &de1 =
  pubdict.GetDictEntryByName("Patient Name", tPatientName);
  std::cout « "Found:  " « tPatientName « std::endl;
  // Indeed the attribute could not be found.  Since DICOM 2003, Patient Name
  // has become Patient's Name.
  Tag tPatientsName;
  //const DictEntry &de2 =
  pubdict.GetDictEntryByName("Patient's Name", tPatientsName);
  std::cout « "Found:  " « tPatientsName « std::endl;
  // Let's try to read an arbitrary DICOM Attribute:
  Tag tDoseGridScaling;
  //const DictEntry &de3 =
  pubdict.GetDictEntryByName("Dose Grid Scaling", tDoseGridScaling);
  std::cout « "Found:  " « tDoseGridScaling « std::endl;
  if( ds.FindDataElement( tDoseGridScaling ) )
    {
    gdcm::StringFilter sf;
    sf.SetFile(file);
    std::cout « "Attribute Value as String:  " « sf.ToString( tDoseGridScaling ) « std::endl;
    // Let's check the name again:
    std::pair<std::string, std::string> pss
      = sf.ToStringPair( tDoseGridScaling );
    std::cout « "Attribute Name Checked:  " « pss.first « std::endl;
    std::cout « "Attribute Value (string):  " « pss.second « std::endl;
    //const DataElement &dgs = ds.GetDataElement( tDoseGridScaling );
    // Let's assume for a moment we knew the tag number:
    Attribute<0x3004,0x000e> at;
    assert( at.GetTag() == tDoseGridScaling );
    at.SetFromDataSet( ds );
    // For the sake of long term maintenance, we will not write
    // that this particular attribute is stored as a double.  What if
    // a user made a mistake.  It is much safer to rely on GDCM internal
    // mechanism to deduce the VR::DS type (represented as a ieee double)
```

```
    Attribute<0x3004,0x000e>::ArrayType v = at.GetValue();
    std::cout « "DoseGridScaling=" « v « std::endl;
    }
  return 0;
}
```

12.89 ReadExplicitLengthSQIVR.cxx

```
/*=========================================================================

Program:  GDCM (Grassroots DICOM). A DICOM library

Copyright (c) 2006-2011 Mathieu Malaterre
All rights reserved.
See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.

This software is distributed WITHOUT ANY WARRANTY; without even
the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
PURPOSE.  See the above copyright notice for more information.

=========================================================================*/
#include "gdcmReader.h"
#include "gdcmImplicitDataElement.h"
#include "gdcmDataSet.h"
#include "gdcmPrivateTag.h"
#include "gdcmPrivateTag.h"
#include "gdcmByteValue.h"
#include "gdcmSequenceOfItems.h"
using namespace gdcm;
int main(int argc, char *argv[])
{
  if ( argc < 2 ) return 1;
  const char *filename = argv[1];
  gdcm::Reader r;
  r.SetFileName( filename );
  r.Read();
  //gdcm::PrivateTag pt(0xe1,0x42,"ELSCINT1");
  //gdcm::Tag pt(0x88,0x200);
  gdcm::Tag pt(0x8,0x1140);
  DataSet &ds = r.GetFile().GetDataSet();
  const DataElement &de = ds.GetDataElement( pt );
  std::cout « de « std::endl;
  const ByteValue *bv = de.GetByteValue();
  SmartPointer<SequenceOfItems> sqi = new SequenceOfItems;
  sqi->SetLength( bv->GetLength() );
  std::stringstream ss;
  ss.str( std::string( bv->GetPointer(), bv->GetLength() ) );
  sqi->Read<ImplicitDataElement,SwapperNoOp>( ss );
  std::cout « *sqi « std::endl;
  return 0;
}
```

12.90 ReadGEMSSDO.cxx

```
/*=========================================================================

Program:  GDCM (Grassroots DICOM). A DICOM library

Copyright (c) 2006-2011 Mathieu Malaterre
All rights reserved.
See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.

This software is distributed WITHOUT ANY WARRANTY; without even
the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
PURPOSE.  See the above copyright notice for more information.

=========================================================================*/
#include "gdcmReader.h"
#include "gdcmDataElement.h"
#include "gdcmPrivateTag.h"
#include <iostream>
#include <string>
using namespace gdcm;
```

```
struct SDOElement
{
  typedef std::vector<std::string>::size_type SizeType;
  const char *GetData(SizeType index)const {
    return Data[index].c_str();
  }
  SizeType GetNumberOfData()const {
    return Data.size();
  }
  void SetData(SizeType index, const char *data) {
    Data[index] = data;
  }
  const char *GetDataFormat()const {
    return DataFormat.c_str();
  }
  void SetDataFormat(const char *dataformat, SizeType num) {
    DataFormat = dataformat;
    Data.resize( num );
  }
  void Print( std::ostream &os )const {
    os « DataFormat « ":" « std::endl;
    std::vector<std::string>::const_iterator it = Data.begin();
    size_t s = 0;
    for( ; it != Data.end(); ++it )
      {
      os « "  (" « s++ « ") " « *it « std::endl;
      }
  }
private:
  std::string DataFormat;
  std::vector<std::string> Data;
};
class SDOHeader
{
public:
  typedef std::vector<SDOElement> SDOElements;
  typedef SDOElements::size_type SizeType;
  SizeType GetNumberOfSDOElements()const {
  return InternalSDODataSet.size();
  }
  void AddSDOElement(SDOElement const &sdoelement) {
  InternalSDODataSet.push_back( sdoelement );
  }
  const SDOElement &GetSDOElement(SizeType index)const {
  return InternalSDODataSet[index];
  }
  const SDOElement &GetSDOElementByName(const char *)const {
    return InternalSDODataSet[0];
  }
  void LoadFromAttributes(std::string const &s1, std::string const &s2)
    {
  std::string tok;
  std::string tok2;
  std::stringstream strstr(s1);
  std::stringstream strstr2(s2);
  SDOElement element;
  // Do format
  size_t count = 0;
  while ( std::getline ( strstr2, tok, '\\' ) )
    {
    //std::cout « tok « " ";
    std::getline ( strstr2, tok2, '\\' );
    //std::cout « tok2 « std::endl;
    count += atoi( tok2.c_str() );
    element.SetDataFormat( tok.c_str(), atoi( tok2.c_str() ) );
    for( size_t t = 0; t < element.GetNumberOfData(); ++t )
      {
      std::getline ( strstr, tok, '\\' );
      element.SetData(t, tok.c_str() );
      }
    AddSDOElement( element );
    }
  //while ( std::getline ( strstr, tok, '^' ) )
//  while ( std::getline ( strstr, tok, '\\' ) )
//    {
//    std::cout « tok « std::endl;
//    count++;
//    }
//  std::cout « "Count:  " « count « std::endl;
//  count = 0;
//  std::cout « "Count:  " « count « std::endl;
```

```
    }
  void Print( std::ostream &os )const {
    SDOElements::const_iterator it = InternalSDODataSet.begin();
    for( ; it != InternalSDODataSet.end(); ++it )
      {
      it->Print ( os );
      }
  }
private:
  SDOElements InternalSDODataSet;
};
bool sdo_decode( DataElement const &stringdata, DataElement const &stringdataformat )
{
  const char *sd = stringdata.GetByteValue()->GetPointer();
  const size_t len_sd = stringdata.GetByteValue()->GetLength();
  std::string s1 = std::string( sd, len_sd );
  const char *sdf = stringdataformat.GetByteValue()->GetPointer();
  const size_t len_sdf = stringdataformat.GetByteValue()->GetLength();
  std::string s2 = std::string( sdf, len_sdf );
//  std::cout « s1 « std::endl;
//  std::cout « s2 « std::endl;
  SDOHeader header;
  header.LoadFromAttributes( s1, s2 );
  header.Print( std::cout );
  return true;
}
int main(int argc, char *argv[])
{
  if( argc < 2 )
    {
    std::cerr « argv[0] « " input.dcm" « std::endl;
    return 1;
    }
  const char *filename = argv[1];
  Reader reader;
  reader.SetFileName( filename );
  if( !reader.Read() )
    {
    return 1;
    }
  File &file = reader.GetFile();
  DataSet &ds = file.GetDataSet();
  // StringData (0033,xx1F) 3 "GEMS_GENIE_1" List of SDO parameters stored as
  // list of strings
  const PrivateTag tstringdata(0x33,0x1f,"GEMS_GENIE_1");
  // StringDataFormat (0033,xx23) 3 "GEMS_GENIE_1" Format of string parameters;
  // contains information about name and number of strings in list
  const PrivateTag tstringdataformat(0x33,0x23,"GEMS_GENIE_1");
  if( !ds.FindDataElement( tstringdata ) ) return 1;
  const DataElement& stringdata = ds.GetDataElement( tstringdata );
  if( !ds.FindDataElement( tstringdataformat ) ) return 1;
  const DataElement& stringdataformat = ds.GetDataElement( tstringdataformat );
  sdo_decode( stringdata, stringdataformat );
  return 0;
}
```

12.91 ReadMultiTimesException.cxx

```
/*=========================================================================

Program:  GDCM (Grassroots DICOM). A DICOM library

Copyright (c) 2006-2011 Mathieu Malaterre
All rights reserved.
See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.

This software is distributed WITHOUT ANY WARRANTY; without even
the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
PURPOSE.  See the above copyright notice for more information.

=========================================================================*/
// The intention of this sample program is to provoke bad_alloc exceptions in gdcm code
#include "gdcmImageReader.h"
int main(int argc, char* argv[])
{
  // We pre-allocate some memory (about 1Gb) to help the issue to show up earlier
  char *dummyBuffer = new char[1024*1024*1100]; (void)dummyBuffer;
```

```
  // Check the number of parameters given
  if (argc < 3)
    {
    std::cerr « "Usage:  " « argv[0] « " Filename numberOfTries" « std::endl;
    return 1;
    }
  std::cout « "We are going to read the file:  " « argv[1] « " " « argv[2] « " times" « std::endl;
  // We hold the pointers in an array to avoid the memory to be released
  // We read the input file n-times
  for (int i = 0; i < atoi(argv[2]); ++i)
    {
    gdcm::ImageReader reader;
    std::cout « "Reading try:  " « i « std::endl;
    // Read files
    reader.SetFileName(argv[1]);
    try
      {
      reader.Read();
      gdcm::Image & img = reader.GetImage();
      unsigned long len = img.GetBufferLength();
      char *buffer = new char[ len ];
      img.GetBuffer( buffer ); // do NOT de-allocate buffer !
      }
    catch (std::bad_alloc &ba)
      {
      (void)ba;
      std::cerr « "BAD ALLOC Exception caught!" « std::endl;
      }
    catch (...)
      {
      std::cerr « "Exception caught!" « std::endl;
      }
    }
  return 0;
}
```

12.92 ReadUTF8QtDir.cxx

```
/*=========================================================================

Program:  GDCM (Grassroots DICOM). A DICOM library

Copyright (c) 2006-2011 Mathieu Malaterre
All rights reserved.
See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.

This software is distributed WITHOUT ANY WARRANTY; without even
the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
PURPOSE.  See the above copyright notice for more information.

=========================================================================*/
/*
* GDCM API expect a const char * as input for SetFileName
* In order to use this API from Qt, here is a simple test that
* shows how to do it in a portable manner:
*
* http://doc.qt.nokia.com/latest/qdir.html#navigation-and-directory-operations
*/
#include "gdcmReader.h"
#include "gdcmDirectory.h"
#include <QDir>
#include <QString>
#include <QCoreApplication>
#include <string>
#include <fstream>
#include <stdio.h> // fopen
static int TestBothFuncs(const char *info , const char *ba_str)
{
  int res = 0;
  FILE *f = fopen( ba_str, "r" );
  if( f )
    {
    std::cout « info « " fopen:  " « ba_str « std::endl;
    fclose(f);
    ++res;
    }
  gdcm::Reader reader;
```

```
  std::ifstream is( ba_str, std::ios::binary );
  if( is.is_open() )
    {
    std::cout « info « " is_open:  " « ba_str « std::endl;
    ++res;
    }
  reader.SetStream( is );
  if( reader.CanRead() == true )
    {
    std::cout « info « " SetStream/CanRead:" « ba_str « std::endl;
    ++res;
    }
  is.close();
  reader.SetFileName( ba_str );
  if( reader.CanRead() == true )
    {
    std::cout « info « " SetFileName/CanRead:" « ba_str « std::endl;
    ++res;
    }
  return 4 - res;
}
static int scanFolder(const char dirname[])
{
  int res = 0;
  gdcm::Directory dir;
  unsigned int nfiles = dir.Load( dirname, true );
  const gdcm::Directory::FilenamesType &filenames = dir.GetFilenames();
  for( unsigned int i = 0; i < nfiles; ++i )
    {
    const char *ba_str = filenames[i].c_str();
    res += TestBothFuncs("GDCM",ba_str);
    }
  return res;
}
static int scanFolderQt(QDir const &dir, QStringList& files)
{
  int res = 0;
  QFileInfoList children = dir.entryInfoList(QDir::AllEntries|QDir::NoDotAndDotDot);
  for ( int i=0; i<children.count(); i++ ) {
    QFileInfo file = children.at(i);
    if ( file.isDir() == true ) {
      res += scanFolderQt(QDir(file.absoluteFilePath()), files);
      continue;
    }
    // Convert back from the internal representation to 8bits
    // toLocal8Bit() returns by copy.  Need to store explicitly the QByteArray
    QByteArray str = file.absoluteFilePath().toLocal8Bit();
    const char *ba_str1 = str.constData();
    res += TestBothFuncs("QString", ba_str1);
    }
  return res;
}
int main(int argc, char *argv[])
{
  // very important:
  QCoreApplication qCoreApp( argc , argv );
  if( argc < 2 )
    {
    std::cerr « argv[0] « " dir " « std::endl;
    return 1;
    }
  int res = 0;
  const char *dirname = argv[1];
  res += scanFolder( dirname );
  QDir dir( QString::fromLocal8Bit(dirname) );
  QStringList files;
  res += scanFolderQt( dir, files);
  if( res )
    std::cerr « "Problem with UTF-8" « std::endl;
  else
    std::cerr « "Success with UTF-8" « std::endl;
  return res;
}
```

12.93 SimpleScanner.cxx

```
/*=========================================================================
```

```
Program:  GDCM (Grassroots DICOM). A DICOM library

Copyright (c) 2006-2011 Mathieu Malaterre
All rights reserved.
See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.

This software is distributed WITHOUT ANY WARRANTY; without even
the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
PURPOSE.  See the above copyright notice for more information.

=========================================================================*/
/*
* Simple example to show how to use Scanner API.
* It exposes the three different cases:
*  - DICOM Attribute is present and has a value
*  - DICOM Attribute is present and has no value
*  - DICOM Attribute is not present at all
* It also shows the purpose of the function 'IsKey' to detect whether or
* not the file has been read by the gdcm::Scanner.  Technically most of the time
* if a file is not a 'Key' this is because it is not a DICOM file.  You need to use
* gdcm::System::FileExists to decide whether or not the file actually exist on the disk.
*
* It was tested on this particular image:
* ./SimpleScanner gdcmData/012345.002.050.dcm
*/
#include "gdcmStrictScanner.h"
#include "gdcmSimpleSubjectWatcher.h"
#include "gdcmFileNameEvent.h"
class MyFileWatcher :  public gdcm::SimpleSubjectWatcher
{
public:
  MyFileWatcher(gdcm::Subject * s, const char *comment = ""):
    gdcm::SimpleSubjectWatcher(s,comment){}
  void ShowFileName(gdcm::Subject *, const gdcm::Event &evt)override
{
    const gdcm::FileNameEvent &pe = dynamic_cast<const gdcm::FileNameEvent&>(evt);
    const char *fn = pe.GetFileName();
    std::cout « "FileName:  " « fn « " FileSize:  " « gdcm::System::FileSize( fn ) « std::endl;
    }
};
int main(int argc, char *argv[])
{
  if( argc < 2 )
    {
    return 1;
    }
  const char *filename = argv[1];
  const char filename_invalid[] = "this is a file that may not exist on this disk.dcm";
  gdcm::SmartPointer<gdcm::StrictScanner> sp = new gdcm::StrictScanner;
  gdcm::StrictScanner &s = *sp;
  //gdcm::SimpleSubjectWatcher w(&s, "TestFileName" );
  MyFileWatcher w(&s, "TestFileName" );
  const gdcm::Tag tag_array[] = {
    gdcm::Tag(0x8,0x50),
    gdcm::Tag(0x8,0x51),
    gdcm::Tag(0x8,0x60),
    gdcm::Tag(0x8,0x80),
  };
  s.AddTag( tag_array[0] );
  s.AddTag( tag_array[1] );
  s.AddTag( tag_array[2] );
  s.AddTag( tag_array[3] );
  gdcm::Directory::FilenamesType filenames;
  filenames.push_back( filename );
  filenames.push_back( filename_invalid );
  if( !s.Scan( filenames ) )
    {
    return 1;
    }
  //s.Print( std::cout );
  for(gdcm::Directory::FilenamesType::const_iterator it = filenames.begin();
    it != filenames.end(); ++it )
    {
    if( s.IsKey( it->c_str() ) )
      {
      std::cout « "INFO:" « it->c_str() « " is a proper Key for the Scanner (this is a DICOM file)" « std::endl;
      }
    else
      {
      std::cout « "INFO:" « it->c_str() « " is not a proper Key for the Scanner (this is either not a DICOM file
```

```
      or file does not exist)" « std::endl;
      }
    }
  gdcm::StrictScanner::TagToValue const &ttv = s.GetMapping(filename);
  const gdcm::Tag *ptag = tag_array;
  for( ; ptag != tag_array + 3; ++ptag )
    {
    gdcm::StrictScanner::TagToValue::const_iterator it = ttv.find( *ptag );
    if( it != ttv.end() )
      {
      std::cout « *ptag « " was properly found in this file" « std::endl;
      // it contains a pair of value.  the first one is the actual tag, so the following is always true:
      //  *ptag == it->first
      // The second part is the actual value (stored as RAW strings).  You will have to reinterpret this string
      // if VR for *ptag is not VR::VRASCII !
      const char *value = it->second;
      if( *value )
        {
        std::cout « "  It has the value:  " « value « std::endl;
        }
      else
        {
        std::cout « "  It has no value (empty)" « std::endl;
        }
      }
    else
      {
      std::cout « "Sorry " « *ptag « " could not be found in this file" « std::endl;
      }
    }
  return 0;
}
```

12.94 SortImage.cxx

```
/*=========================================================================

Program:  GDCM (Grassroots DICOM). A DICOM library

Copyright (c) 2006-2011 Mathieu Malaterre
All rights reserved.
See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.

This software is distributed WITHOUT ANY WARRANTY; without even
the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
PURPOSE.  See the above copyright notice for more information.

=========================================================================*/
/*
*/
#include "gdcmSorter.h"
#include "gdcmScanner.h"
#include "gdcmDataSet.h"
#include "gdcmAttribute.h"
bool mysort(gdcm::DataSet const & ds1, gdcm::DataSet const & ds2 )
{
  //gdcm::Attribute<0x0020,0x0013> at1; // Instance Number
  gdcm::Attribute<0x0018,0x1060> at1; // Trigger Time
  gdcm::Attribute<0x0020,0x0032> at11; // Image Position (Patient)
  at1.Set( ds1 );
  at11.Set( ds1 );
  //gdcm::Attribute<0x0020,0x0013> at2;
  gdcm::Attribute<0x0018,0x1060> at2;
  gdcm::Attribute<0x0020,0x0032> at22;
  at2.Set( ds2 );
  at22.Set( ds2 );
  if( at11 == at22 )
    {
    return at1 < at2;
    }
  return at11 < at22;
}
bool mysort_part1(gdcm::DataSet const & ds1, gdcm::DataSet const & ds2 )
{
  gdcm::Attribute<0x0018,0x1060> at1;
  at1.Set( ds1 );
  gdcm::Attribute<0x0018,0x1060> at2;
```

```
  at2.Set( ds2 );
  return at1 < at2;
}
bool mysort_part2(gdcm::DataSet const & ds1, gdcm::DataSet const & ds2 )
{
  gdcm::Attribute<0x0020,0x0032> at1;
  at1.Set( ds1 );
  gdcm::Attribute<0x0020,0x0032> at2;
  at2.Set( ds2 );
  return at1 < at2;
}
// technically all files are in the same Frame of Reference, so this function
// should be a no-op
bool mysort_dummy(gdcm::DataSet const & ds1, gdcm::DataSet const & ds2 )
{
  gdcm::Attribute<0x0020,0x0052> at1; // FrameOfReferenceUID
  at1.Set( ds1 );
  gdcm::Attribute<0x0020,0x0052> at2;
  at2.Set( ds2 );
  return at1 < at2;
}
int main(int argc, char *argv[])
{
  if (argc < 2 ) return 1;
  const char *dirname = argv[1];
  gdcm::Directory dir;
  unsigned int nfiles = dir.Load( dirname );
  dir.Print( std::cout );
  gdcm::Sorter sorter;
  sorter.SetSortFunction( mysort );
  sorter.Sort( dir.GetFilenames() );
  std::cout « "Sorter:" « std::endl;
  sorter.Print( std::cout );
  gdcm::Sorter sorter2;
  sorter2.SetSortFunction( mysort_part1 );
  sorter2.StableSort( dir.GetFilenames() );
  sorter2.SetSortFunction( mysort_part2 );
  sorter2.StableSort( sorter2.GetFilenames() ); // IMPORTANT
  sorter2.SetSortFunction( mysort_dummy );
  sorter2.StableSort( sorter2.GetFilenames() ); // IMPORTANT
  std::cout « "Sorter2:" « std::endl;
  sorter2.Print( std::cout );
  gdcm::Scanner s;
  s.AddTag( gdcm::Tag(0x20,0x32) ); // Image Position (Patient)
  //s.AddTag( gdcm::Tag(0x20,0x37) ); // Image Orientation (Patient)
  s.Scan( dir.GetFilenames() );
  //s.Print( std::cout );
  // Count how many different IPP there are:
  const gdcm::Scanner::ValuesType &values = s.GetValues();
  size_t nvalues = values.size();
  std::cout « "There are " « nvalues « " different type of values" « std::endl;
  //std::cout « "nfiles=" « nfiles « std::endl;
  if( nfiles % nvalues != 0 )
    {
    std::cerr « "Impossible:  this is a not a proper series" « std::endl;
    return 1;
    }
  std::cout « "Series is composed of " « (nfiles/nvalues) « " different 3D volumes" « std::endl;
  return 0;
}
```

12.95 StreamImageReaderTest.cxx

```
/*=========================================================================

Program:  GDCM (Grassroots DICOM). A DICOM library

Copyright (c) 2006-2011 Mathieu Malaterre
All rights reserved.
See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.

This software is distributed WITHOUT ANY WARRANTY; without even
the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
PURPOSE.  See the above copyright notice for more information.

=========================================================================*/
// This work was realised during the GSOC 2011 by Manoj Alwani
```

```
#include "gdcmStreamImageReader.h"
#include "gdcmFileMetaInformation.h"
#include "gdcmSystem.h"
#include "gdcmFilename.h"
#include "gdcmByteSwap.h"
#include "gdcmTrace.h"
#include "gdcmTesting.h"
#include "gdcmImageHelper.h"
#include "gdcmImageReader.h"
#include "gdcmImage.h"
#include "gdcmMediaStorage.h"
#include "gdcmRAWCodec.h"
#include "gdcmJPEGLSCodec.h"
#include "gdcmUIDGenerator.h"
#include "gdcmStreamImageWriter.h"
#include "gdcmAttribute.h"
#include "gdcmFile.h"
#include "gdcmTag.h"
bool StreamImageRead(gdcm::StreamImageWriter & theStreamWriter,
  const char* filename, const char* outfilename, int resolution)
{
  gdcm::StreamImageReader reader;
  reader.SetFileName( filename );
  if (!reader.ReadImageInformation())
    {
    std::cerr « "unable to read image information" « std::endl;
    return 1; //unable to read tags as expected.
    }
  //let's be tricky; each image will be read in portions, first the top half, then the bottom
  //that way, we can test how the stream handles fragmentation of the data
  //we could also loop this to get various different size combinations, but I'm not sure
  //that's useful, yet.
  std::vector<unsigned int> extent =
    gdcm::ImageHelper::GetDimensionsValue(reader.GetFile());
  // std::cout « extent[0];
  //at this point, these values aren't used, but may be in the future
  //unsigned short xmin = 0;
  //unsigned short xmax = extent[0];
  //unsigned short ymin = 0;
  //unsigned short ymax = extent[1];
  //unsigned short zmin = 0;
  //unsigned short zmax = extent[2];
  std::cout« "\n Row:  "«extent[0] «"\n Col :"« extent[1]« "\n Resolution :"« extent[2] « std::endl;
  int a =1;
  for (int i=1; i<=(extent[2]-resolution);++i)
    a = a*2;
  reader.DefinePixelExtent(0, extent[0]/a, 0, extent[1]/a, resolution-1, resolution);
  unsigned long len = reader.DefineProperBufferLength();
  char* finalBuffer = new char[len];
  memset(finalBuffer, 0, sizeof(char)*len);
  if (reader.CanReadImage())
    {
    bool result = reader.Read(finalBuffer, len);
    if( !result )
      {
      std::cout « "res2 failure:" « filename « std::endl;
      delete [] finalBuffer;
      return 1;
      }
    else
      {
      std::cout« "Able to read";
      }
    }
  else
    {
    std::cerr« "Not able to put in buffer"« std::endl;
    }
/*
//now, read in smaller buffer extents
reader.DefinePixelExtent(xmin, xmax, ymin, ymax);
len = reader.DefineProperBufferLength();

char* buffer = new char[len];
bool res2 = reader.Read(buffer, len);
if( !res2 ){
std::cerr « "res2 failure:" « filename « std::endl;
return 1;
}
//copy the result into finalBuffer
memcpy(finalBuffer, buffer, len);
```

```
//now read the next half of the image
ymin = ymax;
ymax = extent[1];

reader.DefinePixelExtent(xmin, xmax, ymin, ymax);

//std::cerr « "Success to read image from file:  " « filename « std::endl;
unsigned long len2 = reader.DefineProperBufferLength();

char* buffer2 = new char[len2];
bool res3 = reader.Read(buffer2, len2);
if( !res3 ){
std::cerr « "res3 failure:" « filename « std::endl;
return 1;
}
//copy the result into finalBuffer
memcpy(&(finalBuffer[len]), buffer2, len2);

delete [] buffer;
delete [] buffer2;
*/
  gdcm::Writer w;
  gdcm::File &file = w.GetFile();
  gdcm::DataSet &ds = file.GetDataSet();
  file.GetHeader().SetDataSetTransferSyntax( gdcm::TransferSyntax::ExplicitVRLittleEndian );
  gdcm::UIDGenerator uid;
  gdcm::DataElement de( gdcm::Tag(0x8,0x18) ); // SOP Instance UID
  de.SetVR( gdcm::VR::UI );
  const char *u = uid.Generate();
  de.SetByteValue( u, strlen(u) );
  ds.Insert( de );
  gdcm::DataElement de1( gdcm::Tag(0x8,0x16) );
  de1.SetVR( gdcm::VR::UI );
  gdcm::MediaStorage ms( gdcm::MediaStorage::VLWholeSlideMicroscopyImageStorage );
  de1.SetByteValue( ms.GetString(), strlen(ms.GetString()));
  ds.Insert( de1 );
  const char mystr[] = "MONOCHROME2 ";
  gdcm::DataElement de2( gdcm::Tag(0x28,0x04) );
  //de.SetTag(gdcm::Tag(0x28,0x04));
  de2.SetVR( gdcm::VR::CS );
  de2.SetByteValue(mystr, strlen(mystr));
  ds.Insert( de2 );
   gdcm::Attribute<0x0028,0x0008> Number_Of_Frames = {1};
   ds.Insert( Number_Of_Frames.GetAsDataElement() );
   gdcm::Attribute<0x0028,0x0010> row = {extent[0]/a};//
   ds.Insert( row.GetAsDataElement() );
   gdcm::Attribute<0x0028,0x0011> col = {extent[1]/a};//
   ds.Insert( col.GetAsDataElement() );
   gdcm::Attribute<0x0028,0x0100> at = {8};
   ds.Insert( at.GetAsDataElement() );
   gdcm::Attribute<0x0028,0x0002> at1 = {1};//
   ds.Insert( at1.GetAsDataElement() );
   gdcm::Attribute<0x0028,0x0101> at2 = {8};
   ds.Insert( at2.GetAsDataElement() );
   gdcm::Attribute<0x0028,0x0102> at3 = {7};
   ds.Insert( at3.GetAsDataElement() );
    /*
ds1.Remove( gdcm::Tag(0x0028,0x0008) );

gdcm::Attribute<0x0028,0x0008> Number_Of_Frames = {1};
ds1.Insert( Number_Of_Frames.GetAsDataElement() );
*/
   theStreamWriter.SetFile(file);
   if (!theStreamWriter.WriteImageInformation())
     {
     std::cerr « "unable to write image information" « std::endl;
     return 1; //the CanWrite function should prevent getting here, else,
     //that's a test failure
     }
    std::vector<unsigned int> extent1 = gdcm::ImageHelper::GetDimensionsValue(file);
    unsigned short xmax = extent1[0];
    unsigned short ymax = extent1[1];
    unsigned short theChunkSize = 1;
    unsigned short ychunk = extent1[1]/theChunkSize; //go in chunk sizes of theChunkSize
    unsigned short zmax = 1;
    std::cout« "\n Row:  "«extent1[0] «"\n Col :"« extent1[1]« "\n Resolution :"« extent1[2] « std::endl;
    if (xmax == 0 || ymax == 0)
      {
      std::cerr « "Image has no size, unable to write zero-sized image." « std::endl;
      return 0;
```

```
      }
    int z, y, nexty;
    unsigned long prevLen = 0; //when going through the char buffer, make sure to grab
    //the bytes sequentially.    So, store how far you got in the buffer with each iteration.
    for (z = 0; z < zmax; ++z){
      for (y = 0; y < ymax; y += ychunk){
        nexty = y + ychunk;
        if (nexty > ymax) nexty = ymax;
        theStreamWriter.DefinePixelExtent(0, xmax, y, nexty, z, z+1);
        unsigned long len = theStreamWriter.DefineProperBufferLength();
        std::cout « "\n" «len;
        char* finalBuffer1 = new char[len];
        memcpy(finalBuffer1, &(finalBuffer[prevLen]), len);
        std::cout « "\nable to write";
        if (!theStreamWriter.Write(finalBuffer1, len)){
          std::cerr « "writing failure:" « "output.dcm" « " at y = " « y « " and z= " « z « std::endl;
          delete [] finalBuffer1;
          delete [] finalBuffer;
          return 1;
        }
        delete [] finalBuffer1;
        prevLen += len;
      }
    }
   delete [] finalBuffer;
   std::cout « "all is set";
  return true;
}
int main(int argc, char *argv[])
{
  if( argc < 3 )
    {
    std::cerr « argv[0] « " input.dcm output.dcm Resolution" « std::endl;
    return 1;
    }
  const char *filename = argv[1];
  const char *outfilename = argv[2];
  char *res = argv[3];
  int resolution = atoi(res);
  gdcm::StreamImageWriter theStreamWriter;
  std::ofstream of;
  of.open( outfilename, std::ios::out | std::ios::binary );
  theStreamWriter.SetStream(of);
  // else
  // First of get rid of warning/debug message
  gdcm::Trace::DebugOn();
  gdcm::Trace::WarningOn();
  if(!StreamImageRead( theStreamWriter, filename, outfilename, resolution))
    return 1;
  uint16_t firstTag1 =  0xfffe;
  uint16_t secondTag1 = 0xe0dd;
  uint32_t thirdTag1 =  0x00000000;
  //uint16_t fourthTag1 = 0xffff;
  const int theBufferSize1 = 2*sizeof(uint16_t)+sizeof(uint32_t);
  char* tmpBuffer2 = new char[theBufferSize1];
  memcpy(&(tmpBuffer2[0]), &firstTag1, sizeof(uint16_t));
  memcpy(&(tmpBuffer2[sizeof(uint16_t)]), &secondTag1, sizeof(uint16_t));
  memcpy(&(tmpBuffer2[2*sizeof(uint16_t)]), &thirdTag1, sizeof(uint32_t));
  //memcpy(&(tmpBuffer2[3*sizeof(uint16_t)]), &fourthTag1, sizeof(uint16_t));
  assert( of && !of.eof() && of.good() );
  of.write(tmpBuffer2, theBufferSize1);
  of.flush();
  assert( of );
  return 0;
}
```

12.96 TemplateEmptyImage.cxx

```
/*=========================================================================

Program:  GDCM (Grassroots DICOM). A DICOM library

Copyright (c) 2006-2011 Mathieu Malaterre
All rights reserved.
See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.

This software is distributed WITHOUT ANY WARRANTY; without even
```

```
the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
PURPOSE.  See the above copyright notice for more information.

=========================================================================*/
#include "gdcmFileStreamer.h"
#include "gdcmTag.h"
#include "gdcmTrace.h"
#include "gdcmImageRegionReader.h"
#include "gdcmImageHelper.h"
#include "gdcmWriter.h"
#include "gdcmImageWriter.h"
#include "gdcmTagKeywords.h"
#include "gdcmUIDGenerator.h"
int main(int argc, char *argv[])
{
  if( argc < 2 ) return 1;
  const char * filename = argv[1];
  gdcm::ImageRegionReader irr;
  irr.SetFileName( filename );
  const bool b3 = irr.ReadInformation();
  std::cout << b3 << std::endl;
  gdcm::Image & img = irr.GetImage();
  std::cout << img << std::endl;
  //  const gdcm::Region & r = irr.GetRegion();
  //  std::cout << r << std::endl;
  gdcm::ImageWriter w;
  gdcm::File & file = w.GetFile();
  gdcm::DataSet & ds = file.GetDataSet();
  gdcm::UIDGenerator uid;
  namespace kwd = gdcm::Keywords;
  kwd::FrameOfReferenceUID frameref;
  frameref.SetValue( uid.Generate() );
  // ContentDate
  char date[22];
  const size_t datelen = 8;
  int res = gdcm::System::GetCurrentDateTime(date);
  (void)res;
  kwd::ContentDate contentdate;
  // Do not copy the whole cstring:
  contentdate.SetValue( gdcm::DAComp( date, datelen ) );
  ds.Insert( contentdate.GetAsDataElement() );
  // ContentTime
  const size_t timelen = 6 + 1 + 6; // time + milliseconds
  kwd::ContentTime contenttime;
  // Do not copy the whole cstring:
  contenttime.SetValue( gdcm::TMComp(date+datelen, timelen) );
  ds.Insert( contenttime.GetAsDataElement() );
  gdcm::MediaStorage ms0 = w.ComputeTargetMediaStorage();
  std::cout << ms0 << std::endl;
  kwd::SeriesNumber seriesnumber = { 1 };
  kwd::InstanceNumber instancenum = { 1 };
  kwd::StudyID studyid = { "St1" };
  kwd::PatientID patientid = { "P1" };
  kwd::SOPClassUID sopclassuid;
  kwd::PositionReferenceIndicator pri;
  //kwd::Laterality lat;
  //kwd::BodyPartExamined bodypartex = { "HEAD" };
  kwd::BodyPartExamined bodypartex = { "ANKLE" };
  kwd::PatientOrientation pator;
  kwd::BurnedInAnnotation bia = { "NO" };
  kwd::ConversionType convtype = { "SYN" };
  kwd::PresentationLUTShape plutshape = { "IDENTITY" }; // MONOCHROME2
  // gdcm will pick the Word in case Byte class is not compatible:
  gdcm::MediaStorage ms = gdcm::MediaStorage::MultiframeGrayscaleByteSecondaryCaptureImageStorage;
  sopclassuid.SetValue( ms.GetString() );
  ds.Insert( instancenum.GetAsDataElement() );
  ds.Insert( sopclassuid.GetAsDataElement() );
  ds.Insert( seriesnumber.GetAsDataElement() );
  ds.Insert( patientid.GetAsDataElement() );
  ds.Insert( studyid.GetAsDataElement() );
  ds.Insert( frameref.GetAsDataElement() );
  ds.Insert( pri.GetAsDataElement() );
  //ds.Insert( lat.GetAsDataElement() );
  ds.Insert( bodypartex.GetAsDataElement() );
  ds.Insert( pator.GetAsDataElement() );
  ds.Insert( bia.GetAsDataElement() );
  ds.Insert( convtype.GetAsDataElement() );
  ds.Insert( plutshape.GetAsDataElement() );
  //    gdcm::MediaStorage ms1 = w.ComputeTargetMediaStorage();
  //    std::cout << ms1 << std::endl;
  std::cout << ds << std::endl;
```

```
  gdcm::PixelFormat & pf = img.GetPixelFormat();
  pf.SetPixelRepresentation(0); // always overwrite
  img.SetSlope(1);
  img.SetIntercept(0);
  w.SetImage( img );
  w.SetFileName( "TemplateImage.dcm" );
  if( !w.Write() )
  {
    return 1;
  }
  return 0;
}
```

12.97 TraverseModules.cxx

```
/*=========================================================================

Program:  GDCM (Grassroots DICOM). A DICOM library

Copyright (c) 2006-2011 Mathieu Malaterre
All rights reserved.
See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.

This software is distributed WITHOUT ANY WARRANTY; without even
the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
PURPOSE.  See the above copyright notice for more information.

=========================================================================*/
/*
*/
#include "gdcmDefs.h"
#include "gdcmGlobal.h"
#include "gdcmIODs.h"
#include "gdcmIOD.h"
#include "gdcmMacros.h"
#include "gdcmIODEntry.h"
#include "gdcmModules.h"
#include "gdcmModule.h"
#include "gdcmAnonymizer.h"
#include "gdcmDicts.h"
int main(int , char *[])
{
  using namespace gdcm;
  static Global &g = Global::GetInstance();
  if( !g.LoadResourcesFiles() )
    {
    return 1;
    }
  static const Defs &defs = g.GetDefs();
  static const Modules &modules = defs.GetModules();
  static const IODs &iods = defs.GetIODs();
  static const Macros &macros = defs.GetMacros();
  static const Dicts &dicts = g.GetDicts();
  std::vector<Tag> tags = gdcm::Anonymizer::GetBasicApplicationLevelConfidentialityProfileAttributes();
  for( std::vector<Tag>::const_iterator tit = tags.begin(); tit != tags.end(); ++tit )
    {
    const Tag &tag = *tit;
    const DictEntry &dictentry = dicts.GetDictEntry(tag);
    std::cout << "Processing Attribute:  " << tag << " " << dictentry << std::endl;
    IODs::IODMapTypeConstIterator it = iods.Begin();
    for( ; it != iods.End(); ++it )
      {
      const IODs::IODName &name = it->first;
      const IOD &iod = it->second;
      const size_t niods = iod.GetNumberOfIODs();
      // Iterate over each iod entry in order:
      for(unsigned int idx = 0; idx < niods; ++idx)
        {
        const IODEntry &iodentry = iod.GetIODEntry(idx);
        const char *ref = iodentry.GetRef();
        //Usage::UsageType ut = iodentry.GetUsageType();
        const Module &module = modules.GetModule( ref );
        if( module.FindModuleEntryInMacros(macros, tag ) )
          {
          const ModuleEntry &module_entry = module.GetModuleEntryInMacros(macros,tag);
          Type type = module_entry.GetType();
```

```
          std::cout « "IOD Name:  " « name « std::endl;
          std::cout « "Type:  " « type « std::endl;
          }
        }
      }
    }
  return 0;
}
```

12.98 VolumeSorter.cxx

```
/*=========================================================================

Program:  GDCM (Grassroots DICOM). A DICOM library

Copyright (c) 2006-2011 Mathieu Malaterre
All rights reserved.
See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.

This software is distributed WITHOUT ANY WARRANTY; without even
the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
PURPOSE.  See the above copyright notice for more information.

=========================================================================*/
/*
*/
#include "gdcmSorter.h"
#include "gdcmIPPSorter.h"
#include "gdcmScanner.h"
#include "gdcmDataSet.h"
#include "gdcmAttribute.h"
#include "gdcmTesting.h"
bool mysort1(gdcm::DataSet const & ds1, gdcm::DataSet const & ds2 )
{
  gdcm::Attribute<0x0020,0x000d> at1;
  at1.Set( ds1 );
  gdcm::Attribute<0x0020,0x000d> at2;
  at2.Set( ds2 );
  return at1 < at2;
}
bool mysort2(gdcm::DataSet const & ds1, gdcm::DataSet const & ds2 )
{
  gdcm::Attribute<0x0020,0x000e> at1;
  at1.Set( ds1 );
  gdcm::Attribute<0x0020,0x000e> at2;
  at2.Set( ds2 );
  return at1 < at2;
}
bool mysort3(gdcm::DataSet const & ds1, gdcm::DataSet const & ds2 )
{
  // This is a floating point number is the comparison ok ?
  gdcm::Attribute<0x0020,0x0037> at1;
  at1.Set( ds1 );
  gdcm::Attribute<0x0020,0x0037> at2;
  at2.Set( ds2 );
  return at1 < at2;
}
bool mysort4(gdcm::DataSet const & ds1, gdcm::DataSet const & ds2 )
{
  // Do the IPP sorting here
  gdcm::Attribute<0x0020,0x0032> ipp1;
  gdcm::Attribute<0x0020,0x0037> iop1;
  ipp1.Set( ds1 );
  iop1.Set( ds1 );
  gdcm::Attribute<0x0020,0x0032> ipp2;
  gdcm::Attribute<0x0020,0x0037> iop2;
  ipp2.Set( ds2 );
  iop2.Set( ds2 );
  if( iop1 != iop2 )
    {
    return false;
    }
  // else
  double normal[3];
  normal[0] = iop1[1]*iop1[5] - iop1[2]*iop1[4];
  normal[1] = iop1[2]*iop1[3] - iop1[0]*iop1[5];
  normal[2] = iop1[0]*iop1[4] - iop1[1]*iop1[3];
```

```
  double dist1 = 0;
  for (int i = 0; i < 3; ++i) dist1 += normal[i]*ipp1[i];
  double dist2 = 0;
  for (int i = 0; i < 3; ++i) dist2 += normal[i]*ipp2[i];
  std::cout « dist1 « "," « dist2 « std::endl;
  return dist1 < dist2;
}
int main(int argc, char *argv[])
{
  const char *extradataroot = gdcm::Testing::GetDataExtraRoot();
  std::string dir1;
  if( argc < 2 )
    {
    if( !extradataroot )
      {
      return 1;
      }
    dir1 = extradataroot;
    dir1 += "/gdcmSampleData/ForSeriesTesting/VariousIncidences/ST1";
    }
  else
    {
    dir1 = argv[1];
    }
  gdcm::Directory d;
  d.Load( dir1.c_str(), true ); // recursive !
  const gdcm::Directory::FilenamesType &l1 = d.GetFilenames();
  const size_t nfiles = l1.size();
  std::cout « nfiles « std::endl;
  //if( nfiles != 280 )
  //  {
  //  return 1;
  //  }
  //d.Print( std::cout );
  gdcm::Scanner s0;
  const gdcm::Tag t1(0x0020,0x000d); // Study Instance UID
  const gdcm::Tag t2(0x0020,0x000e); // Series Instance UID
  //const gdcm::Tag t3(0x0010,0x0010); // Patient's Name
  s0.AddTag( t1 );
  s0.AddTag( t2 );
  //s0.AddTag( t3 );
  //s0.AddTag( t4 );
  //s0.AddTag( t5 );
  //s0.AddTag( t6 );
  bool b = s0.Scan( d.GetFilenames() );
  if( !b )
    {
    std::cerr « "Scanner failed" « std::endl;
    return 1;
    }
  //s0.Print( std::cout );
  // Only get the DICOM files:
  gdcm::Directory::FilenamesType l2 = s0.GetKeys();
  const size_t nfiles2 = l2.size();
  std::cout « nfiles2 « std::endl;
  if ( nfiles2 > nfiles )
    {
    return 1;
    }
  gdcm::Sorter sorter;
  sorter.SetSortFunction( mysort1 );
  sorter.StableSort( l2 );
  sorter.SetSortFunction( mysort2 );
  sorter.StableSort( sorter.GetFilenames() );
  sorter.SetSortFunction( mysort3 );
  sorter.StableSort( sorter.GetFilenames() );
  sorter.SetSortFunction( mysort4 );
  sorter.StableSort( sorter.GetFilenames() );
  //sorter.Print( std::cout );
  // Let's try to check our result:
  // assume that IPP is precise enough so that we can test floating point equality:
  size_t nvalues = 0;
{
  gdcm::Scanner s;
  s.AddTag( gdcm::Tag(0x20,0x32) ); // Image Position (Patient)
  //s.AddTag( gdcm::Tag(0x20,0x37) ); // Image Orientation (Patient)
  s.Scan( d.GetFilenames() );
  //s.Print( std::cout );
  const gdcm::Scanner::ValuesType &values = s.GetValues();
  nvalues = values.size();
  std::cout « "There are " « nvalues « " different type of values" « std::endl;
```

```
  assert( nfiles2 % nvalues == 0 );
  std::cout « "Series is composed of " « (nfiles/nvalues) « " different 3D volumes" « std::endl;
}
  gdcm::Directory::FilenamesType sorted_files = sorter.GetFilenames();
  // Which means we can take nvalues files at a time and execute gdcm::IPPSorter on it:
  gdcm::IPPSorter ippsorter;
  gdcm::Directory::FilenamesType sub( sorted_files.begin(), sorted_files.begin() + nvalues);
  std::cout « sub.size() « std::endl;
  std::cout « sub[0] « std::endl;
  std::cout « sub[nvalues-1] « std::endl;
  ippsorter.SetComputeZSpacing( false );
  if( !ippsorter.Sort( sub ) )
    {
    std::cerr « "Could not sort" « std::endl;
    return 1;
    }
  std::cout « "IPPSorter:" « std::endl;
  ippsorter.Print( std::cout );
  return 0;
}
```

12.99 csa2img.cxx

```
/*=========================================================================

Program:  GDCM (Grassroots DICOM). A DICOM library

Copyright (c) 2006-2011 Mathieu Malaterre
All rights reserved.
See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.

This software is distributed WITHOUT ANY WARRANTY; without even
the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
PURPOSE.  See the above copyright notice for more information.

=========================================================================*/
/*
* I do not know what the format is, just guessing from info found on the net:
*
* http://atonal.ucdavis.edu/matlab/fmri/spm5/spm_dicom_convert.m
*
* This example is an attempt at understanding the format used by SIEMENS
* their "SIEMENS CSA NON-IMAGE" DICOM file (1.3.12.2.1107.5.9.1)
*
* Everything done in this code is for the sole purpose of writing interoperable
* software under Sect.  1201 (f) Reverse Engineering exception of the DMCA.
* If you believe anything in this code violates any law or any of your rights,
* please contact us (gdcm-developers@lists.sourceforge.net) so that we can
* find a solution.
*
*/
#include "gdcmReader.h"
#include "gdcmImageReader.h"
#include "gdcmImageWriter.h"
#include "gdcmCSAHeader.h"
#include "gdcmAttribute.h"
#include "gdcmPrivateTag.h"
#include <math.h>
int main(int argc, char *argv [])
{
  if( argc < 2 ) return 1;
  // gdcmDataExtra/gdcmNonImageData/exCSA_Non-Image_Storage.dcm
  // PHANTOM.MR.CARDIO_COEUR_S_QUENCE_DE_REP_RAGE.9.257.2008.03.20.14.53.25.578125.43151705.IMA
  const char *filename = argv[1];
  gdcm::Reader reader; // Do not use ImageReader
  reader.SetFileName( filename );
  if( !reader.Read() )
    {
    std::cerr « "Failed to read:  " « filename « std::endl;
    return 1;
    }
  gdcm::CSAHeader csa;
  const gdcm::DataSet& ds = reader.GetFile().GetDataSet();
  const gdcm::PrivateTag &t1 = csa.GetCSAImageHeaderInfoTag();
  //std::cout « t1 « std::endl;
  //const gdcm::PrivateTag &t2 = csa.GetCSASeriesHeaderInfoTag();
  if( ds.FindDataElement( t1 ) )
```

```
  {
  csa.LoadFromDataElement( ds.GetDataElement( t1 ) );
  csa.Print( std::cout );
  }
int dims[2] = {};
if( csa.FindCSAElementByName( "Columns" ) )
  {
  const gdcm::CSAElement &csael = csa.GetCSAElementByName( "Columns" );
  std::cout « csael « std::endl;
  //const gdcm::ByteValue *bv = csael.GetByteValue();
  gdcm::Element<gdcm::VR::IS, gdcm::VM::VM1> el;
  el.Set( csael.GetValue() );
  dims[0] = el.GetValue();
  std::cout « "Columns:" « el.GetValue() « std::endl;
  }
if( csa.FindCSAElementByName( "Rows" ) )
  {
  const gdcm::CSAElement &csael2 = csa.GetCSAElementByName( "Rows" );
  std::cout « csael2 « std::endl;
  gdcm::Element<gdcm::VR::IS, gdcm::VM::VM1> el2;
  el2.Set( csael2.GetValue() );
  dims[1] = el2.GetValue();
  std::cout « "Rows:" « el2.GetValue() « std::endl;
  }
double spacing[2] = { 1.  , 1.  };
bool spacingfound = false;
if( csa.FindCSAElementByName( "PixelSpacing" ) )
  {
  const gdcm::CSAElement &csael3 = csa.GetCSAElementByName( "PixelSpacing" );
  if( !csael3.IsEmpty() )
    {
    std::cout « csael3 « std::endl;
    gdcm::Element<gdcm::VR::DS, gdcm::VM::VM2> el3;
    el3.Set( csael3.GetValue() );
    spacing[0] = el3.GetValue(0);
    spacing[1] = el3.GetValue(1);
    std::cout « "PixelSpacing:" « el3.GetValue() « "," « el3.GetValue(1) « std::endl;
    spacingfound = true;
    }
  }
if( !spacingfound )
  {
  std::cerr « "Problem with PixelSpacing" « std::endl;
  //return 1;
  }
if( !dims[0] || !dims[1] )
  {
  std::cerr « "Problem with dims" « std::endl;
  return 1;
  }
gdcm::ImageWriter writer;
gdcm::Image &image = writer.GetImage();
image.SetNumberOfDimensions( 2 ); // good default
image.SetDimension(0, dims[0] );
image.SetDimension(1, dims[1] );
image.SetSpacing(0, spacing[0] );
image.SetSpacing(1, spacing[1] );
gdcm::PixelFormat pixeltype = gdcm::PixelFormat::INT16; // bytepix = spm_type('int16','bits')/8;
//unsigned long l = image.GetBufferLength();
//const int p =  l / (dims[0] * dims[1]);
//image.SetNumberOfDimensions( 3 );
//image.SetDimension(2, p / pixeltype.GetPixelSize() );
gdcm::PhotometricInterpretation pi;
pi = gdcm::PhotometricInterpretation::MONOCHROME2;
//pixeltype.SetSamplesPerPixel(  );
image.SetPhotometricInterpretation( pi );
image.SetPixelFormat( pixeltype );
//image.SetIntercept( inputimage.GetIntercept() );
//image.SetSlope( inputimage.GetSlope() );
//gdcm::DataElement pixeldata( gdcm::Tag(0x7fe1,0x1010) );
//pixeldata.SetByteValue( &outbuf[0], outbuf.size() );
gdcm::PrivateTag csanonimaget(0x7fe1,0x10,"SIEMENS CSA NON-IMAGE");
const gdcm::DataElement &pixeldata = ds.GetDataElement( csanonimaget );
image.SetDataElement( pixeldata );
std::string outfilename = "outcsa.dcm";
//writer.SetFile( reader.GetFile() );
writer.SetFileName( outfilename.c_str() );
if( !writer.Write() )
  {
  std::cerr « "could not write:  " « outfilename « std::endl;
  return 1;
```

```
    }
  return 0;
}
```

12.100 iU22tomultisc.cxx

```
/*=========================================================================

Program:  GDCM (Grassroots DICOM). A DICOM library

Copyright (c) 2006-2011 Mathieu Malaterre
All rights reserved.
See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.

This software is distributed WITHOUT ANY WARRANTY; without even
the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
PURPOSE.  See the above copyright notice for more information.

=========================================================================*/
/*
* iU22 Raw Data extractor
*/
#include "gdcmReader.h"
#include "gdcmImageWriter.h"
#include "gdcmAttribute.h"
#include "gdcmPrivateTag.h"
#include <math.h>
int main(int argc, char *argv [])
{
  if( argc < 2 ) return 1;
  // IM_001
  const char *filename = argv[1];
  gdcm::Reader reader; // Do not use ImageReader
  reader.SetFileName( filename );
  if( !reader.Read() )
    {
    std::cerr « "Failed to read:  " « filename « std::endl;
    return 1;
    }
 // * The data is simply 8-bit unsigned in the obvious x/y/z order
 // * 200D,300B contains the data
 // * 200D,3001 contains the no.  of voxels (416,412,256 in this case)
 // * 200D,3003 contains the voxel sizes (0.156184527398215 /
 // 0.1223749613981957 / 0.328479990704639 in this case)
  const gdcm::File &file = reader.GetFile();
  const gdcm::DataSet &ds = file.GetDataSet();
  const gdcm::PrivateTag trawdataus( 0x200d, 0x0b, "Philips US Imaging DD 033" );
  const gdcm::DataElement &rawdataus = ds.GetDataElement( trawdataus );
  const gdcm::PrivateTag tcolsrowsframes( 0x200d, 0x01, "Philips US Imaging DD 036" );
  const gdcm::DataElement &colsrowsframes = ds.GetDataElement( tcolsrowsframes );
  // const gdcm::PrivateTag tcolsrowsframes( 0x200d, 0x02, "Philips US Imaging DD 036" );
  // this is just a duplicate previous tag.
  const gdcm::PrivateTag tvoxelspacing( 0x200d, 0x03, "Philips US Imaging DD 036" );
  const gdcm::DataElement &voxelspacing = ds.GetDataElement( tvoxelspacing );
  gdcm::Element<gdcm::VR::DS,gdcm::VM::VM3> dims; // Use DS to interpret value stored in LO
  dims.SetFromDataElement( colsrowsframes );
  gdcm::Element<gdcm::VR::DS,gdcm::VM::VM3> spacing;
  spacing.SetFromDataElement( voxelspacing );
  gdcm::ImageWriter writer;
  gdcm::Image &image = writer.GetImage();
  image.SetNumberOfDimensions( 3 ); // good default
  image.SetDimension(0, (unsigned int)dims[0] );
  image.SetDimension(1, (unsigned int)dims[1] );
  image.SetDimension(2, (unsigned int)dims[2] );
  image.SetSpacing(0, spacing[0] );
  image.SetSpacing(1, spacing[1] );
  image.SetSpacing(2, spacing[2] );
  gdcm::PixelFormat pixeltype = gdcm::PixelFormat::UINT8;
  gdcm::PhotometricInterpretation pi;
  pi = gdcm::PhotometricInterpretation::MONOCHROME2;
  image.SetPhotometricInterpretation( pi );
  image.SetPixelFormat( pixeltype );
  image.SetDataElement( rawdataus );
  std::string outfilename = "outiu22.dcm";
  gdcm::DataElement de( gdcm::Tag(0x8,0x16) ); // SOP Class UID
  de.SetVR( gdcm::VR::UI );
  gdcm::MediaStorage ms(
```

```
    gdcm::MediaStorage::UltrasoundMultiFrameImageStorage );
//    gdcm::MediaStorage::MultiframeGrayscaleByteSecondaryCaptureImageStorage );
  de.SetByteValue( ms.GetString(), (uint32_t)strlen(ms.GetString()));
  writer.GetFile().GetDataSet().Replace( de );
  writer.SetFileName( outfilename.c_str() );
  if( !writer.Write() )
    {
    std::cerr « "could not write:  " « outfilename « std::endl;
    return 1;
    }
  return 0;
}
```

12.101 pmsct_rgb1.cxx

```
/*=========================================================================

Program:  GDCM (Grassroots DICOM). A DICOM library

Copyright (c) 2006-2011 Mathieu Malaterre
All rights reserved.
See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.

This software is distributed WITHOUT ANY WARRANTY; without even
the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
PURPOSE.  See the above copyright notice for more information.

=========================================================================*/
/*
* This example shows how to rewrite a ELSCINT1/PMSCT_RGB1 compressed
* image so that it is readable by most 3rd party software (DICOM does
* not specify this particular encoding).
* This is required for the sake of interoperability with any standard
* conforming DICOM system.
*
* Everything done in this code is for the sole purpose of writing interoperable
* software under Sect.  1201 (f) Reverse Engineering exception of the DMCA.
* If you believe anything in this code violates any law or any of your rights,
* please contact us (gdcm-developers@lists.sourceforge.net) so that we can
* find a solution.
*
* Everything you do with this code is at your own risk, since decompression
* algorithm was not written from specification documents.
*
* Special thanks to:
* Jean-Pierre Roux for providing the sample datasets
*/
#include "gdcmReader.h"
#include "gdcmPrivateTag.h"
#include "gdcmAttribute.h"
#include "gdcmImageWriter.h"
void delta_decode(const unsigned char *data_in, size_t data_size,
  std::vector<unsigned char> &new_stream, unsigned short pc, size_t w, size_t h)
{
  const size_t plane_size = h * w;
  const size_t outputlen = 3 * plane_size;
  new_stream.resize( outputlen );
  assert( data_size != outputlen );
  if( data_size == outputlen )
    {
    return;
    }
  typedef unsigned char byte;
  enum {
    COLORMODE  = 0x81,
    ESCMODE    = 0x82,
    REPEATMODE = 0x83
  };
  const byte* src = (const byte*)data_in;
  byte* dest = (byte*)&new_stream[0];
  union { byte gray; byte rgb[3]; } pixel;
  pixel.rgb[0] = pixel.rgb[1] = pixel.rgb[2] = 0;
  // always start in grayscale mode
  bool graymode = true;
  size_t dx = 1;
  size_t dy = 3;
  // algorithm works with both planar configuration
```

```
// It does produce surprising greenish background color for planar
// configuration is 0, while the nested Icon SQ display a nice black
// background
if (pc)
  {
  dx = plane_size;
  dy = 1;
  }
size_t ps = plane_size;
// The following is highly unoptimized as we have nested if statement in a while loop
// we need to switch from one algorithm to ther other (RGB <-> GRAY)
while (ps)
  {
  // next byte:
  byte b = *src++;
  assert( src < data_in + data_size );
  // mode selection:
  switch ( b )
    {
  case ESCMODE:
    // Used to treat a byte 81/82/83 as a normal byte
    if (graymode)
      {
      pixel.gray += *src++;
      dest[0*dx] = pixel.gray;
      dest[1*dx] = pixel.gray;
      dest[2*dx] = pixel.gray;
      }
    else
      {
      pixel.rgb[0] += *src++;
      pixel.rgb[1] += *src++;
      pixel.rgb[2] += *src++;
      dest[0*dx] = pixel.rgb[0];
      dest[1*dx] = pixel.rgb[1];
      dest[2*dx] = pixel.rgb[2];
      }
    dest += dy;
    ps--;
    break;
  case REPEATMODE:
    // repeat mode (RLE)
    b = *src++;
    ps -= b;
    if (graymode)
      {
      while (b-- > 0)
        {
        dest[0*dx] = pixel.gray;
        dest[1*dx] = pixel.gray;
        dest[2*dx] = pixel.gray;
        dest += dy;
        }
      }
    else
      {
      while (b-- > 0)
        {
        dest[0*dx] = pixel.rgb[0];
        dest[1*dx] = pixel.rgb[1];
        dest[2*dx] = pixel.rgb[2];
        dest += dy;
        }
      }
    break;
  case COLORMODE:
    // We are swithing from one mode to the other.  The stream contains an intermixed
    // compression of RGB codec and GRAY codec.  Each one not knowing of the other
    // reset old value to 0.
    if (graymode)
      {
      graymode = false;
      pixel.rgb[0] = pixel.rgb[1] = pixel.rgb[2] = 0;
      }
    else
      {
      graymode = true;
      pixel.gray = 0;
      }
    break;
  default:
```

```
      // This is identical to ESCMODE, it would be nicer to use fall-through
      if (graymode)
        {
        pixel.gray += b;
        dest[0*dx] = pixel.gray;
        dest[1*dx] = pixel.gray;
        dest[2*dx] = pixel.gray;
        }
      else
        {
        pixel.rgb[0] += b;
        pixel.rgb[1] += *src++;
        pixel.rgb[2] += *src++;
        dest[0*dx] = pixel.rgb[0];
        dest[1*dx] = pixel.rgb[1];
        dest[2*dx] = pixel.rgb[2];
        }
      dest += dy;
      ps--;
      break;
      } // end switch
    } // end while
}
int main(int argc, char *argv [])
{
  if( argc < 2 ) return 1;
  const char *filename = argv[1];
  gdcm::Reader reader;
  reader.SetFileName( filename );
  if( !reader.Read() )
    {
    std::cerr « "Failed to read:  " « filename « std::endl;
    return 1;
    }
  const gdcm::DataSet& ds = reader.GetFile().GetDataSet();
  // (07a1,1011) CS [PMSCT_RGB1]                                      # 10,1 Tamar Compression Type
  const gdcm::PrivateTag tcompressiontype(0x07a1,0x0011,"ELSCINT1");
  if( !ds.FindDataElement( tcompressiontype ) ) return 1;
  const gdcm::DataElement& compressiontype = ds.GetDataElement( tcompressiontype );
  if ( compressiontype.IsEmpty() ) return 1;
  const gdcm::ByteValue * bv = compressiontype.GetByteValue();
  std::string comprle = "PMSCT_RLE1";
  std::string comprgb = "PMSCT_RGB1";
  bool isrle = false;
  bool isrgb = false;
  if( strncmp( bv->GetPointer(), comprle.c_str(), comprle.size() ) == 0 )
    {
    isrle = true;
    return 1;
    }
  if( strncmp( bv->GetPointer(), comprgb.c_str(), comprgb.size() ) == 0 )
    {
    isrgb = true;
    }
  if( !isrgb && !isrle ) return 1;
  const gdcm::PrivateTag tcompressedpixeldata(0x07a1,0x000a,"ELSCINT1");
  if( !ds.FindDataElement( tcompressedpixeldata) ) return 1;
  const gdcm::DataElement& compressionpixeldata = ds.GetDataElement( tcompressedpixeldata);
  if ( compressionpixeldata.IsEmpty() ) return 1;
  const gdcm::ByteValue * bv2 = compressionpixeldata.GetByteValue();
  gdcm::Attribute<0x0028,0x0006> at0;
  at0.SetFromDataSet( ds );
  gdcm::Attribute<0x0028,0x0010> at1;
  at1.SetFromDataSet( ds );
  gdcm::Attribute<0x0028,0x0011> at2;
  at2.SetFromDataSet( ds );
  std::vector<unsigned char> buffer;
  delta_decode((const unsigned char*)bv2->GetPointer(), bv2->GetLength(), buffer,
    at0.GetValue(), at1.GetValue(), at2.GetValue() );
  gdcm::DataElement pixeldata( gdcm::Tag(0x7fe0,0x0010) );
  pixeldata.SetVR( gdcm::VR::OW );
  pixeldata.SetByteValue( (char*)&buffer[0], (uint32_t)buffer.size() );
  // TODO we should check that decompress byte buffer match the expected size (row*col*...)
  // Add the pixel data element
  reader.GetFile().GetDataSet().Replace( pixeldata );
  reader.GetFile().GetHeader().SetDataSetTransferSyntax(
    gdcm::TransferSyntax::ExplicitVRLittleEndian);
  gdcm::Writer writer;
  writer.SetFile( reader.GetFile() );
  // Cleanup stuff:
  // remove the compressed pixel data:
```

```
  // FIXME: should I remove more private tags ?  all of them ?
  // oh well this is just an example
  // use gdcm::Anonymizer::RemovePrivateTags if needed...
  writer.GetFile().GetDataSet().Remove( compressionpixeldata.GetTag() );
  std::string outfilename;
  if (argc > 2)
     outfilename = argv[2];
  else
     outfilename = "outrgb.dcm";
  writer.SetFileName( outfilename.c_str() );
  if( !writer.Write() )
    {
    std::cerr « "Failed to write" « std::endl;
    return 1;
    }
  std::cout « "success !" « std::endl;
  return 0;
}
```

12.102 rle2img.cxx

```
/*=========================================================================

Program:  GDCM (Grassroots DICOM). A DICOM library

Copyright (c) 2006-2011 Mathieu Malaterre
All rights reserved.
See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.

This software is distributed WITHOUT ANY WARRANTY; without even
the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
PURPOSE.  See the above copyright notice for more information.

=========================================================================*/
/*
* This example shows how to rewrite a ELSCINT1/PMSCT_RLE1 compressed
* image so that it is readable by most 3rd party software (DICOM does
* not specify this particular encoding).
* This is required for the sake of interoperability with any standard
* conforming DICOM system.
*
* Everything done in this code is for the sole purpose of writing interoperable
* software under Sect.  1201 (f) Reverse Engineering exception of the DMCA.
* If you believe anything in this code violates any law or any of your rights,
* please contact us (gdcm-developers@lists.sourceforge.net) so that we can
* find a solution.
*
* Everything you do with this code is at your own risk, since decompression
* algorithm was not written from specification documents.
*
* Special thanks to:
* Mauro Maiorca for bringing to our attention on this new ELSCINT1
* compression algorithm :  PMSCT_RLE1 (different from the 'LOSSLESS RICE')
* See post at:
* http://groups.google.com/group/comp.protocols.dicom/msg/f2b99bf706a7f8ca
*
* Thanks to Jesus Spinola, for more datasets,
* http://www.itk.org/pipermail/insight-users/2008-April/025571.html
*
* And last but not least, a very big thank to Ivo van Poorten, without
* whom we would still be looking at this compressed byte stream as if
* it was RLE compressed.
*/
#include "gdcmReader.h"
#include "gdcmPrivateTag.h"
#include "gdcmAttribute.h"
#include "gdcmImageWriter.h"
/* FIXME: Why is PhilipsLosslessRice.dcm a 512x512 image ...  */
void delta_decode(const char *inbuffer, size_t length, std::vector<unsigned short> &output)
{
  // RLE pass
  std::vector<char> temp;
  for(size_t i = 0; i < length; ++i)
    {
    if( inbuffer[i] == (char)0xa5 )
      {
      //unsigned char repeat = (unsigned char)inbuffer[i+1] + 1;
```

```
      //assert( (unsigned char)inbuffer[i+1] != 255 );
      int repeat = (unsigned char)inbuffer[i+1] + 1;
      char value = inbuffer[i+2];
      while(repeat)
        {
        temp.push_back( value );
        --repeat;
        }
      i+=2;
      }
    else
      {
      temp.push_back( inbuffer[i] );
      }
    }
  // Delta encoding pass
  unsigned short delta = 0;
  for(size_t i = 0; i < temp.size(); ++i)
    {
    if( temp[i] == 0x5a )
      {
      unsigned char v1 = (unsigned char)temp[i+1];
      unsigned char v2 = (unsigned char)temp[i+2];
      unsigned short value = (unsigned short)(v2 * 256 + v1);
      output.push_back( value );
      delta = value;
      i+=2;
      }
    else
      {
      unsigned short value = (unsigned short)(temp[i] + delta);
      output.push_back( value );
      delta = value;
      }
    //assert( output[output.size()-1] == ref[output.size()-1] );
    }
  if ( output.size() % 2 )
    {
    output.resize( output.size() - 1 );
    }
  std::cout « length « " -> " « output.size() * 2 « std::endl;
}
int main(int argc, char *argv [])
{
  if( argc < 2 )
    {
    std::cerr « argv[0] « "input.dcm [output.dcm]" « std::endl;
    std::cerr « "will default to 'outrle.dcm' unless output.dcm is specified."
      « std::endl;
    return 1;
    }
  const char *filename = argv[1];
  gdcm::Reader reader;
  reader.SetFileName( filename );
  if( !reader.Read() )
    {
    std::cerr « "Failed to read:  " « filename « std::endl;
    return 1;
    }
  const gdcm::DataSet& ds = reader.GetFile().GetDataSet();
  // (07a1,1011) CS [PMSCT_RLE1]                                    # 10,1 Tamar Compression Type
  const gdcm::PrivateTag tcompressiontype(0x07a1,0x0011,"ELSCINT1");
  if( !ds.FindDataElement( tcompressiontype ) ) return 1;
  const gdcm::DataElement& compressiontype = ds.GetDataElement( tcompressiontype );
  if ( compressiontype.IsEmpty() ) return 1;
  const gdcm::ByteValue * bv = compressiontype.GetByteValue();
  std::string comprle = "PMSCT_RLE1";
  std::string comprgb = "PMSCT_RGB1";
  bool isrle = false;
  bool isrgb = false;
  if( strncmp( bv->GetPointer(), comprle.c_str(), comprle.size() ) == 0 )
    {
    isrle = true;
    }
  if( strncmp( bv->GetPointer(), comprgb.c_str(), comprgb.size() ) == 0 )
    {
    isrgb = true;
    std::cerr « "See:  pmsct_rgb1.cxx instead" « std::endl;
    return 1;
    }
  if( !isrgb && !isrle ) return 1;
```

```
  // check if compressed pixel data reside in private or standard tag
  const gdcm::PrivateTag tprivatepixeldata(0x07a1,0x100a,"ELSCINT1");
  const gdcm::Tag tstandardpixeldata(0x7fe0, 0x0010);
  gdcm::Tag tpixeldata;
  if(ds.FindDataElement(tprivatepixeldata)) tpixeldata = tprivatepixeldata;
  else if(ds.FindDataElement(tstandardpixeldata)) tpixeldata = tstandardpixeldata;
  if(!ds.FindDataElement(tpixeldata)) return 1;
  const gdcm::DataElement& compressionpixeldata = ds.GetDataElement( tpixeldata);
  if ( compressionpixeldata.IsEmpty() ) return 1;
  const gdcm::ByteValue * bv2 = compressionpixeldata.GetByteValue();
  gdcm::Attribute<0x0028,0x0010> at1;
  at1.SetFromDataSet( ds );
  gdcm::Attribute<0x0028,0x0011> at2;
  at2.SetFromDataSet( ds );
  gdcm::DataElement pixeldata;
  // if standard voxel data element does not exist, create it
  if( !reader.GetFile().GetDataSet().FindDataElement( tpixeldata ) )
  {
    pixeldata = gdcm::DataElement( tpixeldata, 0, gdcm::VR::OW );
  }
  else{
    pixeldata = reader.GetFile().GetDataSet().GetDataElement( tpixeldata );
  }
  pixeldata.SetVR( gdcm::VR::OW );
  gdcm::VL bv2l = bv2->GetLength();
  gdcm::VL at1l = at1.GetValue() * at2.GetValue() * 2; /* sizeof(unsigned short) == 2 */
  // Handle special case that is not compressed:
  if( bv2l == at1l )
    {
    pixeldata.SetByteValue( bv2->GetPointer(), bv2->GetLength() );
    }
  else
    {
    std::vector<unsigned short> buffer;
    delta_decode(bv2->GetPointer(), bv2->GetLength(), buffer);
    pixeldata.SetByteValue( (char*)&buffer[0], (uint32_t)(buffer.size() * sizeof( unsigned short )) );
    }
  // TODO we should check that decompress byte buffer match the expected size (row*col*...)
  // Add the pixel data element
  if( reader.GetFile().GetDataSet().FindDataElement( tpixeldata ) )
  {
    reader.GetFile().GetDataSet().Replace( pixeldata );
  }
  else
  {
    reader.GetFile().GetDataSet().ReplaceEmpty( pixeldata );
  }
  reader.GetFile().GetHeader().SetDataSetTransferSyntax(
    gdcm::TransferSyntax::ExplicitVRLittleEndian);
  gdcm::Writer writer;
  writer.SetFile( reader.GetFile() );
  // Cleanup stuff:
  // This makes the code equivalent to Philips workstation IntelliSpace Portal
  if( writer.GetFile().GetDataSet().FindDataElement( tcompressiontype ) )
  {
    writer.GetFile().GetDataSet().Remove( gdcm::Tag(0x07a1,0x1011) );
  }
  if( writer.GetFile().GetDataSet().FindDataElement( tprivatepixeldata ) )
  {
    writer.GetFile().GetDataSet().Remove( gdcm::Tag(0x07a1,0x100a) );
  }
  std::string outfilename;
  if (argc > 2)
     outfilename = argv[2];
  else
     outfilename = "outrle.dcm";
  writer.SetFileName( outfilename.c_str() );
  if( !writer.Write() )
    {
    std::cerr « "Failed to write" « std::endl;
    return 1;
    }
  std::cout « "success !" « std::endl;
  return 0;
}
```

12.103 uid_unique.cxx

```
/*=========================================================================

Program:  GDCM (Grassroots DICOM). A DICOM library

Copyright (c) 2006-2011 Mathieu Malaterre
All rights reserved.
See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.

This software is distributed WITHOUT ANY WARRANTY; without even
the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
PURPOSE.  See the above copyright notice for more information.

=========================================================================*/
#include "gdcmUIDGenerator.h"
#include <iostream>
#include <string>
#include <set>
int main()
{
  gdcm::UIDGenerator uid;
  //const char myroot[] = "9876543210.9876543210.9876543210.9876543210.9876543210"; // fails in ~40000 tries
  const char myroot[] = "9876543210.9876543210.9876543210";
  uid.SetRoot( myroot );
  std::set<std::string> uids;
  uint64_t wrap = 0;
  uint64_t c = 0;
  while(true)
    {
    const char *unique = uid.Generate();
    //std::cout « unique « std::endl;
    if( c % 10000 == 0 )
      {
      std::cout « "wrap=" « wrap « ",c=" « c « std::endl;
      }
    ++c;
    if( c == 0 )
      {
      wrap++;
      }
    if ( uids.count(unique) == 1 )
      {
      std::cerr « "Failed with:  " « unique « std::endl;
      return 1;
      }
    uids.insert( unique );
    }
}
```

12.104 DecompressImage.java

```
/*=========================================================================

Program:  GDCM (Grassroots DICOM). A DICOM library

Copyright (c) 2006-2011 Mathieu Malaterre
All rights reserved.
See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.

This software is distributed WITHOUT ANY WARRANTY; without even
the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
PURPOSE.  See the above copyright notice for more information.

=========================================================================*/
/*
* This example will take in a DICOM file, and tries to decompress it (actually write it
* as ImplicitVRLittleEndian Transfer Syntax).
*
* Compilation:
* $ CLASSPATH=gdcm.jar javac ../../gdcm/Examples/Java/DecompressImage.java -d .
*
* Usage:
* $ LD_LIBRARY_PATH=.  CLASSPATH=gdcm.jar:.  java DecompressImage gdcmData/012345.002.050.dcm out.dcm
*/
import gdcm.*;
public class DecompressImage
```

```
{
  public static void main(String[] args) throws Exception
    {
    String file1 = args[0];
    String file2 = args[1];
    ImageReader reader = new ImageReader();
    reader.SetFileName( file1 );
    boolean ret = reader.Read();
    if( !ret )
      {
      throw new Exception("Could not read:  " + file1 );
      }
    ImageChangeTransferSyntax change = new ImageChangeTransferSyntax();
    change.SetTransferSyntax( new TransferSyntax(TransferSyntax.TSType.ImplicitVRLittleEndian) );
    change.SetInput( reader.GetImage() );
    if( !change.Change() )
      {
      throw new Exception("Could not change:  " + file1 );
      }
    Image out = change.GetOutput();
    System.out.println( out.toString() );
    // Set the Source Application Entity Title
    FileMetaInformation.SetSourceApplicationEntityTitle( "Just For Fun" );
    ImageWriter writer = new ImageWriter();
    writer.SetFileName( file2 );
    writer.SetFile( reader.GetFile() );
    writer.SetImage( out );
    ret = writer.Write();
    if( !ret )
      {
      throw new Exception("Could not write:  " + file2 );
      }
    }
}
```

12.105 DecompressPixmap.java

```
/*=========================================================================

Program:  GDCM (Grassroots DICOM). A DICOM library

Copyright (c) 2006-2011 Mathieu Malaterre
All rights reserved.
See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.

This software is distributed WITHOUT ANY WARRANTY; without even
the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
PURPOSE.  See the above copyright notice for more information.

=========================================================================*/
/*
* This example will take in a DICOM file, and tries to decompress it (actually write it
* as ImplicitVRLittleEndian Transfer Syntax).
*
* Compilation:
* $ CLASSPATH=gdcm.jar javac ../../gdcm/Examples/Java/DecompressPixmap.java -d .
*
* Usage:
* $ LD_LIBRARY_PATH=.  CLASSPATH=gdcm.jar:.  java DecompressPixmap gdcmData/012345.002.050.dcm out.dcm
*/
import gdcm.*;
public class DecompressPixmap
{
  public static void main(String[] args) throws Exception
    {
    String file1 = args[0];
    String file2 = args[1];
    PixmapReader reader = new PixmapReader();
    reader.SetFileName( file1 );
    boolean ret = reader.Read();
    if( !ret )
      {
      throw new Exception("Could not read:  " + file1 );
      }
    ImageChangeTransferSyntax change = new ImageChangeTransferSyntax();
    change.SetTransferSyntax( new TransferSyntax(TransferSyntax.TSType.ImplicitVRLittleEndian) );
    PixmapToPixmapFilter filter = (PixmapToPixmapFilter)change;
```

```
    filter.SetInput( reader.GetPixmap() );
    if( !change.Change() )
      {
      throw new Exception("Could not change:  " + file1 );
      }
    // The following does not work in Java/swig 2.0.7
    //Pixmap p = ((PixmapToPixmapFilter)change).GetOutput();
    Pixmap p = change.GetOutputAsPixmap();  // be explicit
    //System.out.println( p.toString() );
    // Set the Source Application Entity Title
    FileMetaInformation.SetSourceApplicationEntityTitle( "Just For Fun" );
    PixmapWriter writer = new PixmapWriter();
    writer.SetFileName( file2 );
    writer.SetFile( reader.GetFile() );
    writer.SetImage( p );
    ret = writer.Write();
    if( !ret )
      {
      throw new Exception("Could not write:  " + file2 );
      }
    }
}
```

12.106 ExtractImageRegion.java

```
/*=========================================================================

Program:  GDCM (Grassroots DICOM). A DICOM library

Copyright (c) 2006-2011 Mathieu Malaterre
All rights reserved.
See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.

This software is distributed WITHOUT ANY WARRANTY; without even
the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
PURPOSE.  See the above copyright notice for more information.

=========================================================================*/
/*
* This small code shows how to use the gdcm.ImageRegionReader API
* In this example we are taking each frame by frame and dump them to
* /tmp/frame.raw.
*
* Usage:
* $ LD_LIBRARY_PATH=.  CLASSPATH=gdcm.jar:.  java ExtractImageRegion input.dcm
*/
import gdcm.*;
import java.io.FileOutputStream;
public class ExtractImageRegion
{
  public static void main(String[] args) throws Exception
    {
    String filename = args[0];
    // instantiate the reader:
    ImageRegionReader reader = new ImageRegionReader();
    reader.SetFileName( filename );
    // pull DICOM info:
    if (!reader.ReadInformation()) return;
    // Get file infos
    File f = reader.GetFile();
    // get some info about image
    UIntArrayType dims = ImageHelper.GetDimensionsValue(f);
    PixelFormat pf = ImageHelper.GetPixelFormatValue (f);
    int pixelsize = pf.GetPixelSize();
    // buffer to get the pixels
    long buffer_length = dims.get(0) * dims.get(1) * pixelsize;
    byte[] buffer = new byte[ (int)buffer_length ];
    // define a simple box region.
    BoxRegion box = new BoxRegion();
    for (int z = 0; z < dims.get(2); z++)
      {
      // Define that I want the image 0, full size (dimx x dimy pixels)
      // and do that for each z:
      box.SetDomain(0, dims.get(0) - 1, 0, dims.get(1) - 1, z, z);
      //System.Console.WriteLine( box.toString() );
      reader.SetRegion( box );
      // reader will try to load the uncompressed image region into buffer.
```

```
      // the call returns an error when buffer.Length is too small.  For instance
      // one can call:
      // long buf_len = reader.ComputeBufferLength(); // take into account pixel size
      // to get the exact size of minimum buffer
      if (reader.ReadIntoBuffer(buffer, buffer_length))
        {
        FileOutputStream fos = new FileOutputStream("/tmp/frame.raw");
        fos.write(buffer);
        fos.close();
        }
      else
        {
        throw new Exception("can't read pixels error");
        }
      }
    }
}
```

12.107 FileAnonymize.java

```
/*=========================================================================

Program:  GDCM (Grassroots DICOM). A DICOM library

Copyright (c) 2006-2011 Mathieu Malaterre
All rights reserved.
See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.

This software is distributed WITHOUT ANY WARRANTY; without even
the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
PURPOSE.  See the above copyright notice for more information.

=========================================================================*/
import gdcm.*;
public class FileAnonymize
{
  public static class MyWatcher extends SimpleSubjectWatcher
    {
    public MyWatcher(Subject s) { super(s,"Override String"); }
    protected void ShowProgress(Subject caller, Event evt)
      {
      ProgressEvent pe = ProgressEvent.Cast(evt);
      System.out.println( "This is my progress:  " + pe.GetProgress() );
      }
    }
  public static void main(String[] args) throws Exception
    {
    String input = args[0];
    String output = args[1];
    FileAnonymizer fa = new FileAnonymizer();
    fa.SetInputFileName( input );
    fa.SetOutputFileName( output );
    // Empty Operations
    // It will create elements, since those tags are non-registered public elements (2011):
    fa.Empty( new Tag(0x0008,0x1313) );
    fa.Empty( new Tag(0x0008,0x1317) );
    // Remove Operations
    // The following Tag are actually carefully chosen, since they refer to SQ:
    fa.Remove( new Tag(0x0008,0x2112) );
    fa.Remove( new Tag(0x0008,0x9215) );
    // Replace Operations
    // do not call replace operation on SQ attribute !
    fa.Replace( new Tag(0x0018,0x5100), "MYVALUE " );
    fa.Replace( new Tag(0x0008,0x1160), "MYOTHERVAL" );
    if( !fa.Write() )
      {
      System.out.println( "Could not write" );
      return;
      }
    System.out.println( "success" );
    }
}
```

12.108 HelloSimple.java

```
/*=========================================================================

Program:  GDCM (Grassroots DICOM). A DICOM library

Copyright (c) 2006-2011 Mathieu Malaterre
All rights reserved.
See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.

This software is distributed WITHOUT ANY WARRANTY; without even
the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
PURPOSE.  See the above copyright notice for more information.

=========================================================================*/
/*
* Compilation:
* $ CLASSPATH=gdcm.jar javac ../../gdcm/Examples/Java/HelloSimple.java -d .
*
* Usage:
* $ LD_LIBRARY_PATH=.  CLASSPATH=gdcm.jar:.  java HelloSimple gdcmData/012345.002.050.dcm
*/
import gdcm.*;
public class HelloSimple
{
  public static void main(String[] args) throws Exception
    {
    String filename = args[0];
    Reader reader = new Reader();
    reader.SetFileName( filename );
    boolean ret = reader.Read();
    if( !ret )
      {
      throw new Exception("Could not read:  " + filename );
      }
    File f = reader.GetFile();
    DataSet ds = f.GetDataSet();
    System.out.println( ds.toString() );
    System.out.println("Success reading:  " + filename );
    }
}
```

12.109 ReadFiles.java

```
/*=========================================================================

Program:  GDCM (Grassroots DICOM). A DICOM library

Copyright (c) 2006-2011 Mathieu Malaterre
All rights reserved.
See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.

This software is distributed WITHOUT ANY WARRANTY; without even
the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
PURPOSE.  See the above copyright notice for more information.

=========================================================================*/
import gdcm.*;
import java.io.File;
public class ReadFiles
{
  static int i = 0;
  public static void process(String path)
    {
    //String path = file.getPath();
    assert PosixEmulation.FileExists(path) :  "Problem converting to 8bits";
    System.out.println("Reading:  " + path );
    System.out.println("File:  " + i++);
    Reader r = new Reader();
    try
      {
      r.SetFileName( path );
      TagSetType skip = new TagSetType();
      skip.insert( new Tag(0x7fe0,0x10) );
      boolean b = r.ReadUpToTag( new Tag(0x88,0x200), skip );
      //System.out.println("DS:\n" + r.GetFile().GetDataSet().toString() );
      }
```

```
    finally
      {
      r.delete(); // will properly call C++ destructor and close file descriptor
      }
    }
  // Process only files under dir
  public static void visitAllFiles(File dir)
    {
    if (dir.isDirectory())
      {
      String[] children = dir.list();
      for (int i=0; i<children.length; i++)
        {
        visitAllFiles(new File(dir, children[i]));
        }
      }
    else
      {
      process(dir.getPath());
      }
    }
  public static void waiting (int n)
    {
    long t0, t1;
    t0 =  System.currentTimeMillis();
    do
      {
      t1 = System.currentTimeMillis();
      }
    while ((t1 - t0) < (n * 1000));
    }
  public static void main(String[] args) throws Exception
    {
    String directory = args[0];
    Directory gdir = new Directory();
    long n = gdir.Load( directory, true );
    System.out.println( gdir.toString() );
    FilenamesType files = gdir.GetFilenames();
    for( long i = 0; i < n; ++i )
      {
      String path = files.get( (int)i );
      process( path );
      }
    System.out.println( "Java API" );
    //waiting( 10 );
    for( int i = 0; i < 2; ++i )
      {
      File dir = new File(directory);
      visitAllFiles(dir);
      }
    }
}
```

12.110 ScanDirectory.java

```
/*=========================================================================

Program:  GDCM (Grassroots DICOM). A DICOM library

Copyright (c) 2006-2011 Mathieu Malaterre
All rights reserved.
See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.

This software is distributed WITHOUT ANY WARRANTY; without even
the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
PURPOSE.  See the above copyright notice for more information.

=========================================================================*/
import gdcm.*;
import gdcm.Reader;
import gdcm.LookupTable;
import java.io.File;
import java.io.*;
import java.awt.image.*;
import javax.imageio.ImageIO;
public class ScanDirectory
{
```

```
  public static class MyWatcher extends SimpleSubjectWatcher
    {
    public MyWatcher(Subject s) { super(s,"Override String"); }
    protected void ShowProgress(Subject caller, Event evt)
      {
      ProgressEvent pe = ProgressEvent.Cast(evt);
      System.out.println( "This is my progress:  " + pe.GetProgress() );
      }
    }
  public static byte[] GetAsByte(Bitmap input)
    {
    long len = input.GetBufferLength();
    byte[] buffer = new byte[ (int)len ];
    PhotometricInterpretation pi = input.GetPhotometricInterpretation();
    if( pi.GetType() == PhotometricInterpretation.PIType.MONOCHROME1 )
      {
      ImageChangePhotometricInterpretation icpi = new ImageChangePhotometricInterpretation();
      icpi.SetInput( input );
      icpi.SetPhotometricInterpretation(
        new PhotometricInterpretation(
          PhotometricInterpretation.PIType.MONOCHROME2 ) );
      if( icpi.Change() )
        {
        Bitmap output = icpi.GetOutput();
        output.GetArray( buffer );
        }
      return buffer;
      }
    else
      {
      input.GetArray( buffer );
      return buffer;
      }
    }
  public static short[] GetAsShort(Bitmap input)
    {
    long len = input.GetBufferLength(); // length in bytes
    short[] buffer = new short[ (int)len / 2 ];
    PhotometricInterpretation pi = input.GetPhotometricInterpretation();
    if( pi.GetType() == PhotometricInterpretation.PIType.MONOCHROME1 )
      {
      ImageChangePhotometricInterpretation icpi = new ImageChangePhotometricInterpretation();
      icpi.SetInput( input );
      icpi.SetPhotometricInterpretation(
        new PhotometricInterpretation(
          PhotometricInterpretation.PIType.MONOCHROME2 ) );
      if( icpi.Change() )
        {
        Bitmap output = icpi.GetOutput();
        output.GetArray( buffer );
        }
      return buffer;
      }
    else
      {
      input.GetArray( buffer );
      return buffer;
      }
    }
  public static boolean WritePNG(Bitmap input, String outfilename )
    {
    int imageType = BufferedImage.TYPE_CUSTOM;
    PixelFormat pf = input.GetPixelFormat();
    PhotometricInterpretation pi = input.GetPhotometricInterpretation();
    // We need to handle both public and private icon
    // It could well be that we are getting an RGB Icon or 16 bits Icon:
    ColorModel colorModel = null;
    if( pf.GetSamplesPerPixel() == 1 )
      {
      if( pi.GetType() == PhotometricInterpretation.PIType.MONOCHROME1
         || pi.GetType() == PhotometricInterpretation.PIType.MONOCHROME2 )
        {
        if( pf.GetScalarType() == PixelFormat.ScalarType.UINT8 )
          {
          imageType = BufferedImage.TYPE_BYTE_GRAY;
          }
        else if( pf.GetScalarType() == PixelFormat.ScalarType.UINT12 )
          {
          imageType = BufferedImage.TYPE_USHORT_GRAY;
          }
        else if( pf.GetScalarType() == PixelFormat.ScalarType.UINT16 )
```

```
          {
          imageType = BufferedImage.TYPE_USHORT_GRAY;
          }
        }
      else if( pi.GetType() == PhotometricInterpretation.PIType.PALETTE_COLOR )
        {
        LookupTable lut = input.GetLUT();
        long rl = lut.GetLUTLength( LookupTable.LookupTableType.RED );
        byte[] rbuf = new byte[ (int)rl ];
        long rl2 = lut.GetLUT( LookupTable.LookupTableType.RED, rbuf );
        assert rl == rl2;
        long gl = lut.GetLUTLength( LookupTable.LookupTableType.GREEN );
        byte[] gbuf = new byte[ (int)gl ];
        long gl2 = lut.GetLUT( LookupTable.LookupTableType.GREEN, gbuf );
        assert gl == gl2;
        long bl = lut.GetLUTLength( LookupTable.LookupTableType.BLUE );
        byte[] bbuf = new byte[ (int)bl ];
        long bl2 = lut.GetLUT( LookupTable.LookupTableType.BLUE, bbuf );
        assert bl == bl2;
        colorModel = new IndexColorModel(8, (int)rl, rbuf, gbuf, bbuf);
        // For code below
        imageType = BufferedImage.TYPE_BYTE_GRAY;
        }
      }
    else if( pf.GetSamplesPerPixel() == 3 )
      {
      if( pf.GetScalarType() == PixelFormat.ScalarType.UINT8 )
        {
        // FIXME should be TYPE_3BYTE_RGB
        imageType = BufferedImage.TYPE_3BYTE_BGR;
        }
      }
    //System.out.println( "pf:  " + pf.toString() );
    //System.out.println( "pi:  " + pi.toString() );
    long width  = input.GetDimension(0);
    long height = input.GetDimension(0);
    BufferedImage bi;
    if( pi.GetType() == PhotometricInterpretation.PIType.PALETTE_COLOR )
      {
      bi = new BufferedImage(colorModel,
        colorModel.createCompatibleWritableRaster((int)width, (int)height),
        false, null);
      }
    else
      {
      bi = new BufferedImage((int)width,(int)height,imageType);
      }
    WritableRaster wr = bi.getRaster();
    //System.out.println( "imagetype:  " + imageType );
    if( imageType == BufferedImage.TYPE_BYTE_GRAY
      || imageType == BufferedImage.TYPE_3BYTE_BGR )
      {
      byte[] buffer = GetAsByte( input );
      wr.setDataElements (0, 0, (int)width, (int)height, buffer);
      }
    else if( imageType == BufferedImage.TYPE_USHORT_GRAY )
      {
      short[] buffer = GetAsShort( input );
      wr.setDataElements (0, 0, (int)width, (int)height, buffer);
      }
    File outputfile = new File( outfilename );
    try {
      ImageIO.write(bi, "png", outputfile);
    } catch (IOException e) {
      return false;
    }
    return true;
    }
  public static void main(String[] args) throws Exception
    {
    String directory = args[0];
    Directory d = new Directory();
    long nfiles = d.Load( directory, true );
    if(nfiles == 0)
      {
      throw new Exception("No files found");
      }
//    System.out.println( "Files:\n" + d.toString() );
    FilenamesType fns = d.GetFilenames();
    //Scanner s = new Scanner();
    SmartPtrScan sscan = Scanner.New();
```

```
Scanner s = sscan.__ref__();
//SimpleSubjectWatcher watcher = new SimpleSubjectWatcher(s, "MySimple");
MyWatcher watcher = new MyWatcher(s);
Tag[] tagarray = {
  new Tag(0x0010, 0x0010),    // PatientName
  new Tag(0x0010, 0x0020),    // PatientID
  new Tag(0x0010, 0x0030),    // PatientBirthDate
  new Tag(0x0010, 0x0040),    // PatientSex
  new Tag(0x0010, 0x1010),    // PatientAge
  new Tag(0x0020, 0x000d),    // StudyInstanceUID
  new Tag(0x0020, 0x0010),    // StudyID
  new Tag(0x0008, 0x0020),    // StudyDate
  new Tag(0x0008, 0x1030),    // StudyDescription
  new Tag(0x0020, 0x000e),    // SeriesInstanceUID
  new Tag(0x0020, 0x0011),    // SeriesNumber
  new Tag(0x0008, 0x0021),    // SeriesDate
  new Tag(0x0008, 0x103e),    // SeriesDescription
  new Tag(0x0008, 0x0090),    // ReferringPhysicianName
  new Tag(0x0008, 0x0060),    // Modality
  new Tag(0x0054, 0x0400),    // ImageID ??  Should be Instance number ??
  new Tag(0x0008, 0x0018),    // SOPInstanceUID
  new Tag(0x0008, 0x0032),    // AcquisitionTime
  new Tag(0x0008, 0x0033),    // ContentTime
  new Tag(0x0020, 0x0013),    // InstanceNumber
  new Tag(0x0020, 0x1041),    // SliceLocation
  new Tag(0x0018, 0x0050),    // SliceThickness ??  Eg.  Enhanced MR Image Storage
  new Tag(0x0008, 0x0080),    // InstitutionName
  new Tag(0x0028, 0x1050),    // WindowCenter
  new Tag(0x0028, 0x1051),    // WindowWidth
};
for( Tag t :  tagarray ) {
  //System.out.println( "Tag:  " + t.toString() );
  s.AddTag( t );
}
boolean b = s.Scan( fns );
if(!b)
  {
  throw new Exception("Could not scan");
  }
String fn0 = fns.get(0);
TagToValue mappings = s.GetMapping( fn0 );
System.out.println( "mappings size:  " + mappings.size() );
for( Tag tag :  tagarray ) {
  if( mappings.has_key( tag ) ) {
    String val = mappings.get( tag );
    System.out.println( "tag/val:  " + tag + "->" + val );
  }
}
for( long idx = 0; idx < fns.size(); ++idx )
  {
  Reader r = new Reader();
  String fn = fns.get( (int)idx );
  String outfn = fn + ".png";
  r.SetFileName( fn );
  TagSetType tst = new TagSetType();
  tst.insert( new Tag(0x7fe0,0x10) );
  b = r.ReadUpToTag( new Tag(0x88,0x200), tst );
  UIntArrayType dims = ImageHelper.GetDimensionsValue( r.GetFile() );
  if( b )
    {
    IconImageFilter iif = new IconImageFilter();
    System.out.println( "Processing:  " + fn );
    iif.SetFile( r.GetFile() );
    b = iif.Extract();
    if( b )
      {
      Bitmap icon = iif.GetIconImage(0);
      WritePNG(icon, outfn);
      }
    else
      {
      ImageReader ir = new ImageReader();
      ir.SetFileName( fn );
      if( ir.Read() )
        {
        Image img = ir.GetImage();
        StringFilter sf = new StringFilter();
        sf.SetFile( r.GetFile() );
        String strval = sf.ToString( new Tag(0x0028,0x0120) );
        IconImageGenerator iig = new IconImageGenerator();
        iig.SetPixmap( img );
```

```
            iig.AutoPixelMinMax( true );
            try {
              double val = Double.parseDouble( strval );
              iig.SetOutsideValuePixel( val );
            }
            catch ( NumberFormatException e) {
              }
            iig.ConvertRGBToPaletteColor( false );
            long idims[] = { 128, 128};
            iig.SetOutputDimensions( idims );
            iig.Generate();
            Bitmap icon = iig.GetIconImage();
            WritePNG(icon, outfn);
            }
          }
        }
      }
    System.out.println( "Scan:\n" + s.toString() );
    System.out.println( "success" );
    }
}
```

12.111 SimplePrint.java

```
/*=========================================================================

Program:  GDCM (Grassroots DICOM). A DICOM library

Copyright (c) 2006-2011 Mathieu Malaterre
All rights reserved.
See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.

This software is distributed WITHOUT ANY WARRANTY; without even
the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
PURPOSE.  See the above copyright notice for more information.

=========================================================================*/
/*
* Compilation:
* $ CLASSPATH=gdcm.jar javac ../../gdcm/Examples/Java/SimplePrint.java -d .
*
* Usage:
* $ LD_LIBRARY_PATH=.  CLASSPATH=gdcm.jar:.  java SimplePrint gdcmData/012345.002.050.dcm
*/
import gdcm.*;
public class SimplePrint
{
  public static void RecurseDataSet(File f, DataSet ds, String indent)
    {
    JavaDataSet cds = new JavaDataSet(ds);
    while(!cds.IsAtEnd())
      {
      DataElement de = cds.GetCurrent();
      // Compute VR from the toplevel file, and the currently processed dataset:
      VR vr = DataSetHelper.ComputeVR(f, ds, de.GetTag() );
      if( vr.Compatible( new VR(VR.VRType.SQ) ) )
        {
        long uvl = de.GetVL().GetValueLength(); // Test cast is ok
        System.out.println(  indent + de.GetTag().toString() + ":" + uvl ); // why not ?
        //SequenceOfItems sq = de.GetSequenceOfItems();
        // GetValueAsSQ handle more cases than GetSequenceOfItems
        SmartPtrSQ sq = de.GetValueAsSQ();
        long n = sq.GetNumberOfItems();
        for( long i = 1; i <= n; i++) // item starts at 1, not 0
          {
          Item item = sq.GetItem( i );
          DataSet nested = item.GetNestedDataSet();
          RecurseDataSet( f, nested, indent + "  " );
          }
        }
      else
        {
        System.out.println( indent + de.toString() );
        }
      cds.Next();
      }
    }
```

```
  public static void main(String[] args) throws Exception
    {
    String filename = args[0];
    Reader reader = new Reader();
    reader.SetFileName( filename );
    boolean ret = reader.Read();
    if( !ret )
      {
      throw new Exception("Could not read:  " + filename );
      }
    File f = reader.GetFile();
    DataSet ds = f.GetDataSet();
    RecurseDataSet( f, ds, "" );
    }
}
```

12.112 AddPrivateAttribute.py

```
1
14
15 """
16 Usage:
17
18 python AddPrivateAttribute.py input.dcm output.dcm
19
20
21 """
22
23 import sys
24 import gdcm
25
26 if __name__ == "__main__":
27
28 file1 = sys.argv[1]
29 file2 = sys.argv[2]
30
31 r = gdcm.Reader()
32 r.SetFileName( file1 )
33 if not r.Read():
34     sys.exit(1)
35
36   f = r.GetFile()
37   ds = f.GetDataSet()
38
39   # Create a dataelement
40   de = gdcm.DataElement(gdcm.Tag(0x0051, 0x1011))
41   de.SetByteStringValue("p2")
42   de.SetVR(gdcm.VR(gdcm.VR.SH))
43
44   ds.Insert(de)
45
46   w = gdcm.Writer()
47   w.SetFile( f )
48   w.SetFileName( file2 )
49   if not w.Write():
50     sys.exit(1)
```

12.113 ConvertMPL.py

```
1
14
15 """
16 display a DICOM image with matPlotLib via numpy
17
18 Caveats:
19 - Does not support UINT12/INT12
20
21 Usage:
22
23 python ConvertNumpy.py "IM000000"
24
25 Thanks:
26 plotting example - Ray Schumacher 2009
```

```
27 """
28
29 import gdcm
30 import numpy
31 from pylab import *
32
33
34 def get_gdcm_to_numpy_typemap():
35 """Returns the GDCM Pixel Format to numpy array type mapping."""
36     _gdcm_np = {gdcm.PixelFormat.UINT8  :numpy.int8,
37                 gdcm.PixelFormat.INT8   :numpy.uint8,
38                 gdcm.PixelFormat.UINT16 :numpy.uint16,
39                 gdcm.PixelFormat.INT16  :numpy.int16,
40                 gdcm.PixelFormat.UINT32 :numpy.uint32,
41                 gdcm.PixelFormat.INT32  :numpy.int32,
42                 gdcm.PixelFormat.FLOAT32:numpy.float32,
43                 gdcm.PixelFormat.FLOAT64:numpy.float64 }
44     return _gdcm_np
45
46 def get_numpy_array_type(gdcm_pixel_format):
47     """Returns a numpy array typecode given a GDCM Pixel Format."""
48     return get_gdcm_to_numpy_typemap()[gdcm_pixel_format]
49
50 def gdcm_to_numpy(image):
51     """Converts a GDCM image to a numpy array.
52 """
53 pf = image.GetPixelFormat().GetScalarType()
54 print 'pf', pf
55     print image.GetPixelFormat().GetScalarTypeAsString()
56     assert pf in get_gdcm_to_numpy_typemap().keys(), \
57            "Unsupported array type %s"%pf
58     d = image.GetDimension(0), image.GetDimension(1)
59     print 'Image Size:  %d x %d' % (d[0], d[1])
60     dtype = get_numpy_array_type(pf)
61     gdcm_array = image.GetBuffer()
62
63     result = numpy.frombuffer(gdcm_array, dtype=dtype).astype(float)
64
68     result.shape = d
69     return result
70
71 if __name__ == "__main__":
72   import sys
73   r = gdcm.ImageReader()
74   filename = sys.argv[1]
75   r.SetFileName( filename )
76   if not r.Read():    sys.exit(1)
77   numpy_array = gdcm_to_numpy( r.GetImage() )
78
79   subplot(111)# one plot, on left
80   title(filename)
81
82   imshow(numpy_array, interpolation='bilinear', cmap=cm.jet)
83
84   subplots_adjust(bottom=0.1, right=0.8, top=0.9)
85   cax = axes([0.85, 0.1, 0.075, 0.8])
86   colorbar(cax=cax)
87   title('values')
88   get_current_fig_manager().window.title('plot')
89   show()
```

12.114 ConvertNumpy.py

```
1
14
15 """
16 This module add support for converting a gdcm.Image to a numpy array.
17
18 Caveats:
19 - Does not support UINT12/INT12
20
21 Removed:
22 - float16 is defined in GDCM API but no implementation exist for it ...
23 """
24
25 import gdcm
26 import numpy
```

```
27
28 def get_gdcm_to_numpy_typemap():
29     """Returns the GDCM Pixel Format to numpy array type mapping."""
30     _gdcm_np = {gdcm.PixelFormat.UINT8  :numpy.uint8,
31                 gdcm.PixelFormat.INT8   :numpy.int8,
32                 #gdcm.PixelFormat.UINT12 :numpy.uint12,
33                 #gdcm.PixelFormat.INT12  :numpy.int12,
34                 gdcm.PixelFormat.UINT16 :numpy.uint16,
35                 gdcm.PixelFormat.INT16  :numpy.int16,
36                 gdcm.PixelFormat.UINT32 :numpy.uint32,
37                 gdcm.PixelFormat.INT32  :numpy.int32,
38                 #gdcm.PixelFormat.FLOAT16:numpy.float16,
39                 gdcm.PixelFormat.FLOAT32:numpy.float32,
40                 gdcm.PixelFormat.FLOAT64:numpy.float64 }
41     return _gdcm_np
42
43 def get_numpy_array_type(gdcm_pixel_format):
44     """Returns a numpy array typecode given a GDCM Pixel Format."""
45     return get_gdcm_to_numpy_typemap()[gdcm_pixel_format]
46
47 def gdcm_to_numpy(image):
48     """Converts a GDCM image to a numpy array.
49 """
50 pf = image.GetPixelFormat()
51
52 assert pf.GetScalarType() in get_gdcm_to_numpy_typemap().keys(), \
53            "Unsupported array type %s"%pf
54
55     shape = image.GetDimension(0) * image.GetDimension(1), pf.GetSamplesPerPixel()
56     if image.GetNumberOfDimensions() == 3:
57       shape = shape[0] * image.GetDimension(2), shape[1]
58
59     dtype = get_numpy_array_type(pf.GetScalarType())
60     gdcm_array = image.GetBuffer()
61     result = numpy.frombuffer(gdcm_array, dtype=dtype)
62     result.shape = shape
63     return result
64
65 if __name__ == "__main__":
66   import sys
67   r = gdcm.ImageReader()
68   filename = sys.argv[1]
69   r.SetFileName( filename )
70   if not r.Read():
71     sys.exit(1)
72
73   numpy_array = gdcm_to_numpy( r.GetImage() )
74   print numpy_array
```

12.115 ConvertPIL.py

```
1
14
15 """
16 save a DICOM image with PIL via numpy
17
18 Caveats:
19 - Does not support UINT12/INT12
20
21 Usage:
22
23 python ConvertNumpy.py "IM000000"
24
25 Thanks:
26 plotting example - Ray Schumacher 2009
27 """
28
29 import gdcm
30 import numpy
31 from PIL import Image, ImageOps
32
33
34 def get_gdcm_to_numpy_typemap():
35 """Returns the GDCM Pixel Format to numpy array type mapping."""
36     _gdcm_np = {gdcm.PixelFormat.UINT8  :numpy.int8,
37                 gdcm.PixelFormat.INT8   :numpy.uint8,
38                 gdcm.PixelFormat.UINT16 :numpy.uint16,
```

```
39                 gdcm.PixelFormat.INT16  :numpy.int16,
40                 gdcm.PixelFormat.UINT32 :numpy.uint32,
41                 gdcm.PixelFormat.INT32  :numpy.int32,
42                 gdcm.PixelFormat.FLOAT32:numpy.float32,
43                 gdcm.PixelFormat.FLOAT64:numpy.float64 }
44     return _gdcm_np
45
46 def get_numpy_array_type(gdcm_pixel_format):
47     """Returns a numpy array typecode given a GDCM Pixel Format."""
48     return get_gdcm_to_numpy_typemap()[gdcm_pixel_format]
49
50 def gdcm_to_numpy(image):
51     """Converts a GDCM image to a numpy array.
52 """
53 pf = image.GetPixelFormat().GetScalarType()
54 print 'pf', pf
55     print image.GetPixelFormat().GetScalarTypeAsString()
56     assert pf in get_gdcm_to_numpy_typemap().keys(), \
57            "Unsupported array type %s"%pf
58     d = image.GetDimension(0), image.GetDimension(1)
59     print 'Image Size:  %d x %d' % (d[0], d[1])
60     dtype = get_numpy_array_type(pf)
61     gdcm_array = image.GetBuffer()
62     result = numpy.frombuffer(gdcm_array, dtype=dtype)
63     maxV = float(result[result.argmax()])
64
67     result = numpy.log(result+50)
68     maxV = float(result[result.argmax()])
69     result = result*(2.**8/maxV)
70     result.shape = d
71     return result
72
73 if __name__ == "__main__":
74   import sys
75   r = gdcm.ImageReader()
76   filename = sys.argv[1]
77   r.SetFileName( filename )
78   if not r.Read():    sys.exit(1)
79   numpy_array = gdcm_to_numpy( r.GetImage() )
80
82   pilImage = Image.frombuffer('L',
83                               numpy_array.shape,
84                               numpy_array.astype(numpy.uint8),
85                               'raw','L',0,1)
86
87   pilImage = ImageOps.autocontrast(pilImage, cutoff=.1)
88   pilImage.save(sys.argv[1]+'.jpg')
```

12.116 CreateRAWStorage.py

```
1
14
15 """
16 <uid value="1.2.840.10008.5.1.4.1.1.66" name="Raw Data Storage" type="SOP Class" part="PS 3.4"
       retired="false"/>
17 """
18
19 import gdcm
20 import sys,os
21
22 if __name__ == "__main__":
23 r = gdcm.Reader()
24 # Will require Testing...
25   dataroot = gdcm.Testing.GetDataRoot()
26   filename = os.path.join( dataroot, '012345.002.050.dcm' )
27   r.SetFileName( filename )
28   r.Read()
29   f = r.GetFile()
30   ds = f.GetDataSet()
31
32   uid = "1.2.840.10008.5.1.4.1.1.66"
33 #  f = gdcm.File()
34 #  ds = f.GetDataSet()
35   de = gdcm.DataElement( gdcm.Tag(0x0008,0x0016) )
36   de.SetByteStringValue( uid )
37   vr = gdcm.VR( gdcm.VR.UI )
38   de.SetVR( vr )
```

```
39   ds.Replace( de )
40
41   ano = gdcm.Anonymizer()
42   ano.SetFile( r.GetFile() )
43   ano.RemovePrivateTags()
44   ano.RemoveGroupLength()
45   taglist = [
46   gdcm.Tag(0x0008,0x0008),
47   gdcm.Tag(0x0008,0x0022),
48   gdcm.Tag(0x0008,0x0032),
49   gdcm.Tag(0x0008,0x2111),
50   gdcm.Tag(0x0008,0x1150),
51   gdcm.Tag(0x0008,0x1155),
52   gdcm.Tag(0x0008,0x0100),
53   gdcm.Tag(0x0008,0x0102),
54   gdcm.Tag(0x0008,0x0104),
55   gdcm.Tag(0x0040,0xa170),
56   gdcm.Tag(0x0008,0x2112),
57   gdcm.Tag(0x0008,0x0100),
58   gdcm.Tag(0x0008,0x0102),
59   gdcm.Tag(0x0008,0x0104),
60   gdcm.Tag(0x0008,0x9215),
61   gdcm.Tag(0x0018,0x0010),
62   gdcm.Tag(0x0018,0x0022),
63   gdcm.Tag(0x0018,0x0050),
64   gdcm.Tag(0x0018,0x0060),
65   gdcm.Tag(0x0018,0x0088),
66   gdcm.Tag(0x0018,0x0090),
67   gdcm.Tag(0x0018,0x1040),
68   gdcm.Tag(0x0018,0x1100),
69   gdcm.Tag(0x0018,0x1110),
70   gdcm.Tag(0x0018,0x1111),
71   gdcm.Tag(0x0018,0x1120),
72   gdcm.Tag(0x0018,0x1130),
73   gdcm.Tag(0x0018,0x1150),
74   gdcm.Tag(0x0018,0x1151),
75   gdcm.Tag(0x0018,0x1152),
76   gdcm.Tag(0x0018,0x1160),
77   gdcm.Tag(0x0018,0x1190),
78   gdcm.Tag(0x0018,0x1210),
79   gdcm.Tag(0x0020,0x0012),
80   gdcm.Tag(0x0020,0x0032),
81   gdcm.Tag(0x0020,0x0037),
82   gdcm.Tag(0x0020,0x1041),
83   gdcm.Tag(0x0020,0x4000),
84   gdcm.Tag(0x0028,0x0002),
85   gdcm.Tag(0x0028,0x0004),
86   gdcm.Tag(0x0028,0x0010),
87   gdcm.Tag(0x0028,0x0011),
88   gdcm.Tag(0x0028,0x0030),
89   gdcm.Tag(0x0028,0x0100),
90   gdcm.Tag(0x0028,0x0101),
91   gdcm.Tag(0x0028,0x0102),
92   gdcm.Tag(0x0028,0x0103),
93   gdcm.Tag(0x0028,0x1052),
94   gdcm.Tag(0x0028,0x1053),
95   gdcm.Tag(0x0028,0x2110),
96   gdcm.Tag(0x0028,0x2112),
97   gdcm.Tag(0x7fe0,0x0010),
98   gdcm.Tag(0x0018,0x0020),
99   gdcm.Tag(0x0018,0x0021),
100   gdcm.Tag(0x0018,0x0023),
101   gdcm.Tag(0x0018,0x0025),
102   gdcm.Tag(0x0018,0x0080),
103   gdcm.Tag(0x0018,0x0081),
104   gdcm.Tag(0x0018,0x0083),
105   gdcm.Tag(0x0018,0x0084),
106   gdcm.Tag(0x0018,0x0085),
107   gdcm.Tag(0x0018,0x0086),
108   gdcm.Tag(0x0018,0x0087),
109   gdcm.Tag(0x0018,0x0091),
110   gdcm.Tag(0x0018,0x0093),
111   gdcm.Tag(0x0018,0x0094),
112   gdcm.Tag(0x0018,0x0095),
113   gdcm.Tag(0x0018,0x1088),
114   gdcm.Tag(0x0018,0x1090),
115   gdcm.Tag(0x0018,0x1094),
116   gdcm.Tag(0x0018,0x1250),
117   gdcm.Tag(0x0018,0x1251),
118   gdcm.Tag(0x0018,0x1310),
119   gdcm.Tag(0x0018,0x1312),
```

```
120   gdcm.Tag(0x0018,0x1314),
121   gdcm.Tag(0x0018,0x1315),
122   gdcm.Tag(0x0018,0x1316),
123   gdcm.Tag(0x0020,0x0110),
124   gdcm.Tag(0x0028,0x0120),
125   gdcm.Tag(0x0028,0x1050),
126   gdcm.Tag(0x0028,0x1051)
127   ]
128   for tag in taglist:
129     #print tag
130     ano.Remove( tag )
131
132   # special handling
133   gen = gdcm.UIDGenerator()
134   ano.Replace( gdcm.Tag(0x0008,0x9123), gen.Generate() )
135   #ano.Empty( gdcm.Tag(0x0040,0x0555) )
136
137
138 #
139 #  uid = gen.Generate()
140 #  de.SetTag( gdcm.Tag(0x0008,0x0018) )
141 #  de.SetByteStringValue( uid )
142 #  ds.Insert( de )
143
144   # init FMI now:
145   #fmi = f.GetHeader()
146   #ts = gdcm.TransferSyntax()
147   #print ts
148   #fmi.SetDataSetTransferSyntax( ts ) # default
149   #print fmi.GetDataSetTransferSyntax()
150   #de.SetTag( gdcm.Tag(0x0002,0x0010) )
151   #uid = "1.2.840.10008.1.2"
152   #de.SetByteStringValue( uid )
153   #fmi.Insert( de )
154 #  f.SetHeader( r.GetFile().GetHeader() )
155
156   writer = gdcm.Writer()
157   writer.SetFile( ano.GetFile() )
158   writer.SetFileName( "rawstorage.dcm" );
159   writer.Write()
```

12.117 DecompressImage.py

```
1
14
15 """
16 Usage:
17
18 python DecompressImage.py gdcmData/012345.002.050.dcm decompress.dcm
19 """
20
21 import gdcm
22 import sys
23
24 if __name__ == "__main__":
25
26   file1 = sys.argv[1]
27   file2 = sys.argv[2]
28
29   r = gdcm.ImageReader()
30   r.SetFileName( file1 )
31   if not r.Read():
32     sys.exit(1)
33
34   # check GetFragment API:
35   pd = r.GetFile().GetDataSet().GetDataElement(gdcm.Tag(0x7fe0, 0x0010))
36   frags = pd.GetSequenceOfFragments();
37   frags.GetFragment(0);
38
39   ir = r.GetImage()
40   w = gdcm.ImageWriter()
41   image = w.GetImage()
42
43   image.SetNumberOfDimensions( ir.GetNumberOfDimensions() );
44   dims = ir.GetDimensions();
45   print ir.GetDimension(0);
46   print ir.GetDimension(1);
```

```
47   print "Dims:",dims
48
49   #  Just for fun:
50   dircos =  ir.GetDirectionCosines()
51   t = gdcm.Orientation.GetType(tuple(dircos))
52   l = gdcm.Orientation.GetLabel(t)
53   print "Orientation label:",l
54
55   image.SetDimension(0, ir.GetDimension(0) );
56   image.SetDimension(1, ir.GetDimension(1) );
57
58   pixeltype = ir.GetPixelFormat();
59   image.SetPixelFormat( pixeltype );
60
61   pi = ir.GetPhotometricInterpretation();
62   image.SetPhotometricInterpretation( pi );
63
64   pixeldata = gdcm.DataElement( gdcm.Tag(0x7fe0,0x0010) )
65   str1 = ir.GetBuffer()
66   #print ir.GetBufferLength()
67   pixeldata.SetByteStringValue( str1 )
68   image.SetDataElement( pixeldata )
69
70   w.SetFileName( file2 )
71   w.SetFile( r.GetFile() )
72   w.SetImage( image )
73   if not w.Write():
74     sys.exit(1)
```

12.118 DumbAnonymizer.py

```
1
14
15 """
16 This example shows how one can use the gdcm.Anonymizer in 'dumb' mode.
17 This class becomes really handy when one knows which particular tag to fill in.
18
19 Usage:
20
21 python DumbAnonymizer.py gdcmData/012345.002.050.dcm out.dcm
22
23 """
24
25 import gdcm
26
27 # http://www.oid-info.com/get/1.3.6.1.4.17434
28 THERALYS_ORG_ROOT = "1.3.6.1.4.17434"
29
30 tag_rules={
31   # Value
32   (0x0012,0x0010):("Value","MySponsorName"),
33   (0x0012,0x0020):("Value","MyProtocolID"),
34   (0x0012,0x0021):("Value","MyProtocolName"),
35   (0x0012,0x0062):("Value","YES"),
36   (0x0012,0x0063):("Value","MyDeidentificationMethod"),
37
38   # Method
39   #(0x0002,0x0003):("Method","GenerateMSOPId"),
40   #(0x0008,0x1155):("Method","GenerateMSOPId"),
41   (0x0008,0x0018):("Method","GenerateMSOPId"),
42   (0x0010,0x0010):("Method","GetSponsorInitials"),
43   (0x0010,0x0020):("Method","GetSponsorId"),
44   (0x0012,0x0030):("Method","GetSiteId"),
45   (0x0012,0x0031):("Method","GetSiteName"),
46   (0x0012,0x0040):("Method","GetSponsorId"),
47   (0x0012,0x0050):("Method","GetTPId"),
48   (0x0018,0x0022):("Method","KeepIfExist"),
49   (0x0018,0x1315):("Method","KeepIfExist"),
50   (0x0020,0x000d):("Method","GenerateStudyId"),
51   (0x0020,0x000e):("Method","GenerateSeriesId"),
52   (0x0020,0x1002):("Method","GetNumberOfFrames"),
53   (0x0020,0x0020):("Method","GetPatientOrientation"),
54   # Other:
55   (0x0012,0x0051):("Patient Field","Type Examen"),
56   (0x0018,0x1250):("Sequence Field","Receive Coil"),
57   (0x0018,0x0088):("Sequence Field","Spacing Between Slice"),
58   (0x0018,0x0095):("Sequence Field","Pixel Bandwidth"),
```

```
59   (0x0018,0x0082):("Sequence Field","Invertion Time"),
60 }
61
62 class MyAnon:
63   def __init__(self):
64     self.studyuid = None
65     self.seriesuid = None
66     generator = gdcm.UIDGenerator()
67     if not self.studyuid:
68       self.studyuid = generator.Generate()
69     if not self.seriesuid:
70       self.seriesuid = generator.Generate()
71   def GetSponsorInitials(self):
72     return "dummy^foobar"
73   def GenerateStudyId(self):
74     return self.studyuid
75   def GenerateSeriesId(self):
76     return self.seriesuid
77   #def GenerateMSOPId(self):
78   def GenerateMSOPId(self):
79     generator = gdcm.UIDGenerator()
80     return generator.Generate()
81   def GetSiteId(self):
82     return "MySiteId"
83   def GetSiteName(self):
84     return "MySiteName"
85   def GetSponsorId(self):
86     return "MySponsorId"
87   def GetTPId(self):
88     return "MyTP"
89
90 if __name__ == "__main__":
91   import sys
92   gdcm.FileMetaInformation.SetSourceApplicationEntityTitle( "DumbAnonymizer" )
93   gdcm.UIDGenerator.SetRoot( THERALYS_ORG_ROOT )
94
95   r = gdcm.Reader()
96   filename = sys.argv[1]
97   r.SetFileName( filename )
98   if not r.Read():  sys.exit(1)
99
100   obj = MyAnon()
101
102   w = gdcm.Writer()
103   ano = gdcm.Anonymizer()
104   ano.SetFile( r.GetFile() )
105   ano.RemoveGroupLength()
106   for tag,rule in tag_rules.items():
107     if rule[0] == 'Value':
108       print tag,rule
109       ano.Replace( gdcm.Tag( tag[0], tag[1] ), rule[1] )
110     elif rule[0] == 'Method':
111       print tag,rule
112       # result = locals()[rule[1]]()
113       methodname = rule[1]
114       if hasattr(obj, methodname):
115         _member = getattr(obj, methodname)
116         result = _member()
117         ano.Replace( gdcm.Tag( tag[0], tag[1] ), result )
118       else:
119         print "Problem with:  ", methodname
120
121   outfilename = sys.argv[2]
122   w.SetFileName( outfilename )
123   w.SetFile( ano.GetFile() )
124   if not w.Write():  sys.exit(1)
```

12.119 ExtractImageRegion.py

```
1
14
15 """
16
17 This small code shows how to use the gdcm.ImageRegionReader API
18 In this example we are taking each frame by frame and dump them to
19 /tmp/frame.raw.
20
```

```
21 Usage:
22 $ ExtractImageRegion.py input.dcm
23
24 Example:
25 $ ExtractImageRegion.py gdcmData/012345.002.050.dcm
26 $ md5sum /tmp/frame.raw
27 d594a5e2fde12f32b6633ca859b4d4a6  /tmp/frame.raw
28 $ gdcminfo --md5sum gdcmData/012345.002.050.dcm
29 [...]
30 md5sum:  d594a5e2fde12f32b6633ca859b4d4a6
31 """
32
33 import gdcm
34
35 if __name__ == "__main__":
36 import sys
37   filename = sys.argv[1]
38
39   file_size = gdcm.System.FileSize(filename);
40
41   # instantiate the reader:
42   reader = gdcm.ImageRegionReader();
43   reader.SetFileName( filename );
44
45   # pull DICOM info:
46   if not reader.ReadInformation():
47     sys.exit(1)
48
49   # store current offset:
50   cur_pos = reader.GetStreamCurrentPosition();
51
52   remaining = file_size - cur_pos;
53
54   print("Remaining bytes to read (Pixel Data):  %d" % remaining );
55
56   # Get file infos
57   f = reader.GetFile();
58
59   # get some info about image
60   dims = gdcm.ImageHelper.GetDimensionsValue(f);
61   print(dims)
62   pf = gdcm.ImageHelper.GetPixelFormatValue (f);
63   pixelsize = pf.GetPixelSize();
64   pi = gdcm.ImageHelper.GetPhotometricInterpretationValue(f);
65   print( pi );
66
67   # buffer to get the pixels
68   buffer = bytearray( dims[0] * dims[1] * pixelsize )
69
70   # define a simple box region.
71   box = gdcm.BoxRegion();
72   for z in range(0, dims[2]):
73     # Define that I want the image 0, full size (dimx x dimy pixels)
74     # and do that for each z:
75     box.SetDomain(0, dims[0] - 1, 0, dims[1] - 1, z, z);
76     #print( box.toString() );
77     reader.SetRegion( box );
78
79     # reader will try to load the uncompressed image region into buffer.
80     # the call returns an error when buffer.Length is too small.  For instance
81     # one can call:
82     # uint buf_len = reader.ComputeBufferLength(); // take into account pixel size
83     # to get the exact size of minimum buffer
84     if reader.ReadIntoBuffer(buffer):
85       open('/tmp/frame.raw', 'wb').write(buffer)
86     else:
87       #throw new Exception("can't read pixels error");
88       sys.exit(1)
```

12.120 FindAllPatientName.py

```
1
14 """
15 This example shows how one can use the gdcm.CompositeNetworkFunctions class
16 for executing a C-FIND query
17 It will print the list of patient name found
18
```

```
Usage:

python FindAllPatientName.py

"""

import gdcm

# Patient Name
tag = gdcm.Tag(0x10,0x10)
de = gdcm.DataElement(tag)

# Search all patient name where string match 'F*'
de.SetByteStringValue('F*')

ds = gdcm.DataSet()
ds.Insert(de)

cnf = gdcm.CompositeNetworkFunctions()
theQuery = cnf.ConstructQuery (gdcm.ePatientRootType,gdcm.ePatient,ds)

#print theQuery.ValidateQuery()

# prepare the variable for output
ret = gdcm.DataSetArrayType()

# Execute the C-FIND query
cnf.CFind('dicom.example.com',11112,theQuery,ret,'GDCM_PYTHON','ANY-SCP')

for i in range(0,ret.size()):
  print "Patient #",i
  print ret[i]
```

12.121 FixCommaBug.py

```

"""
Using LC_NUMERIC set to something not compatible with "C" it is possible to write out "," instead of
"." as required by the DICOM standard
Issue is still current (IMHO) with gdcm 2.0.9
"""

import gdcm
import sys

filename = sys.argv[1]
outname = sys.argv[2]

# read
r = gdcm.Reader()
r.SetFileName( filename )
if not r.Read():
  print "not valid"
  sys.exit(1)

file = r.GetFile()
dataset = file.GetDataSet()

ano = gdcm.Anonymizer()
ano.SetFile( file )

tags = [
gdcm.Tag(0x0018,0x1164),
gdcm.Tag(0x0018,0x0088),
gdcm.Tag(0x0018,0x0050),
gdcm.Tag(0x0028,0x0030),
]

for tag in tags:
  print tag
  if dataset.FindDataElement( tag ):
    pixelspacing = dataset.GetDataElement( tag )
    #print pixelspacing
    bv = pixelspacing.GetByteValue()
    str = bv.GetBuffer()
    #print bv.GetLength()
```

```
55     #print len(str)
56     new_str = str.replace(",",".")
57     # Need to explicitly pass bv.GetLength() to remove any trailing garbage
58     ano.Replace( tag, new_str, bv.GetLength() )
59
60 #print dataset
61
62 w = gdcm.Writer()
63 w.SetFile( file )
64 w.SetFileName( outname )
65 if not w.Write():
66   print "Cannot write"
67   sys.exit(1)
68
69 # paranoid:
70 image_reader = gdcm.ImageReader()
71 image_reader.SetFileName( outname )
72 if not image_reader.Read():
73   print "there is still a comma"
74   sys.exit(1)
75
76 print "Sucess!"
77 sys.exit(0) # success
```

12.122 GetPortionCSAHeader.py

```
1
14
15 """
16 Usage:
17
18 python GetPortionCSAHeader.py input.dcm
19
20 Footnote:
21 SIEMENS is not publishing any information on the CSA header.  So any info extracted
22   is at your own risk.
23 """
24
25 import sys
26 import gdcm
27
28 if __name__ == "__main__":
29
30 file = sys.argv[1]
31
32 r = gdcm.Reader()
33 r.SetFileName( file )
34 if not r.Read():
35     sys.exit(1)
36
37   ds = r.GetFile().GetDataSet()
38   csa_t1 = gdcm.CSAHeader()
39   csa_t2 = gdcm.CSAHeader()
40   #print csa
41   t1 = csa_t1.GetCSAImageHeaderInfoTag();
42   print t1
43   t2 = csa_t2.GetCSASeriesHeaderInfoTag();
44   print t2
45   # Let's do it for t1:
46   if ds.FindDataElement( t1 ):
47     csa_t1.LoadFromDataElement( ds.GetDataElement( t1 ) )
48     print csa_t1
49
50   # Now let's pretend we are only interested in B_value and DiffusionGradientDirection entries:
51   bvalues = csa_t1.GetCSAElementByName( "B_value" ) # WARNING: it is case sensitive !
52   print bvalues
53
54   diffgraddir = csa_t1.GetCSAElementByName( "DiffusionGradientDirection" ) # WARNING: it is case sensitive !
55   print diffgraddir
56
57   # repeat for t2 if you like it:
58   if ds.FindDataElement( t2 ):
59     csa_t2.LoadFromDataElement( ds.GetDataElement( t2 ) )
60     # print csa_t2
61
62   gdt = csa_t2.GetCSAElementByName( "GradientDelayTime" )
63   print gdt
```

```
64
65   bv = gdt.GetByteValue();
66   #print bv
67   str = bv.GetPointer()
68   print str.split("\\")
```

12.123 HelloWorld.py

```
1
14
15 """
16 Hello World !
17 """
18
19 import gdcm
20 import sys
21
22 if __name__ == "__main__":
23
24   # verbosity:
25   #gdcm.Trace.DebugOn()
26   #gdcm.Trace.WarningOn()
27   #gdcm.Trace.ErrorOn()
28
29   # Get the filename from the command line
30   filename = sys.argv[1]
31
32   # Instanciate a gdcm.Reader
33   # This is the main class to handle any type of DICOM object
34   # You should check for gdcm.ImageReader for reading specifically DICOM Image file
35   r = gdcm.Reader()
36   r.SetFileName( filename )
37   # If the reader fails to read the file, we should stop !
38   if not r.Read():
39     print "Not a valid DICOM file"
40     sys.exit(1)
41
42   # Get the DICOM File structure
43   file = r.GetFile()
44
45   # Get the DataSet part of the file
46   dataset = file.GetDataSet()
47
48   # Ok let's print it !
49   print dataset
50
51   # Use StringFilter to print a particular Tag:
52   sf = gdcm.StringFilter()
53   sf.SetFile(r.GetFile())
54
55   # Check if Attribute exist
56   print dataset.FindDataElement( gdcm.Tag(0x0028,0x0010))
57
58   # Let's print it as string pair:
59   print sf.ToStringPair(gdcm.Tag(0x0028,0x0010))
```

12.124 ManipulateFile.py

```
1
14
15 """
16 Usage:
17
18 python ManipulateFile.py input.dcm output.dcm
19
20 Footnote:
21 GDCM 1.2.x would create incorrect Multiframes MR Image Storage file.  Try to recover from
22   the issues to recreate a MultiframeGrayscaleByteSecondaryCaptureImageStorage file.
23   e.g:
24
25   python ManipulateFile.py Insight/Testing/Temporary/itkGDCMImageIOTest5-j2k.dcm manipulated.dcm
26 """
27
```

```
28 import sys
29 import gdcm
30
31 if __name__ == "__main__":
32
33 file1 = sys.argv[1]
34 file2 = sys.argv[2]
35
36 r = gdcm.Reader()
37 r.SetFileName( file1 )
38 if not r.Read():
39     sys.exit(1)
40
41   ano = gdcm.Anonymizer()
42   ano.SetFile( r.GetFile() )
43   ano.RemovePrivateTags()
44   ano.Remove( gdcm.Tag(0x0032,0x1030) )
45   ano.Remove( gdcm.Tag(0x008,0x14) )
46   ano.Remove( gdcm.Tag(0x008,0x1111) )
47   ano.Remove( gdcm.Tag(0x008,0x1120) )
48   ano.Remove( gdcm.Tag(0x008,0x1140) )
49   ano.Remove( gdcm.Tag(0x10,0x21b0) )
50   ano.Empty( gdcm.Tag(0x10,0x10) )
51   ano.Empty( gdcm.Tag(0x10,0x20) )
52   ano.Empty( gdcm.Tag(0x10,0x30) )
53   ano.Empty( gdcm.Tag(0x20,0x10) )
54   ano.Empty( gdcm.Tag(0x32,0x1032) )
55   ano.Empty( gdcm.Tag(0x32,0x1033) )
56   ano.Empty( gdcm.Tag(0x40,0x241) )
57   ano.Empty( gdcm.Tag(0x40,0x254) )
58   ano.Empty( gdcm.Tag(0x40,0x253) )
59   ano.Empty( gdcm.Tag(0x40,0x1001) )
60   ano.Empty( gdcm.Tag(0x8,0x80) )
61   ano.Empty( gdcm.Tag(0x8,0x50) )
62   ano.Empty( gdcm.Tag(0x8,0x1030) )
63   ano.Empty( gdcm.Tag(0x8,0x103e) )
64   ano.Empty( gdcm.Tag(0x18,0x1030) )
65   ano.Empty( gdcm.Tag(0x38,0x300) )
66   g = gdcm.UIDGenerator()
67   ano.Replace( gdcm.Tag(0x0008,0x0018), g.Generate() )
68   ano.Replace( gdcm.Tag(0x0020,0x00d), g.Generate() )
69   ano.Replace( gdcm.Tag(0x0020,0x00e), g.Generate() )
70   ano.Replace( gdcm.Tag(0x0020,0x052), g.Generate() )
71   #ano.Replace( gdcm.Tag(0x0008,0x0016), "1.2.840.10008.5.1.4.1.1.7.2" )
72   """
73 ano.Remove( gdcm.Tag(0x0018,0x0020) ) # ScanningSequence
74   ano.Remove( gdcm.Tag(0x0018,0x0021) ) # SequenceVariant
75   ano.Remove( gdcm.Tag(0x0018,0x0022) ) # ScanOptions
76   ano.Remove( gdcm.Tag(0x0018,0x0023) ) # MRAcquisitionType
77   ano.Remove( gdcm.Tag(0x0018,0x0050) ) # SliceThickness
78   ano.Remove( gdcm.Tag(0x0018,0x0080) ) # RepetitionTime
79   ano.Remove( gdcm.Tag(0x0018,0x0081) ) # EchoTime
80   ano.Remove( gdcm.Tag(0x0018,0x0088) ) # SpacingBetweenSlices
81   ano.Remove( gdcm.Tag(0x0018,0x0091) ) # EchoTrainLength
82   ano.Remove( gdcm.Tag(0x0018,0x1164) ) # ImagerPixelSpacing
83
84   ano.Remove( gdcm.Tag(0x0020,0x0032) ) # Image Position (Patient)
85   ano.Remove( gdcm.Tag(0x0020,0x0037) ) # Image Orientation (Patient)
86   ano.Remove( gdcm.Tag(0x0020,0x0052) ) # Frame of Reference UID
87   ano.Remove( gdcm.Tag(0x0020,0x1040) ) # Position Reference Indicator
88
89   ano.Replace( gdcm.Tag(0x0028,0x0301), "NO" ) # Burned In Annotation
90
91   ano.Empty( gdcm.Tag(0x0020,0x0020) )
92
93   ano.Remove( gdcm.Tag(0x7fe0,0x0000) )
94
95   #ano.Empty( gdcm.Tag(0x0028,0x0009) ) # Frame Increment Pointer
96
97   #ano.Empty( gdcm.Tag(0x0028,0x1052) )  #<entry group="0028" element="1052" vr="DS" vm="1" name="Rescale
      Intercept"/>
98   #ano.Empty( gdcm.Tag(0x0028,0x1053) )  #<entry group="0028" element="1053" vr="DS" vm="1" name="Rescale
      Slope"/>
99   #ano.Replace( gdcm.Tag(0x0028,0x1054), "US" )  #<entry group="0028" element="1054" vr="LO" vm="1"
      name="Rescale Type"/>
100
101   ano.Replace( gdcm.Tag(0x2050, 0x0020), "IDENTITY")
102   """
103
104 w = gdcm.Writer()
105 w.SetFile( ano.GetFile() )
```

```
106 w.SetFileName( file2 )
107 if not w.Write():
108     sys.exit(1)
```

12.125 ManipulateSequence.py

```
1
14
15 """
16 Usage:
17
18 python ManipulateSequence.py input.dcm output.dcm
19
20 This was tested using:
21
22 python ManipulateSequence.py gdcmData/D_CLUNIE_CT1_J2KI.dcm myoutput.dcm
23
24 This is a dummy example on how to modify a value set in a nested-nested dataset
25
26 WARNING:
27 Do not use as-is in production, this is just an example
28 This example works in an undefined length Item only (you need to explicitly recompute the length otherwise)
29 """
30
31 import sys
32 import gdcm
33
34 if __name__ == "__main__":
35
36   file1 = sys.argv[1]
37   file2 = sys.argv[2]
38
39   r = gdcm.Reader()
40   r.SetFileName( file1 )
41   if not r.Read():
42     sys.exit(1)
43
44   f = r.GetFile()
45   ds = f.GetDataSet()
46   tsis = gdcm.Tag(0x0008,0x2112) # SourceImageSequence
47   if ds.FindDataElement( tsis ):
48     sis = ds.GetDataElement( tsis )
49     #sqsis = sis.GetSequenceOfItems()
50     # GetValueAsSQ handle more cases
51     sqsis = sis.GetValueAsSQ()
52     if sqsis.GetNumberOfItems():
53       item1 = sqsis.GetItem(1)
54       nestedds = item1.GetNestedDataSet()
55       tprcs = gdcm.Tag(0x0040,0xa170) # PurposeOfReferenceCodeSequence
56       if nestedds.FindDataElement( tprcs ):
57         prcs = nestedds.GetDataElement( tprcs )
58         sqprcs = prcs.GetSequenceOfItems()
59         if sqprcs.GetNumberOfItems():
60           item2 = sqprcs.GetItem(1)
61           nestedds2 = item2.GetNestedDataSet()
62           # (0008,0104) LO [Uncompressed predecessor]               #  24, 1 CodeMeaning
63           tcm = gdcm.Tag(0x0008,0x0104)
64           if nestedds2.FindDataElement( tcm ):
65             cm = nestedds2.GetDataElement( tcm )
66             mystr = "GDCM was here"
67             cm.SetByteStringValue( mystr )
68
69   w = gdcm.Writer()
70   w.SetFile( f )
71   w.SetFileName( file2 )
72   if not w.Write():
73     sys.exit(1)
```

12.126 MergeFile.py

```
1
14
15 """
```

```
16 Usage:
17
18 python MergeFile.py input1.dcm input2.dcm
19
20 It will produce a 'merge.dcm' output file, which contains all meta information from input1.dcm
21   and copy the Stored Pixel values from input2.dcm
22   This script even works when input2.dcm is a Secondary Capture and does not contains information
23   such as IOP and IPP...
24 """
25
26 import sys
27 import gdcm
28
29 if __name__ == "__main__":
30
31 file1 = sys.argv[1]
32 file2 = sys.argv[2]
33
34 r1 = gdcm.ImageReader()
35 r1.SetFileName( file1 )
36 if not r1.Read():
37     sys.exit(1)
38
39   r2 = gdcm.ImageReader()
40   r2.SetFileName( file2 )
41   if not r2.Read():
42     sys.exit(1)
43
44   # Image from r2 could be Secondary Capture and thus would not contains neither IPP nor IOP
45   # Instead always prefer to only copy the Raw Data Element.
46   # Warning !  Image need to be identical !  Only the value of Stored Pixel can be different.
47   r1.GetImage().SetDataElement( r2.GetImage().GetDataElement() )
48
49   w = gdcm.ImageWriter()
50   w.SetFile( r1.GetFile() )
51   #w.SetImage( r2.GetImage() )  # See comment above
52   w.SetImage( r1.GetImage() )
53
54   w.SetFileName( "merge.dcm" )
55   if not w.Write():
56     sys.exit(1)
57
58   sys.exit(0)
```

12.127 NewSequence.py

```
1
14
15 """
16 Usage:
17
18 python NewSequence.py input.dcm output.dcm
19
20
21 Thanks to Robert Irie for code
22 """
23
24 import sys
25 import gdcm
26
27 if __name__ == "__main__":
28
29 file1 = sys.argv[1]
30 file2 = sys.argv[2]
31
32 r = gdcm.Reader()
33 r.SetFileName( file1 )
34 if not r.Read():
35     sys.exit(1)
36
37   f = r.GetFile()
38   ds = f.GetDataSet()
39   #tsis = gdcm.Tag(0x0008,0x2112) # SourceImageSequence
40
41   # Create a dataelement
42   de = gdcm.DataElement(gdcm.Tag(0x0010, 0x2180))
43   de.SetByteStringValue("Occupation")
```

```
44   de.SetVR(gdcm.VR(gdcm.VR.SH))
45
46   # Create an item
47   it=gdcm.Item()
48   it.SetVLToUndefined()      # Needed to not popup error message
49   #it.InsertDataElement(de)
50   nds=it.GetNestedDataSet()
51   nds.Insert(de)
52
53   # Create a Sequence
54   sq=gdcm.SequenceOfItems().New()
55   sq.SetLengthToUndefined()
56   sq.AddItem(it)
57
58   # Insert sequence into data set
59   des=gdcm.DataElement(gdcm.Tag(0x0400,0x0550))
60   des.SetVR(gdcm.VR(gdcm.VR.SQ))
61   des.SetValue(sq.__ref__())
62   des.SetVLToUndefined()
63
64   ds.Insert(des)
65
66   w = gdcm.Writer()
67   w.SetFile( f )
68   w.SetFileName( file2 )
69   if not w.Write():
70     sys.exit(1)
```

12.128 PhilipsPrivateRescaleInterceptSlope.py

```
1
14
15 """
16 Usage:
17
18 python
19 """
20
21 import gdcm
22 import sys
23
24 filename = sys.argv[1]
25 tmpfile = "/tmp/philips_rescaled.dcm"
26
27
28 # Need to access some private tags, read the file :
29 reader = gdcm.Reader()
30 reader.SetFileName( filename )
31 if not reader.Read():
32   sys.exit(1)
33
34 ds = reader.GetFile().GetDataSet()
35
36 #print ds
37 # (2005,1409)     DS      4       0.0
38 # (2005,140a)     DS      16      1.52283272283272
39
40 # (2005,0014)     LO      26      Philips MR Imaging DD 005
41 tag1 = gdcm.PrivateTag(0x2005,0x09,"Philips MR Imaging DD 005")
42 tag2 = gdcm.PrivateTag(0x2005,0x0a,"Philips MR Imaging DD 005")
43 print tag1
44 print tag2
45
46 # make sure to do a copy, we want the private tag to remain
47 # otherwise gdcm gives us a reference
48 el1 = gdcm.DataElement( ds.GetDataElement( tag1 ) )
49 print el1
50 el2 = gdcm.DataElement( ds.GetDataElement( tag2 ) )
51 print el2
52
53 # (0028,1052) DS [-1000]                                  #   6, 1 RescaleIntercept
54 # (0028,1053) DS [1]                                      #   2, 1 RescaleSlope
55
56 el1.SetTag( gdcm.Tag(0x0028,0x1052) )
57 el2.SetTag( gdcm.Tag(0x0028,0x1053) )
58
59 ds.Insert( el1 )
```

```
60 ds.Insert( el2 )
61
62 w = gdcm.Writer()
63 w.SetCheckFileMetaInformation( False )
64 w.SetFileName( tmpfile )
65 w.SetFile( reader.GetFile() )
66 if not w.Write():
67   sys.exit(1)
68
69 print "success"
```

12.129 PlaySound.py

```
1
14
15 """
16 Usage:
17
18 python PlaySound.py input.dcm
19 """
20
21 import gdcm
22 import sys
23
24 #filename = "/home/mmalaterre/Creatis/gdcmDataExtra/gdcmNonImageData/audio_from_rafael_sanguinetti.dcm"
25 filename = sys.argv[1]
26 print filename
27
28 r = gdcm.Reader()
29 r.SetFileName( filename )
30 if not r.Read():
31   sys.exit(1)
32
33 ds = r.GetFile().GetDataSet()
34
35 waveformtag = gdcm.Tag(0x5400,0x0100)
36 waveformsq = ds.GetDataElement( waveformtag )
37 #print waveformsq
38
39 #print dir(waveformsq)
40
41 items = waveformsq.GetSequenceOfItems()
42
43 if not items.GetNumberOfItems():
44   sys.exit(1)
45
46 item = items.GetItem(1)
47 #print item
48
49 waveformds = item.GetNestedDataSet()
50 #print waveformds
51
52 waveformdatatag = gdcm.Tag(0x5400,0x1010)
53 waveformdata = waveformds.GetDataElement( waveformdatatag )
54
55 #print waveformdata.GetPointer()
56 bv = waveformdata.GetByteValue()
57 print dir(bv)
58
59 #print bv.GetPointer()
60 print bv.GetLength()
61 l = 116838
62
63 file='test.wav'
64 myfile = open(file, "wb")
65 s = bv.GetPointer()
66 for i in range(0, l):
67   myfile.write(s[i])
68 myfile.close()
69
70 # http://mail.python.org/pipermail/python-list/2004-October/288905.html
71 if sys.platform.startswith('win'):
72    from winsound import PlaySound, SND_FILENAME, SND_ASYNC
73    PlaySound(file, SND_FILENAME|SND_ASYNC)
74 elif sys.platform.find('linux')>-1:
75    from wave import open as waveOpen
76    from ossaudiodev import open as ossOpen
```

```
77    s = waveOpen(file,'rb')
78    (nc,sw,fr,nf,comptype, compname) = s.getparams( )
79    dsp = ossOpen('/dev/dsp','w')
80    try:
81      from ossaudiodev import AFMT_S16_NE
82    except ImportError:
83      if byteorder == "little":
84        AFMT_S16_NE = ossaudiodev.AFMT_S16_LE
85      else:
86        AFMT_S16_NE = ossaudiodev.AFMT_S16_BE
87    dsp.setparameters(AFMT_S16_NE, nc, fr)
88    data = s.readframes(nf)
89    s.close()
90    dsp.write(data)
91    dsp.close()
```

12.130 PrivateDict.py

```
1
14
15 """
16 """
17
18 import gdcm
19 import sys,os
20
21 if __name__ == "__main__":
22   #gdcm.Trace.DebugOn()
23   globInst = gdcm.Global.GetInstance()
24   # Try to load Part3.xml file
25   # This fils is too big for being accessible directly at runtime.
26   globInst.LoadResourcesFiles()
27
28
29   # Get a private tag from the runtime dicts.  LoadResourcesFiles could
30   # have failed but this has no impact on the private dict
31
32   d = globInst.GetDicts()
33   print d.GetDictEntry( gdcm.Tag(0x0029,0x0010) ,"SIEMENS CSA HEADER" )
34   pd = d.GetPrivateDict()
35   print pd.GetDictEntry( gdcm.PrivateTag(0x0029,0x0010,"SIEMENS CSA HEADER") )
```

12.131 ReWriteSCAsMR.py

```
1
14
15 """
16 GDCM 1.x would write out MR Image Storage as Secondary Capture Object while still setting Rescale
       Slope/Intercept
17 and saving the Pixel Spacing in (0028,0030)
18 """
19
20 import gdcm
21 import sys,os
22
23 def CheckSecondaryCaptureObjectIsMRImageStorage(r):
24   ds = r.GetFile().GetDataSet()
25   # Check Source Image Sequence
26   if ds.FindDataElement( gdcm.Tag(0x0008,0x2112) ):
27     sis = ds.GetDataElement( gdcm.Tag(0x0008,0x2112) )
28     sqsis = sis.GetSequenceOfItems()
29     if sqsis.GetNumberOfItems():
30       item1 = sqsis.GetItem(1)
31       nestedds = item1.GetNestedDataSet()
32       if nestedds.FindDataElement( gdcm.Tag(0x0008,0x1150) ):
33         ReferencedSOPClassUID = nestedds.GetDataElement( gdcm.Tag(0x0008,0x1150) )
34         raw = ReferencedSOPClassUID.GetByteValue().GetPointer()
35         uids = gdcm.UIDs()
36         # what is the actual object we are looking at ?
37         ms = gdcm.MediaStorage()
38         ms.SetFromDataSet(ds)
39         msuid = ms.GetString()
40         uids.SetFromUID( msuid )
```

```
41          msuidname = uids.GetName() # real Media Storage Name
42          uids.SetFromUID( raw )
43          sqmsuidname = uids.GetName() # Source Image Sequence Media Storage Name
44          # If object is SC and Source derivation is MRImageStorage then we can assume 'Pixel Spacing' is
     correct
45          if( sqmsuidname == 'MR Image Storage' and msuidname == 'Secondary Capture Image Storage' ):
46            return True
47    # in all other case simply return the currentspacing:
48    return False
49
50  if __name__ == "__main__":
51    r = gdcm.ImageReader()
52    filename = sys.argv[1]
53    r.SetFileName( filename )
54    if not r.Read():
55      sys.exit(1)
56    f = r.GetFile()
57
58    if( CheckSecondaryCaptureObjectIsMRImageStorage(r) ):
59      # Special handling of the spacing:
60      # GDCM 1.2.0 would not rewrite correcly DICOM Object and would always set them as 'Secondary Capture
     Image Storage'
61      # while we would rather have 'MR Image Storage'
62      gdcm.ImageHelper.SetForcePixelSpacing( True )
63      mrspacing = gdcm.ImageHelper.GetSpacingValue( r.GetFile() )
64      # TODO: I cannot do simply the following:
65      #image.SetSpacing( mrspacing )
66      image.SetSpacing(0, mrspacing[0] )
67      image.SetSpacing(1, mrspacing[1] )
68      image.SetSpacing(2, mrspacing[2] )
69      gdcm.ImageHelper.SetForceRescaleInterceptSlope( True )
70      ris = gdcm.ImageHelper.GetRescaleInterceptSlopeValue( r.GetFile() )
71      image.SetIntercept( ris[0] )
72      image.SetSlope( ris[1] )
73
74    outfilename = sys.argv[2]
75    w = gdcm.ImageWriter()
76    w.SetFileName( outfilename )
77    w.SetFile( r.GetFile() )
78    w.SetImage( image )
79    if not w.Write():
80      sys.exit(1)
81
82    sys.exit(0)
```

12.132 ReadAndDumpDICOMDIR.py

```
1
23
24
25
26  import sys
27  import gdcm
28
29  if __name__ == "__main__":
30      # Check arguments
31      if (len(sys.argv) < 2):
32          # No filename passed
33          print "No input filename found"
34          quit()
35
36      filename = sys.argv[1]
37
38
39      # Read file
40      reader = gdcm.Reader()
41      reader.SetFileName(filename)
42      if (not reader.Read()):
43          print "Unable to read %s" % (filename)
44          quit()
45
46      file = reader.GetFile()
47
48      # Retrieve header information
49      fileMetaInformation = file.GetHeader()
50      print fileMetaInformation
51
```

```
52      # Retrieve data set
53      dataSet = file.GetDataSet()
54      #print dataSet
55
56      # Check media storage
57      mediaStorage = gdcm.MediaStorage()
58      mediaStorage.SetFromFile(file)
59      if (gdcm.MediaStorage.GetMSType(str(mediaStorage)) != gdcm.MediaStorage.MediaStorageDirectoryStorage):
60          # File is not a DICOMDIR
61          print "This file is not a DICOMDIR (Media storage type:  %s)" % (str(mediaStorage))
62          quit()
63
64      # Check Media Storage SOP Class
65      if (fileMetaInformation.FindDataElement(gdcm.Tag(0x0002, 0x0002))):
66          sopClassUid = str(fileMetaInformation.GetDataElement(gdcm.Tag(0x0002, 0x0002)).GetValue())
67          # Check SOP UID
68          if (sopClassUid != "1.2.840.10008.1.3.10"):
69              # File is not a DICOMDIR
70              print "This file is not a DICOMDIR"
71      else:
72          # Not present
73          print "Media Storage SOP Class not present"
74          quit()
75
76      # Iterate through the DICOMDIR data set
77      iterator = dataSet.GetDES().begin()
78      while (not iterator.equal(dataSet.GetDES().end())):
79          dataElement = iterator.next()
80
81          # Check the element tag
82          if (dataElement.GetTag() == gdcm.Tag(0x004, 0x1220)):
83              # The 'Directory Record Sequence' element
84              sequence = dataElement.GetValueAsSQ()
85
86              # Loop through the sequence items
87              itemNr = 1
88              while (itemNr < sequence.GetNumberOfItems()):
89                  item = sequence.GetItem(itemNr)
90
91                  # Check the element tag
92                  if (item.FindDataElement(gdcm.Tag(0x0004, 0x1430))):
93                      # The 'Directory Record Type' element
94                      value = str(item.GetDataElement(gdcm.Tag(0x0004, 0x1430)).GetValue())
95
96                      # PATIENT
97                      while (value.strip() == "PATIENT"):
98                          print value.strip()
99                          # Print patient name
100                         if (item.FindDataElement(gdcm.Tag(0x0010, 0x0010))):
101                             value = str(item.GetDataElement(gdcm.Tag(0x0010, 0x0010)).GetValue())
102                             print value
103
104                         # Print patient ID
105                         if (item.FindDataElement(gdcm.Tag(0x0010, 0x0020))):
106                             value = str(item.GetDataElement(gdcm.Tag(0x0010, 0x0020)).GetValue())
107                             print value
108
109                         # Next
110                         itemNr = itemNr + 1
111                         item = sequence.GetItem(itemNr)
112                         if (item.FindDataElement(gdcm.Tag(0x0004, 0x1430))):
113                             value = str(item.GetDataElement(gdcm.Tag(0x0004, 0x1430)).GetValue())
114
115                         # STUDY
116                         while (value.strip() == "STUDY"):
117                             print value.strip()
118
119                             # Print study UID
120                             if (item.FindDataElement(gdcm.Tag(0x0020, 0x000d))):
121                                 value = str(item.GetDataElement(gdcm.Tag(0x0020, 0x000d)).GetValue())
122                                 print value
123
124                             # Print study date
125                             if (item.FindDataElement(gdcm.Tag(0x0008, 0x0020))):
126                                 value = str(item.GetDataElement(gdcm.Tag(0x0008, 0x0020)).GetValue())
127                                 print value
128
129                             # Print study description
130                             if (item.FindDataElement(gdcm.Tag(0x0008, 0x1030))):
131                                 value = str(item.GetDataElement(gdcm.Tag(0x0008, 0x1030)).GetValue())
132                                 print value
```

```
133
134                             # Next
135                             itemNr = itemNr + 1
136                             item = sequence.GetItem(itemNr)
137                             if (item.FindDataElement(gdcm.Tag(0x0004, 0x1430))):
138                                     value = str(item.GetDataElement(gdcm.Tag(0x0004, 0x1430)).GetValue())
139
140                             # SERIES
141                             while (value.strip() == "SERIES"):
142                                 print value.strip()
143
144                                 # Print series UID
145                                 if (item.FindDataElement(gdcm.Tag(0x0020, 0x000e))):
146                                     value = str(item.GetDataElement(gdcm.Tag(0x0020, 0x000e)).GetValue())
147                                     print value
148
149                                 # Print series modality
150                                 if (item.FindDataElement(gdcm.Tag(0x0008, 0x0060))):
151                                     value = str(item.GetDataElement(gdcm.Tag(0x0008, 0x0060)).GetValue())
152                                     print "Modality"
153                                     print value
154
155                                 # Print series description
156                                 if (item.FindDataElement(gdcm.Tag(0x0008, 0x103e))):
157                                     value = str(item.GetDataElement(gdcm.Tag(0x0008, 0x103e)).GetValue())
158                                     print "Description"
159                                     print value
160
161                                 # Next
162                                 itemNr = itemNr + 1
163                                 item = sequence.GetItem(itemNr)
164                                 if (item.FindDataElement(gdcm.Tag(0x0004, 0x1430))):
165                                     value = str(item.GetDataElement(gdcm.Tag(0x0004, 0x1430)).GetValue())
166
167                                 # IMAGE
168                                 while (value.strip() == "IMAGE"):
169                                     print value.strip()
170
171                                     # Print image UID
172                                     if (item.FindDataElement(gdcm.Tag(0x0004, 0x1511))):
173                                         value = str(item.GetDataElement(gdcm.Tag(0x0004,
      0x1511)).GetValue())
174                                         print value
175
176                                     # Next
177                                     if (itemNr < sequence.GetNumberOfItems()):
178                                         itemNr = itemNr + 1
179                                     else:
180                                         break
181
182                                     item = sequence.GetItem(itemNr)
183                                     if (item.FindDataElement(gdcm.Tag(0x0004, 0x1430))):
184                                             value = str(item.GetDataElement(gdcm.Tag(0x0004,
      0x1430)).GetValue())
185
186                 # Next
187                 itemNr = itemNr + 1
```

12.133 RemovePrivateTags.py

```
1
14
15 """
16 Usage:
17
18 python RemovePrivateTags.py input.dcm output.dcm
19 """
20
21 import sys
22 import gdcm
23
24
25 if __name__ == "__main__":
26
27   file1 = sys.argv[1]
28   file2 = sys.argv[2]
29
```

```
30   # Instanciate the reader.
31   r = gdcm.Reader()
32   r.SetFileName( file1 )
33   if not r.Read():
34     sys.exit(1)
35
36   # Remove private tags
37   ano = gdcm.Anonymizer()
38   ano.SetFile( r.GetFile() )
39   if not ano.RemovePrivateTags():
40     sys.exit(1)
41
42   # Write DICOM file
43   w = gdcm.Writer()
44   w.SetFile( ano.GetFile() )
45   #w.CheckFileMetaInformationOff() # Do not attempt to check meta header
46   w.SetFileName( file2 )
47   if not w.Write():
48     sys.exit(1)
49
50   # It is usually a good idea to exit the script with an error, as gdcm does not remove partial (incorrect)
      DICOM file
51   # (application level)
```

12.134 ScanDirectory.py

```
1
14
15 import gdcm
16 import sys,os
17
18 class ProgressWatcher(gdcm.SimpleSubjectWatcher):
19   def ShowProgress(self, sender, event):
20     pe = gdcm.ProgressEvent.Cast(event)
21     print pe.GetProgress()
22   def EndFilter(self):
23     print "Yay !  I am done"
24
25 if __name__ == "__main__":
26   directory = sys.argv[1]
27
28   # Define the set of tags we are interested in
29   t1 = gdcm.Tag(0x8,0x8);
30   t2 = gdcm.Tag(0x10,0x10);
31
32   # Iterate over directory
33   d = gdcm.Directory();
34   nfiles = d.Load( directory );
35   if(nfiles == 0):  sys.exit(1);
36   # System.Console.WriteLine( "Files:\n" + d.toString() );
37
38   filenames = d.GetFilenames()
39
40   #  Get rid of any Warning while parsing the DICOM files
41   gdcm.Trace.WarningOff()
42
43   # instanciate Scanner:
44   sp = gdcm.Scanner.New();
45   s = sp.__ref__()
46   w = ProgressWatcher(s, 'Watcher')
47
48   s.AddTag( t1 );
49   s.AddTag( t2 );
50   b = s.Scan( filenames );
51   if(not b):  sys.exit(1);
52
53   print "success" ;
54   #print s
55
56   pttv = gdcm.PythonTagToValue( s.GetMapping( filenames[1] ) )
57   pttv.Start()
58   # iterate until the end:
59   while( not pttv.IsAtEnd() ):
60     # get current value for tag and associated value:
61     # if tag was not found, then it was simply not added to the internal std::map
62     # Warning value can be None
63     tag = pttv.GetCurrentTag()
```

```
64     value = pttv.GetCurrentValue()
65     print tag,"->",value
66     # increment iterator
67     pttv.Next()
68
69   sys.exit(0)
```

12.135 SortImage.py

```
1
14
15 """
16 Usage:
17
18 python SortImage.py dirname
19 """
20
21 import gdcm
22 import sys
23
24 def PrintProgress(object, event):
25   assert event == "ProgressEvent"
26   print "Progress:", object.GetProgress()
27
28 def MySort(ds1, ds2):
29   # compare ds1
30   return False
31
32 if __name__ == "__main__":
33
34   dirname = sys.argv[1]
35   d = gdcm.Directory()
36   d.Load( dirname )
37
38   print d
39
40   sorter = gdcm.Sorter()
41   sorter.SetSortFunction( MySort )
42   #sorter.AddObserver( "ProgressEvent", PrintProgress )
43   sorter.Sort( d.GetFilenames() )
44
45   print "Sorter:"
46   print sorter
```

12.136 WriteBuffer.py

```
1
14
15 """
16 Usage:
17
18 http://chuckhahm.com/Ischem/Zurich/XX_0134
19
20 (2005,1132) SQ (Sequence with undefined length #=8)     # u/l, 1 Unknown Tag & Data
21   (fffe,e000) na (Item with undefined length #=9)        # u/l, 1 Item
22     (2005,0011) LO [Philips MR Imaging DD 002]             #  26, 1 PrivateCreator
23     (2005,1137) PN [PDF_CONTROL_GEN_PARS]                  #  20, 1 Unknown Tag & Data
24     (2005,1138) PN (no value available)                    #   0, 0 Unknown Tag & Data
25     (2005,1139) PN [IEEE_PDF]                              #   8, 1 Unknown Tag & Data
26     (2005,1140) PN (no value available)                    #   0, 0 Unknown Tag & Data
27     (2005,1141) PN (no value available)                    #   0, 0 Unknown Tag & Data
28     (2005,1143) SL 3103                                    #   4, 1 Unknown Tag & Data
29     (2005,1144) OW 0566\0000\013b\0000\0a4a\0000\000e\0000\0a7a\0000\0195\0000\0008...  # 3104, 1 Unknown Tag
    & Data
30     (2005,1147) CS [Y]                                     #   2, 1 Unknown Tag & Data
31   (fffe,e00d) na (ItemDelimitationItem)                  #   0, 0 ItemDelimitationItem
32   (fffe,e000) na (Item with undefined length #=9)        # u/l, 1 Item
33     (2005,0011) LO [Philips MR Imaging DD 002]             #  26, 1 PrivateCreator
34     (2005,1137) PN [PDF_CONTROL_PREP_PARS]                 #  22, 1 Unknown Tag & Data
35     (2005,1138) PN (no value available)                    #   0, 0 Unknown Tag & Data
36     (2005,1139) PN [IEEE_PDF]                              #   8, 1 Unknown Tag & Data
37     (2005,1140) PN (no value available)                    #   0, 0 Unknown Tag & Data
38     (2005,1141) PN (no value available)                    #   0, 0 Unknown Tag & Data
```

```
39     (2005,1143) SL 7934                                      #   4, 1 Unknown Tag & Data
40     (2005,1144) OW 19b6\0000\005f\0000\1b2a\0000\00f3\0000\1eee\0000\0000\0000\0008...  # 7934, 1 Unknown Tag
     & Data
41     (2005,1147) CS [Y]                                       #   2, 1 Unknown Tag & Data
42   (fffe,e00d) na (ItemDelimitationItem)                    #   0, 0 ItemDelimitationItem
43 ...
44 """
45
46 import sys
47 import gdcm
48
49 if __name__ == "__main__":
50
51 file1 = sys.argv[1]
52 file2 = sys.argv[2]
53
54 r = gdcm.Reader()
55 r.SetFileName( file1 )
56 if not r.Read():
57     sys.exit(1)
58
59   fg = gdcm.FilenameGenerator()
60   f = r.GetFile()
61   ds = f.GetDataSet()
62   tsis = gdcm.Tag(0x2005,0x1132) #
63   if ds.FindDataElement( tsis ):
64     sis = ds.GetDataElement( tsis )
65     #sqsis = sis.GetSequenceOfItems()
66     # GetValueAsSQ handle more cases
67     sqsis = sis.GetValueAsSQ()
68     if sqsis.GetNumberOfItems():
69       nitems = sqsis.GetNumberOfItems();
70       fg.SetNumberOfFilenames( nitems )
71       fg.SetPrefix( file2 )
72       if not fg.Generate():
73         print "problem"
74         sys.exit(1)
75       for i in range(0,nitems):
76         item1 = sqsis.GetItem(i+1) # Item start at 1
77         nestedds = item1.GetNestedDataSet()
78         tprcs = gdcm.Tag(0x2005,0x1144) #
79         if nestedds.FindDataElement( tprcs ):
80           prcs = nestedds.GetDataElement( tprcs )
81           bv = prcs.GetByteValue()
82           print bv
83           f = open( fg.GetFilename(i) , "w" )
84           f.write( bv.WriteBuffer() )
```

12.137 HelloActiviz.cs

```
/*=========================================================================

Program:  GDCM (Grassroots DICOM). A DICOM library

Copyright (c) 2006-2011 Mathieu Malaterre
All rights reserved.
See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.

This software is distributed WITHOUT ANY WARRANTY; without even
the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
PURPOSE.  See the above copyright notice for more information.

=========================================================================*/
using vtkgdcm;
using Kitware.VTK;
using System;
using System.Runtime.InteropServices;
/*
* This example shows how vtkgdcm can be connected to Kitware.VTK Activiz product.
* Three (3) arguments are required:
* 1.  Input DICOM file                    (SWIG)
* 2.  Temporary PNG (intermediate) file   (Activiz)
* 3.  Final DICOM file                    (SWIG)
*
* $ export MONO_PATH=/usr/lib/cli/ActiViz.NET/:/usr/lib/cli/Kitware.mummy.Runtime-1.0
* $ mono ./bin/HelloActiviz.exe ~/Creatis/gdcmData/test.acr out.png toto.dcm
*
```

```
* Footnote:
* this test originally used vtkBMPWriter / vtkBMPReader combination to store intermediate
* image file, but BMP file are 24bits by default.  Instead use PNG format which supports seems
* to be closer to what was expected in this simple test.
*/
public class HelloActiviz
{
  // Does not work with ActiViz.NET-5.4.0.455-Linux-x86_64-Personal
/*
static void ConnectSWIGToActiviz(Kitware.VTK.vtkImageExport imgin, Kitware.VTK.vtkImageImport imgout)
{
imgout.SetUpdateInformationCallback(imgin.GetUpdateInformationCallback());
imgout.SetPipelineModifiedCallback(imgin.GetPipelineModifiedCallback());
imgout.SetWholeExtentCallback(imgin.GetWholeExtentCallback());
imgout.SetSpacingCallback(imgin.GetSpacingCallback());
imgout.SetOriginCallback(imgin.GetOriginCallback());
imgout.SetScalarTypeCallback(imgin.GetScalarTypeCallback());
imgout.SetNumberOfComponentsCallback(imgin.GetNumberOfComponentsCallback());
imgout.SetPropagateUpdateExtentCallback(imgin.GetPropagateUpdateExtentCallback());
imgout.SetUpdateDataCallback(imgin.GetUpdateDataCallback());
imgout.SetDataExtentCallback(imgin.GetDataExtentCallback());
imgout.SetBufferPointerCallback(imgin.GetBufferPointerCallback());
imgout.SetCallbackUserData(imgin.GetCallbackUserData());
}
*/
  static Kitware.VTK.vtkImageData ConnectSWIGToActiviz(vtkgdcm.vtkImageData imgin)
    {
    HandleRef rawCppThis = imgin.GetCppThis();
    Kitware.VTK.vtkImageData imgout = new Kitware.VTK.vtkImageData( rawCppThis.Handle, false, false);
    return imgout;
    }
  static vtkgdcm.vtkImageData ConnectActivizToSWIG(Kitware.VTK.vtkImageData imgin)
    {
    HandleRef rawCppThis = imgin.GetCppThis();
    vtkgdcm.vtkImageData imgout = new vtkgdcm.vtkImageData( rawCppThis );
    return imgout;
    }
  public static int Main(string[] args)
    {
    string filename = args[0];
    string outfilename = args[1];
    // Step 1.  Test SWIG -> Activiz
    vtkGDCMImageReader reader = vtkGDCMImageReader.New();
    reader.SetFileName( filename );
    //reader.Update(); // DO NOT call Update to check pipeline execution
    Kitware.VTK.vtkImageData imgout = ConnectSWIGToActiviz(reader.GetOutput());
    System.Console.WriteLine( imgout.ToString() ); // not initialized as expected
    vtkPNGWriter writer = new vtkPNGWriter();
    writer.SetInput( imgout );
    writer.SetFileName( outfilename );
    writer.Write();
    // Step 2.  Test Activiz -> SWIG
    vtkPNGReader bmpreader = new vtkPNGReader();
    bmpreader.SetFileName( outfilename );
    //bmpreader.Update(); // DO NOT update to check pipeline execution
    System.Console.WriteLine( bmpreader.GetOutput().ToString() ); // not initialized as expected
    vtkgdcm.vtkImageData imgout2 = ConnectActivizToSWIG(bmpreader.GetOutput());
    System.Console.WriteLine( imgout2.ToString() ); // not initialized as expected
    Kitware.VTK.vtkMedicalImageProperties prop = new Kitware.VTK.vtkMedicalImageProperties();
    prop.SetModality( "MR" );
    string outfilename2 = args[2];
    vtkGDCMImageWriter writer2 = vtkGDCMImageWriter.New();
    writer2.SetMedicalImageProperties( prop.CastToActiviz() );
    writer2.SetFileName( outfilename2 );
    writer2.SetInput( imgout2 );
    writer2.Write();
    return 0;
    }
}
```

12.138 HelloActiviz2.cs

```
/*=========================================================================

Program:  GDCM (Grassroots DICOM). A DICOM library

Copyright (c) 2006-2011 Mathieu Malaterre
```

```
All rights reserved.
See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.

This software is distributed WITHOUT ANY WARRANTY; without even
the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
PURPOSE.  See the above copyright notice for more information.

=========================================================================*/
using Kitware.VTK;
using Kitware.VTK.GDCM;
/*
* Usage:
* $ export MONO_PATH=/usr/lib/cli/ActiViz.NET/:/usr/lib/cli/Kitware.mummy.Runtime-1.0
* $ mono ./bin/HelloActiviz2.exe gdcmData/test.acr bla.png bla2.dcm
*/
/*
* From the outside view, no-one can detect that object pass to/from
* vtkGDCMImageWriter/vtkGDCMImageReader are not Activiz object.
*
* TODO: Test Command/Observer
*/
public class HelloActiviz2
{
  public static int Main(string[] args)
    {
    string filename = args[0];
    string outfilename = args[1];
    string outfilename2 = args[2];
    vtkGDCMImageReader reader = new Kitware.VTK.GDCM.vtkGDCMImageReader();
    reader.SetFileName( filename );
    // When calling multiple times creation of C# object from the same C++ object it triggers a:
//error:  potential refcounting error:  Duplicate rawCppThis - weak reference that is still alive.  Attempting
      to add '0x00b2dc10' again.
//       Allowing new wrapped object to take over table key...
//       Original object should *not* have been destroyed while we still had it in our table without notifying
      us...
    //reader.GetOutput();
    //reader.GetOutput();
    System.Console.WriteLine( reader.ToString() ); // Test the ToString compat with Activiz
    vtkGDCMImageWriter writer = new vtkGDCMImageWriter();
    writer.SetInput( reader.GetOutput() );
    writer.SetFileName( outfilename2 );
    writer.Write();
    System.Console.WriteLine( reader.GetOutput().ToString() ); // Test the ToString compat with Activiz
    System.Console.WriteLine( writer.ToString() ); // Test the ToString compat with Activiz
    vtkPNGWriter pngwriter = new vtkPNGWriter();
    pngwriter.SetInput( reader.GetOutput() );
    pngwriter.SetFileName( outfilename );
    pngwriter.Write();
    // at that point the .Write() should have triggered an Update() on the reader:
    if( reader.GetImageFormat() == vtkgdcm.VTK_LUMINANCE ) // MONOCHROME2
      {
      System.Console.WriteLine( "Image is MONOCHROME2" ); //
      }
    vtkPNGReader bmpreader = new vtkPNGReader();
    bmpreader.SetFileName( outfilename );
    vtkMedicalImageProperties prop = new vtkMedicalImageProperties();
    prop.SetModality( "MR" );
    vtkMatrix4x4 dircos = reader.GetDirectionCosines();
    dircos.Invert();
    vtkGDCMImageWriter writer2 = new vtkGDCMImageWriter();
    writer2.SetFileName( outfilename2 );
    writer2.SetDirectionCosines( dircos );
    writer2.SetMedicalImageProperties( prop );
    writer2.SetInput( bmpreader.GetOutput() );
    writer2.Write();
    return 0;
    }
}
```

12.139 HelloActiviz3.cs

```
/*=========================================================================

Program:  GDCM (Grassroots DICOM). A DICOM library

Copyright (c) 2006-2011 Mathieu Malaterre
```

```
All rights reserved.
See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.

This software is distributed WITHOUT ANY WARRANTY; without even
the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
PURPOSE.  See the above copyright notice for more information.

=========================================================================*/
using Kitware.VTK;
using Kitware.VTK.GDCM;
/*
* $ export MONO_PATH=/usr/lib/cli/ActiViz.NET/:/usr/lib/cli/Kitware.mummy.Runtime-1.0
* $ mono ./bin/HelloActiviz3.exe ~/Creatis/gdcmData/test.acr
*/
public class HelloActiviz3
{
  public static int Main(string[] args)
    {
    string filename = args[0];
    vtkGDCMImageReader reader = vtkGDCMImageReader.New();
    vtkStringArray array = vtkStringArray.New();
    array.InsertNextValue(filename);
    reader.SetFileNames(array);
    reader.Update();
    //System.Console.Write(reader.GetOutput());
    vtkRenderWindowInteractor iren = vtkRenderWindowInteractor.New();
    vtkImageViewer2 viewer = vtkImageViewer2.New();
    viewer.SetInput(reader.GetOutput());
    viewer.SetupInteractor(iren);
    viewer.SetSize(600, 600);
    viewer.Render();
    iren.Initialize();
    iren.Start();
    return 0;
    }
}
```

12.140 HelloActiviz4.cs

```
/*=========================================================================

Program:  GDCM (Grassroots DICOM). A DICOM library

Copyright (c) 2006-2011 Mathieu Malaterre
All rights reserved.
See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.

This software is distributed WITHOUT ANY WARRANTY; without even
the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
PURPOSE.  See the above copyright notice for more information.

=========================================================================*/
using Kitware.VTK;
using Kitware.VTK.GDCM;
/*
* $ export MONO_PATH=/usr/lib/cli/ActiViz.NET/:/usr/lib/cli/Kitware.mummy.Runtime-1.0
* $ mono ./bin/HelloActiviz4.exe ~/Creatis/gdcmData/test.acr
*/
public class HelloActiviz4
{
  public static int Main(string[] args)
    {
    string filename = args[0];
    vtkGDCMImageReader reader = new vtkGDCMImageReader();
    vtkStringArray array = vtkStringArray.New();
    array.InsertNextValue(filename);
    reader.SetFileNames(array);
    reader.Update();
    //System.Console.Write(reader.GetOutput());
    vtkRenderWindowInteractor iren = vtkRenderWindowInteractor.New();
    vtkImageViewer viewer = vtkImageViewer.New();
    viewer.SetInput(reader.GetOutput());
    viewer.SetupInteractor(iren);
    viewer.SetSize(600, 600);
    viewer.Render();
    iren.Initialize();
    iren.Start();
```

```
    return 0;
    }
}
```

12.141 HelloActiviz5.cs

```
/*=========================================================================

Program:  GDCM (Grassroots DICOM). A DICOM library

Copyright (c) 2006-2011 Mathieu Malaterre
All rights reserved.
See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.

This software is distributed WITHOUT ANY WARRANTY; without even
the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
PURPOSE.  See the above copyright notice for more information.

=========================================================================*/
using Kitware.VTK;
using Kitware.VTK.GDCM;
// The command line arguments are:
// -I        => run in interactive mode; unless this is used, the program will
//              not allow interaction and exit
// -D <path> => path to the data; the data should be in <path>/Data/
/*
* $ export MONO_PATH=/usr/lib/cli/ActiViz.NET/:/usr/lib/cli/Kitware.mummy.Runtime-1.0
* $ mono ./bin/HelloActiviz5.exe -I
*/
public class HelloActiviz5
{
  public static int Main(string[] args)
    {
    vtkTesting testHelper = vtkTesting.New();
    for ( int cc = 0; cc < args.Length; cc++ )
      {
      //testHelper.AddArguments(argc,const_cast<const char **>(argv));
      //System.Console.Write( "args:  " + args[cc] + "\n" );
      testHelper.AddArgument( args[cc] );
      }
    if ( testHelper.IsFlagSpecified("-D") != 0 )
      {
      string VTK_DATA_ROOT = vtkGDCMTesting.GetVTKDataRoot();
      if( VTK_DATA_ROOT != null )
        {
        //System.Console.Write( "VTK_DATA_ROOT: " + VTK_DATA_ROOT  + "\n" );
        testHelper.SetDataRoot(VTK_DATA_ROOT);
        testHelper.AddArgument("-D");
        testHelper.AddArgument(VTK_DATA_ROOT);
        }
      }
    string dataRoot = testHelper.GetDataRoot();
    string filename = dataRoot;
    filename += "/Data/mr.001";
    vtkDirectory dir = vtkDirectory.New();
    if( dir.FileIsDirectory( dataRoot ) == 0 )
      {
      filename = vtkGDCMTesting.GetGDCMDataRoot() + "/test.acr";
      }
    //System.Console.Write( "dataRoot:  " + dataRoot + "\n" );
    System.Console.Write( "filename being used is:  " + filename + "\n" );
    vtkGDCMImageReader reader = vtkGDCMImageReader.New();
    vtkStringArray array = vtkStringArray.New();
    array.InsertNextValue(filename);
    reader.SetFileNames(array);
    reader.Update();
    System.Console.Write(reader.GetOutput());
    vtkRenderWindowInteractor iren = vtkRenderWindowInteractor.New();
    vtkRenderer ren1 = vtkRenderer.New();
    vtkRenderWindow renWin = vtkRenderWindow.New();
    renWin.AddRenderer(ren1);
    vtkImageActor actor = vtkImageActor.New();
    vtkImageMapToWindowLevelColors coronalColors = vtkImageMapToWindowLevelColors.New();
    coronalColors.SetInput(reader.GetOutput());
    actor.SetInput(coronalColors.GetOutput());
    ren1.AddActor(actor);
    iren.SetRenderWindow(renWin);
```

```
    iren.Initialize();
    renWin.Render();
    int retVal = testHelper.IsInteractiveModeSpecified();
    if( retVal != 0 )
      {
      iren.Start();
      }
    return 0;
    }
}
```

12.142 HelloVTKWorld.cs

```
/*=========================================================================

Program:  GDCM (Grassroots DICOM). A DICOM library

Copyright (c) 2006-2011 Mathieu Malaterre
All rights reserved.
See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.

This software is distributed WITHOUT ANY WARRANTY; without even
the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
PURPOSE.  See the above copyright notice for more information.

=========================================================================*/
using vtkgdcm;
/*
* This test only test the SWIG/VTK part, you do not need Activiz
*/
public class HelloVTKWorld
{
  public static int Main(string[] args)
    {
    string filename = args[0];
    vtkGDCMImageReader reader = vtkGDCMImageReader.New();
    reader.SetFileName( filename );
    reader.Update();
    vtkMedicalImageProperties prop = reader.GetMedicalImageProperties();
    System.Console.WriteLine( prop.GetPatientName() ); //
    if( reader.GetImageFormat() == vtkgdcm.vtkgdcm.VTK_LUMINANCE ) // MONOCHROME2
      {
      System.Console.WriteLine( "Image is MONOCHROME2" ); //
      }
    // Just for fun, invert the direction cosines, output should reflect that:
    vtkMatrix4x4 dircos = reader.GetDirectionCosines();
    dircos.Invert();
    string outfilename = args[1];
    vtkGDCMImageWriter writer = vtkGDCMImageWriter.New();
    writer.SetMedicalImageProperties( reader.GetMedicalImageProperties() );
    writer.SetDirectionCosines( dircos );
    writer.SetShift( reader.GetShift() );
    writer.SetScale( reader.GetScale() );
    writer.SetImageFormat( reader.GetImageFormat() );
    writer.SetFileName( outfilename );
    writer.SetInputConnection( reader.GetOutputPort() );
    writer.Write();
    return 0;
    }
}
```

12.143 HelloVTKWorld2.cs

```
/*=========================================================================

Program:  GDCM (Grassroots DICOM). A DICOM library

Copyright (c) 2006-2011 Mathieu Malaterre
All rights reserved.
See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.

This software is distributed WITHOUT ANY WARRANTY; without even
the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
```

```
PURPOSE.  See the above copyright notice for more information.

=========================================================================*/
using vtkgdcm;
/*
* This test only test the SWIG/VTK part, you do not need Activiz
*/
public class HelloVTKWorld2
{
  public static int Main(string[] args)
    {
    string VTK_DATA_ROOT = vtkGDCMTesting.GetVTKDataRoot();
    vtkVolume16Reader reader = vtkVolume16Reader.New();
    reader.SetDataDimensions(64, 64);
    reader.SetDataByteOrderToLittleEndian();
    reader.SetFilePrefix(VTK_DATA_ROOT + "/Data/headsq/quarter");
    reader.SetImageRange(1, 93);
    reader.SetDataSpacing(3.2, 3.2, 1.5);
    vtkImageCast cast = vtkImageCast.New();
    cast.SetInputConnection( reader.GetOutputPort() );
    cast.SetOutputScalarTypeToUnsignedChar();
    // By default this is creating a Multiframe Grayscale Word Secondary Capture Image Storage
    vtkGDCMImageWriter writer = vtkGDCMImageWriter.New();
    writer.SetFileName( "headsq.dcm" );
    writer.SetInputConnection( reader.GetOutputPort() );
    // cast -> Multiframe Grayscale Byte Secondary Capture Image Storage
    // writer.SetInputConnection( cast.GetOutputPort() );
    writer.SetFileDimensionality( 3 );
    writer.Write();
    return 0;
    }
}
```

12.144 MetaImageMD5Activiz.cs

```
/*=========================================================================

Program:  GDCM (Grassroots DICOM). A DICOM library

Copyright (c) 2006-2011 Mathieu Malaterre
All rights reserved.
See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.

This software is distributed WITHOUT ANY WARRANTY; without even
the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
PURPOSE.  See the above copyright notice for more information.

=========================================================================*/
using Kitware.VTK;
using Kitware.VTK.GDCM;
using gdcm;
/*
* $ export MONO_PATH=/usr/lib/cli/ActiViz.NET/:/usr/lib/cli/Kitware.mummy.Runtime-1.0
* $ mono ./bin/MetaImageMD5Activiz.exe gdcmData/012345.002.050.dcm
*/
public class MetaImageMD5Activiz
{
  public static int ProcessOneMHDMD5(string filename)
    {
    vtkGDCMImageReader reader = vtkGDCMImageReader.New();
    reader.FileLowerLeftOn();
    reader.DebugOff();
    int canread = reader.CanReadFile( filename );
    if( canread == 0 )
      {
      string refms = gdcm.Testing.GetMediaStorageFromFile(filename);
      if( gdcm.MediaStorage.IsImage( gdcm.MediaStorage.GetMSType(refms) ) )
        {
        System.Console.Write( "Problem with file:  " + filename + "\n" );
        return 1;
        }
      // not an image
      return 0;
      }
    reader.SetFileName( filename );
    reader.Update();
    // System.Console.Write(reader.GetOutput());
```

```
    vtkMetaImageWriter writer = vtkMetaImageWriter.New();
    writer.SetCompression( false );
    writer.SetInput( reader.GetOutput() );
    string subdir = "MetaImageMD5Activiz";
    string tmpdir = gdcm.Testing.GetTempDirectory( subdir );
    if( !gdcm.PosixEmulation.FileIsDirectory( tmpdir ) )
      {
      gdcm.PosixEmulation.MakeDirectory( tmpdir );
      }
    string mhdfile = gdcm.Testing.GetTempFilename( filename, subdir );
    string rawfile = mhdfile;
    mhdfile += ".mhd";
    rawfile += ".raw";
    writer.SetFileName( mhdfile );
    writer.Write();
    string digestmhd = gdcm.Testing.ComputeFileMD5( mhdfile );
    string digestraw = gdcm.Testing.ComputeFileMD5( rawfile );
    string mhdref = vtkGDCMTesting.GetMHDMD5FromFile(filename);
    string rawref = vtkGDCMTesting.GetRAWMD5FromFile(filename);
    if( mhdref != digestmhd )
      {
      System.Console.Write( "Problem with mhd file:  " + filename + "\n" );
      System.Console.Write( digestmhd );
      System.Console.Write( "\n" );
      System.Console.Write( mhdref );
      System.Console.Write( "\n" );
      return 1;
      }
    if( rawref != digestraw )
      {
      System.Console.Write( "Problem with raw file:  " + filename + "\n" );
      System.Console.Write( digestraw );
      System.Console.Write( "\n" );
      System.Console.Write( rawref );
      System.Console.Write( "\n" );
      return 1;
      }
    return 0;
    }
  public static int Main(string[] args)
    {
    if ( args.Length  == 1 )
      {
      string filename = args[0];
      return ProcessOneMHDMD5( filename );
      }
    // Loop over all gdcmData
    gdcm.Trace.DebugOff();
    gdcm.Trace.WarningOff();
    gdcm.Trace.ErrorOff();
    uint n = gdcm.Testing.GetNumberOfFileNames();
    int ret = 0;
    for( uint i = 0; i < n; ++i )
      {
      string filename = gdcm.Testing.GetFileName( i );
      ret += ProcessOneMHDMD5( filename );
      }
    return ret;
    }
}
```

12.145 RefCounting.cs

```
/*=========================================================================

Program:  GDCM (Grassroots DICOM). A DICOM library

Copyright (c) 2006-2011 Mathieu Malaterre
All rights reserved.
See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.

This software is distributed WITHOUT ANY WARRANTY; without even
the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
PURPOSE.  See the above copyright notice for more information.

=========================================================================*/
using Kitware.VTK;
```

```
using Kitware.VTK.GDCM;
/*
* this is not so much an example but simply a test to make sure cstor / dstor work as expected
* and call the ::New and ->Delete() of VTK style.
*/
public class RefCounting
{
  public static int Main(string[] args)
    {
    vtkGDCMTesting testing1 = vtkGDCMTesting.New();
    vtkGDCMTesting testing2 = new vtkGDCMTesting(); // just in case people do not read STYLE documentation
    vtkGDCMImageReader reader1 = vtkGDCMImageReader.New();
    vtkGDCMImageReader reader2 = new vtkGDCMImageReader();
    vtkGDCMImageWriter writer1 = vtkGDCMImageWriter.New();
    vtkGDCMImageWriter writer2 = new vtkGDCMImageWriter();
    using (vtkGDCMTesting testing3 = new vtkGDCMTesting())
      {
      System.Console.Write( "GetReferenceCount:  " + testing1.GetReferenceCount() + "\n");
      System.Console.Write( "GetReferenceCount:  " + testing2.GetReferenceCount() + "\n");
      System.Console.Write( "GetReferenceCount:  " + testing3.GetReferenceCount() + "\n");
      }
    using (vtkGDCMImageReader reader3 = new vtkGDCMImageReader())
      {
      System.Console.Write( "GetReferenceCount:  " + reader3.GetReferenceCount() + "\n");
      }
    using (vtkGDCMImageWriter writer3 = vtkGDCMImageWriter.New())
      {
      System.Console.Write( "GetReferenceCount:  " + writer3.GetReferenceCount() + "\n");
      }
    // C# destructor will call ->Delete on all C++ object as expected.
    return 0;
    }
}
```

12.146 Compute3DSpacing.cxx

```
/*=========================================================================

Program:  GDCM (Grassroots DICOM). A DICOM library

Copyright (c) 2006-2011 Mathieu Malaterre
All rights reserved.
See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.

This software is distributed WITHOUT ANY WARRANTY; without even
the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
PURPOSE.  See the above copyright notice for more information.

=========================================================================*/
#include "vtkGDCMImageReader2.h"
#include "vtkImageChangeInformation.h"
#include "vtkStringArray.h"
#include "gdcmIPPSorter.h"
#ifndef vtkFloatingPointType
#define vtkFloatingPointType double
#endif
/*
* Simple example to check computation of spacing within vtkGDCMImageReader2
* This is a direct implementation of:
*
*
      http://gdcm.sourceforge.net/wiki/index.php/Using_GDCM_API#Automatic_ordering_of_slices_for_vtkGDCMImageReader.SetFileNames
*
* For more advanced information on how 3D spacing is being computed see:
*
*  - http://gdcm.sourceforge.net/html/classgdcm_1_1IPPSorter.html
*
* Usage:
*
* $ Compute3DSpacing SIEMENS_MAGNETOM-12-MONO2-FileSeq0.dcm \
*     SIEMENS_MAGNETOM-12-MONO2-FileSeq1.dcm \
*     SIEMENS_MAGNETOM-12-MONO2-FileSeq2.dcm \
*     SIEMENS_MAGNETOM-12-MONO2-FileSeq3.dcm
*/
int main(int argc, char *argv[])
{
  if( argc < 2 ) return 1;
```

```
  std::vector<std::string> filenames;
  for( int  i = 1; i < argc; ++i )
    {
    filenames.push_back( argv[i] );
    }
  gdcm::IPPSorter s;
  s.SetComputeZSpacing( true );
  s.SetZSpacingTolerance( 1e-3 );
  bool b = s.Sort( filenames );
  if( !b )
    {
    std::cerr « "Failed to sort files" « std::endl;
    return 1;
    }
  std::cout « "Sorting succeeded:" « std::endl;
  //s.Print( std::cout );
  std::cout « "Found z-spacing:" « std::endl;
  std::cout « s.GetZSpacing() « std::endl;
  const double ippzspacing = s.GetZSpacing();
  const std::vector<std::string> & sorted = s.GetFilenames();
  vtkGDCMImageReader2 * reader = vtkGDCMImageReader2::New();
  vtkStringArray *files = vtkStringArray::New();
  std::vector< std::string >::const_iterator it = sorted.begin();
  for( ; it != sorted.end(); ++it)
    {
    const std::string &f = *it;
    files->InsertNextValue( f.c_str() );
    }
  reader->SetFileNames( files );
  reader->Update();
  const vtkFloatingPointType *spacing = reader->GetOutput()->GetSpacing();
  vtkImageChangeInformation *v16 = vtkImageChangeInformation::New();
#if (VTK_MAJOR_VERSION >= 6)
  v16->SetInputConnection( reader->GetOutputPort() );
#else
  v16->SetInput( reader->GetOutput() );
#endif
  v16->SetOutputSpacing( spacing[0], spacing[1], ippzspacing );
  v16->Update();
  v16->GetOutput()->Print( std::cout );
  return 0;
}
```

12.147 Convert16BitsTo8Bits.cxx

```
/*=========================================================================

Program:  GDCM (Grassroots DICOM). A DICOM library

Copyright (c) 2006-2011 Mathieu Malaterre
All rights reserved.
See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.

This software is distributed WITHOUT ANY WARRANTY; without even
the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
PURPOSE.  See the above copyright notice for more information.

=========================================================================*/
#include "vtkGDCMImageReader.h"
#include "vtkGDCMImageWriter.h"
#include "vtkImageData.h"
#include "vtkImageCast.h"
#include "gdcmTesting.h"
// The following file is 16/16/15 but the scalar range of the image is [0,192]
// it could be safely stored as 8bits instead:
// gdcmData/012345.002.050.dcm
int main(int, char *[])
{
  const char *directory = gdcm::Testing::GetDataRoot();
  if(!directory) return 1;
  std::string file = std::string(directory) + "/012345.002.050.dcm";
  std::cout « file « std::endl;
  vtkGDCMImageReader *reader = vtkGDCMImageReader::New();
  reader->SetFileName( file.c_str() );
  reader->Update();
  //reader->GetOutput()->Print( std::cout );
  vtkImageCast *cast = vtkImageCast::New();
```

```
#if (VTK_MAJOR_VERSION >= 6)
  cast->SetInputConnection( reader->GetOutputPort() );
#else
  cast->SetInput( reader->GetOutput() );
#endif
  cast->SetOutputScalarTypeToUnsignedChar();
  vtkGDCMImageWriter *writer = vtkGDCMImageWriter::New();
  writer->SetFileName( "/tmp/cast.dcm" );
#if (VTK_MAJOR_VERSION >= 6)
  writer->SetInputConnection( cast->GetOutputPort() );
#else
  writer->SetInput( cast->GetOutput() );
#endif
  writer->SetImageFormat( reader->GetImageFormat() );
  writer->SetMedicalImageProperties( reader->GetMedicalImageProperties() );
  writer->SetDirectionCosines( reader->GetDirectionCosines() );
  writer->SetShift( reader->GetShift() );
  writer->SetScale( reader->GetScale() );
  writer->Write();
  reader->Delete();
  cast->Delete();
  writer->Delete();
  return 0;
}
```

12.148 ConvertMultiFrameToSingleFrame.cxx

```
/*=========================================================================

Program:  GDCM (Grassroots DICOM). A DICOM library

Copyright (c) 2006-2011 Mathieu Malaterre
All rights reserved.
See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.

This software is distributed WITHOUT ANY WARRANTY; without even
the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
PURPOSE.  See the above copyright notice for more information.

=========================================================================*/
#include "vtkGDCMImageReader.h"
#include "vtkGDCMImageWriter.h"
#include "vtkImageData.h"
#include "vtkStringArray.h"
#include "gdcmTesting.h"
#include "gdcmFilenameGenerator.h"
int main(int argc, char *argv[])
{
  std::string filename;
  if( argc <= 1 )
    {
    const char *directory = gdcm::Testing::GetDataRoot();
    if(!directory) return 1;
    std::string file = std::string(directory) + "/US-PAL-8-10x-echo.dcm";
    filename = file;
    }
  else
    {
    filename = argv[1];
    }
  std::cout « "file:  " « filename « std::endl;
  vtkGDCMImageReader *reader = vtkGDCMImageReader::New();
  reader->SetFileName( filename.c_str() );
  reader->Update();
  //reader->GetOutput()->Print( std::cout );
  int dims[3];
  reader->GetOutput()->GetDimensions( dims );
  std::ostringstream os;
  os « "singleframe";
  os « "%04d.dcm";
  gdcm::FilenameGenerator fg;
  fg.SetPattern( os.str().c_str() );
  unsigned int nfiles = dims[2];
  fg.SetNumberOfFilenames( nfiles );
  bool b = fg.Generate();
  if( !b )
    {
```

```
    std::cerr « "FilenameGenerator::Generate() failed" « std::endl;
    return 1;
    }
  if( !fg.GetNumberOfFilenames() )
    {
    std::cerr « "FilenameGenerator::Generate() failed somehow..." « std::endl;
    return 1;
    }
  // By default write them as Secondary Capture (for portability)
  vtkGDCMImageWriter *writer = vtkGDCMImageWriter::New();
  vtkStringArray *filenames = vtkStringArray::New();
  for(unsigned int i = 0; i < fg.GetNumberOfFilenames(); ++i)
    {
    filenames->InsertNextValue( fg.GetFilename(i) );
    }
  assert( filenames->GetNumberOfValues() == (int)fg.GetNumberOfFilenames() );
  writer->SetFileNames( filenames );
  filenames->Delete();
  writer->SetFileDimensionality( 2 );
#if (VTK_MAJOR_VERSION >= 6)
  writer->SetInputConnection( reader->GetOutputPort() );
#else
  writer->SetInput( reader->GetOutput() );
#endif
  writer->SetImageFormat( reader->GetImageFormat() );
  writer->Write();
  reader->Delete();
  writer->Delete();
  return 0;
}
```

12.149 ConvertRGBToLuminance.cxx

```
/*=========================================================================

Program:  GDCM (Grassroots DICOM). A DICOM library

Copyright (c) 2006-2011 Mathieu Malaterre
All rights reserved.
See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.

This software is distributed WITHOUT ANY WARRANTY; without even
the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
PURPOSE.  See the above copyright notice for more information.

=========================================================================*/
#include "vtkGDCMImageReader.h"
#include "vtkGDCMImageWriter.h"
#include "vtkImageData.h"
#include "vtkImageLuminance.h"
#include "gdcmTesting.h"
// There is no such thing as MR Image Storage + Photometric Interpretation = RGB
// let's rewrite that into a proper single component image:
int main(int, char *[])
{
  const char *directory = gdcm::Testing::GetDataRoot();
  if(!directory) return 1;
  std::string file = std::string(directory) + "/SIEMENS-MR-RGB-16Bits.dcm";
  std::cout « file « std::endl;
  vtkGDCMImageReader *reader = vtkGDCMImageReader::New();
  reader->SetFileName( file.c_str() );
  reader->Update();
  //reader->GetOutput()->Print( std::cout );
  vtkImageLuminance *luminance = vtkImageLuminance::New();
#if (VTK_MAJOR_VERSION >= 6)
  luminance->SetInputConnection( reader->GetOutputPort() );
#else
  luminance->SetInput( reader->GetOutput() );
#endif
  vtkGDCMImageWriter *writer = vtkGDCMImageWriter::New();
  writer->SetFileName( "/tmp/bla.dcm" );
#if (VTK_MAJOR_VERSION >= 6)
  writer->SetInputConnection( luminance->GetOutputPort() );
#else
  writer->SetInput( luminance->GetOutput() );
#endif
  //writer->SetImageFormat( reader->GetImageFormat() ); // Do NOT pass image format
```

```
  writer->SetMedicalImageProperties( reader->GetMedicalImageProperties() );
  writer->SetDirectionCosines( reader->GetDirectionCosines() );
  writer->SetShift( reader->GetShift() );
  writer->SetScale( reader->GetScale() );
  writer->Write();
  // TODO:
  //vtkImageAppendComponents.h
  reader->Delete();
  luminance->Delete();
  writer->Delete();
  return 0;
}
```

12.150 ConvertSingleBitTo8Bits.cxx

```
/*=========================================================================

Program:  GDCM (Grassroots DICOM). A DICOM library

Copyright (c) 2006-2011 Mathieu Malaterre
All rights reserved.
See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.

This software is distributed WITHOUT ANY WARRANTY; without even
the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
PURPOSE.  See the above copyright notice for more information.

=========================================================================*/
#include "vtkGDCMImageReader.h"
#include "vtkGDCMImageWriter.h"
#include "vtkImageData.h"
#include "vtkImageCast.h"
#include "vtkPointData.h"
#include "vtkBitArray.h"
#include "vtkUnsignedCharArray.h"
int main(int argc, char *argv[])
{
  if( argc < 3 )
    {
    return 1;
    }
  const char *filename = argv[1];
  const char *outfilename = argv[2];
  vtkGDCMImageReader *reader = vtkGDCMImageReader::New();
  reader->SetFileName( filename );
  reader->Update();
  //reader->GetOutput()->Print( std::cout );
  vtkDataArray* array = reader->GetOutput()->GetPointData()->GetScalars();
  vtkBitArray *barray = vtkBitArray::SafeDownCast( array );
  if( !barray ) return false;
  vtkIdType nvalues = array->GetNumberOfTuples();
  vtkUnsignedCharArray *uarray = vtkUnsignedCharArray::New();
  uarray->SetNumberOfTuples( nvalues );
  for(vtkIdType i = 0; i < nvalues; ++i)
    {
    uarray->SetValue( i, (unsigned char)barray->GetValue(i) );
    }
  vtkImageData *copy = vtkImageData::New();
  // http://www.vtk.org/Wiki/VTK/VTK_6_Migration/Changes_to_Scalars_Manipulation_Functions#AllocateScalars.28.29
  copy->SetExtent( reader->GetOutput()->GetExtent() );
#if (VTK_MAJOR_VERSION >= 6)
  copy->AllocateScalars(VTK_UNSIGNED_CHAR, 3);
#else
  copy->SetScalarType( VTK_UNSIGNED_CHAR );
  copy->AllocateScalars();
#endif
  //uarray->Print( std::cout );
  //copy->GetPointData()->GetScalars()->Print( std::cout );
  copy->GetPointData()->SetScalars( uarray );
  uarray->Delete();
  vtkGDCMImageWriter *writer = vtkGDCMImageWriter::New();
  writer->SetFileName( outfilename );
  //writer->SetInput( cast->GetOutput() );
#if (VTK_MAJOR_VERSION >= 6)
  writer->SetInputData( copy );
#else
  writer->SetInput( copy );
```

```
#endif
  writer->SetImageFormat( reader->GetImageFormat() );
  writer->SetMedicalImageProperties( reader->GetMedicalImageProperties() );
  writer->SetDirectionCosines( reader->GetDirectionCosines() );
  writer->SetShift( reader->GetShift() );
  writer->SetScale( reader->GetScale() );
  writer->SetFileDimensionality( reader->GetFileDimensionality( ) );
  writer->Write();
  reader->Delete();
  copy->Delete();
  writer->Delete();
  return 0;
}
```

12.151 CreateFakePET.cxx

```
/*=========================================================================

Program:  GDCM (Grassroots DICOM). A DICOM library

Copyright (c) 2006-2011 Mathieu Malaterre
All rights reserved.
See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.

This software is distributed WITHOUT ANY WARRANTY; without even
the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
PURPOSE.  See the above copyright notice for more information.

=========================================================================*/
#include "vtkGDCMImageWriter.h"
#include "vtkImageReader.h"
#include "vtkImageCast.h"
#include "vtkImageData.h"
#include "vtkPointData.h"
#include "vtkDataArray.h"
#include "vtkMedicalImageProperties.h"
#include "vtkStringArray.h"
#include "gdcmTrace.h"
#include "gdcmReader.h"
#include "gdcmWriter.h"
#include "gdcmAttribute.h"
#include "gdcmFilenameGenerator.h"
/*
* Minimal example to create a fake RTDOSE file.  The data contains a sphere
* just for testing.
* The vtkMedicalImageProperties is not properly filled, but only contains a
* single field which is required to set the proper SOP Class
*/
int main(int, char *[])
{
  gdcm::Trace::DebugOn();
  const vtkIdType xSize = 512;
  const vtkIdType ySize = 512;
  const vtkIdType zSize = 512;
  // Create the filenames in advance to supply to the vtkGDCMImageWriter
  std::ostringstream os;
  os « "PT";
  os « "%03d.dcm";
  gdcm::FilenameGenerator fg;
  fg.SetPattern( os.str().c_str() );
  unsigned int nfiles = zSize;
  fg.SetNumberOfFilenames( nfiles );
  bool b = fg.Generate();
  if( !b )
    {
    std::cerr « "FilenameGenerator::Generate() failed" « std::endl;
    return 1;
    }
  if( !fg.GetNumberOfFilenames() )
    {
    std::cerr « "FilenameGenerator::Generate() failed somehow..." « std::endl;
    return 1;
    }
  vtkStringArray *filenames = vtkStringArray::New();
  for(unsigned int i = 0; i < fg.GetNumberOfFilenames(); ++i)
    {
    filenames->InsertNextValue( fg.GetFilename(i) );
```

```
    }
  vtkImageData *image = vtkImageData::New();
  image->SetDimensions(xSize,ySize,zSize);
  image->SetOrigin(-350.684,350.0,890.76);
  image->SetSpacing(5.4688,-5.4688,-3.27);
#if VTK_MAJOR_VERSION <= 5
  image->SetNumberOfScalarComponents(1);
  image->SetScalarTypeToDouble();
#else
  image->AllocateScalars(VTK_DOUBLE ,1);
#endif
  double pt[3];
  for( int z = 0; z < zSize; ++z )
    for( int y = 0; y < ySize; ++y )
      for( int x = 0; x < xSize; ++x )
        {
        pt[0] = x;
        pt[1] = y;
        pt[2] = z;
        pt[0] -= xSize / 2;
        pt[1] -= ySize / 2;
        pt[2] -= zSize / 2;
        pt[0] /= xSize / 2;
        pt[1] /= ySize / 2;
        pt[2] /= zSize / 2;
        const double unit = pt[0] * pt[0] + pt[1] * pt[1] + pt[2] * pt[2];
        const double inval = unit <= 1.  ?  (3 * unit + 7) :  0.; // just for fun => max == 10.
        double* pixel= static_cast<double*>(image->GetScalarPointer(x,y,z));
        pixel[0] = inval;
        }
  vtkGDCMImageWriter *writer = vtkGDCMImageWriter::New();
  writer->SetFileDimensionality( 2 );
  writer->SetFileNames(filenames);
#if (VTK_MAJOR_VERSION >= 6)
  writer->SetInputData( image );
#else
  writer->SetInput( image );
#endif
  writer->GetMedicalImageProperties()->SetSliceThickness("1.5");
  writer->GetMedicalImageProperties()->SetModality( "PT" );
  writer->SetScale( 0.0042 ); // why not
  writer->Write();
  image->Delete();
  writer->Delete();
  return 0;
}
```

12.152 CreateFakeRTDOSE.cxx

```
/*=========================================================================

Program:  GDCM (Grassroots DICOM). A DICOM library

Copyright (c) 2006-2011 Mathieu Malaterre
All rights reserved.
See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.

This software is distributed WITHOUT ANY WARRANTY; without even
the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
PURPOSE.  See the above copyright notice for more information.

=========================================================================*/
#include "vtkGDCMImageWriter.h"
#include "vtkImageReader.h"
#include "vtkImageCast.h"
#include "vtkImageData.h"
#include "vtkPointData.h"
#include "vtkDataArray.h"
#include "vtkMedicalImageProperties.h"
#include "gdcmTrace.h"
#include "gdcmReader.h"
#include "gdcmWriter.h"
#include "gdcmAttribute.h"
/*
* Minimal example to create a fake RTDOSE file.  The data contains a sphere
* just for testing.
* The vtkMedicalImageProperties is not properly filled, but only contains a
```

```
* single field which is required to set the proper SOP Class
*/
int main(int, char *[])
{
  //gdcm::Trace::DebugOn();
  const vtkIdType xSize = 512;
  const vtkIdType ySize = 512;
  const vtkIdType zSize = 512;
  vtkImageData *image = vtkImageData::New();
  image->SetDimensions(xSize,ySize,zSize);
  image->SetOrigin(-350.684,350.0,890.76);
  image->SetSpacing(5.4688,-5.4688,-3.27);
#if VTK_MAJOR_VERSION <= 5
  image->SetNumberOfScalarComponents(1);
  image->SetScalarTypeToDouble();
#else
  image->AllocateScalars(VTK_DOUBLE ,1);
#endif
  double pt[3];
  for( int z = 0; z < zSize; ++z )
    for( int y = 0; y < ySize; ++y )
      for( int x = 0; x < xSize; ++x )
        {
        pt[0] = x;
        pt[1] = y;
        pt[2] = z;
        pt[0] -= xSize / 2;
        pt[1] -= ySize / 2;
        pt[2] -= zSize / 2;
        pt[0] /= xSize / 2;
        pt[1] /= ySize / 2;
        pt[2] /= zSize / 2;
        const double unit = pt[0] * pt[0] + pt[1] * pt[1] + pt[2] * pt[2];
        const double inval = unit <= 1.  ?  (3 * unit + 7) :  0.; // just for fun => max == 10.
        double* pixel= static_cast<double*>(image->GetScalarPointer(x,y,z));
        pixel[0] = inval;
        }
  vtkGDCMImageWriter *writer = vtkGDCMImageWriter::New();
  writer->SetFileDimensionality( 3 );
  writer->SetFileName( "rtdose.dcm" );
#if (VTK_MAJOR_VERSION >= 6)
  writer->SetInputData( image );
#else
  writer->SetInput( image );
#endif
  writer->GetMedicalImageProperties()->SetSliceThickness("1.5");
  writer->GetMedicalImageProperties()->AddUserDefinedValue( "Dose Units", "GY");
  writer->GetMedicalImageProperties()->AddUserDefinedValue( "Dose Summation Type", "PLAN");
  writer->GetMedicalImageProperties()->AddUserDefinedValue( "Dose Type", "PHYSICAL");
  writer->GetMedicalImageProperties()->AddUserDefinedValue( "Frame of Reference UID",
      "1.3.12.2.1107.5.6.1.68100.30270111041215391275000000001");
  writer->GetMedicalImageProperties()->SetModality( "RTDOSE" );
  //writer->GetMedicalImageProperties()->SetModality( "PT" ); // debug
  writer->SetScale( 0.0042 ); // why not
  writer->Write();
  image->Delete();
  writer->Delete();
  // BEGIN HACK
  // In GDCM version 2.4.3 and before, the following tag was missing which caused issue with some RTDose
      software:
  // Open the DICOM file that was temporarily created.  This will allows me to used
  // GDCM to append specific tags that allows the RTDOSE to be associated with the
  // relevant CT images.
  gdcm::Reader reader2;
  reader2.SetFileName("rtdose.dcm" );
  reader2.Read();
  gdcm::File &file = reader2.GetFile();
  gdcm::DataSet &ds = file.GetDataSet();
  // Required by some software and not automagically added by GDCM in old version
  gdcm::Attribute<0x0028,0x0009> framePointer;
  framePointer.SetNumberOfValues(1);
  framePointer.SetValue( gdcm::Tag(0x3004,0x000C) );
  ds.Replace( framePointer.GetAsDataElement() );
  gdcm::Writer writer2;
  writer2.CheckFileMetaInformationOff();
  writer2.SetFileName("rtdose2.dcm");
  writer2.SetFile( file );
  writer2.Write();
  // END HACK
  return 0;
}
```

12.153 GenerateRTSTRUCT.cxx

```
/*=========================================================================

Program:  GDCM (Grassroots DICOM). A DICOM library

Copyright (c) 2006-2011 Mathieu Malaterre
All rights reserved.
See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.

This software is distributed WITHOUT ANY WARRANTY; without even
the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
PURPOSE.  See the above copyright notice for more information.

=========================================================================*/
#include "vtkGDCMPolyDataWriter.h"
#include "vtkGDCMPolyDataReader.h"
#include "vtkPolyData.h"
#include "vtkPolyDataReader.h"
#include "vtkMedicalImageProperties.h"
#include "vtkRTStructSetProperties.h"
#include "vtkStringArray.h"
#include "vtkAppendPolyData.h"
#include "vtkPolyDataWriter.h"
#include "vtkPolyDataMapper.h"
#include "vtkPolyDataMapper2D.h"
#include "vtkActor2D.h"
#include "vtkRenderWindowInteractor.h"
#include "vtkMedicalImageProperties.h"
#include "vtkRenderWindow.h"
#include "vtkRenderer.h"
#include "vtkCamera.h"
#include "vtkProperty.h"
#include "vtkProperty2D.h"
#include "vtkImageData.h"
#include <algorithm> //for std::find
#include "gdcmDirectoryHelper.h"
using namespace gdcm;
//view each organ independently of the others, to make sure that
//organ names correspond to actual segmentations.
void ShowOrgan(vtkPolyData* inData)
{
  // Now we'll look at it.
  vtkPolyDataMapper *cubeMapper = vtkPolyDataMapper::New();
#if (VTK_MAJOR_VERSION >= 6)
  cubeMapper->SetInputData( inData );
#else
  cubeMapper->SetInput( inData );
#endif
  cubeMapper->SetScalarRange(0,7);
  vtkActor *cubeActor = vtkActor::New();
  cubeActor->SetMapper(cubeMapper);
  vtkProperty * property = cubeActor->GetProperty();
  property->SetRepresentationToWireframe();
  vtkRenderer *renderer = vtkRenderer::New();
  vtkRenderWindow *renWin = vtkRenderWindow::New();
  renWin->AddRenderer(renderer);
  vtkRenderWindowInteractor *iren = vtkRenderWindowInteractor::New();
  iren->SetRenderWindow(renWin);
  renderer->AddActor(cubeActor);
  renderer->ResetCamera();
  renderer->SetBackground(1,1,1);
  renWin->SetSize(300,300);
  renWin->Render();
  iren->Start();
  cubeMapper->Delete();
  cubeActor->Delete();
  renderer->Delete();
  renWin->Delete();
  iren->Delete();
}
/*
* Full application which ...  RTSTUCT
*/
int main(int argc, char *argv[])
{
  if( argc < 2 )
    {
    std::cerr « argv[0] « " directory-with-rtstruct-and-ct-images\n";
    return 1;
```

```
    }
  std::string theDirName(argv[1]);
  Directory::FilenamesType theRTSeries =
    DirectoryHelper::GetRTStructSeriesUIDs(theDirName);
  gdcm::Directory theDir;
  theDir.Load(argv[1]);
  if (theRTSeries.empty())
    {
    std::cerr « "No RTStructs found for the test, ending." « std::endl;
    return 1;
    }
  for (size_t q = 0; q < theRTSeries.size(); q++)
    {
    Directory::FilenamesType theRTNames =
      DirectoryHelper::GetFilenamesFromSeriesUIDs(theDirName, theRTSeries[q]);
    if (theRTNames.empty()){
      std::cerr « "Unable to load RT Series " « theRTSeries[q] « ", continuing.  " « std::endl;
      continue;
    }
    vtkGDCMPolyDataReader * reader = vtkGDCMPolyDataReader::New();
    reader->SetFileName( theRTNames[0].c_str() );
    reader->Update();
    //std::cout « reader->GetMedicalImageProperties()->GetStudyDate() « std::endl;
    vtkGDCMPolyDataWriter * writer = vtkGDCMPolyDataWriter::New();
    int numMasks = reader->GetNumberOfOutputPorts() + 1;//add a blank one in
    writer->SetNumberOfInputPorts( numMasks );
    std::string thePotentialName = theDirName + "/" + "GDCMTestRTStruct." +  theRTSeries[q] + ".dcm";
    gdcm::Directory::FilenamesType theFileNames = theDir.GetFilenames();
    //keep renaming the output until we get something that doesn't overwrite what was there already
    int count = 0;
    while (std::find(theFileNames.begin(), theFileNames.end(), thePotentialName) != theFileNames.end())
      {
        char buff[255];
        sprintf(buff,"%d",count);
        thePotentialName = theDirName + "/" + "GDCMTestRTStruct." + buff + "." + theRTSeries[q] + ".dcm";
      }
    writer->SetFileName( thePotentialName.c_str());
    writer->SetMedicalImageProperties( reader->GetMedicalImageProperties() );
    //this line is cheating, we won't have the same stuff, and may not have a struct
    //to start with.
    //have to go back to the original data to reconstruct the RTStructureSetProperties
    //writer->SetRTStructSetProperties( reader->GetRTStructSetProperties() );
    //writer->Write();
    //loop through the outputs in order to write them out as if they had been created and appended
    vtkStringArray* roiNames = vtkStringArray::New();
    vtkStringArray* roiAlgorithms = vtkStringArray::New();
    vtkStringArray* roiTypes = vtkStringArray::New();
    roiNames->SetNumberOfValues(numMasks);
    roiAlgorithms->SetNumberOfValues(numMasks);
    roiTypes->SetNumberOfValues(numMasks);
    vtkAppendPolyData* append = vtkAppendPolyData::New();
    //ok, now we'll add a blank organ
    //the blank organ is to test to ensure that blank organs work; there have been crash reports
    //this code is added at the beginning to ensure that the blank organs are read
    //and preserved as individual organs.
    vtkPolyData* blank = vtkPolyData::New();
#if (VTK_MAJOR_VERSION >= 6)
    writer->SetInputData(0, blank);
#else
    writer->SetInput(0, blank);
#endif
    roiNames->InsertValue(0, "blank");
    roiAlgorithms->InsertValue(0, "blank");
    roiTypes->InsertValue(0, "ORGAN");
    //note the offsets used to place the blank rtstruct at the beginning of the newly generated RT.
    //the idea is to run the program twice; first to generate an rtstruct with a blank mask (making
    //sure that that functionality works), and then a second time to make sure that everything is
    //being read properly.    Multiple organs with the same name could cause some strangenesses.
    for (int i = 1; i < numMasks; ++i)
      {
#if (VTK_MAJOR_VERSION >= 6)
      writer->SetInputConnection(i, reader->GetOutputPort(i-1));
      append->AddInputConnection(reader->GetOutputPort(i-1));
#else
      writer->SetInput(i, reader->GetOutput(i-1));
      append->AddInput(reader->GetOutput(i-1));
#endif
      std::string theString = reader->GetRTStructSetProperties()->GetStructureSetROIName(i-1);
      roiNames->InsertValue(i, theString);
      theString = reader->GetRTStructSetProperties()->GetStructureSetROIGenerationAlgorithm(i-1);
      roiAlgorithms->InsertValue(i, theString);
```

```
      theString = reader->GetRTStructSetProperties()->GetStructureSetRTROIInterpretedType(i-1);
      roiTypes->InsertValue(i, theString);
      ShowOrgan(reader->GetOutput(i-1));
      }
    vtkRTStructSetProperties* theProperties = vtkRTStructSetProperties::New();
    writer->SetRTStructSetProperties(theProperties);
    writer->InitializeRTStructSet(theDirName,
      reader->GetRTStructSetProperties()->GetStructureSetLabel(),
      reader->GetRTStructSetProperties()->GetStructureSetName(),
      roiNames, roiAlgorithms, roiTypes);
    writer->SetRTStructSetProperties(theProperties);
    writer->Write();
    // print reader output:
    reader->Print( std::cout );
    // print first output:
    reader->GetOutput()->Print( std::cout );
    reader->Delete();
    append->Delete();
    roiNames->Delete();
    roiTypes->Delete();
    theProperties->Delete();
    roiAlgorithms->Delete();
    blank->Delete();
    writer->Delete();
  }
  return 0;
}
```

12.154 MagnifyFile.cxx

```
/*=========================================================================

Program:  GDCM (Grassroots DICOM). A DICOM library

Copyright (c) 2006-2011 Mathieu Malaterre
All rights reserved.
See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.

This software is distributed WITHOUT ANY WARRANTY; without even
the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
PURPOSE.  See the above copyright notice for more information.

=========================================================================*/
#include "vtkGDCMImageReader.h"
#include "vtkGDCMImageWriter.h"
#include "vtkImageData.h"
#include "vtkImageMagnify.h"
#include "vtkImageCast.h"
#include "gdcmTesting.h"
#include "gdcmSystem.h"
// This is a simple test to magnify an image that is known to give excellent
// compression ratio.  This will be our test for those large image
int main(int, char *[])
{
  const char *directory = gdcm::Testing::GetDataRoot();
  if(!directory) return 1;
  std::string file = std::string(directory) + "/test.acr";
  std::cout « file « std::endl;
  if( !gdcm::System::FileExists( file.c_str() ) ) return 1;
  vtkGDCMImageReader *reader = vtkGDCMImageReader::New();
  reader->SetFileName( file.c_str() );
  reader->Update();
  //reader->GetOutput()->Print( std::cout );
  vtkImageCast *cast = vtkImageCast::New();
#if (VTK_MAJOR_VERSION >= 6)
  cast->SetInputConnection( reader->GetOutputPort() );
#else
  cast->SetInput( reader->GetOutput() );
#endif
  cast->SetOutputScalarTypeToUnsignedShort();
  vtkImageMagnify *magnify = vtkImageMagnify::New();
#if (VTK_MAJOR_VERSION >= 6)
  magnify->SetInputConnection( cast->GetOutputPort() );
#else
  magnify->SetInput( cast->GetOutput() );
#endif
  magnify->SetInterpolate( 1 );
```

```
  magnify->SetInterpolate( 0 );
  int factor = 100;
  magnify->SetMagnificationFactors (factor, factor, 1);
  vtkGDCMImageWriter *writer = vtkGDCMImageWriter::New();
  writer->SetFileName( "/tmp/bla.dcm" );
#if (VTK_MAJOR_VERSION >= 6)
  writer->SetInputConnection( magnify->GetOutputPort() );
#else
  writer->SetInput( magnify->GetOutput() );
#endif
  writer->SetImageFormat( reader->GetImageFormat() );
  writer->SetMedicalImageProperties( reader->GetMedicalImageProperties() );
  writer->SetDirectionCosines( reader->GetDirectionCosines() );
  writer->SetShift( reader->GetShift() );
  writer->SetScale( reader->GetScale() );
  writer->Write();
  // TODO:
  //vtkImageAppendComponents.h
  reader->Delete();
  magnify->Delete();
  writer->Delete();
  return 0;
}
```

12.155 gdcmorthoplanes.cxx

```
/*=========================================================================

Program:  GDCM (Grassroots DICOM). A DICOM library

Copyright (c) 2006-2011 Mathieu Malaterre
All rights reserved.
See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.

This software is distributed WITHOUT ANY WARRANTY; without even
the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
PURPOSE.  See the above copyright notice for more information.

=========================================================================*/
#include "vtkActor.h"
#include "vtkCamera.h"
#include "vtkMatrix4x4.h"
#include "vtkTransform.h"
#include "vtkAssembly.h"
#include "vtkCellPicker.h"
#include "vtkCommand.h"
#include "vtkImageActor.h"
#include "vtkImageMapToColors.h"
#include "vtkImageOrthoPlanes.h"
#include "vtkImagePlaneWidget.h"
#include "vtkImageReader.h"
#include "vtkInteractorEventRecorder.h"
#include "vtkLookupTable.h"
#include "vtkOutlineFilter.h"
#include "vtkPolyDataMapper.h"
#include "vtkProperty.h"
#include "vtkRenderWindow.h"
#include "vtkRenderWindowInteractor.h"
#include "vtkRenderer.h"
#include "vtkVolume16Reader.h"
#include "vtkImageData.h"
#include "vtkImageChangeInformation.h"
#include "vtkOrientationMarkerWidget.h"
#include "vtkAnnotatedCubeActor.h"
#include "vtkAxesActor.h"
#include "vtkCaptionActor2D.h"
#include "vtkTextProperty.h"
#include "vtkPropAssembly.h"
#include "vtkGDCMImageReader.h"
#include "vtkGDCMImageWriter.h"
#include "vtkStringArray.h"
#include "gdcmSystem.h"
#include "gdcmDirectory.h"
#include "gdcmIPPSorter.h"
#ifndef vtkFloatingPointType
#define vtkFloatingPointType double
#endif
```

```
//----------------------------------------------------------------------------
class vtkOrthoPlanesCallback :  public vtkCommand
{
public:
  static vtkOrthoPlanesCallback *New()
  { return new vtkOrthoPlanesCallback; }
  void Execute( vtkObject *caller, unsigned long vtkNotUsed( event ),
                void *callData )
  {
    vtkImagePlaneWidget* self =
      reinterpret_cast< vtkImagePlaneWidget* >( caller );
    if(!self) return;
    double* wl = static_cast<double*>( callData );
    if ( self == this->WidgetX )
      {
      this->WidgetY->SetWindowLevel(wl[0],wl[1],1);
      this->WidgetZ->SetWindowLevel(wl[0],wl[1],1);
      }
    else if( self == this->WidgetY )
      {
      this->WidgetX->SetWindowLevel(wl[0],wl[1],1);
      this->WidgetZ->SetWindowLevel(wl[0],wl[1],1);
      }
    else if (self == this->WidgetZ)
      {
      this->WidgetX->SetWindowLevel(wl[0],wl[1],1);
      this->WidgetY->SetWindowLevel(wl[0],wl[1],1);
      }
  }
  vtkOrthoPlanesCallback():WidgetX( 0 ), WidgetY( 0 ), WidgetZ ( 0 ) {}
  vtkImagePlaneWidget* WidgetX;
  vtkImagePlaneWidget* WidgetY;
  vtkImagePlaneWidget* WidgetZ;
};
int main( int argc, char *argv[] )
{
  //char* fname = vtkTestUtilities::ExpandDataFileName(argc, argv, "Data/headsq/quarter");
  //vtkVolume16Reader* v16 =  vtkVolume16Reader::New();
  //  v16->SetDataDimensions( 64, 64);
  //  v16->SetDataByteOrderToLittleEndian();
  //  v16->SetImageRange( 1, 93);
  //  v16->SetDataSpacing( 3.2, 3.2, 1.5);
  //  v16->SetFilePrefix( fname );
  //  v16->SetDataMask( 0x7fff);
  //  v16->Update();
  std::vector<std::string> filenames;
  if( argc < 2 )
    {
    std::cerr « argv[0] « " filename1.dcm [filename2.dcm ...]\n";
    return 1;
    }
  else
    {
    // Is it a single directory ?  If so loop over all files contained in it:
    const char *filename = argv[1];
    if( argc == 2 && gdcm::System::FileIsDirectory( filename ) )
      {
      std::cout « "Loading directory:  " « filename « std::endl;
      bool recursive = false;
      gdcm::Directory d;
      d.Load(filename, recursive);
      gdcm::Directory::FilenamesType const &files = d.GetFilenames();
      for( gdcm::Directory::FilenamesType::const_iterator it = files.begin(); it != files.end(); ++it )
        {
        filenames.push_back( it->c_str() );
        }
      }
    else // list of files passed directly on the cmd line:
        // discard non-existing or directory
      {
      for(int i=1; i < argc; ++i)
        {
        filename = argv[i];
        if( gdcm::System::FileExists( filename ) )
          {
          if( gdcm::System::FileIsDirectory( filename ) )
            {
            std::cerr « "Discarding directory:  " « filename « std::endl;
            }
          else
            {
```

```
            filenames.push_back( filename );
            }
          }
        else
          {
          std::cerr « "Discarding non existing file:  " « filename « std::endl;
          }
        }
      }
    //names->Print( std::cout );
    }
  vtkGDCMImageReader * reader = vtkGDCMImageReader::New();
  double ippzspacing;
  if( filenames.size() > 1 )
    {
    //gdcm::Trace::DebugOn();
    //gdcm::Trace::WarningOn();
    gdcm::IPPSorter s;
    s.SetComputeZSpacing( true );
    s.SetZSpacingTolerance( 1e-3 );
    bool b = s.Sort( filenames );
    if( !b )
      {
      std::cerr « "Failed to sort files" « std::endl;
      return 1;
      }
    std::cout « "Sorting succeeded:" « std::endl;
    s.Print( std::cout );
    std::cout « "Found z-spacing:" « std::endl;
    std::cout « s.GetZSpacing() « std::endl;
    ippzspacing = s.GetZSpacing();
    const std::vector<std::string> & sorted = s.GetFilenames();
    vtkStringArray *files = vtkStringArray::New();
    std::vector< std::string >::const_iterator it = sorted.begin();
    for( ; it != sorted.end(); ++it)
      {
      const std::string &f = *it;
      files->InsertNextValue( f.c_str() );
      }
    reader->SetFileNames( files );
    //reader->SetFileLowerLeft( 1 );
    reader->Update(); // important
    files->Delete();
    }
  else
    {
    reader->SetFileName( argv[1] );
    reader->Update(); // important
    ippzspacing = reader->GetOutput()->GetSpacing()[2];
    ippzspacing = 4;
    }
  //reader->GetOutput()->Print( std::cout );
  //vtkFloatingPointType range[2];
  //reader->GetOutput()->GetScalarRange(range);
  //std::cout « "Range:  " « range[0] « " " « range[1] « std::endl;
  const vtkFloatingPointType *spacing = reader->GetOutput()->GetSpacing();
  vtkImageChangeInformation *v16 = vtkImageChangeInformation::New();
#if (VTK_MAJOR_VERSION >= 6)
  v16->SetInputConnection( reader->GetOutputPort() );
#else
  v16->SetInput( reader->GetOutput() );
#endif
  v16->SetOutputSpacing( spacing[0], spacing[1], ippzspacing );
  v16->Update();
#if 0
    vtkGDCMImageWriter *writer = vtkGDCMImageWriter::New();
    writer->SetInput( v16->GetOutput() );
    writer->SetFileLowerLeft( reader->GetFileLowerLeft() );
    writer->SetDirectionCosines( reader->GetDirectionCosines() );
    writer->SetImageFormat( reader->GetImageFormat() );
    writer->SetFileDimensionality( 3); //reader->GetFileDimensionality() );
    writer->SetMedicalImageProperties( reader->GetMedicalImageProperties() );
    writer->SetShift( reader->GetShift() );
    writer->SetScale( reader->GetScale() );
    writer->SetFileName( "out.dcm" );
    writer->Write();
#endif
  vtkOutlineFilter* outline = vtkOutlineFilter::New();
    outline->SetInputConnection(v16->GetOutputPort());
  vtkPolyDataMapper* outlineMapper = vtkPolyDataMapper::New();
    outlineMapper->SetInputConnection(outline->GetOutputPort());
```

```
  vtkActor* outlineActor =  vtkActor::New();
    outlineActor->SetMapper( outlineMapper);
  vtkRenderer* ren1 = vtkRenderer::New();
  vtkRenderer* ren2 = vtkRenderer::New();
  vtkRenderWindow* renWin = vtkRenderWindow::New();
    renWin->AddRenderer(ren2);
    renWin->AddRenderer(ren1);
  vtkRenderWindowInteractor* iren = vtkRenderWindowInteractor::New();
    iren->SetRenderWindow(renWin);
  vtkCellPicker* picker = vtkCellPicker::New();
    picker->SetTolerance(0.005);
  vtkProperty* ipwProp = vtkProperty::New();
   //assign default props to the ipw's texture plane actor
  vtkImagePlaneWidget* planeWidgetX = vtkImagePlaneWidget::New();
    planeWidgetX->SetInteractor( iren);
    planeWidgetX->SetKeyPressActivationValue('x');
    planeWidgetX->SetPicker(picker);
    planeWidgetX->RestrictPlaneToVolumeOn();
    planeWidgetX->GetPlaneProperty()->SetColor(1,0,0);
    planeWidgetX->SetTexturePlaneProperty(ipwProp);
    planeWidgetX->TextureInterpolateOff();
    planeWidgetX->SetResliceInterpolateToNearestNeighbour();
#if (VTK_MAJOR_VERSION >= 6)
    planeWidgetX->SetInputConnection(v16->GetOutputPort());
#else
    planeWidgetX->SetInput(v16->GetOutput());
#endif
    planeWidgetX->SetPlaneOrientationToXAxes();
    //planeWidgetX->SetSliceIndex(32);
    planeWidgetX->DisplayTextOn();
    planeWidgetX->On();
    planeWidgetX->InteractionOff();
    planeWidgetX->InteractionOn();
  vtkImagePlaneWidget* planeWidgetY = vtkImagePlaneWidget::New();
    planeWidgetY->SetInteractor( iren);
    planeWidgetY->SetKeyPressActivationValue('y');
    planeWidgetY->SetPicker(picker);
    planeWidgetY->GetPlaneProperty()->SetColor(1,1,0);
    planeWidgetY->SetTexturePlaneProperty(ipwProp);
    planeWidgetY->TextureInterpolateOn();
    planeWidgetY->SetResliceInterpolateToLinear();
#if (VTK_MAJOR_VERSION >= 6)
    planeWidgetY->SetInputConnection(v16->GetOutputPort());
#else
    planeWidgetY->SetInput(v16->GetOutput());
#endif
    planeWidgetY->SetPlaneOrientationToYAxes();
    //planeWidgetY->SetSlicePosition(102.4);
    planeWidgetY->SetLookupTable( planeWidgetX->GetLookupTable());
    planeWidgetY->DisplayTextOn();
    planeWidgetY->UpdatePlacement();
    planeWidgetY->On();
  vtkImagePlaneWidget* planeWidgetZ = vtkImagePlaneWidget::New();
    planeWidgetZ->SetInteractor( iren);
    planeWidgetZ->SetKeyPressActivationValue('z');
    planeWidgetZ->SetPicker(picker);
    planeWidgetZ->GetPlaneProperty()->SetColor(0,0,1);
    planeWidgetZ->SetTexturePlaneProperty(ipwProp);
    planeWidgetZ->TextureInterpolateOn();
    planeWidgetZ->SetResliceInterpolateToCubic();
#if (VTK_MAJOR_VERSION >= 6)
    planeWidgetZ->SetInputConnection(v16->GetOutputPort());
#else
    planeWidgetZ->SetInput(v16->GetOutput());
#endif
    planeWidgetZ->SetPlaneOrientationToZAxes();
    //planeWidgetZ->SetSliceIndex(25);
    planeWidgetZ->SetLookupTable( planeWidgetX->GetLookupTable());
    planeWidgetZ->DisplayTextOn();
    planeWidgetZ->On();
  vtkImageOrthoPlanes *orthoPlanes = vtkImageOrthoPlanes::New();
    orthoPlanes->SetPlane(0, planeWidgetX);
    orthoPlanes->SetPlane(1, planeWidgetY);
    orthoPlanes->SetPlane(2, planeWidgetZ);
    orthoPlanes->ResetPlanes();
   vtkOrthoPlanesCallback* cbk = vtkOrthoPlanesCallback::New();
   cbk->WidgetX = planeWidgetX;
   cbk->WidgetY = planeWidgetY;
   cbk->WidgetZ = planeWidgetZ;
   planeWidgetX->AddObserver( vtkCommand::EndWindowLevelEvent, cbk );
   planeWidgetY->AddObserver( vtkCommand::EndWindowLevelEvent, cbk );
```

```
   planeWidgetZ->AddObserver( vtkCommand::EndWindowLevelEvent, cbk );
   cbk->Delete();
  double wl[2];
  planeWidgetZ->GetWindowLevel(wl);
  // Add a 2D image to test the GetReslice method
  //
  vtkImageMapToColors* colorMap = vtkImageMapToColors::New();
    colorMap->PassAlphaToOutputOff();
    colorMap->SetActiveComponent(0);
    colorMap->SetOutputFormatToLuminance();
#if (VTK_MAJOR_VERSION >= 6)
    colorMap->SetInputData(planeWidgetZ->GetResliceOutput());
#else
    colorMap->SetInput(planeWidgetZ->GetResliceOutput());
#endif
    colorMap->SetLookupTable(planeWidgetX->GetLookupTable());
  vtkImageActor* imageActor = vtkImageActor::New();
    imageActor->PickableOff();
#if (VTK_MAJOR_VERSION >= 6)
    imageActor->SetInputData(colorMap->GetOutput());
#else
    imageActor->SetInput(colorMap->GetOutput());
#endif
  // Add the actors
  //
  ren1->AddActor( outlineActor);
  ren2->AddActor( imageActor);
  ren1->SetBackground( 0.1, 0.1, 0.2);
  ren2->SetBackground( 0.2, 0.1, 0.2);
  renWin->SetSize( 600, 350);
  ren1->SetViewport(0,0,0.58333,1);
  ren2->SetViewport(0.58333,0,1,1);
  // Set the actors' postions
  //
  renWin->Render();
  //iren->SetEventPosition( 175,175);
  //iren->SetKeyCode('r');
  //iren->InvokeEvent(vtkCommand::CharEvent,NULL);
  //iren->SetEventPosition( 475,175);
  //iren->SetKeyCode('r');
  //iren->InvokeEvent(vtkCommand::CharEvent,NULL);
  //renWin->Render();
  //ren1->GetActiveCamera()->Elevation(110);
  //ren1->GetActiveCamera()->SetViewUp(0, 0, -1);
  //ren1->GetActiveCamera()->Azimuth(45);
  //ren1->GetActiveCamera()->Dolly(1.15);
  ren1->ResetCameraClippingRange();
  vtkAnnotatedCubeActor* cube = vtkAnnotatedCubeActor::New();
  cube->SetXPlusFaceText ( "R" );
  cube->SetXMinusFaceText( "L" );
  cube->SetYPlusFaceText ( "A" );
  cube->SetYMinusFaceText( "P" );
  cube->SetZPlusFaceText ( "H" );
  cube->SetZMinusFaceText( "F" );
  cube->SetFaceTextScale( 0.666667 );
  vtkAxesActor* axes2 = vtkAxesActor::New();
  vtkMatrix4x4 *invert = vtkMatrix4x4::New();
  invert->DeepCopy( reader->GetDirectionCosines() );
  invert->Invert();
  // simulate a left-handed coordinate system
  //
  vtkTransform *transform = vtkTransform::New();
  transform->Identity();
  //transform->RotateY(90);
  transform->Concatenate(invert);
  axes2->SetShaftTypeToCylinder();
  axes2->SetUserTransform( transform );
  cube->GetAssembly()->SetUserTransform( transform );
  axes2->SetTotalLength( 1.5, 1.5, 1.5 );
  axes2->SetCylinderRadius( 0.500 * axes2->GetCylinderRadius() );
  axes2->SetConeRadius    ( 1.025 * axes2->GetConeRadius() );
  axes2->SetSphereRadius  ( 1.500 * axes2->GetSphereRadius() );
  vtkTextProperty* tprop = axes2->GetXAxisCaptionActor2D()->
    GetCaptionTextProperty();
  tprop->ItalicOn();
  tprop->ShadowOn();
  tprop->SetFontFamilyToTimes();
  axes2->GetYAxisCaptionActor2D()->GetCaptionTextProperty()->ShallowCopy( tprop );
  axes2->GetZAxisCaptionActor2D()->GetCaptionTextProperty()->ShallowCopy( tprop );
  vtkPropAssembly* assembly = vtkPropAssembly::New();
  assembly->AddPart( axes2 );
```

```
  assembly->AddPart( cube );
  vtkOrientationMarkerWidget* widget = vtkOrientationMarkerWidget::New();
  widget->SetOutlineColor( 0.9300, 0.5700, 0.1300 );
  widget->SetOrientationMarker( assembly );
  widget->SetInteractor( iren );
  widget->SetViewport( 0.0, 0.0, 0.4, 0.4 );
  widget->SetEnabled( 1 );
  widget->InteractiveOff();
  widget->InteractiveOn();
  // Playback recorded events
  //
  //vtkInteractorEventRecorder *recorder = vtkInteractorEventRecorder::New();
  //recorder->SetInteractor(iren);
  //recorder->ReadFromInputStringOn();
  //recorder->SetInputString(IOPeventLog);
  // Interact with data
  // Render the image
  //
  iren->Initialize();
  renWin->Render();
  // Test SetKeyPressActivationValue for one of the widgets
  //
  //iren->SetKeyCode('z');
  //iren->InvokeEvent(vtkCommand::CharEvent,NULL);
  //iren->SetKeyCode('z');
  //iren->InvokeEvent(vtkCommand::CharEvent,NULL);
  //int retVal = vtkRegressionTestImage( renWin );
  //
  //if ( retVal == vtkRegressionTester::DO_INTERACTOR)
    {
    iren->Start();
    }
  // Clean up
  //
  //recorder->Off();
  //recorder->Delete();
  ipwProp->Delete();
  orthoPlanes->Delete();
  planeWidgetX->Delete();
  planeWidgetY->Delete();
  planeWidgetZ->Delete();
  colorMap->Delete();
  imageActor->Delete();
  picker->Delete();
  outlineActor->Delete();
  outlineMapper->Delete();
  outline->Delete();
  iren->Delete();
  renWin->Delete();
  ren1->Delete();
  ren2->Delete();
  v16->Delete();
  reader->Delete();
  return 0;
}
```

12.156 gdcmreslice.cxx

```
/*=========================================================================

Program:  GDCM (Grassroots DICOM). A DICOM library

Copyright (c) 2006-2011 Mathieu Malaterre
All rights reserved.
See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.

This software is distributed WITHOUT ANY WARRANTY; without even
the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
PURPOSE.  See the above copyright notice for more information.

=========================================================================*/
#include "vtkGDCMImageReader.h"
#include "vtkRenderer.h"
#include "vtkAssembly.h"
#include "vtkImageFlip.h"
#include "vtkImageReslice.h"
#include "vtkRenderWindow.h"
```

```
#include "vtkAnnotatedCubeActor.h"
#include "vtkTransform.h"
#include "vtkAxesActor.h"
#include "vtkTextProperty.h"
#include "vtkCaptionActor2D.h"
#include "vtkPropAssembly.h"
#include "vtkOrientationMarkerWidget.h"
#include "vtkRenderWindowInteractor.h"
#include "vtkPolyDataMapper.h"
#include "vtkActor.h"
#include "vtkImageData.h"
#include "vtkLookupTable.h"
#include "vtkTexture.h"
#include "vtkPlaneSource.h"
int main( int argc, char *argv[] )
{
  if( argc < 2 ) return 1;
  vtkGDCMImageReader *reader = vtkGDCMImageReader::New();
  reader->SetFileName( argv[1] );
  //reader->FileLowerLeftOn();
  reader->Update();
  vtkImageFlip *flip = vtkImageFlip::New();
#if (VTK_MAJOR_VERSION >= 6)
  flip->SetInputConnection(reader->GetOutputPort());
#else
  flip->SetInput(reader->GetOutput());
#endif
  flip->SetFilteredAxis(0);
  flip->Update();
  vtkImageReslice *reslice = vtkImageReslice::New();
  //reslice->SetInput(reader->GetOutput());
#if (VTK_MAJOR_VERSION >= 6)
  reslice->SetInputConnection(flip->GetOutputPort());
#else
  reslice->SetInput(flip->GetOutput());
#endif
  //reslice->SetResliceAxesDirectionCosines()
  reader->GetDirectionCosines()->Print(std::cout);
  vtkMatrix4x4 *invert = vtkMatrix4x4::New();
  invert->DeepCopy( reader->GetDirectionCosines() );
  invert->Invert();
  //reslice->SetResliceAxes( reader->GetDirectionCosines() );
  reslice->SetResliceAxes( invert );
  reslice->Update();
  vtkImageData* ima = reslice->GetOutput();
  vtkLookupTable* table = vtkLookupTable::New();
  table->SetNumberOfColors(1000);
  table->SetTableRange(0,1000);
  table->SetSaturationRange(0,0);
  table->SetHueRange(0,1);
  table->SetValueRange(0,1);
  table->SetAlphaRange(1,1);
  table->Build();
  // Texture
  vtkTexture* texture = vtkTexture::New();
#if (VTK_MAJOR_VERSION >= 6)
  texture->SetInputData(ima);
#else
  texture->SetInput(ima);
#endif
  texture->InterpolateOn();
  texture->SetLookupTable(table);
  // PlaneSource
  vtkPlaneSource* plane = vtkPlaneSource::New();
  // PolyDataMapper
  vtkPolyDataMapper *planeMapper = vtkPolyDataMapper::New();
#if (VTK_MAJOR_VERSION >= 6)
  planeMapper->SetInputConnection(plane->GetOutputPort());
#else
  planeMapper->SetInput(plane->GetOutput());
#endif
  // Actor
  vtkActor* planeActor = vtkActor::New();
  planeActor->SetTexture(texture);
  planeActor->SetMapper(planeMapper);
  planeActor->PickableOn();
  // Final rendering with simple interactor:
  vtkRenderer        *ren = vtkRenderer::New();
  vtkRenderWindow *renwin = vtkRenderWindow::New();
  renwin->AddRenderer(ren);
  vtkRenderWindowInteractor *iren = vtkRenderWindowInteractor::New();
```

```
  iren->SetRenderWindow(renwin);
  ren->AddActor(planeActor);
  ren->SetBackground(0,0,0.5);
  // DICOM is RAH:
  vtkAnnotatedCubeActor* cube = vtkAnnotatedCubeActor::New();
  cube->SetXPlusFaceText ( "R" );
  cube->SetXMinusFaceText( "L" );
  cube->SetYPlusFaceText ( "A" );
  cube->SetYMinusFaceText( "P" );
  cube->SetZPlusFaceText ( "H" );
  cube->SetZMinusFaceText( "F" );
  vtkAxesActor* axes2 = vtkAxesActor::New();
  vtkTransform *transform = vtkTransform::New();
  transform->Identity();
  //reader->GetDirectionCosines()->Print(std::cout);
  transform->Concatenate(invert);
  //axes2->SetShaftTypeToCylinder();
  axes2->SetUserTransform( transform );
  cube->GetAssembly()->SetUserTransform( transform ); // cant get it to work
  vtkPropAssembly* assembly = vtkPropAssembly::New();
  assembly->AddPart( axes2 );
  assembly->AddPart( cube );
  vtkOrientationMarkerWidget* widget = vtkOrientationMarkerWidget::New();
  widget->SetOrientationMarker( assembly );
  widget->SetInteractor( iren );
  widget->SetEnabled( 1 );
  widget->InteractiveOff();
  widget->InteractiveOn();
  renwin->Render();
  iren->Start();
  // Clean up:
  reader->Delete();
  table->Delete();
  texture->Delete();
  plane->Delete();
  planeMapper->Delete();
  planeActor->Delete();
  ren->Delete();
  renwin->Delete();
  iren->Delete();
  return 0;
}
```

12.157 gdcmrtionplan.cxx

```
/*=========================================================================

Program:  GDCM (Grassroots DICOM). A DICOM library

Copyright (c) 2006-2011 Mathieu Malaterre
All rights reserved.
See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.

This software is distributed WITHOUT ANY WARRANTY; without even
the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
PURPOSE.  See the above copyright notice for more information.

=========================================================================*/
#include "vtkImageData.h"
#include "vtkPointData.h"
#include "vtkPolyData.h"
#include "vtkProperty.h"
#include "vtkPolyDataMapper.h"
#include "vtkActor.h"
#include "vtkRenderer.h"
#include "vtkCellArray.h"
#include "vtkPoints.h"
#include "vtkDoubleArray.h"
#include <vtkXMLImageDataWriter.h>
#include <vtkXMLPolyDataWriter.h>
#include <vtkRenderWindowInteractor.h>
#include <vtkImageColorViewer.h>
#include "gdcmReader.h"
#include "gdcmAttribute.h"
/*
This example is just for fun.  We found a RT Ion Plan Storage and simply extracted the viz stuff for VTK
```

```
RTIonPlanStorage, // 1.2.840.10008.5.1.4.1.1.481.8
*/
int main(int argc, char *argv[])
{
  if( argc < 3 )
    {
    std::cerr « argv[0] « " filename.dcm outfile.vti\n";
    return 1;
    }
  const char * filename = argv[1];
  const char * outfilename = argv[2];
  const char * outfilename2 = argv[3];
  gdcm::Reader reader;
  reader.SetFileName( filename );
  if( !reader.Read() )
    {
    return 1;
    }
  gdcm::MediaStorage ms;
  ms.SetFromFile( reader.GetFile() );
  if( ms != gdcm::MediaStorage::RTIonPlanStorage )
    {
    return 1;
    }
/*
(300a,03a2) SQ                                                  # u/l,1 Ion Beam Sequence
(fffe,e000) na (Item with undefined length)
(0008,1040) LO [Test]                                         # 4,1 Institutional Department Name
(300a,00b2) SH (no value)                                     # 0,1 Treatment Machine Name
(300a,00b3) CS [MU]                                           # 2,1 Primary Dosimeter Unit
(300a,00c0) IS [1 ]                                           # 2,1 Beam Number
(300a,00c2) LO [1 ]                                           # 2,1 Beam Name
(300a,00c4) CS [STATIC]                                       # 6,1 Beam Type
(300a,00c6) CS [PROTON]                                       # 6,1 Radiation Type
(300a,00ce) CS [TREATMENT ]                                   # 10,1 Treatment Delivery Type
(300a,00d0) IS [0 ]                                           # 2,1 Number of Wedges
(300a,00e0) IS [1 ]                                           # 2,1 Number of Compensators
(300a,00ed) IS [0 ]                                           # 2,1 Number of Boli
(300a,00f0) IS [1 ]                                           # 2,1 Number of Blocks
(300a,0110) IS [2 ]                                           # 2,1 Number of Control Points
(300a,02ea) SQ                                                # u/l,1 Ion Range Compensator Sequence
(fffe,e000) na (Item with undefined length)
(300a,00e1) SH [lucite]                                     # 6,1 Material ID
(300a,00e4) IS [1 ]                                         # 2,1 Compensator Number
(300a,00e5) SH [75hdhe5 ]                                   # 8,1 Compensator ID
(300a,00e7) IS [35]                                         # 2,1 Compensator Rows
(300a,00e8) IS [37]                                         # 2,1 Compensator Columns
(300a,00e9) DS [3.679991\4.249288 ]                         # 18,2 Compensator Pixel Spacing
(300a,00ea) DS [-76.00\62.50]                               # 12,2 Compensator Position
(300a,00ec) DS
      [52.13\52.13\52.13\53.18\54.04\54.04\47.11\40.06\40.06\38.79\34.87\33.28\33.28\33.28\33.28\35.43\35.43\34.54\34.54\34.71\36.
      # 7618,1-n Compensator Thickness Data
(300a,02e0) CS [ABSENT]                                     # 6,1 Compensator Divergence
(300a,02e1) CS [SOURCE_SIDE ]                               # 12,1 Compensator Mounting Position
(300a,02e4) FL 39.2                                         # 4,1 Isocenter to Compensator Tray Distance
(300a,02e5) FL 2.12                                         # 4,1 Compensator Column Offset
(300a,02e8) FL 4.76                                         # 4,1 Compensator Milling Tool Diameter
(fffe,e00d)
*/
  const gdcm::DataSet& ds = reader.GetFile().GetDataSet();
  gdcm::Tag tbeamsq(0x300a,0x03a2);
  if( !ds.FindDataElement( tbeamsq ) )
    {
    return 1;
    }
  const gdcm::DataElement &beamsq = ds.GetDataElement( tbeamsq );
  //std::cout « beamsq « std::endl;
  gdcm::SmartPointer<gdcm::SequenceOfItems> sqi = beamsq.GetValueAsSQ();
  if( !sqi || !sqi->GetNumberOfItems() )
    {
    return 1;
    }
  //for(unsigned int pd = 0; pd < sqi->GetNumberOfItems(); ++pd)
  //  {
    //const gdcm::Item & item = sqi->GetItem(1); // Item start at #1
    const gdcm::Item & item = sqi->GetItem(1); // Item start at #1
    const gdcm::DataSet& nestedds = item.GetNestedDataSet();
    //std::cout « nestedds « std::endl;
    gdcm::Tag tcompensatorsq(0x300a,0x02ea);
    if( !nestedds.FindDataElement( tcompensatorsq ) )
      {
```

```
      return 1;
      }
    const gdcm::DataElement &compensatorsq = nestedds.GetDataElement( tcompensatorsq );
    //std::cout « compensatorsq « std::endl;
    gdcm::SmartPointer<gdcm::SequenceOfItems> ssqi = compensatorsq.GetValueAsSQ();
    const gdcm::Item & item2 = ssqi->GetItem(1); // Item start at #1
    const gdcm::DataSet& nestedds2 = item2.GetNestedDataSet();
    //std::cout « nestedds2 « std::endl;
    gdcm::Tag tcompensatorthicknessdata(0x300a,0x00ec);
    if( !nestedds2.FindDataElement( tcompensatorthicknessdata ) )
      {
      return 1;
      }
    const gdcm::DataElement &compensatorthicknessdata = nestedds2.GetDataElement( tcompensatorthicknessdata );
    //  std::cout « compensatorthicknessdata « std::endl;
    gdcm::Attribute<0x300a,0x00ec> at;
    at.SetFromDataElement( compensatorthicknessdata );
    const double* pts = at.GetValues();
    //        (300a,00e7) IS [35]                                     # 2,1 Compensator Rows
    gdcm::Attribute<0x300a,0x00e7> at1;
    const gdcm::DataElement &compensatorrows = nestedds2.GetDataElement( at1.GetTag() );
    at1.SetFromDataElement( compensatorrows );
    std::cout « at1.GetValue() « std::endl;
    //        (300a,00e8) IS [37]                                     # 2,1 Compensator Columns
    gdcm::Attribute<0x300a,0x00e8> at2;
    const gdcm::DataElement &compensatorcols = nestedds2.GetDataElement( at2.GetTag() );
    at2.SetFromDataElement( compensatorcols );
    std::cout « at2.GetValue() « std::endl;
        // (300a,00e9) DS [3.679991\4.249288 ]                      # 18,2 Compensator Pixel Spacing
    gdcm::Attribute<0x300a,0x00e9> at3;
    const gdcm::DataElement &compensatorpixelspacing = nestedds2.GetDataElement( at3.GetTag() );
    at3.SetFromDataElement( compensatorpixelspacing );
    std::cout « at3.GetValue(0) « std::endl;
        // (300a,00ea) DS [-76.00\62.50]                            # 12,2 Compensator Position
    gdcm::Attribute<0x300a,0x00ea> at4;
    const gdcm::DataElement &compensatorposition = nestedds2.GetDataElement( at4.GetTag() );
    at4.SetFromDataElement( compensatorposition );
    std::cout « at4.GetValue(0) « std::endl;
    vtkDoubleArray *d = vtkDoubleArray::New();
    d->SetArray( const_cast<double*>(pts) , at1.GetValue() * at2.GetValue() , 0 );
    vtkImageData *img = vtkImageData::New();
    img->Initialize();
    img->SetDimensions( at2.GetValue(), at1.GetValue(), 1 );
    //imgb->SetExtent(1, xdim, 1, ydim, 1, zdim);
#if (VTK_MAJOR_VERSION >= 6)
    assert(0);
#else
    img->SetScalarTypeToDouble();
#endif
    img->SetSpacing( at3.GetValue(1), at3.GetValue(0), 1); // FIXME image is upside down
    img->SetOrigin( at4.GetValue(0), at4.GetValue(1), 1);
#if (VTK_MAJOR_VERSION >= 6)
    assert(0);
#else
    img->SetNumberOfScalarComponents(1);
#endif
    img->GetPointData()->SetScalars(d);
#if (VTK_MAJOR_VERSION >= 6)
#else
    img->Update();
#endif
    img->Print(std::cout);
    vtkXMLImageDataWriter *writeb= vtkXMLImageDataWriter::New();
#if (VTK_MAJOR_VERSION >= 6)
    writeb->SetInputData( img );
#else
    writeb->SetInput( img );
#endif
    writeb->SetFileName( outfilename );
    writeb->Write( );
/*
(300a,03a6) SQ                                      # u/1,1 Ion Block Sequence
(fffe,e000) na (Item with undefined length)
(300a,00e1) SH [brass ]                           # 6,1 Material ID
(300a,00f7) FL 95.03                              # 4,1 Isocenter to Block Tray Distance
(300a,00f8) CS [APERTURE]                         # 8,1 Block Type
(300a,00fa) CS [ABSENT]                           # 6,1 Block Divergence
(300a,00fb) CS [SOURCE_SIDE ]                     # 12,1 Block Mounting Position
(300a,00fc) IS [1 ]                               # 2,1 Block Number
(300a,0100) DS [50.00 ]                           # 6,1 Block Thickness
(300a,0104) IS [179 ]                             # 4,1 Block Number of Points
```

```
(300a,0106) DS
      [1.7\50.0\14.3\50.0\16.7\49.4\18.7\48.2\19.4\47.7\20.1\47.1\21.0\47.0\22.3\47.0\23.7\46.8\25.7\46.2\27.0\45.6\27.2\45.4\28.2
2\37.4\43.0\37.1\44.7\36] # 1934,2-2n Block Data
(fffe,e00d)
(fffe,e0dd)

*/
    gdcm::Tag tblocksq(0x300a,0x03a6);
    if( !nestedds.FindDataElement( tblocksq ) )
      {
      return 1;
      }
    const gdcm::DataElement &blocksq = nestedds.GetDataElement( tblocksq );
    //std::cout « blocksq « std::endl;
    gdcm::SmartPointer<gdcm::SequenceOfItems> sssqi = blocksq.GetValueAsSQ();
    const gdcm::Item & item3 = sssqi->GetItem(1); // Item start at #1
    const gdcm::DataSet& nestedds3 = item3.GetNestedDataSet();
    gdcm::Tag tblockdata(0x300a,0x0106);
    if( !nestedds3.FindDataElement( tblockdata ) )
      {
      return 1;
      }
    const gdcm::DataElement &blockdata = nestedds3.GetDataElement( tblockdata );
    //  std::cout « blockdata « std::endl;
    gdcm::Attribute<0x300a,0x0106> at_;
    at_.SetFromDataElement( blockdata );
    vtkDoubleArray *scalars = vtkDoubleArray::New();
    scalars->SetNumberOfComponents(3);
    gdcm::Attribute<0x300a,0x0104> bnpts; //  IS [179 ]                                          # 4,1 Block Number
      of Points
    if( !nestedds3.FindDataElement( bnpts.GetTag() ) )
      {
      return 1;
      }
    const gdcm::DataElement &blocknpts = nestedds3.GetDataElement( bnpts.GetTag() );
    bnpts.SetFromDataElement(  blocknpts );
    //std::cout « bnpts.GetValue() « std::endl;
    vtkPolyData *output = vtkPolyData::New();
    vtkPoints *newPts = vtkPoints::New();
    vtkCellArray *polys = vtkCellArray::New();
    const double *ptr = at_.GetValues();
    //unsigned int npts = bnpts.GetNumberOfValues() / 2;
    unsigned int npts = bnpts.GetValue();
    vtkIdType *ptIds =  new vtkIdType[npts];
    for(unsigned int i = 0; i < npts; ++i)
      {
      float x[3] = {};
      x[0] = (float)ptr[2*i+0];
      x[1] = (float)ptr[2*i+1];
      //x[2] = pts[i+2];
      vtkIdType ptId = newPts->InsertNextPoint( x );
      //std::cout « x[0] « "," « x[1] « "," « x[2] « std::endl;
      ptIds[i ] = ptId;
      }
    vtkIdType cellId = polys->InsertNextCell(npts , ptIds);
    (void)cellId;
    delete[] ptIds;
    output->SetPoints(newPts);
    newPts->Delete();
    output->SetPolys(polys);
    polys->Delete();
    //output->GetCellData()->SetScalars(scalars);
    //scalars->Delete();
#if (VTK_MAJOR_VERSION >= 6)
#else
    output->Update();
#endif
    output->Print( std::cout );
  //  }
   vtkRenderWindowInteractor *iren = vtkRenderWindowInteractor::New();
   vtkImageColorViewer *viewer = vtkImageColorViewer::New();
#if (VTK_MAJOR_VERSION >= 6)
   viewer->SetInputData(img);
#else
   viewer->SetInput(img);
#endif
   viewer->SetupInteractor(iren);
   viewer->SetSize(600, 600);
   viewer->GetRenderer()->ResetCameraClippingRange();
   viewer->Render();
   viewer->GetRenderer()->ResetCameraClippingRange();
```

```
  vtkPolyDataMapper *cubeMapper = vtkPolyDataMapper::New();
  //vtkPolyDataMapper2D* cubeMapper = vtkPolyDataMapper2D::New();
#if (VTK_MAJOR_VERSION >= 6)
      cubeMapper->SetInputData( output );
#else
      cubeMapper->SetInput( output );
#endif
      cubeMapper->SetScalarRange(0,7);
  vtkActor *cubeActor = vtkActor::New();
  //vtkActor2D* cubeActor = vtkActor2D::New();
      cubeActor->SetMapper(cubeMapper);
  vtkProperty * property = cubeActor->GetProperty();
  property->SetRepresentationToWireframe();
viewer->GetRenderer()->AddActor( cubeActor );
    vtkXMLPolyDataWriter *writec= vtkXMLPolyDataWriter::New();
#if (VTK_MAJOR_VERSION >= 6)
    writec->SetInputData( output );
#else
    writec->SetInput( output );
#endif
    writec->SetFileName( outfilename2 );
    writec->Write( );
   iren->Initialize();
   iren->Start();
  return 0;
}
```

12.158 gdcmrtplan.cxx

```
/*=========================================================================

Program:  GDCM (Grassroots DICOM). A DICOM library

Copyright (c) 2006-2011 Mathieu Malaterre
All rights reserved.
See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.

This software is distributed WITHOUT ANY WARRANTY; without even
the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
PURPOSE.  See the above copyright notice for more information.

=========================================================================*/
#include "vtkImageData.h"
#include "vtkPointData.h"
#include "vtkPolyData.h"
#include "vtkProperty.h"
#include "vtkPolyDataMapper.h"
#include "vtkActor.h"
#include "vtkRenderer.h"
#include "vtkCellArray.h"
#include "vtkPoints.h"
#include "vtkDoubleArray.h"
#include <vtkXMLImageDataWriter.h>
#include <vtkRenderWindowInteractor.h>
#include <vtkImageColorViewer.h>
#include "gdcmReader.h"
#include "gdcmAttribute.h"
/*
This example is just for fun.  We found a fake RT Ion Plan Storage and simply extracted the viz stuff for VTK
but this is rather a RT Plan storage
*/
int main(int argc, char *argv[])
{
  if( argc < 3 )
    {
    std::cerr « argv[0] « " filename.dcm outfile.vti\n";
    return 1;
    }
  const char * filename = argv[1];
  const char * outfilename = argv[2];
  gdcm::Reader reader;
  reader.SetFileName( filename );
  if( !reader.Read() )
    {
    return 1;
    }
  gdcm::MediaStorage ms;
```

```
  ms.SetFromFile( reader.GetFile() );
  if( ms != gdcm::MediaStorage::RTIonPlanStorage )
    {
    return 1;
    }
/*
(300a,00b0) SQ                                                  # u/l,1 Beam Sequence
(fffe,e000) na (Item with undefined length)
(300a,00b2) SH (no value)                                     # 0,1 Treatment Machine Name
(300a,00c0) IS [1 ]                                           # 2,1 Beam Number
(300a,00c2) LO [1 ]                                           # 2,1 Beam Name
(300a,00c4) CS [STATIC]                                       # 6,1 Beam Type
(300a,00c6) CS [PROTON]                                       # 6,1 Radiation Type
(300a,00ce) CS [TREATMENT ]                                   # 10,1 Treatment Delivery Type
(300a,00e0) IS [1 ]                                           # 2,1 Number of Compensators
(300a,00e3) SQ                                                # u/l,1 Compensator Sequence
(fffe,e000) na (Item with undefined length)
(300a,00e1) SH [lucite]                                     # 6,1 Material ID
(300a,00e4) IS [1 ]                                         # 2,1 Compensator Number
(300a,00e5) SH [75hdhe5 ]                                   # 8,1 Compensator ID
(300a,00e7) IS [35]                                         # 2,1 Compensator Rows
(300a,00e8) IS [37]                                         # 2,1 Compensator Columns
(300a,00e9) DS [3.679991\4.249288 ]                         # 18,2 Compensator Pixel Spacing
(300a,00ea) DS [-76.00\62.50]                               # 12,2 Compensator Position
(300a,00ec) DS
      [52.13\52.13\52.13\53.18\54.04\54.04\47.11\40.06\40.06\38.79\34.87\33.28\33.28\33.28\33.28\35.43\35.43\34.54\34.54\34.71\36.
      # 7618,1-n Compensator Thickness Data
(300a,02e0) CS [ABSENT]                                     # 6,1 Compensator Divergence
(300a,02e1) CS [SOURCE_SIDE ]                               # 12,1 Compensator Mounting Position
(fffe,e00d)
(fffe,e000) na (Item with undefined length)
(fffe,e00d)
(fffe,e0dd)
*/
  const gdcm::DataSet& ds = reader.GetFile().GetDataSet();
  gdcm::Tag tbeamsq(0x300a,0x00b0);
  if( !ds.FindDataElement( tbeamsq ) )
    {
    return 1;
    }
  const gdcm::DataElement &beamsq = ds.GetDataElement( tbeamsq );
  //std::cout « beamsq « std::endl;
  gdcm::SmartPointer<gdcm::SequenceOfItems> sqi = beamsq.GetValueAsSQ();
  if( !sqi || !sqi->GetNumberOfItems() )
    {
    return 1;
    }
  //for(unsigned int pd = 0; pd < sqi->GetNumberOfItems(); ++pd)
  //  {
    //const gdcm::Item & item = sqi->GetItem(1); // Item start at #1
    const gdcm::Item & item = sqi->GetItem(2); // Item start at #1
    const gdcm::DataSet& nestedds = item.GetNestedDataSet();
    //std::cout « nestedds « std::endl;
    gdcm::Tag tcompensatorsq(0x300a,0x00e3);
    if( !nestedds.FindDataElement( tcompensatorsq ) )
      {
      return 1;
      }
    const gdcm::DataElement &compensatorsq = nestedds.GetDataElement( tcompensatorsq );
    //std::cout « compensatorsq « std::endl;
    gdcm::SmartPointer<gdcm::SequenceOfItems> ssqi = compensatorsq.GetValueAsSQ();
    const gdcm::Item & item2 = ssqi->GetItem(1); // Item start at #1
    const gdcm::DataSet& nestedds2 = item2.GetNestedDataSet();
    //std::cout « nestedds2 « std::endl;
    gdcm::Tag tcompensatorthicknessdata(0x300a,0x00ec);
    if( !nestedds2.FindDataElement( tcompensatorthicknessdata ) )
      {
      return 1;
      }
    const gdcm::DataElement &compensatorthicknessdata = nestedds2.GetDataElement( tcompensatorthicknessdata );
    //  std::cout « compensatorthicknessdata « std::endl;
    gdcm::Attribute<0x300a,0x00ec> at;
    at.SetFromDataElement( compensatorthicknessdata );
    const double* pts = at.GetValues();
    //        (300a,00e7) IS [35]                                     # 2,1 Compensator Rows
    gdcm::Attribute<0x300a,0x00e7> at1;
    const gdcm::DataElement &compensatorrows = nestedds2.GetDataElement( at1.GetTag() );
    at1.SetFromDataElement( compensatorrows );
    std::cout « at1.GetValue() « std::endl;
    //        (300a,00e8) IS [37]                                     # 2,1 Compensator Columns
    gdcm::Attribute<0x300a,0x00e8> at2;
```

```
    const gdcm::DataElement &compensatorcols = nestedds2.GetDataElement( at2.GetTag() );
    at2.SetFromDataElement( compensatorcols );
    std::cout « at2.GetValue() « std::endl;
        // (300a,00e9) DS [3.679991\4.249288 ]                      # 18,2 Compensator Pixel Spacing
    gdcm::Attribute<0x300a,0x00e9> at3;
    const gdcm::DataElement &compensatorpixelspacing = nestedds2.GetDataElement( at3.GetTag() );
    at3.SetFromDataElement( compensatorpixelspacing );
    std::cout « at3.GetValue(0) « std::endl;
        // (300a,00ea) DS [-76.00\62.50]                            # 12,2 Compensator Position
    gdcm::Attribute<0x300a,0x00ea> at4;
    const gdcm::DataElement &compensatorposition = nestedds2.GetDataElement( at4.GetTag() );
    at4.SetFromDataElement( compensatorposition );
    std::cout « at4.GetValue(0) « std::endl;
    vtkDoubleArray *d = vtkDoubleArray::New();
    d->SetArray( const_cast<double*>(pts) , at1.GetValue() * at2.GetValue() , 0 );
    vtkImageData *img = vtkImageData::New();
    img->Initialize();
    img->SetDimensions( at2.GetValue(), at1.GetValue(), 1 );
    //imgb->SetExtent(1, xdim, 1, ydim, 1, zdim);
#if (VTK_MAJOR_VERSION >= 6)
    assert(0);
#else
    img->SetScalarTypeToDouble();
#endif
    img->SetSpacing( at3.GetValue(1), at3.GetValue(0), 1); // FIXME image is upside down
    img->SetOrigin( at4.GetValue(0), at4.GetValue(1), 1);
#if (VTK_MAJOR_VERSION >= 6)
    assert(0);
#else
    img->SetNumberOfScalarComponents(1);
#endif
    img->GetPointData()->SetScalars(d);
    vtkXMLImageDataWriter *writeb= vtkXMLImageDataWriter::New();
#if (VTK_MAJOR_VERSION >= 6)
    writeb->SetInputData( img );
#else
    writeb->SetInput( img );
#endif
    writeb->SetFileName( outfilename );
    writeb->Write( );
/*
(300a,00f4) SQ                                          # u/l,1 Block Sequence
(fffe,e000) na (Item with undefined length)
(300a,00e1) SH [brass ]                                 # 6,1 Material ID
(300a,00f8) CS [APERTURE]                               # 8,1 Block Type
(300a,00fa) CS [ABSENT]                                 # 6,1 Block Divergence
(300a,00fb) CS [SOURCE_SIDE ]                           # 12,1 Block Mounting Position
(300a,00fc) IS [1 ]                                     # 2,1 Block Number
(300a,0100) DS [50.00 ]                                 # 6,1 Block Thickness
(300a,0104) IS [179 ]                                   # 4,1 Block Number of Points
(300a,0106) DS
      [1.7\50.0\14.3\50.0\16.7\49.4\18.7\48.2\19.4\47.7\20.1\47.1\21.0\47.0\22.3\47.0\23.7\46.8\25.7\46.2\27.0\45.6\27.2\45.4\28.2
      # 1934,2-2n Block Data
(fffe,e00d)
(fffe,e000) na (Item with undefined length)
(fffe,e00d)
(fffe,e0dd)
*/
    gdcm::Tag tblocksq(0x300a,0x00f4);
    if( !nestedds.FindDataElement( tblocksq ) )
      {
      return 1;
      }
    const gdcm::DataElement &blocksq = nestedds.GetDataElement( tblocksq );
    //std::cout « blocksq « std::endl;
    gdcm::SmartPointer<gdcm::SequenceOfItems> sssqi = blocksq.GetValueAsSQ();
    const gdcm::Item & item3 = sssqi->GetItem(1); // Item start at #1
    const gdcm::DataSet& nestedds3 = item3.GetNestedDataSet();
    gdcm::Tag tblockdata(0x300a,0x0106);
    if( !nestedds3.FindDataElement( tblockdata ) )
      {
      return 1;
      }
    const gdcm::DataElement &blockdata = nestedds3.GetDataElement( tblockdata );
    //  std::cout « blockdata « std::endl;
    gdcm::Attribute<0x300a,0x0106> at_;
    at_.SetFromDataElement( blockdata );
    vtkDoubleArray *scalars = vtkDoubleArray::New();
    scalars->SetNumberOfComponents(3);
    gdcm::Attribute<0x300a,0x0104> bnpts; //  IS [179 ] # 4,1 Block Number of Points
    if( !nestedds3.FindDataElement( bnpts.GetTag() ) )
```

```
      {
      return 1;
      }
    const gdcm::DataElement &blocknpts = nestedds3.GetDataElement( bnpts.GetTag() );
    bnpts.SetFromDataElement(  blocknpts );
    std::cout « bnpts.GetValue() « std::endl;
    vtkPolyData *output = vtkPolyData::New();
    vtkPoints *newPts = vtkPoints::New();
    vtkCellArray *polys = vtkCellArray::New();
    const double *ptr = at_.GetValues();
    //unsigned int npts = bnpts.GetNumberOfValues() / 2;
    unsigned int npts = bnpts.GetValue();
    vtkIdType *ptIds =  new vtkIdType[npts];
    for(unsigned int i = 0; i < npts; ++i)
      {
      float x[3] = {};
      x[0] = (float)ptr[2*i+0];
      x[1] = (float)ptr[2*i+1];
      //x[2] = pts[i+2];
      vtkIdType ptId = newPts->InsertNextPoint( x );
      //std::cout « x[0] « "," « x[1] « "," « x[2] « std::endl;
      ptIds[i ] = ptId;
      }
    vtkIdType cellId = polys->InsertNextCell(npts , ptIds);
    (void)cellId;
    delete[] ptIds;
    output->SetPoints(newPts);
    newPts->Delete();
    output->SetPolys(polys);
    polys->Delete();
    //output->GetCellData()->SetScalars(scalars);
    //scalars->Delete();
#if (VTK_MAJOR_VERSION >= 6)
#else
    output->Update();
#endif
    output->Print( std::cout );
  //  }
   vtkRenderWindowInteractor *iren = vtkRenderWindowInteractor::New();
   vtkImageColorViewer *viewer = vtkImageColorViewer::New();
#if (VTK_MAJOR_VERSION >= 6)
   viewer->SetInputData(img);
#else
   viewer->SetInput(img);
#endif
   viewer->SetupInteractor(iren);
   viewer->SetSize(600, 600);
   viewer->Render();
  vtkPolyDataMapper *cubeMapper = vtkPolyDataMapper::New();
  //vtkPolyDataMapper2D* cubeMapper = vtkPolyDataMapper2D::New();
#if (VTK_MAJOR_VERSION >= 6)
      cubeMapper->SetInputData( output );
#else
      cubeMapper->SetInput( output );
#endif
      cubeMapper->SetScalarRange(0,7);
  vtkActor *cubeActor = vtkActor::New();
  //vtkActor2D* cubeActor = vtkActor2D::New();
      cubeActor->SetMapper(cubeMapper);
  vtkProperty * property = cubeActor->GetProperty();
  property->SetRepresentationToWireframe();
  viewer->GetRenderer()->AddActor( cubeActor );
   iren->Initialize();
   iren->Start();
  return 0;
}
```

12.159 gdcmscene.cxx

```
/*=========================================================================

Program:  GDCM (Grassroots DICOM). A DICOM library

Copyright (c) 2006-2011 Mathieu Malaterre
All rights reserved.
See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.
```

```
This software is distributed WITHOUT ANY WARRANTY; without even
the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
PURPOSE.  See the above copyright notice for more information.

=========================================================================*/
#include "vtkGDCMPolyDataReader.h"
//#include "vtkGDCMPolyDataWriter.h"
#include "vtkAppendPolyData.h"
#include "vtkPolyDataWriter.h"
#include "vtkPolyDataMapper.h"
#include "vtkPolyDataMapper2D.h"
#include "vtkActor2D.h"
#include "vtkRenderWindowInteractor.h"
#include "vtkRenderWindow.h"
#include "vtkRenderer.h"
#include "vtkCamera.h"
#include "vtkProperty.h"
#include "vtkProperty2D.h"
// gdcmDataExtra/gdcmNonImageData/exRT_Structure_Set_Storage.dcm
// gdcmDataExtra/gdcmNonImageData/RTSTRUCT_1.3.6.1.4.1.22213.1.1396.2.dcm
// gdcmDataExtra/gdcmNonImageData/RT/RTStruct.dcm
int main(int argc, char *argv[])
{
  if( argc < 2 )
    {
    std::cerr « argv[0] « " filename1.dcm\n";
    return 1;
    }
  const char * filename = argv[1];
  vtkGDCMPolyDataReader * reader = vtkGDCMPolyDataReader::New();
  reader->SetFileName( filename );
  reader->Update();
//  vtkGDCMPolyDataWriter * writer2 = vtkGDCMPolyDataWriter::New();
//  for(int num = 0; num < reader->GetNumberOfOutputPorts(); ++num )
//    writer2->SetInput( num, reader->GetOutput(num) );
//  writer2->SetFileName( "rtstruct.dcm" );
//  writer2->Write();
  // print reader output:
  reader->Print( std::cout );
  // print first output:
  reader->GetOutput()->Print( std::cout );
  vtkAppendPolyData *append = vtkAppendPolyData::New();
  int n = reader->GetNumberOfOutputPorts();
  for(int i = 0; i < n; ++i)
    {
#if (VTK_MAJOR_VERSION >= 6)
    append->AddInputConnection( reader->GetOutputPort(i) );
#else
    append->AddInput( reader->GetOutput(i) );
#endif
    }
  vtkPolyDataWriter * writer = vtkPolyDataWriter::New();
#if (VTK_MAJOR_VERSION >= 6)
  writer->SetInputConnection( reader->GetOutputPort() );
#else
  writer->SetInput( reader->GetOutput() );
#endif
  writer->SetFileName( "rtstruct.vtk" );
  //writer->Write();
  // Now we'll look at it.
  vtkPolyDataMapper *cubeMapper = vtkPolyDataMapper::New();
  //vtkPolyDataMapper2D* cubeMapper = vtkPolyDataMapper2D::New();
      //cubeMapper->SetInput( reader->GetOutput() );
#if (VTK_MAJOR_VERSION >= 6)
      cubeMapper->SetInputConnection( append->GetOutputPort() );
#else
      cubeMapper->SetInput( append->GetOutput() );
#endif
      cubeMapper->SetScalarRange(0,7);
  vtkActor *cubeActor = vtkActor::New();
  //vtkActor2D* cubeActor = vtkActor2D::New();
      cubeActor->SetMapper(cubeMapper);
  vtkProperty * property = cubeActor->GetProperty();
  property->SetRepresentationToWireframe();
  //cubeActor->GetProperty()->SetColor(1, 0, 0);
  // The usual rendering stuff.
//  vtkCamera *camera = vtkCamera::New();
//      camera->SetPosition(1,1,1);
//      camera->SetFocalPoint(0,0,0);
  vtkRenderer *renderer = vtkRenderer::New();
  vtkRenderWindow *renWin = vtkRenderWindow::New();
```

```
    renWin->AddRenderer(renderer);
  vtkRenderWindowInteractor *iren = vtkRenderWindowInteractor::New();
    iren->SetRenderWindow(renWin);
  renderer->AddActor(cubeActor);
  //renderer->AddActor2D(cubeActor);
      //renderer->SetActiveCamera(camera);
      renderer->ResetCamera();
      renderer->SetBackground(1,1,1);
  renWin->SetSize(300,300);
  // interact with data
  renWin->Render();
  iren->Start();
  reader->Delete();
  append->Delete();
  cubeMapper->Delete();
  cubeActor->Delete();
//  camera->Delete();
  renderer->Delete();
  renWin->Delete();
  iren->Delete();
  writer->Delete();
  return 0;
}
```

12.160 gdcmtexture.cxx

```
/*=========================================================================

Program:  GDCM (Grassroots DICOM). A DICOM library

Copyright (c) 2006-2011 Mathieu Malaterre
All rights reserved.
See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.

This software is distributed WITHOUT ANY WARRANTY; without even
the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
PURPOSE.  See the above copyright notice for more information.

=========================================================================*/
#include "vtkGDCMImageReader.h"
#include "vtkRenderer.h"
#include "vtkAssembly.h"
#include "vtkRenderWindow.h"
#include "vtkAnnotatedCubeActor.h"
#include "vtkTransform.h"
#include "vtkAxesActor.h"
#include "vtkTextProperty.h"
#include "vtkCaptionActor2D.h"
#include "vtkPropAssembly.h"
#include "vtkOrientationMarkerWidget.h"
#include "vtkRenderWindowInteractor.h"
#include "vtkPolyDataMapper.h"
#include "vtkActor.h"
#include "vtkImageData.h"
#include "vtkLookupTable.h"
#include "vtkTexture.h"
#include "vtkPlaneSource.h"
int main( int argc, char *argv[] )
{
  if( argc < 2 ) return 1;
  vtkGDCMImageReader *reader = vtkGDCMImageReader::New();
  reader->SetFileName( argv[1] );
  reader->Update();
  vtkImageData* ima = reader->GetOutput();
  vtkLookupTable* table = vtkLookupTable::New();
  table->SetNumberOfColors(1000);
  table->SetTableRange(0,1000);
  table->SetSaturationRange(0,0);
  table->SetHueRange(0,1);
  table->SetValueRange(0,1);
  table->SetAlphaRange(1,1);
  table->Build();
  // Texture
  vtkTexture* texture = vtkTexture::New();
#if (VTK_MAJOR_VERSION >= 6)
  texture->SetInputData(ima);
#else
```

```
  texture->SetInput(ima);
#endif
  texture->InterpolateOn();
  texture->SetLookupTable(table);
  // PlaneSource
  vtkPlaneSource* plane = vtkPlaneSource::New();
  plane->SetOrigin( -0.5, -0.5, 0.0);
  plane->SetPoint1(  0.5, -0.5, 0.0);
  plane->SetPoint2( -0.5,  0.5, 0.0);
  // PolyDataMapper
  vtkPolyDataMapper *planeMapper = vtkPolyDataMapper::New();
#if (VTK_MAJOR_VERSION >= 6)
  planeMapper->SetInputConnection(plane->GetOutputPort());
#else
  planeMapper->SetInput(plane->GetOutput());
#endif
  // Actor
  vtkActor* planeActor = vtkActor::New();
  planeActor->SetTexture(texture);
  planeActor->SetMapper(planeMapper);
  planeActor->PickableOn();
  // Final rendering with simple interactor:
  vtkRenderer        *ren = vtkRenderer::New();
  vtkRenderWindow *renwin = vtkRenderWindow::New();
  renwin->AddRenderer(ren);
  vtkRenderWindowInteractor *iren = vtkRenderWindowInteractor::New();
  iren->SetRenderWindow(renwin);
  ren->AddActor(planeActor);
  ren->SetBackground(0,0,0.5);
  vtkAnnotatedCubeActor* cube = vtkAnnotatedCubeActor::New();
  cube->SetXPlusFaceText ( "L" );
  cube->SetXMinusFaceText( "R" );
  cube->SetYPlusFaceText ( "A" );
  cube->SetYMinusFaceText( "P" );
  cube->SetZPlusFaceText ( "H" );
  cube->SetZMinusFaceText( "F" );
  vtkAxesActor* axes2 = vtkAxesActor::New();
  // simulate a left-handed coordinate system
  //
  vtkTransform *transform = vtkTransform::New();
  transform->Identity();
  //transform->RotateY(180);
  reader->GetDirectionCosines()->Print(std::cout);
  transform->Concatenate(reader->GetDirectionCosines());
  //axes2->SetShaftTypeToCylinder();
  axes2->SetUserTransform( transform );
  //cube->SetUserTransform( transform ); // cant get it to work
  cube->GetAssembly()->SetUserTransform( transform ); // cant get it to work
  vtkPropAssembly* assembly = vtkPropAssembly::New();
  assembly->AddPart( axes2 );
  assembly->AddPart( cube );
  vtkOrientationMarkerWidget* widget = vtkOrientationMarkerWidget::New();
  //widget->SetOutlineColor( 0.9300, 0.5700, 0.1300 );
  widget->SetOrientationMarker( assembly );
  widget->SetInteractor( iren );
  //widget->SetViewport( 0.0, 0.0, 0.4, 0.4 );
  widget->SetEnabled( 1 );
  widget->InteractiveOff();
  widget->InteractiveOn();
  renwin->Render();
  iren->Start();
  // Clean up:
  reader->Delete();
  table->Delete();
  texture->Delete();
  plane->Delete();
  planeMapper->Delete();
  planeActor->Delete();
  ren->Delete();
  renwin->Delete();
  iren->Delete();
  return 0;
}
```

12.161 gdcmvolume.cxx

```
/*=========================================================================
```

```
Program:  GDCM (Grassroots DICOM). A DICOM library

Copyright (c) 2006-2011 Mathieu Malaterre
All rights reserved.
See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.

This software is distributed WITHOUT ANY WARRANTY; without even
the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
PURPOSE.  See the above copyright notice for more information.

=========================================================================*/
#include "vtkVersion.h"
#include "vtkGDCMImageReader.h"
#include "vtkPiecewiseFunction.h"
#include "vtkColorTransferFunction.h"
#include "vtkVolume.h"
#include "vtkVolumeProperty.h"
#if VTK_MAJOR_VERSION < 7
#include "vtkVolumeTextureMapper3D.h"
#endif
#include "vtkFixedPointVolumeRayCastMapper.h"
#include "vtkInteractorStyleTrackballCamera.h"
#include "vtkRenderer.h"
#include "vtkRenderWindow.h"
#include "vtkImageClip.h"
#include "vtkRenderWindowInteractor.h"
// gdcmvolume gdcmData/GE_DLX-8-MONO2-Multiframe-Jpeg_Lossless.dcm
int main(int argc, char *argv[])
{
  if( argc < 2 ) return 1;
  vtkGDCMImageReader *reader = vtkGDCMImageReader::New();
  reader->SetFileName( argv[1] );
  reader->Update();
  // Create the renderers, render window, and interactor
  vtkRenderWindow *renWin = vtkRenderWindow::New();
  vtkRenderWindowInteractor *iren = vtkRenderWindowInteractor::New();
  iren->SetRenderWindow(renWin);
  vtkRenderer *ren = vtkRenderer::New();
  renWin->AddRenderer(ren);
  // Create a transfer function mapping scalar value to opacity
  vtkPiecewiseFunction *oTFun = vtkPiecewiseFunction::New();
  //oTFun->AddSegment(0, 1.0, 256, 0.1);
  oTFun->AddSegment(0, 1.0, 240, 0.1);
  vtkColorTransferFunction *cTFun = vtkColorTransferFunction::New();
  cTFun->AddRGBPoint(   0, 1.0, 1.0, 1.0 );
  //cTFun->AddRGBPoint( 255, 1.0, 1.0, 1.0 );
  cTFun->AddRGBPoint( 240, 1.0, 1.0, 1.0 );
  // Need to crop to actually see minimum intensity
  vtkImageClip *clip = vtkImageClip::New();
  clip->SetInputConnection( reader->GetOutputPort() );
  clip->SetOutputWholeExtent(0,66,0,66,30,37);
  clip->ClipDataOn();
  vtkVolumeProperty *property = vtkVolumeProperty::New();
  property->SetScalarOpacity(oTFun);
  property->SetColor(cTFun);
  property->SetInterpolationTypeToLinear();
  vtkFixedPointVolumeRayCastMapper *mapper = vtkFixedPointVolumeRayCastMapper::New();
  mapper->SetBlendModeToMinimumIntensity();
  mapper->SetInputConnection( reader->GetOutputPort() );
  vtkVolume *volume = vtkVolume::New();
  volume->SetMapper(mapper);
  volume->SetProperty(property);
  ren->AddViewProp(volume);
  renWin->Render();
    {
    iren->Start();
    }
  volume->Delete();
  mapper->Delete();
  property->Delete();
  clip->Delete();
  cTFun->Delete();
  oTFun->Delete();
  reader->Delete();
  renWin->Delete();
  iren->Delete();
  ren->Delete();
 return 0;
}
```

12.162 offscreenimage.cxx

```
/*=========================================================================

Program:  GDCM (Grassroots DICOM). A DICOM library

Copyright (c) 2006-2011 Mathieu Malaterre
All rights reserved.
See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.

This software is distributed WITHOUT ANY WARRANTY; without even
the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
PURPOSE.  See the above copyright notice for more information.

=========================================================================*/
#include "vtkGDCMImageReader.h"
#include "vtkRenderWindow.h"
#include "vtkRenderer.h"
#include "vtkImageMapToWindowLevelColors.h"
#include "vtkImageActor.h"
#include "vtkPNGWriter.h"
#include "vtkWindowToImageFilter.h"
#include "vtkMedicalImageProperties.h"
int main(int argc, char *argv[])
{
  if( argc < 2 )
    {
    return 1;
    }
  const char *filename = argv[1];
  vtkGDCMImageReader *reader = vtkGDCMImageReader::New();
  reader->SetFileName( filename );
  reader->Update(); // important to read the window/level info
  vtkMedicalImageProperties *prop = reader->GetMedicalImageProperties();
  vtkRenderWindow *renWin = vtkRenderWindow::New();
  renWin->OffScreenRenderingOn();
  vtkRenderer *renderer = vtkRenderer::New();
  renWin->AddRenderer(renderer);
  vtkImageMapToWindowLevelColors *windowlevel = vtkImageMapToWindowLevelColors::New();
#if (VTK_MAJOR_VERSION >= 6)
  windowlevel->SetInputConnection( reader->GetOutputPort() );
#else
  windowlevel->SetInput( reader->GetOutput() );
#endif
  unsigned int n = prop->GetNumberOfWindowLevelPresets();
  if( n )
    {
    // Take the first one by default:
    const double *wl = prop->GetNthWindowLevelPreset(0);
    windowlevel->SetWindow( wl[0] );
    windowlevel->SetLevel( wl[1] );
    }
  vtkImageActor *actor = vtkImageActor::New();
#if (VTK_MAJOR_VERSION >= 6)
  actor->SetInputData( windowlevel->GetOutput() );
#else
  actor->SetInput( windowlevel->GetOutput() );
#endif
  renderer->AddActor( actor );
  renWin->Render();
  vtkWindowToImageFilter *w2if = vtkWindowToImageFilter::New();
  w2if->SetInput ( renWin );
  vtkPNGWriter *wr = vtkPNGWriter::New();
#if (VTK_MAJOR_VERSION >= 6)
  wr->SetInputConnection( w2if->GetOutputPort() );
#else
  wr->SetInput( w2if->GetOutput() );
#endif
  wr->SetFileName ( "offscreenimage.png" );
  wr->Write();
  reader->Delete();
  renWin->Delete();
  renderer->Delete();
  windowlevel->Delete();
  actor->Delete();
  w2if->Delete();
  wr->Delete();
  return 0;
}
```

12.163 reslicesphere.cxx

```
/*=========================================================================

Program:  GDCM (Grassroots DICOM). A DICOM library

Copyright (c) 2006-2011 Mathieu Malaterre
All rights reserved.
See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.

This software is distributed WITHOUT ANY WARRANTY; without even
the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
PURPOSE.  See the above copyright notice for more information.

=========================================================================*/
//
// Load a DICOM series.
// Position a sphere within the volume.
// Allow the user to change between Axial, Sagittal, Coronal, and
//     Oblique view of the images and move through the slices.
// The display should show the resliced image and the cross section
// of the sphere intersecting that plane.
//
/*
from  Scott Johnson /Scott Johnson neuwave com/
to  VTK /vtkusers vtk.org/
date  Tue, May 11, 2010 at 7:01 PM
*/
#include <string>
#include <vtkDICOMImageReader.h>
#include <vtkStringArray.h>
#include <vtkDirectory.h>
#include <vtkImageThreshold.h>
#include <vtkImageShiftScale.h>
#include <vtkImageReslice.h>
#include <vtkRenderWindowInteractor.h>
#include <vtkImageViewer2.h>
#include <vtkSphereSource.h>
#include <vtkPolyDataMapper.h>
#include <vtkPlane.h>
#include <vtkCutter.h>
#include <vtkActor.h>
#include <vtkCommand.h>
#include <vtkSmartPointer.h>
#include <vtkMatrix4x4.h>
#include <vtkInteractorObserver.h>
#include <vtkProperty.h>
#include <vtkRenderer.h>
#include <vtkImageData.h>
#include <vtkImageActor.h>
#include "vtkTransformPolyDataFilter.h"
#include <vtkCamera.h>
#include <vtkMath.h>
#include <vtkTransform.h>
#include <vtkTextActor.h>
#include <vtkActor2D.h>
#include <vtkPolyDataMapper2D.h>
#include <vtkProperty2D.h>
#include <vtkGDCMImageReader.h>
#include <vtkImageChangeInformation.h>
#include "gdcmDirectory.h"
#include "gdcmTesting.h"
#include "gdcmIPPSorter.h"
// Change to match the path to find Raw_0.vti or provide
// the parameter when starting ResliceSphere.
const double sphereCenter[3]={74, 219, 70};
// Angles (0, 0, 0)
const double AxialMatrix[]    = { 1.0,  0.0,  0.0,  0.0,
                                  0.0,  1.0,  0.0,  0.0,
                                  0.0,  0.0,  1.0,  0.0,
                                  0.0,  0.0,  0.0,  1.0 };
// Angles (0, 90, 0)
const double SagittalMatrix[] = { 0.0,  0.0,  1.0,  0.0,
                                  0.0,  1.0,  0.0,  0.0,
                                 -1.0,  0.0,  0.0,  0.0,
                                  0.0,  0.0,  0.0,  1.0 };
// Angles (-90, 0, 0)
const double CoronalMatrix[]  = { 1.0,  0.0,  0.0,  0.0,
                                  0.0,  0.0,  1.0,  0.0,
                                  0.0, -1.0,  0.0,  0.0,
```

```
                                  0.0,  0.0,  0.0,  1.0 };
// Angles (0, 90, 31)
const double ObliqueMatrix[] =  { 0.0, -0.515038, 0.857167, 0.0,
                                  0.0,  0.857167, 0.515038, 0.0,
                                 -1.0,  0.0,      0.0,      0.0,
                                  0.0,  0.0,      0.0,      1.0 };
class ResliceRender;
// Class to handle key press events.
class KeyCallback :  public vtkCommand
{
public:
    static KeyCallback* New()
    {
        return new KeyCallback();
    }
    void Execute(vtkObject* caller, unsigned long eventId, void *calldata);
    void SetCallbackData(ResliceRender* reslice);
protected:
    ResliceRender* _reslice;
};
class ResliceRender
{
public:
    typedef enum _ORIENTATION
    {
        AXIAL = 0,
        SAGITTAL = 1,
        CORONAL = 2,
        OBLIQUE = 3
    } ORIENTATION;
    ResliceRender()
    {
        _orientation=AXIAL;
    }
    ~ResliceRender()
    {
        _transform->Delete();
        _reader->Delete();
        _reslice->Delete();
        _interactor->Delete();
        _imageViewer->Delete();
        _sphere->Delete();
        _sphereMapper->Delete();
        _sphereActor->Delete();
        _plane->Delete();
        _cutter->Delete();
        _polyTransform->Delete();
        _ROIMapper->Delete();
        _ROIActor->Delete();
        _annotation->Delete();
    }
    void CreatePipeline(const char* fileName)
    {
        vtkProperty2D* props;
        //_reader=vtkXMLImageDataReader::New();
        //_reader->SetFileName(fileName);
        //_reader->Update();
        //_reader=qzDICOMImageReader::New();
        _reader=vtkGDCMImageReader::New();
      //vtkDirectory *d = vtkDirectory::New();
      //d->Open(fileName);
      //d->Print( std::cout );
      gdcm::Directory d;
      d.Load(fileName);
      gdcm::Directory::FilenamesType const &files = d.GetFilenames();
  gdcm::IPPSorter s;
  s.SetComputeZSpacing( true );
  s.SetZSpacingTolerance( 1e-3 );
  bool b = s.Sort( files );
  if( !b )
    {
    std::cerr « "Failed to sort:" « fileName « std::endl;
    //return ;
    }
  //std::cout « "Sorting succeeded:" « std::endl;
  //s.Print( std::cout );
  //std::cout « "Found z-spacing:" « std::endl;
  //std::cout « s.GetZSpacing() « std::endl;
  double ippzspacing = s.GetZSpacing();
  const std::vector<std::string> & sorted = s.GetFilenames();
  vtkStringArray *vtkfiles = vtkStringArray::New();
```

```
  std::vector< std::string >::const_iterator it = sorted.begin();
  for( ; it != sorted.end(); ++it)
    {
    const std::string &f = *it;
    vtkfiles->InsertNextValue( f.c_str() );
    }
        //_reader->SetDirectoryName(fileName);
        //_reader->SetFileNames( d->GetFiles() );
        _reader->SetFileNames( vtkfiles );
        _reader->Update();
#ifndef vtkFloatingPointType
#define vtkFloatingPointType double
#endif
  const vtkFloatingPointType *spacing = _reader->GetOutput()->GetSpacing();
  vtkImageChangeInformation *v16 = vtkImageChangeInformation::New();
#if (VTK_MAJOR_VERSION >= 6)
  v16->SetInputConnection( _reader->GetOutputPort() );
#else
  v16->SetInput( _reader->GetOutput() );
#endif
  v16->SetOutputSpacing( spacing[0], spacing[1], ippzspacing );
  v16->Update();
        _threshold=vtkImageThreshold::New();
        _threshold->ThresholdByUpper(-3024.0);
        _threshold->ReplaceOutOn();
        _threshold->SetOutValue(0.0);
        _threshold->SetInputConnection(v16->GetOutputPort());
        _shift=vtkImageShiftScale::New();
        _shift->SetShift(0);
        _shift->SetScale(1);
        _shift->SetInputConnection(_threshold->GetOutputPort());
        // Initialize the reslice with an axial orientation.
        vtkSmartPointer<vtkMatrix4x4> matrix =
            vtkSmartPointer<vtkMatrix4x4>::New();
        matrix->Identity();
        _transform = vtkTransform::New();
        _transform->SetMatrix(matrix);
        _reslice = vtkImageReslice::New();
        _reslice->SetOutputDimensionality(3);
        // PROBLEM:
        // The original intent was to connect the same transform
        // to the vtkImageReslice and vtkTransformPolyDataFilter,
        // but the resulting reslices appear different using the
        // vtkTransform as opposed to explicitly setting the
        // reslice axes via SetResliceAxes.    Also, if the vtkTransform
        // is connected and orientated other than axial, the extents
        // don't seem to update resulting in VTK believing the slice
        // is out of range.
        //_reslice->SetResliceTransform(_transform);
        _reslice->SetResliceAxes(matrix);
        //_reslice->SetInputConnection(_reader->GetOutputPort());
        _reslice->SetInputConnection(_shift->GetOutputPort());
        // Create the sphere target shape.
        _sphere=vtkSphereSource::New();
        _sphere->SetRadius(7.0);
        _sphere->SetThetaResolution(16);
        _sphere->SetPhiResolution(16);
        _sphere->SetCenter(sphereCenter[0], sphereCenter[1], sphereCenter[2]);
        _sphereMapper=vtkPolyDataMapper::New();
        _sphereMapper->SetInputConnection(_sphere->GetOutputPort());
        _sphereActor=vtkActor::New();
        _sphereActor->SetMapper(_sphereMapper);
        _sphereActor->PickableOff();
        _sphereActor->GetProperty()->SetColor(1.0, 0.0, 0.0);
        _sphereActor->GetProperty()->SetEdgeColor(1.0, 0.0, 0.0);
        _sphereActor->GetProperty()->SetDiffuseColor(1.0, 0.0, 0.0);
        _sphereActor->SetVisibility(true);
        // Create the cutting pipeline.
        // This plane will be positioned in the original image coordinate system.
        _plane = vtkPlane::New();
        _plane->SetNormal(0.0, 0.0, 1.0);
        _cutter = vtkCutter::New();
        _cutter->SetInputConnection(_sphere->GetOutputPort());
        _cutter->SetCutFunction(_plane);
        _cutter->GenerateCutScalarsOn();
        _cutter->SetValue(0, 0.5);
        // The transform attached to _polyTransform should move the cut
        // ROI into the resliced coordinate system, which should be the
        // same as the coordinate system of the resliced images.
        // PROBLEM:  It doesn't.
        _polyTransform = vtkTransformPolyDataFilter::New();
```

```
        _polyTransform->SetTransform(_transform);
        _polyTransform->SetInputConnection(_cutter->GetOutputPort());
        _ROIMapper = vtkPolyDataMapper2D::New();
        _ROIMapper->SetInputConnection(_polyTransform->GetOutputPort());
    vtkCoordinate* coordinate = vtkCoordinate::New();
    coordinate->SetCoordinateSystemToWorld();
    _ROIMapper->SetTransformCoordinate(coordinate);
        _ROIActor = vtkActor2D::New();
        _ROIActor->SetMapper(_ROIMapper);
        // Make sure the cut can be seen, especially the edges.
        props=_ROIActor->GetProperty();
        props->SetLineWidth(2);
        props->SetOpacity(1.0);
//        props->EdgeVisibilityOn();
//        props->SetDiffuse(0.8);
//        props->SetSpecular(0.3);
//        props->SetSpecularPower(20);
//        props->SetRepresentationToSurface();
//        props->SetDiffuseColor(1.0, 0.0, 0.0);
//        props->SetEdgeColor(1.0, 0.0, 0.0);
        props->SetColor(1.0, 0.0, 0.0);
        _interactor = vtkRenderWindowInteractor::New();
        // Create the image viewer and add the actor with the cut ROI.
        _imageViewer = vtkImageViewer2::New();
        _imageViewer->SetupInteractor(_interactor);
        _imageViewer->SetSize(400, 400);
        _imageViewer->SetColorWindow(1024);
        _imageViewer->SetColorLevel(800);
        _imageViewer->SetInputConnection(_reslice->GetOutputPort());
        _imageViewer->GetImageActor()->SetOpacity(0.5);
        _annotation = vtkTextActor::New();
        _annotation->SetTextScaleModeToViewport();
        _imageViewer->GetRenderer()->AddActor(_annotation);
        // Add the cut shape actor to the renderer.
        _imageViewer->GetRenderer()->AddActor(_ROIActor);
        // Set up the key handler.
        vtkSmartPointer<KeyCallback> callback = vtkSmartPointer<KeyCallback>::New();
        callback->SetCallbackData(this);
        _interactor->AddObserver(vtkCommand::KeyPressEvent, callback);
        _interactor->Initialize();
    }
    void Start()
    {
        _interactor->Start();
    }
    void ResetOrientation()
    {
        vtkSmartPointer<vtkMatrix4x4> matrix =
            vtkSmartPointer<vtkMatrix4x4>::New();
        matrix->Identity();
        SetOrientation(matrix);
    }
    // Make sure the orientation of the vtkImageReslice and
    // vtkTransform are in sync.
    void SetOrientation(vtkMatrix4x4* matrix)
    {
        _reslice->SetResliceAxes(matrix);
        _reslice->Update();
    vtkMatrix4x4* inverse = vtkMatrix4x4::New();
    vtkMatrix4x4::Invert(matrix, inverse);
        _transform->SetMatrix(inverse);
        _transform->Update();
    }
    // Set the current slice of the current view.
    void SetSlice(int slice)
    {
        std::stringstream posString;
        double    center[3];
        double    spacing[3];
        double    origin[3];
        double    point[4];
        double    newPoint[4];
        vtkImageData* imageData;
        int newSlice;
        // Try to make sure the extents of the reslice are updated.
        // PROBLEM:  It doesn't seem to work when changing the orientation.
        imageData=vtkImageData::SafeDownCast(_reslice->GetOutput());
#if (VTK_MAJOR_VERSION >= 6)
        assert(0);
#else
        imageData->UpdateInformation();
```

```
#endif
        // Let vtkImageViewer2 handle the slice limits.
        _imageViewer->SetSlice(slice);
        newSlice=GetSlice();
        imageData->GetCenter(center);
        imageData->GetSpacing(spacing);
        imageData->GetOrigin(origin);
        // Compute the position of the center of the slice based on the
        // spacing of the slices.    The resliced axis will always
        // be the "Z" axis.
        point[0]=center[0];
        point[1]=center[1];
        point[2]=(newSlice * spacing[2]) + origin[2];
        point[3]=1.0;
        // Convert the coordinate from the reslice coordinate system to the
        // original image coordinate system.
        // PROBLEM:  Logically this seems like it should have been multiplied
        // by the inverse to translate from the resliced coordinate system to
        // the original coordinate system.    However, multiplying by the inverse
        // sticks the plane in the wrong place completely.    Using the original
        // matrix at least gets the Z coordinate right.
        vtkMatrix4x4* matrix=_reslice->GetResliceAxes();
        vtkSmartPointer<vtkMatrix4x4> inverse =
            vtkSmartPointer<vtkMatrix4x4>::New();
        vtkMatrix4x4::Invert(matrix, inverse);
        matrix->MultiplyPoint(point, newPoint);
        _plane->SetOrigin(newPoint[0], newPoint[1], newPoint[2]);
        // Annotate the image.
        posString << "Position:  (" << newPoint[0] << ", " << newPoint[1]
                  << ", " << newPoint[2] << ")  Slice:  " << newSlice;
        _annotation->SetInput(posString.str().c_str());
        _imageViewer->Render();
    }
    int GetSlice()
    {
        return _imageViewer->GetSlice();
    }
    // Set the orientation of the view.
    void SetOrientation(ResliceRender::ORIENTATION orientation)
    {
        vtkCamera* camera=_imageViewer->GetRenderer()->GetActiveCamera();
        double spacing[3];
        double origin[3];
        double point[4];
        double newPoint[4];
        double initialPosition;
        double xDirCosine[3];
        double yDirCosine[3];
        double zDirCosine[3];
        double normal[3];
        vtkImageData* imageData;
        vtkSmartPointer<vtkMatrix4x4> matrix =
            vtkSmartPointer<vtkMatrix4x4>::New();
        _orientation=orientation;
        // Reset ViewUp
        camera->SetViewUp(0.0, 1.0, 0.0);
        // Compute the cut plane position to the input coordinate system.
        imageData=vtkImageData::SafeDownCast(_reslice->GetInput());
#if (VTK_MAJOR_VERSION >= 6)
        assert(0);
#else
        imageData->UpdateInformation();
#endif
        imageData->GetSpacing(spacing);
        imageData->GetOrigin(origin);
        point[0]=origin[0];
        point[1]=origin[1];
        point[2]=origin[2];
        point[3]=1.0;
        switch (_orientation)
        {
        case AXIAL:
            matrix->DeepCopy(AxialMatrix);
            initialPosition=sphereCenter[2];
            break;
        case CORONAL:
            matrix->DeepCopy(CoronalMatrix);
            initialPosition=sphereCenter[1];
            break;
        case SAGITTAL:
            matrix->DeepCopy(SagittalMatrix);
```

```
            initialPosition=sphereCenter[0];
            break;
        case OBLIQUE:
            matrix->DeepCopy(ObliqueMatrix);
            initialPosition=sphereCenter[2];
            break;
        }
        // Move the origin from the original image coordinate system to the
        // resliced image coordinate system.
        matrix->MultiplyPoint(point, newPoint);
        matrix->SetElement(0, 3, newPoint[0]);
        matrix->SetElement(1, 3, newPoint[1]);
        matrix->SetElement(2, 3, newPoint[2]);
        ResetOrientation();
        SetOrientation(matrix);
        // Compute the cutting plane normal and set it.
        // PROBLEM:  If the transformation is connected rather than
        // using SetResliceAxes, the Direction Cosines do not reflect
        // the orientation of the vtkImageReslice.
        _reslice->GetResliceAxesDirectionCosines(xDirCosine, yDirCosine,
                                                 zDirCosine);
        vtkMath::Cross(xDirCosine, yDirCosine, normal);
        _plane->SetNormal(normal);
        // Set the extents and spacing of the reslice to account for
        // all of the data.
        _reslice->SetOutputExtentToDefault();
        _reslice->SetOutputSpacing(spacing[0], spacing[0], spacing[0]);
        // Force the vtkImageViewer2 to update.
        // PROBLEM:  The whole extent does not seem to be set in time
        // for the first render.    This results in an error because the
        // slice is positioned outside the old bounds.
#if (VTK_MAJOR_VERSION >= 6)
        _imageViewer->SetInputData(NULL);
#else
        _imageViewer->SetInput(NULL);
#endif
        _imageViewer->SetInputConnection(_reslice->GetOutputPort());
        _imageViewer->GetRenderer()->ResetCameraClippingRange();
        _imageViewer->GetRenderer()->ResetCamera();
        // Set the initial slice to be at the center of the sphere.
        // Divide by the spacing because this will be undone in SetSlice.
        SetSlice( (int)(initialPosition / spacing[0]));
    }
    vtkRenderWindowInteractor* GetInteractor()
    {
        return _interactor;
    }
protected:
    ORIENTATION                _orientation;
    //qzDICOMImageReader*       _reader;
    vtkGDCMImageReader*       _reader;
    vtkImageThreshold*         _threshold;
    vtkImageShiftScale*        _shift;
    vtkImageReslice*           _reslice;
    vtkRenderWindowInteractor* _interactor;
    vtkImageViewer2*           _imageViewer;
    vtkSphereSource*           _sphere;
    vtkPolyDataMapper*         _sphereMapper;
    vtkActor*                  _sphereActor;
    vtkPlane*                  _plane;
    vtkCutter*                 _cutter;
    vtkTransform*              _transform;
    vtkTransformPolyDataFilter* _polyTransform;
    vtkPolyDataMapper2D*         _ROIMapper;
    vtkActor2D*                  _ROIActor;
    vtkTextActor*              _annotation;
};
// Catch KeyPress events.
// Up Arrow   - increases the slice
// Down Arrow - decreases the slice
// 'A'        - sets the view to Axial
// 'S'        - sets the view to Sagittal
// 'C'        - sets the view to Coronal
// 'O'        - set the view to Oblique
void KeyCallback::Execute(vtkObject* caller, unsigned long eventId, void *calldata)
{
  (void)caller;
  (void)eventId;
  (void)calldata;
    std::string sym=_reslice->GetInteractor()->GetKeySym();
    if (!sym.compare("Up"))
```

```
    {
        _reslice->SetSlice(_reslice->GetSlice() + 1);
    }
    else if (!sym.compare("Down"))
    {
        _reslice->SetSlice(_reslice->GetSlice() - 1);
    }
    else if ((!sym.compare("A")) || (!sym.compare("a")))
    {
        _reslice->SetOrientation(ResliceRender::AXIAL);
    }
    else if ((!sym.compare("C")) || (!sym.compare("c")))
    {
        _reslice->SetOrientation(ResliceRender::CORONAL);
    }
    else if ((!sym.compare("S")) || (!sym.compare("s")))
    {
        _reslice->SetOrientation(ResliceRender::SAGITTAL);
    }
    else if ((!sym.compare("O")) || (!sym.compare("o")))
    {
        _reslice->SetOrientation(ResliceRender::OBLIQUE);
    }
}
void KeyCallback::SetCallbackData(ResliceRender* reslice)
{
    _reslice=reslice;
}
// Usage:    ResliceSphere [fileName]
int main(int argc, char *argv[])
{
    ResliceRender render;
    if (argc == 1)
      {
      const char *root = gdcm::Testing::GetDataExtraRoot();
      std::string dir3 = root;
      dir3 += "/gdcmSampleData/ForSeriesTesting/Dentist/images/";
      render.CreatePipeline(dir3.c_str());
      }
    else
      {
      render.CreatePipeline(argv[1]);
      }
    render.SetOrientation(ResliceRender::AXIAL);
    render.Start();
    return EXIT_SUCCESS;
}
```

12.164 rtstructapp.cxx

```
/*=========================================================================

Program:  GDCM (Grassroots DICOM). A DICOM library

Copyright (c) 2006-2011 Mathieu Malaterre
All rights reserved.
See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.

This software is distributed WITHOUT ANY WARRANTY; without even
the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
PURPOSE.  See the above copyright notice for more information.

=========================================================================*/
#include "vtkGDCMPolyDataReader.h"
#include "vtkGDCMPolyDataWriter.h"
#include "vtkPolyDataWriter.h"
#include "vtkPolyDataMapper.h"
#include "vtkPolyDataMapper2D.h"
#include "vtkActor2D.h"
#include "vtkRenderWindowInteractor.h"
#include "vtkMedicalImageProperties.h"
#include "vtkRenderWindow.h"
#include "vtkRenderer.h"
#include "vtkCamera.h"
#include "vtkProperty.h"
#include "vtkProperty2D.h"
#include "vtkAppendPolyData.h"
```

```
#include "vtkImageData.h"
/*
* Small example to read in a RTSTUCT and write it out (displays it too).
*/
// gdcmDataExtra/gdcmNonImageData/exRT_Structure_Set_Storage.dcm
// gdcmDataExtra/gdcmNonImageData/RTSTRUCT_1.3.6.1.4.1.22213.1.1396.2.dcm
// gdcmDataExtra/gdcmNonImageData/RT/RTStruct.dcm
int main(int argc, char *argv[])
{
  if( argc < 3 )
    {
    std::cerr « argv[0] « " input.dcm output.dcm\n";
    return 1;
    }
  const char * filename = argv[1];
  const char * outfilename = argv[2];
  vtkGDCMPolyDataReader * reader = vtkGDCMPolyDataReader::New();
  reader->SetFileName( filename );
  reader->Update();
  //std::cout « reader->GetMedicalImageProperties()->GetStudyDate() « std::endl;
  vtkGDCMPolyDataWriter * writer = vtkGDCMPolyDataWriter::New();
  writer->SetNumberOfInputPorts( reader->GetNumberOfOutputPorts() );
  writer->SetFileName( outfilename );
  for(int num = 0; num < reader->GetNumberOfOutputPorts(); ++num )
#if (VTK_MAJOR_VERSION >= 6)
    writer->SetInputConnection( num, reader->GetOutputPort(num) );
#else
    writer->SetInput( num, reader->GetOutput(num) );
#endif
  //doesn't look like the medical properties are actually written out
  writer->SetMedicalImageProperties( reader->GetMedicalImageProperties() );
  writer->SetRTStructSetProperties( reader->GetRTStructSetProperties() );
  writer->Write();
  // print reader output:
  reader->Print( std::cout );
  // print first output:
  reader->GetOutput()->Print( std::cout );
 vtkAppendPolyData *append = vtkAppendPolyData::New();
 int n = reader->GetNumberOfOutputPorts();
 for(int i = 0; i < n; ++i)
   {
#if (VTK_MAJOR_VERSION >= 6)
   append->AddInputConnection( reader->GetOutputPort(i) );
#else
   append->AddInput( reader->GetOutput(i) );
#endif
   }
  // Now we'll look at it.
  vtkPolyDataMapper *cubeMapper = vtkPolyDataMapper::New();
#if (VTK_MAJOR_VERSION >= 6)
  cubeMapper->SetInputConnection( append->GetOutputPort());
#else
  cubeMapper->SetInput( append->GetOutput());
#endif
  cubeMapper->SetScalarRange(0,7);
  vtkActor *cubeActor = vtkActor::New();
  cubeActor->SetMapper(cubeMapper);
  vtkProperty * property = cubeActor->GetProperty();
  property->SetRepresentationToWireframe();
  vtkRenderer *renderer = vtkRenderer::New();
  vtkRenderWindow *renWin = vtkRenderWindow::New();
  renWin->AddRenderer(renderer);
  vtkRenderWindowInteractor *iren = vtkRenderWindowInteractor::New();
  iren->SetRenderWindow(renWin);
  renderer->AddActor(cubeActor);
  renderer->ResetCamera();
  renderer->SetBackground(1,1,1);
  renWin->SetSize(300,300);
  renWin->Render();
  iren->Start();
  reader->Delete();
  append->Delete();
  cubeMapper->Delete();
  cubeActor->Delete();
  renderer->Delete();
  renWin->Delete();
  iren->Delete();
  writer->Delete();
  return 0;
}
```

12.165 threadgdcm.cxx

```
/*=========================================================================

Program:  GDCM (Grassroots DICOM). A DICOM library

Copyright (c) 2006-2011 Mathieu Malaterre
All rights reserved.
See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.

This software is distributed WITHOUT ANY WARRANTY; without even
the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
PURPOSE.  See the above copyright notice for more information.

=========================================================================*/
#include "gdcmReader.h"
#include "gdcmImageReader.h"
#include "gdcmDirectory.h"
#include "gdcmSystem.h"
#include "vtkImageData.h"
#include "vtkStructuredPointsWriter.h"
#include <pthread.h>
struct threadparams
{
  const char **filenames;
  size_t nfiles;
  char *scalarpointer;
// TODO I should also pass in the dim of the reference image just in case
};
void *ReadFilesThread(void *voidparams)
{
  const threadparams *params = static_cast<const threadparams *> (voidparams);
  const size_t nfiles = params->nfiles;
  for(unsigned int file = 0; file < nfiles; ++file)
    {
    /*
// TODO: update progress
pthread_mutex_lock(&params->lock);
//section critique
ReadingProgress+=params->stepProgress;
pthread_mutex_unlock(&params->lock);
*/
    const char *filename = params->filenames[file];
    //std::cerr « filename « std::endl;
    gdcm::ImageReader reader;
    reader.SetFileName( filename );
    try
      {
      if( !reader.Read() )
        {
        std::cerr « "Failed to read:  " « filename « std::endl;
        break;
        }
      }
    catch( ...  )
      {
      std::cerr « "Failed to read:  " « filename « std::endl;
      break;
      }
    const gdcm::Image &image = reader.GetImage();
    unsigned long len = image.GetBufferLength();
    char * pointer = params->scalarpointer;
#if 0
    char *tempimage = new char[len];
    image.GetBuffer(tempimage);
    memcpy(pointer + file*len, tempimage, len);
    delete[] tempimage;
#else
    char *tempimage = pointer + file * len;
    image.GetBuffer(tempimage);
#endif
    }
  return voidparams;
}
void ShowFilenames(const threadparams &params)
{
  std::cout « "start" « std::endl;
  for(unsigned int i = 0; i < params.nfiles; ++i)
    {
    const char *filename = params.filenames[i];
```

```
    std::cout « filename « std::endl;
    }
  std::cout « "end" « std::endl;
}
void ReadFiles(size_t nfiles, const char *filenames[])
{
  // \precondition:  nfiles > 0
  assert( nfiles > 0 );
  const char *reference= filenames[0]; // take the first image as reference
  gdcm::ImageReader reader;
  reader.SetFileName( reference );
  if( !reader.Read() )
    {
    // That would be very bad...
    assert(0);
    }
  const gdcm::Image &image = reader.GetImage();
  gdcm::PixelFormat pixeltype = image.GetPixelFormat();
  unsigned long len = image.GetBufferLength();
  const unsigned int *dims = image.GetDimensions();
  unsigned short pixelsize = pixeltype.GetPixelSize();
  (void)pixelsize;
  assert( image.GetNumberOfDimensions() == 2 );
  vtkImageData *output = vtkImageData::New();
  output->SetDimensions(dims[0], dims[1], (int)nfiles);
#if (VTK_MAJOR_VERSION >= 6)
  int numscal = pixeltype.GetSamplesPerPixel();
  switch( pixeltype )
    {
  case gdcm::PixelFormat::INT8:
    output->AllocateScalars( VTK_SIGNED_CHAR, numscal );
    break;
  case gdcm::PixelFormat::UINT8:
    output->AllocateScalars( VTK_UNSIGNED_CHAR, numscal );
    break;
  case gdcm::PixelFormat::INT16:
    output->AllocateScalars( VTK_SHORT, numscal );
    break;
  case gdcm::PixelFormat::UINT16:
    output->AllocateScalars( VTK_UNSIGNED_SHORT, numscal );
    break;
  case gdcm::PixelFormat::INT32:
    output->AllocateScalars( VTK_INT, numscal );
    break;
  case gdcm::PixelFormat::UINT32:
    output->AllocateScalars( VTK_UNSIGNED_INT, numscal );
    break;
  default:
    assert(0);
    }
#else
  switch( pixeltype )
    {
  case gdcm::PixelFormat::INT8:
#if (VTK_MAJOR_VERSION >= 5) || ( VTK_MAJOR_VERSION == 4 && VTK_MINOR_VERSION > 5 )
    output->SetScalarType ( VTK_SIGNED_CHAR );
#else
    output->SetScalarType ( VTK_CHAR );
#endif
    break;
  case gdcm::PixelFormat::UINT8:
    output->SetScalarType ( VTK_UNSIGNED_CHAR );
    break;
  case gdcm::PixelFormat::INT16:
    output->SetScalarType ( VTK_SHORT );
    break;
  case gdcm::PixelFormat::UINT16:
    output->SetScalarType ( VTK_UNSIGNED_SHORT );
    break;
  case gdcm::PixelFormat::INT32:
    output->SetScalarType ( VTK_INT );
    break;
  case gdcm::PixelFormat::UINT32:
    output->SetScalarType ( VTK_UNSIGNED_INT );
    break;
  default:
    assert(0);
    }
  output->SetNumberOfScalarComponents ( pixeltype.GetSamplesPerPixel() );
  output->AllocateScalars();
#endif
```

```
  char * scalarpointer = static_cast<char*>(output->GetScalarPointer());
  const unsigned int nthreads = 4;
  threadparams params[nthreads];
  //pthread_mutex_t lock;
  //pthread_mutex_init(&lock, NULL);
  pthread_t *pthread = new pthread_t[nthreads];
  // There is nfiles, and nThreads
  assert( nfiles > nthreads );
  const size_t partition = nfiles / nthreads;
  for (unsigned int thread=0; thread < nthreads; ++thread)
    {
    params[thread].filenames = filenames + thread * partition;
    params[thread].nfiles = partition;
    if( thread == nthreads - 1 )
      {
      // There is slightly more files to process in this thread:
      params[thread].nfiles += nfiles % nthreads;
      }
    assert( thread * partition < nfiles );
    params[thread].scalarpointer = scalarpointer + thread * partition * len;
    //assert( params[thread].scalarpointer < scalarpointer + 2 * dims[0] * dims[1] * dims[2] );
    // start thread:
    int res = pthread_create( &pthread[thread], NULL, ReadFilesThread, &params[thread]);
    if( res )
      {
      std::cerr << "Unable to start a new thread, pthread returned:  " << res << std::endl;
      assert(0);
      }
    //ShowFilenames(params[thread]);
    }
// DEBUG
  size_t total = 0;
  for (unsigned int thread=0; thread < nthreads; ++thread)
    {
    total += params[thread].nfiles;
    }
  assert( total == nfiles );
// END DEBUG
  for (unsigned int thread=0;thread<nthreads;thread++)
    {
    pthread_join( pthread[thread], NULL);
    }
  delete[] pthread;
  //pthread_mutex_destroy(&lock);
  // For some reason writing down the file is painfully slow...
  vtkStructuredPointsWriter *writer = vtkStructuredPointsWriter::New();
#if (VTK_MAJOR_VERSION >= 6)
  writer->SetInputData( output );
#else
  writer->SetInput( output );
#endif
  writer->SetFileName( "/tmp/threadgdcm.vtk" );
  writer->SetFileTypeToBinary();
  //writer->Write();
  writer->Delete();
  //output->Print( std::cout );
  output->Delete();
}
int main(int argc, char *argv[])
{
  if( argc < 2 )
    {
    std::cerr << argv[0] << " [directory|list of filenames]\n";
    return 1;
    }
  // Check if user pass in a single directory
  if( argc == 2 && gdcm::System::FileIsDirectory( argv[1] ) )
    {
    gdcm::Directory d;
    d.Load( argv[1] );
    gdcm::Directory::FilenamesType l = d.GetFilenames();
    const size_t nfiles = l.size();
    const char **filenames = new const char* [ nfiles ];
    for(unsigned int i = 0; i < nfiles; ++i)
      {
      filenames[i] = l[i].c_str();
      }
    ReadFiles(nfiles, filenames);
    delete[] filenames;
    }
  else
```

```
    {
    // Simply copy all filenames into the vector:
    const char **filenames = const_cast<const char**>(argv+1);
    const size_t nfiles = argc - 1;
    ReadFiles(nfiles, filenames);
    }
  return 0;
}
```

12.166 AWTMedical3.java

```
/*=========================================================================

Program:  GDCM (Grassroots DICOM). A DICOM library

Copyright (c) 2006-2011 Mathieu Malaterre
All rights reserved.
See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.

This software is distributed WITHOUT ANY WARRANTY; without even
the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
PURPOSE.  See the above copyright notice for more information.

=========================================================================*/
package examples;
import vtk.*;
//import gdcm.*;
import vtk.util.VtkPanelContainer;
import vtk.util.VtkPanelUtil;
import vtk.util.VtkUtil;
import java.util.ArrayList;
import javax.swing.*;
import java.awt.*;
import java.io.File;
public class AWTMedical3  extends JComponent implements VtkPanelContainer {
  private vtkPanel renWin;
    vtkImageData ReadDataFile(File inSelectedFile){
        vtkImageData outImageData = null;
        Directory theDir = new Directory();
        String theInputDirectory = inSelectedFile.getPath();
        theDir.Load(theInputDirectory);
        Scanner theScanner = new Scanner();
        Tag theStudyTag = new Tag(0x0020,0x000d);
        Tag theSeriesTag = new Tag(0x0020,0x000e);
        theScanner.AddTag(theStudyTag);//get studies,
        theScanner.AddTag(theSeriesTag);//get studies,
        theScanner.Scan(theDir.GetFilenames());
        FilenamesType theStudyValues = theScanner.GetOrderedValues(theStudyTag);
        long theNumStudies = theStudyValues.size();
        //for now, take the first study, and nothing else.
        //and the return is actually not FilenamesType, just a
        //vector of strings
        if (theNumStudies != 1)
            return outImageData;
        String theStudyVal = theStudyValues.get(0);
        //now, get all the values from the scanner that are in that
        //study, then from that get their different series
        FilenamesType theFilenames =
                theScanner.GetAllFilenamesFromTagToValue(theStudyTag, theStudyVal);
        //from that set of filenames, isolate individual series
        //conclude that singleton series = RT struct (can do further
        //checking for things like MIPs and the like)
        //and multiple series entries = volumetric data
        theScanner.Scan(theFilenames);
        FilenamesType theSeriesValues = theScanner.GetOrderedValues(theSeriesTag);
        String studyUID = theScanner.GetValue(theScanner.GetFilenames().get(0), theStudyTag);
        long theNumSeries = theSeriesValues.size();
        for (int i = 0; i < theNumSeries; i++) {
            FilenamesType theSeriesFiles =
                theScanner.GetAllFilenamesFromTagToValue(theSeriesTag, theSeriesValues.get(i));
            long theNumFilesInSeries = theSeriesFiles.size();
            if (theNumFilesInSeries > 1) {//assume it's CT or volumetric data
                //for now, assume a single volume
                //could have multiples, like PET and CT
                IPPSorter sorter = new IPPSorter();
                sorter.SetComputeZSpacing(true);
                sorter.SetZSpacingTolerance(0.001);
```

```
                Boolean sorted = sorter.Sort(theSeriesFiles);
                if (!sorted){
                    //need some better way to handle failures here
                    return outImageData;
                }
                FilenamesType sortedFT = sorter.GetFilenames();
                long theSize = sortedFT.size();
                vtkStringArray sa = new vtkStringArray();
                ArrayList<String> theStrings = new ArrayList<String>();
                vtkGDCMImageReader gdcmReader = new vtkGDCMImageReader();
                for (int j = 0; j < theSize; j++) {
                    String theFileName = sortedFT.get(j);
                    if (gdcmReader.CanReadFile(theFileName) > 0){
                        theStrings.add(theFileName);
                        sa.InsertNextValue(theFileName);
                    } else {
                        //this is a busted series
                        //need some more appropriate error here
                        return outImageData;
                    }
                }
                gdcmReader.SetFileNames(sa);
                gdcmReader.Update();
                outImageData = gdcmReader.GetOutput();//the zeroth output should be the image
            }
        }
        String theImageInfo = "";
        if (outImageData != null){
            theImageInfo = outImageData.Print();
        }
        return outImageData;
    }
    //this function is a rewrite of Medical3 to see if data can
    //be loaded via gdcm easily
  public AWTMedical3(File inFile) {
    // Create the buttons.
    renWin = new vtkPanel();
    vtkImageData theImageData = ReadDataFile(inFile);
    // An isosurface, or contour value of 500 is known to correspond to the
    // skin of the patient.  Once generated, a vtkPolyDataNormals filter is
    // is used to create normals for smooth surface shading during rendering.
    // The triangle stripper is used to create triangle strips from the
    // isosurface these render much faster on some systems.
    vtkContourFilter skinExtractor = new vtkContourFilter();
    skinExtractor.SetInput(theImageData);
    skinExtractor.SetValue(0, 500);
    vtkPolyDataNormals skinNormals = new vtkPolyDataNormals();
    skinNormals.SetInput(skinExtractor.GetOutput());
    skinNormals.SetFeatureAngle(60.0);
//        vtkStripper skinStripper = new vtkStripper();
//        skinStripper.SetInput(skinNormals.GetOutput());
    vtkPolyDataMapper skinMapper = new vtkPolyDataMapper();
    skinMapper.SetInput(skinNormals.GetOutput());
    skinMapper.ScalarVisibilityOff();
    vtkActor skin = new vtkActor();
    skin.SetMapper(skinMapper);
    skin.GetProperty().SetDiffuseColor(1, .49, .25);
    skin.GetProperty().SetSpecular(.3);
    skin.GetProperty().SetSpecularPower(20);
    // An isosurface, or contour value of 1150 is known to correspond to the
    // skin of the patient.  Once generated, a vtkPolyDataNormals filter is
    // is used to create normals for smooth surface shading during rendering.
    // The triangle stripper is used to create triangle strips from the
    // isosurface these render much faster on some systems.
    vtkContourFilter boneExtractor = new vtkContourFilter();
    boneExtractor.SetInput(theImageData);
    boneExtractor.SetValue(0, 1150);
    vtkPolyDataNormals boneNormals = new vtkPolyDataNormals();
    boneNormals.SetInput(boneExtractor.GetOutput());
    boneNormals.SetFeatureAngle(60.0);
    vtkStripper boneStripper = new vtkStripper();
    boneStripper.SetInput(boneNormals.GetOutput());
    vtkPolyDataMapper boneMapper = new vtkPolyDataMapper();
    boneMapper.SetInput(boneStripper.GetOutput());
    boneMapper.ScalarVisibilityOff();
    vtkActor bone = new vtkActor();
    bone.SetMapper(boneMapper);
    bone.GetProperty().SetDiffuseColor(1, 1, .9412);
    // An outline provides context around the data.
    vtkOutlineFilter outlineData = new vtkOutlineFilter();
    outlineData.SetInput(theImageData);
```

```
vtkPolyDataMapper mapOutline = new vtkPolyDataMapper();
mapOutline.SetInput(outlineData.GetOutput());
vtkActor outline = new vtkActor();
outline.SetMapper(mapOutline);
outline.GetProperty().SetColor(0, 0, 0);
// Now we are creating three orthogonal planes passing through the
// volume.  Each plane uses a different texture map and therefore has
// diferent coloration.
// Start by creatin a black/white lookup table.
vtkLookupTable bwLut = new vtkLookupTable();
bwLut.SetTableRange(0, 2000);
bwLut.SetSaturationRange(0, 0);
bwLut.SetHueRange(0, 0);
bwLut.SetValueRange(0, 1);
bwLut.Build();
// Now create a lookup table that consists of the full hue circle (from
// HSV);.
vtkLookupTable hueLut = new vtkLookupTable();
hueLut.SetTableRange(0, 2000);
hueLut.SetHueRange(0, 1);
hueLut.SetSaturationRange(1, 1);
hueLut.SetValueRange(1, 1);
hueLut.Build();
// Finally, create a lookup table with a single hue but having a range
// in the saturation of the hue.
vtkLookupTable satLut = new vtkLookupTable();
satLut.SetTableRange(0, 2000);
satLut.SetHueRange(.6, .6);
satLut.SetSaturationRange(0, 1);
satLut.SetValueRange(1, 1);
satLut.Build();
// Create the first of the three planes.  The filter vtkImageMapToColors
// maps the data through the corresponding lookup table created above.
// The vtkImageActor is a type of vtkProp and conveniently displays an
// image on a single quadrilateral plane.  It does this using texture
// mapping and as a result is quite fast.  (Note:  the input image has to
// be unsigned char values, which the vtkImageMapToColors produces.);
// Note also that by specifying the DisplayExtent, the pipeline
// requests data of this extent and the vtkImageMapToColors only
// processes a slice of data.
vtkImageMapToColors saggitalColors = new vtkImageMapToColors();
saggitalColors.SetInput(theImageData);
saggitalColors.SetLookupTable(bwLut);
vtkImageActor saggital = new vtkImageActor();
saggital.SetInput(saggitalColors.GetOutput());
saggital.SetDisplayExtent(32, 32, 0, 63, 0, 92);
// Create the second (axial); plane of the three planes.  We use the same
// approach as before except that the extent differs.
vtkImageMapToColors axialColors = new vtkImageMapToColors();
axialColors.SetInput(theImageData);
axialColors.SetLookupTable(hueLut);
vtkImageActor axial = new vtkImageActor();
axial.SetInput(axialColors.GetOutput());
axial.SetDisplayExtent(0, 63, 0, 63, 46, 46);
// Create the third (coronal); plane of the three planes.  We use the same
// approach as before except that the extent differs.
vtkImageMapToColors coronalColors = new vtkImageMapToColors();
coronalColors.SetInput(theImageData);
coronalColors.SetLookupTable(satLut);
vtkImageActor coronal = new vtkImageActor();
coronal.SetInput(coronalColors.GetOutput());
coronal.SetDisplayExtent(0, 63, 32, 32, 0, 92);
// It is convenient to create an initial view of the data.  The FocalPoint
// and Position form a vector direction.  Later on (ResetCamera() method)
// this vector is used to position the camera to look at the data in
// this direction.
vtkCamera aCamera = new vtkCamera();
aCamera.SetViewUp(0, 0, -1);
aCamera.SetPosition(0, 1, 0);
aCamera.SetFocalPoint(0, 0, 0);
aCamera.ComputeViewPlaneNormal();
// Actors are added to the renderer.  An initial camera view is created.
// The Dolly() method moves the camera towards the FocalPoint,
// thereby enlarging the image.
renWin.GetRenderer().AddActor(saggital);
renWin.GetRenderer().AddActor(axial);
renWin.GetRenderer().AddActor(coronal);
renWin.GetRenderer().AddActor(outline);
renWin.GetRenderer().AddActor(skin);
renWin.GetRenderer().AddActor(bone);
// Turn off bone for this example.
```

```
    bone.VisibilityOff();
    // Set skin to semi-transparent.
    skin.GetProperty().SetOpacity(0.5);
    // An initial camera view is created.    The Dolly() method moves
    // the camera towards the FocalPoint, thereby enlarging the image.
    renWin.GetRenderer().SetActiveCamera(aCamera);
    renWin.GetRenderer().ResetCamera();
    aCamera.Dolly(1.5);
    // Set a background color for the renderer and set the size of the
    // render window (expressed in pixels).
    renWin.GetRenderer().SetBackground(1, 1, 1);
    VtkPanelUtil.setSize(renWin, 640, 480);
    // Note that when camera movement occurs (as it does in the Dolly()
    // method), the clipping planes often need adjusting.  Clipping planes
    // consist of two planes:  near and far along the view direction.  The
    // near plane clips out objects in front of the plane the far plane
    // clips out objects behind the plane.  This way only what is drawn
    // between the planes is actually rendered.
    renWin.GetRenderer().ResetCameraClippingRange();
    // Setup panel
    setLayout(new BorderLayout());
    add(renWin, BorderLayout.CENTER);
  }
  public vtkPanel getRenWin() {
    return renWin;
  }
  public static void main(String s[]) {
    if (s.length == 0){
      return; //need a filename here
    }
    File theFile = new File(s[0]);
    //File theFile = new
      File("/Users/mmroden/Documents/MVSDownloadDirectory/Documents/1.2.840.113704.1.111.3384.1271766367.5/");
    AWTMedical3 panel = new AWTMedical3(theFile);
    JFrame frame = new JFrame("AWTMedical3");
    frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
    frame.getContentPane().add("Center", panel);
    frame.pack();
    frame.setVisible(true);
  }
}
```

12.167 HelloVTKWorld.java

```
/*=========================================================================

Program:  GDCM (Grassroots DICOM). A DICOM library

Copyright (c) 2006-2011 Mathieu Malaterre
All rights reserved.
See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.

This software is distributed WITHOUT ANY WARRANTY; without even
the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
PURPOSE.  See the above copyright notice for more information.

=========================================================================*/
// We are required to call the package 'vtk' eventhough I (MM) would have prefered
// an import statement along the line of:
// import vtkgdcm.*;
import vtk.*;
/*
* Compilation:
* CLASSPATH=vtkgdcm.jar:/usr/share/java/vtk.jar javac HelloVTKWorld.java
*
* Usage:
* LD_LIBRARY_PATH=/usr/lib/jvm/java-6-openjdk/jre/lib/amd64/xawt:/usr/lib/jni:.
     CLASSPATH=/usr/share/java/vtk.jar:vtkgdcm.jar:gdcm.jar:.  java HelloVTKWorld gdcmData/012345.002.050.dcm
     bla.dcm
*
*/
public class HelloVTKWorld
{
  static {
    System.loadLibrary("vtkCommonJava");
    System.loadLibrary("vtkFilteringJava");
    System.loadLibrary("vtkIOJava");
```

```
    System.loadLibrary("vtkImagingJava");
    System.loadLibrary("vtkGraphicsJava");
    System.loadLibrary("vtkgdcmJava");
    try {
      System.loadLibrary("vtkRenderingJava");
    } catch (Throwable e) {
      System.out.println("cannot load vtkHybrid, skipping...");
    }
    try {
      System.loadLibrary("vtkHybridJava");
    } catch (Throwable e) {
      System.out.println("cannot load vtkHybrid, skipping...");
    }
    try {
      System.loadLibrary("vtkVolumeRenderingJava");
    } catch (Throwable e) {
      System.out.println("cannot load vtkVolumeRendering, skipping...");
    }
  }
  public static void main(String[] args)
    {
    String filename = args[0];
    vtkGDCMImageReader reader = new vtkGDCMImageReader();
    reader.SetFileName( filename );
    reader.Update();
    vtkMedicalImageProperties prop = reader.GetMedicalImageProperties();
    System.out.println( prop.GetPatientName() ); //
//    if( reader.GetImageFormat() == vtkgdcm.vtkgdcm.VTK_LUMINANCE ) // MONOCHROME2
//      {
//      System.out.println( "Image is MONOCHROME2" ); //
//      }
    // Just for fun, invert the direction cosines, output should reflect that:
    vtkMatrix4x4 dircos = reader.GetDirectionCosines();
    dircos.Invert();
    // We need to maintain in sync information stored in vtkMedicalImageProperties:
    double[] cosines = new double[6];
    cosines[0] = dircos.GetElement(0,0);
    cosines[1] = dircos.GetElement(1,0);
    cosines[2] = dircos.GetElement(2,0);
    cosines[3] = dircos.GetElement(0,1);
    cosines[4] = dircos.GetElement(1,1);
    cosines[5] = dircos.GetElement(2,1);
    reader.GetMedicalImageProperties().SetDirectionCosine( cosines );
    String outfilename = args[1];
    vtkGDCMImageWriter writer = new vtkGDCMImageWriter();
    writer.SetMedicalImageProperties( reader.GetMedicalImageProperties() );
    writer.SetDirectionCosines( dircos );
    writer.SetShift( reader.GetShift() );
    writer.SetScale( reader.GetScale() );
    writer.SetImageFormat( reader.GetImageFormat() );
    writer.SetFileName( outfilename );
    writer.SetInputConnection( reader.GetOutputPort() ); // new
    //writer.SetInput( reader.GetOutput() ); // old
    writer.Write();
    System.out.println("Success reading:  " + filename );
    }
}
```

12.168 MIPViewer.java

```
/*=========================================================================

Program:  GDCM (Grassroots DICOM). A DICOM library

Copyright (c) 2006-2011 Mathieu Malaterre
All rights reserved.
See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.

This software is distributed WITHOUT ANY WARRANTY; without even
the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
PURPOSE.  See the above copyright notice for more information.

=========================================================================*/
import vtk.*;
import gdcm.*;
import java.io.File;
import java.awt.Canvas;
```

```
/*
* Compilation:
* CLASSPATH=vtkgdcm.jar:/usr/share/java/vtk.jar javac MIPViewer.java
*
* Usage:
* LD_LIBRARY_PATH=/usr/lib/jvm/java-6-openjdk/jre/lib/amd64/xawt:/usr/lib/jni:.
      CLASSPATH=/usr/share/java/vtk.jar:vtkgdcm.jar:gdcm.jar:.  java MIPViewer BRAINX
*
*/
public class MIPViewer extends Canvas
{
  static {
    // VTK
    System.loadLibrary("vtkCommonJava");
    System.loadLibrary("vtkFilteringJava");
    System.loadLibrary("vtkIOJava");
    System.loadLibrary("vtkImagingJava");
    System.loadLibrary("vtkGraphicsJava");
    System.loadLibrary("vtkRenderingJava");
    System.loadLibrary("vtkVolumeRenderingJava"); // vtkSmartVolumeMapper
    System.loadLibrary("vtkWidgetsJava"); // vtkBoxWidget
    // VTK-GDCM
    System.loadLibrary("vtkgdcmJava");
  }
  static FilenamesType fns = new FilenamesType();
  protected native int Lock();
  protected native int UnLock();
  public static void process(String path)
    {
    fns.add( path );
    }
  // Process only files under dir
  public static void visitAllFiles(File dir)
    {
    if (dir.isDirectory())
      {
      String[] children = dir.list();
      for (int i=0; i<children.length; i++)
        {
        visitAllFiles(new File(dir, children[i]));
        }
      }
    else
      {
      process(dir.getPath());
      }
    }
  public static void main(String[] args) throws Exception
    {
    String dirname = args[0];
    if( !PosixEmulation.FileIsDirectory( dirname ) )
      {
      return;
      }
    File dir = new File(dirname);
    visitAllFiles(dir);
    IPPSorter ipp = new IPPSorter();
    ipp.SetComputeZSpacing( true );
    ipp.SetZSpacingTolerance( 1e-3 );
    boolean b = ipp.Sort( fns );
    if(!b)
      {
      throw new Exception("Could not scan");
      }
    double ippzspacing = ipp.GetZSpacing();
    FilenamesType sorted = ipp.GetFilenames();
    vtkStringArray files = new vtkStringArray();
    long nfiles = sorted.size();
    //for( String f :  sorted )
    for (int i = 0; i < nfiles; i++) {
      String f = sorted.get(i);
      files.InsertNextValue( f );
      }
    vtkGDCMImageReader reader = new vtkGDCMImageReader();
    reader.SetFileNames( files );
    reader.Update(); // get spacing value
    double[] spacing = reader.GetOutput().GetSpacing();
    vtkImageChangeInformation change = new vtkImageChangeInformation();
    change.SetInputConnection( reader.GetOutputPort() );
    change.SetOutputSpacing( spacing[0], spacing[1], ippzspacing );
    // Create our volume and mapper
```

```
    vtkVolume volume = new vtkVolume();
    vtkSmartVolumeMapper mapper = new vtkSmartVolumeMapper();
    vtkRenderWindowInteractor iren = new vtkRenderWindowInteractor();
    // Add a box widget if the clip option was selected
    vtkBoxWidget box = new vtkBoxWidget();
    box.SetInteractor(iren);
    box.SetPlaceFactor(1.01);
    box.SetInputConnection(change.GetOutputPort());
    //box.SetDefaultRenderer(renderer);
    box.InsideOutOn();
    box.PlaceWidget();
    //vtkBoxWidgetCallback callback = vtkBoxWidgetCallback::New();
    //callback.SetMapper(mapper);
    //box.AddObserver(vtkCommand::InteractionEvent, callback);
    //callback.Delete();
//    Lock();
//    box.EnabledOn();
//    UnLock();
    box.GetSelectedFaceProperty().SetOpacity(0.0);
    mapper.SetInputConnection( change.GetOutputPort() );
    // Create our transfer function
    vtkColorTransferFunction colorFun = new vtkColorTransferFunction();
    vtkPiecewiseFunction opacityFun = new vtkPiecewiseFunction();
    // Create the property and attach the transfer functions
    vtkVolumeProperty property = new vtkVolumeProperty();
    property.IndependentComponentsOn();
    property.SetColor( colorFun );
    property.SetScalarOpacity( opacityFun );
    property.SetInterpolationTypeToLinear();
    // connect up the volume to the property and the mapper
    volume.SetProperty( property );
    volume.SetMapper( mapper );
    vtkMedicalImageProperties medprop = reader.GetMedicalImageProperties();
    int n = medprop.GetNumberOfWindowLevelPresets();
    double opacityWindow = 4096;
    double opacityLevel = 2048;
    // Override default with value from DICOM files:
    for( int i = 0; i < n; ++i )
      {
      double wl[] = medprop.GetNthWindowLevelPreset(i);
      //System.out.println( "W/L: " + wl[0] + " " + wl[1] );
      opacityWindow = wl[0];
      opacityLevel  = wl[1];
      }
    colorFun.AddRGBSegment(0.0, 1.0, 1.0, 1.0, 255.0, 1.0, 1.0, 1.0 );
    opacityFun.AddSegment( opacityLevel - 0.5*opacityWindow, 0.0,
      opacityLevel + 0.5*opacityWindow, 1.0 );
    mapper.SetBlendModeToMaximumIntensity();
    // Create the RenderWindow, Renderer
    vtkRenderer ren1 = new vtkRenderer();
    vtkRenderWindow renWin = new vtkRenderWindow();
    renWin.AddRenderer(ren1);
    // Set the default window size
    renWin.SetSize(600,600);
    // Add the volume to the scene
    ren1.AddVolume( volume );
    ren1.ResetCamera();
    iren.SetRenderWindow( renWin );
    // interact with data
    renWin.Render();
    iren.Start();
    }
}
```

12.169 MPRViewer.java

```
/*=========================================================================

Program:  GDCM (Grassroots DICOM). A DICOM library

Copyright (c) 2006-2011 Mathieu Malaterre
All rights reserved.
See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.

This software is distributed WITHOUT ANY WARRANTY; without even
the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
PURPOSE.  See the above copyright notice for more information.
```

```
=========================================================================*/
import vtk.*;
import gdcm.*;
import java.io.File;
/*
* Compilation:
* CLASSPATH=vtkgdcm.jar:/usr/share/java/vtk.jar javac MPRViewer.java
*
* Usage:
* LD_LIBRARY_PATH=/usr/lib/jvm/java-6-openjdk/jre/lib/amd64/xawt:/usr/lib/jni:.
     CLASSPATH=/usr/share/java/vtk.jar:vtkgdcm.jar:gdcm.jar:.  java MPRViewer BRAINX
*
*/
public class MPRViewer
{
  static {
    // VTK
    System.loadLibrary("vtkCommonJava");
    System.loadLibrary("vtkFilteringJava");
    System.loadLibrary("vtkIOJava");
    System.loadLibrary("vtkImagingJava");
    System.loadLibrary("vtkGraphicsJava");
    System.loadLibrary("vtkRenderingJava");
    // VTK-GDCM
    System.loadLibrary("vtkgdcmJava");
  }
  static FilenamesType fns = new FilenamesType();
  public static void process(String path)
    {
    fns.add( path );
    }
  // Process only files under dir
  public static void visitAllFiles(File dir)
    {
    if (dir.isDirectory())
      {
      String[] children = dir.list();
      for (int i=0; i<children.length; i++)
        {
        visitAllFiles(new File(dir, children[i]));
        }
      }
    else
      {
      process(dir.getPath());
      }
    }
  public static void main(String[] args) throws Exception
    {
    String dirname = args[0];
    if( !PosixEmulation.FileIsDirectory( dirname ) )
      {
      return;
      }
    File dir = new File(dirname);
    visitAllFiles(dir);
    IPPSorter ipp = new IPPSorter();
    ipp.SetComputeZSpacing( true );
    ipp.SetZSpacingTolerance( 1e-3 );
    boolean b = ipp.Sort( fns );
    if(!b)
      {
      throw new Exception("Could not scan");
      }
    double ippzspacing = ipp.GetZSpacing();
    FilenamesType sorted = ipp.GetFilenames();
    vtkStringArray files = new vtkStringArray();
    long nfiles = sorted.size();
    //for( String f :  sorted )
    for (int i = 0; i < nfiles; i++) {
      String f = sorted.get(i);
      files.InsertNextValue( f );
      }
    vtkGDCMImageReader reader = new vtkGDCMImageReader();
    reader.SetFileNames( files );
    reader.Update(); // get spacing value
    double[] spacing = reader.GetOutput().GetSpacing();
    vtkImageChangeInformation change = new vtkImageChangeInformation();
    change.SetInputConnection( reader.GetOutputPort() );
    change.SetOutputSpacing( spacing[0], spacing[1], ippzspacing );
```

```
    // A simple vtkInteractorStyleImage example for
    // 3D image viewing with the vtkImageResliceMapper.
    //
    // Drag Left mouse button to window/level
    // Shift-Left drag to rotate (oblique slice)
    // Shift-Middle drag to slice through image
    // OR Ctrl-Right drag to slice through image
    // Create the RenderWindow, Renderer
    vtkRenderer ren1 = new vtkRenderer();
    vtkRenderWindow renWin = new vtkRenderWindow();
    renWin.AddRenderer(ren1);
    vtkImageResliceMapper im = new vtkImageResliceMapper();
    im.SetInputConnection(change.GetOutputPort());
    im.SliceFacesCameraOn();
    im.SliceAtFocalPointOn();
    im.BorderOff();
    vtkImageProperty ip = new vtkImageProperty();
    ip.SetColorWindow(2000);
    ip.SetColorLevel(1000);
    ip.SetAmbient(0.0);
    ip.SetDiffuse(1.0);
    ip.SetOpacity(1.0);
    ip.SetInterpolationTypeToLinear();
    vtkImageSlice ia = new vtkImageSlice();
    ia.SetMapper(im);
    ia.SetProperty(ip);
    ren1.AddViewProp(ia);
    ren1.SetBackground(0.1,0.2,0.4);
    renWin.SetSize(300,300);
    vtkRenderWindowInteractor iren = new vtkRenderWindowInteractor();
    vtkInteractorStyleImage style = new vtkInteractorStyleImage();
    style.SetInteractionModeToImage3D();
    iren.SetInteractorStyle(style);
    renWin.SetInteractor(iren);
    // render the image
    renWin.Render();
    vtkCamera cam1 = ren1.GetActiveCamera();
    cam1.ParallelProjectionOn();
    ren1.ResetCameraClippingRange();
    renWin.Render();
    iren.Start();
    }
}
```

12.170 MPRViewer2.java

```
/*=========================================================================

Program:  GDCM (Grassroots DICOM). A DICOM library

Copyright (c) 2006-2011 Mathieu Malaterre
All rights reserved.
See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.

This software is distributed WITHOUT ANY WARRANTY; without even
the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
PURPOSE.  See the above copyright notice for more information.

=========================================================================*/
import vtk.*;
import gdcm.*;
import java.io.File;
/*
* Compilation:
* CLASSPATH=vtkgdcm.jar:/usr/share/java/vtk.jar javac MPRViewer2.java
*
* Usage:
* LD_LIBRARY_PATH=/usr/lib/jvm/java-6-openjdk/jre/lib/amd64/xawt:/usr/lib/jni:.
     CLASSPATH=/usr/share/java/vtk.jar:vtkgdcm.jar:gdcm.jar:.  java MPRViewer2 BRAINX
*
*/
public class MPRViewer2
{
  static {
    // VTK
    System.loadLibrary("vtkCommonJava");
    System.loadLibrary("vtkFilteringJava");
```

```
    System.loadLibrary("vtkIOJava");
    System.loadLibrary("vtkImagingJava");
    System.loadLibrary("vtkGraphicsJava");
    System.loadLibrary("vtkRenderingJava");
    System.loadLibrary("vtkHybridJava");
    System.loadLibrary("vtkWidgetsJava");
    // VTK-GDCM
    System.loadLibrary("vtkgdcmJava");
  }
  static FilenamesType fns = new FilenamesType();
  public static void process(String path)
    {
    fns.add( path );
    }
  // Process only files under dir
  public static void visitAllFiles(File dir)
    {
    if (dir.isDirectory())
      {
      String[] children = dir.list();
      for (int i=0; i<children.length; i++)
        {
        visitAllFiles(new File(dir, children[i]));
        }
      }
    else
      {
      process(dir.getPath());
      }
    }
  public void dointer(vtkImagePlaneWidget  current_widget)
    {
    int cstat = current_widget.GetCursorDataStatus();
    double[] v = current_widget.GetCurrentCursorPosition();
    //System.out.println( cstat );
    //System.out.println( v[0] );
    //System.out.println( v[1] );
    //System.out.println( v[2] );
    planeWidgetX.SetSliceIndex( (int)v[0] );
    planeWidgetY.SetSliceIndex( (int)v[1] );
    planeWidgetZ.SetSliceIndex( (int)v[2] );
    planeWidgetX.GetCurrentRenderer().ResetCameraClippingRange();
    planeWidgetY.GetCurrentRenderer().ResetCameraClippingRange();
    planeWidgetZ.GetCurrentRenderer().ResetCameraClippingRange();
    }
  public void startinterX()
    {
    dointer( planeWidgetX );
    }
  public void interX()
    {
    dointer( planeWidgetX );
    }
  public void endinterX()
    {
    }
  public void startinterY()
    {
    dointer( planeWidgetY );
    }
  public void interY()
    {
    dointer( planeWidgetY );
    }
  public void endinterY()
    {
    }
  public void startinterZ()
    {
    dointer( planeWidgetZ );
    }
  public void interZ()
    {
    dointer( planeWidgetZ );
    }
  public void endinterZ()
    {
    //System.out.println( "endinter" );
    }
  public static void AlignCamera(int slice_number, vtkImagePlaneWidget current_widget)
    {
```

```
    vtkImageData image = (vtkImageData)current_widget.GetInput();
    vtkRenderer ren = current_widget.GetCurrentRenderer();
    double[] origin = image.GetOrigin();
    double ox = origin[0];
    double oy = origin[1];
    double oz = origin[2];
    int dims[] = image.GetDimensions();
    int xMin = 0;
    int xMax = 1;
    int yMin = 2;
    int yMax = dims[0]-1;
    int zMin = dims[1]-1;
    int zMax = dims[2]-1;
    double[] spacing = image.GetSpacing();
    double sx = spacing[0];
    double sy = spacing[1];
    double sz = spacing[2];
    double cx = ox+(0.5*(xMax-xMin))*sx;
    double cy = oy+(0.5*(yMax-yMin))*sy;
    double cz = oy+(0.5*(zMax-zMin))*sz;
    double vx = 0, vy = 0, vz = 0;
    double nx = 0, ny = 0, nz = 0;
    int iaxis = current_widget.GetPlaneOrientation();
    if ( iaxis == 0 ) {
      vz = -1;
      nx = ox + xMax*sx;
      cx = ox + slice_number*sx;
      }
    else if ( iaxis == 1 ) {
      vz = -1;
      ny = oy+yMax*sy;
      cy = oy+slice_number*sy;
      }
    else {
      vy = 1;
      nz = oz+zMax*sz;
      cz = oz+slice_number*sz;
    }
    double px = cx+nx*2;
    double py = cy+ny*2;
    double pz = cz+nz*3;
    vtkCamera camera = ren.GetActiveCamera();
    camera.SetViewUp(vx, vy, vz);
    camera.SetFocalPoint(cx, cy, cz);
    camera.SetPosition(px, py, pz);
    camera.OrthogonalizeViewUp();
    ren.ResetCameraClippingRange();
    }
  private vtkImagePlaneWidget planeWidgetX = new vtkImagePlaneWidget();
  private vtkImagePlaneWidget planeWidgetY = new vtkImagePlaneWidget();
  private vtkImagePlaneWidget planeWidgetZ = new vtkImagePlaneWidget();
  public void config()
    {
    //System.out.println( "config" );
    planeWidgetX.GetCurrentRenderer().ResetCamera();
    planeWidgetY.GetCurrentRenderer().ResetCamera();
    planeWidgetZ.GetCurrentRenderer().ResetCamera();
    }
  public void Run(String dirname)
    {
    File dir = new File(dirname);
    visitAllFiles(dir);
    IPPSorter ipp = new IPPSorter();
    ipp.SetComputeZSpacing( true );
    ipp.SetZSpacingTolerance( 1e-3 );
    boolean b = ipp.Sort( fns );
    if(!b)
      {
      //throw new Exception("Could not scan");
      }
    double ippzspacing = ipp.GetZSpacing();
    FilenamesType sorted = ipp.GetFilenames();
    vtkStringArray files = new vtkStringArray();
    long nfiles = sorted.size();
    //for( String f :  sorted )
    for (int i = 0; i < nfiles; i++) {
      String f = sorted.get(i);
      files.InsertNextValue( f );
      }
    vtkGDCMImageReader reader = new vtkGDCMImageReader();
    reader.SetFileNames( files );
```

```
reader.Update(); // get spacing value
double[] spacing = reader.GetOutput().GetSpacing();
vtkImageChangeInformation change = new vtkImageChangeInformation();
change.SetInputConnection( reader.GetOutputPort() );
change.SetOutputSpacing( spacing[0], spacing[1], ippzspacing );
change.Update();
System.out.println( change.GetOutput().toString() );
vtkRenderer ren1 = new vtkRenderer();
ren1.SetViewport(0., 0., 0.333, 1);
ren1.SetBackground(0.1,0.2,0.4);
vtkRenderer ren2 = new vtkRenderer();
ren2.SetViewport(0.333, 0., 0.667, 1);
ren2.SetBackground(0.1,0.2,0.4);
vtkRenderer ren3 = new vtkRenderer();
ren3.SetViewport(0.667, 0., 1., 1.);
ren3.SetBackground(0.1,0.2,0.4);
vtkRenderWindow renWin = new vtkRenderWindow();
renWin.AddRenderer(ren1);
renWin.AddRenderer(ren2);
renWin.AddRenderer(ren3);
vtkRenderWindowInteractor iren = new vtkRenderWindowInteractor();
iren.SetRenderWindow(renWin);
vtkInteractorStyleImage style = new vtkInteractorStyleImage();
iren.SetInteractorStyle( style );
vtkCellPicker picker = new vtkCellPicker();
picker.SetTolerance(0.005);
vtkProperty ipwProp = new vtkProperty();
//vtkImagePlaneWidget planeWidgetX = new vtkImagePlaneWidget();
planeWidgetX.SetInteractor(iren);
planeWidgetX.SetCurrentRenderer(ren1);
planeWidgetX.SetDefaultRenderer(ren1);
planeWidgetX.RestrictPlaneToVolumeOn();
planeWidgetX.SetTexturePlaneProperty(ipwProp);
//planeWidgetX.GetPlaneProperty().SetColor(1,0,0);
//planeWidgetX.TextureInterpolateOff();
//planeWidgetX.SetResliceInterpolateToNearestNeighbour();
planeWidgetX.SetInputConnection(change.GetOutputPort());
planeWidgetX.SetPlaneOrientationToXAxes();
planeWidgetX.SetSliceIndex(62);
planeWidgetX.SetPicker(picker);
planeWidgetX.SetKeyPressActivationValue('x');
planeWidgetX.On();
planeWidgetX.InteractionOn();
//vtkImagePlaneWidget planeWidgetY = new vtkImagePlaneWidget();
planeWidgetY.SetInteractor(iren);
planeWidgetY.SetCurrentRenderer(ren2);
planeWidgetY.SetDefaultRenderer(ren2);
planeWidgetY.RestrictPlaneToVolumeOn();
planeWidgetY.SetTexturePlaneProperty(ipwProp);
//planeWidgetY.GetPlaneProperty().SetColor(1,0,0);
//planeWidgetY.TextureInterpolateOff();
//planeWidgetY.SetResliceInterpolateToNearestNeighbour();
planeWidgetY.SetInputConnection(change.GetOutputPort());
planeWidgetY.SetLookupTable( planeWidgetX.GetLookupTable() );
planeWidgetY.SetPlaneOrientationToYAxes();
planeWidgetY.SetSliceIndex(32);
planeWidgetY.SetPicker(picker);
planeWidgetY.SetKeyPressActivationValue('y');
planeWidgetY.On();
//vtkImagePlaneWidget planeWidgetZ = new vtkImagePlaneWidget();
planeWidgetZ.SetInteractor(iren);
planeWidgetZ.SetCurrentRenderer(ren3);
planeWidgetZ.SetDefaultRenderer(ren3);
planeWidgetZ.RestrictPlaneToVolumeOn();
planeWidgetZ.SetTexturePlaneProperty(ipwProp);
//planeWidgetZ.GetPlaneProperty().SetColor(1,0,0);
//planeWidgetZ.TextureInterpolateOff();
//planeWidgetZ.SetResliceInterpolateToNearestNeighbour();
planeWidgetZ.SetInputConnection(change.GetOutputPort());
planeWidgetZ.SetLookupTable( planeWidgetX.GetLookupTable() );
planeWidgetZ.SetPlaneOrientationToZAxes();
planeWidgetZ.SetSliceIndex(32);
planeWidgetZ.SetPicker(picker);
planeWidgetZ.SetKeyPressActivationValue('z');
planeWidgetZ.On();
iren.Initialize();
renWin.Render();
AlignCamera(52, planeWidgetX);
AlignCamera(32, planeWidgetY);
AlignCamera(32, planeWidgetZ);
planeWidgetX.GetCurrentRenderer().ResetCamera();
```

```
    planeWidgetY.GetCurrentRenderer().ResetCamera();
    planeWidgetZ.GetCurrentRenderer().ResetCamera();
    renWin.Render();
    planeWidgetX.AddObserver("StartInteractionEvent", this,"startinterX");
    planeWidgetX.AddObserver("InteractionEvent", this,"interX");
    planeWidgetX.AddObserver("EndInteractionEvent", this,"endinterX");
    planeWidgetY.AddObserver("StartInteractionEvent", this,"startinterY");
    planeWidgetY.AddObserver("InteractionEvent", this,"interY");
    planeWidgetY.AddObserver("EndInteractionEvent", this,"endinterY");
    planeWidgetZ.AddObserver("StartInteractionEvent", this,"startinterZ");
    planeWidgetZ.AddObserver("InteractionEvent", this,"interZ");
    planeWidgetZ.AddObserver("EndInteractionEvent", this,"endinterZ");
    iren.AddObserver("ConfigureEvent", this,"config");
    iren.Start();
    }
  public static void main(String[] args) throws Exception
    {
    String dirname = args[0];
    if( !PosixEmulation.FileIsDirectory( dirname ) )
      {
      return;
      }
    MPRViewer2 me = new MPRViewer2();
    me.Run( dirname );
    }
}
```

12.171 ReadSeriesIntoVTK.java

```
/*=========================================================================

Program:  GDCM (Grassroots DICOM). A DICOM library

Copyright (c) 2006-2011 Mathieu Malaterre
All rights reserved.
See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.

This software is distributed WITHOUT ANY WARRANTY; without even
the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
PURPOSE.  See the above copyright notice for more information.

=========================================================================*/
// We are required to call the package 'vtk' eventhough I (MM) would have prefered
// an import statement along the line of:
// import vtkgdcm.*;
import vtk.*;
/*
* Usage:
* export LD_LIBRARY_PATH=/usr/lib/jvm/java-6-openjdk/jre/lib/amd64/xawt:.
* java -classpath `pwd`/vtkgdcm.jar:/usr/share/java/vtk.jar:.  ReadSeriesIntoVTK
*/
public class ReadSeriesIntoVTK
{
  static {
    System.loadLibrary("vtkCommonJava");
    System.loadLibrary("vtkFilteringJava");
    System.loadLibrary("vtkIOJava");
    System.loadLibrary("vtkImagingJava");
    System.loadLibrary("vtkGraphicsJava");
    System.loadLibrary("vtkgdcmJava");
    try {
      System.loadLibrary("vtkRenderingJava");
    } catch (Throwable e) {
      System.out.println("cannot load vtkHybrid, skipping...");
    }
    try {
      System.loadLibrary("vtkHybridJava");
    } catch (Throwable e) {
      System.out.println("cannot load vtkHybrid, skipping...");
    }
    try {
      System.loadLibrary("vtkVolumeRenderingJava");
    } catch (Throwable e) {
      System.out.println("cannot load vtkVolumeRendering, skipping...");
    }
  }
  public static void main(String[] args)
```

```
    {
    vtkFileOutputWindow outWin = new vtkFileOutputWindow();
    outWin.SetInstance(outWin);
    outWin.SetFileName("MVSVTKViewer.log");
    // See:  http://review.source.kitware.com/#change,888
    // vtkWrapJava does not handle static keyword
    // String directory = vtkGDCMTesting.GetGDCMDataRoot();
    vtkGDCMTesting t = new vtkGDCMTesting();
    String directory = t.GetGDCMDataRoot();
    String file0 = directory + "/SIEMENS_MAGNETOM-12-MONO2-FileSeq0.dcm";
    String file1 = directory + "/SIEMENS_MAGNETOM-12-MONO2-FileSeq1.dcm";
    String file2 = directory + "/SIEMENS_MAGNETOM-12-MONO2-FileSeq2.dcm";
    String file3 = directory + "/SIEMENS_MAGNETOM-12-MONO2-FileSeq3.dcm";
    vtkStringArray s = new vtkStringArray();
    System.out.println("adding :  " + file0 );
    s.InsertNextValue( file0 );
    s.InsertNextValue( file1 );
    s.InsertNextValue( file2 );
    s.InsertNextValue( file3 );
    vtkGDCMImageReader reader = new vtkGDCMImageReader();
    reader.SetFileNames( s );
    reader.Update();
    System.out.println("Success reading:  " + file0 );
    vtkMetaImageWriter writer = new vtkMetaImageWriter();
    writer.DebugOn();
    writer.SetCompression( false );
    writer.SetInputConnection( reader.GetOutputPort() );
    writer.SetFileName( "ReadSeriesIntoVTK.mhd" );
    writer.Write();
    System.out.println("Success writing:  " + writer.GetFileName() );
    }
}
```

12.172 CastConvertPhilips.py

```
1
14
15 """
16 Usage:
17
18 python --public /path/to/directory/
19 or
20 python --private /path/to/directory/
21
22 python --public --extension bak /path/to/directory/
23
24 rename -f 's/\.bak$//' *.bak
25
26 TODO:
27 http://docs.python.org/library/optparse.html#module-optparse
28 """
29
30 import vtkgdcm
31 import vtk
32 import sys
33 import gdcm
34
35 def ProcessOneFilePublic(filename, outfilename, tmpfile):
36   gdcm.ImageHelper.SetForceRescaleInterceptSlope(True)
37   vtkreader = vtkgdcm.vtkGDCMImageReader()
38   vtkreader.SetFileName( filename )
39   vtkreader.Update()
40
41   cast = vtk.vtkImageCast()
42   cast.SetInput( vtkreader.GetOutput() )
43   cast.SetOutputScalarTypeToUnsignedShort()
44
45   # vtkGDCMImageWriter does not support Sequence, so let's write a tmp file first:
46   # Some operation will actually be discarded (we simply need a temp storage)
47   vtkwriter = vtkgdcm.vtkGDCMImageWriter()
48   vtkwriter.SetFileName( tmpfile )
49   vtkwriter.SetMedicalImageProperties( vtkreader.GetMedicalImageProperties() )
50   vtkwriter.SetDirectionCosines( vtkreader.GetDirectionCosines() )
51   print "Format:",vtkreader.GetImageFormat()
52   vtkwriter.SetImageFormat( vtkreader.GetImageFormat() )
53   vtkwriter.SetInput( cast.GetOutput() )
54   #vtkwriter.Update()
```

```
55   vtkwriter.Write()
56
57   # ok now rewrite the exact same file as the original (keep all info)
58   # but use the Pixel Data Element from the written file
59   tmpreader = gdcm.ImageReader()
60   tmpreader.SetFileName( tmpfile )
61   if not tmpreader.Read():
62     sys.exit(1)
63
64   reader = gdcm.Reader()
65   reader.SetFileName( filename )
66   if not reader.Read():
67     sys.exit(1)
68
69   # Make sure to remove Slope/Rescale to avoid re-execution
70   ds = reader.GetFile().GetDataSet()
71   tags = [
72     gdcm.Tag(0x0028,0x1052),
73     gdcm.Tag(0x0028,0x1053),
74     gdcm.Tag(0x0028,0x1053),
75   ]
76   for tag in tags:
77     ds.Remove( tag )
78
79   writer = gdcm.ImageWriter()
80   writer.SetFileName( outfilename )
81   # Pass image from vtk written file
82   writer.SetImage( tmpreader.GetImage() )
83   # pass dataset from initial 'reader'
84   writer.SetFile( reader.GetFile() )
85   if not writer.Write():
86     sys.exit(1)
87
88 def ProcessOneFilePrivate(filename, outfilename, tmpfile):
89   vtkreader = vtkgdcm.vtkGDCMImageReader()
90   vtkreader.SetFileName( filename )
91   vtkreader.Update()
92
93
94   # (2005,1409)     DS      4       0.0
95   # (2005,140a)     DS      16      1.52283272283272
96
97   # (2005,0014)     LO      26      Philips MR Imaging DD 005
98   tag1 = gdcm.PrivateTag(0x2005,0x09,"Philips MR Imaging DD 005")
99   tag2 = gdcm.PrivateTag(0x2005,0x0a,"Philips MR Imaging DD 005")
100
101
102
103    # Need to access some private tags, reread the file (for now):
104    reader = gdcm.Reader()
105    reader.SetFileName( filename )
106    if not reader.Read():
107      sys.exit(1)
108
109    ds = reader.GetFile().GetDataSet()
110
111    el1 = ds.GetDataElement( tag1 )
112    el2 = ds.GetDataElement( tag2 )
113
114
115    #pf = gdcm.PythonFilter()
116    #pf.SetFile( reader.GetFile() )
117    #print el1.GetTag()
118
119    print el1.GetByteValue()
120    v1 = eval(el1.GetByteValue().GetBuffer())
121    print el2.GetByteValue()
122    v2 = eval(el2.GetByteValue().GetBuffer())
123
124    print v1
125    shift = v1
126    print v2
127    scale = v2
128
129    ss = vtk.vtkImageShiftScale()
130    ss.SetInput( vtkreader.GetOutput() )
131    # because VTK image shift / scale convention is inverted from DICOM make sure shift is 0
132    assert shift == 0
133    ss.SetShift( shift )
134    ss.SetScale( scale )
135    ss.SetOutputScalarTypeToUnsignedShort ()
```

```
136   ss.Update()
137
138   # vtkGDCMImageWriter does not support Sequence, so let's write a tmp file first:
139   # Some operation will actually be discarded (we simply need a temp storage)
140   vtkwriter = vtkgdcm.vtkGDCMImageWriter()
141   vtkwriter.SetFileName( tmpfile )
142   vtkwriter.SetMedicalImageProperties( vtkreader.GetMedicalImageProperties() )
143   vtkwriter.SetDirectionCosines( vtkreader.GetDirectionCosines() )
144   vtkwriter.SetImageFormat( reader.GetImageFormat() )
145   # do not pass shift/scale again
146   vtkwriter.SetInput( ss.GetOutput() )
147   #vtkwriter.Update()
148   vtkwriter.Write()
149
150   # ok now rewrite the exact same file as the original (keep all info)
151   # but use the Pixel Data Element from the written file
152   tmpreader = gdcm.ImageReader()
153   tmpreader.SetFileName( tmpfile )
154   if not tmpreader.Read():
155     sys.exit(1)
156
157   writer = gdcm.ImageWriter()
158   writer.SetFileName( outfilename )
159   # Pass image from vtk written file
160   writer.SetImage( tmpreader.GetImage() )
161   # pass dataset from initial 'reader'
162   writer.SetFile( reader.GetFile() )
163   if not writer.Write():
164     sys.exit(1)
165
166 if __name__ == "__main__":
167
168   gdcm.Trace.DebugOff()
169   gdcm.Trace.WarningOff()
170   #filename = sys.argv[1]
171   #outfilename = sys.argv[2]
172   tmpfile = "/tmp/philips_rescaled.dcm"
173   #ProcessOneFile( filename, outfilename, tmpfile )
174   rescaletype = sys.argv[1]
175   assert rescaletype == "--public" or rescaletype == "--private"
176   dirname = sys.argv[2]
177   d = gdcm.Directory()
178   d.Load( dirname )
179
180   for f in d.GetFilenames():
181     #print f
182     ProcessOneFilePublic( f, f + ".bak", tmpfile )
183
184
185 print "success"
```

12.173 headsq2dcm.py

```
1
14
15 """
16 Usage:
17 python headsq2dcm.py -D /path/to/VTKData
18 """
19
20 import vtk
21 import vtkgdcm
22 from vtk.util.misc import vtkGetDataRoot
23 VTK_DATA_ROOT = vtkGetDataRoot()
24
25 reader = vtk.vtkVolume16Reader()
26 reader.SetDataDimensions(64, 64)
27 reader.SetDataByteOrderToLittleEndian()
28 reader.SetFilePrefix(VTK_DATA_ROOT + "/Data/headsq/quarter")
29 reader.SetImageRange(1, 93)
30 reader.SetDataSpacing(3.2, 3.2, 1.5)
31
32 cast = vtk.vtkImageCast()
33 cast.SetInput( reader.GetOutput() )
34 cast.SetOutputScalarTypeToUnsignedChar()
35
36 # By default this is creating a Multiframe Grayscale Word Secondary Capture Image Storage
```

```
writer = vtkgdcm.vtkGDCMImageWriter()
writer.SetFileName( "headsq.dcm" )
writer.SetInput( reader.GetOutput() )
# cast -> Multiframe Grayscale Byte Secondary Capture Image Storage
#writer.SetInput( cast.GetOutput() )
writer.SetFileDimensionality( 3 )
writer.Write()
```

Index

~ASN1
gdcm::ASN1, 133
~AnonymizeEvent
gdcm::AnonymizeEvent, 109
~Anonymizer
gdcm::Anonymizer, 114
~Attribute
gdcm::Attribute< Group, Element, TVR, VM::VM1_n >, 157
~AudioCodec
gdcm::AudioCodec, 169
~BaseCompositeMessage
gdcm::network::BaseCompositeMessage, 174
~BaseNormalizedMessage
gdcm::network::BaseNormalizedMessage, 176
~BasePDU
gdcm::network::BasePDU, 178
~BaseQuery
gdcm::BaseQuery, 181
~BaseRootQuery
gdcm::BaseRootQuery, 187
~Bitmap
gdcm::Bitmap, 199
~BitmapToBitmapFilter
gdcm::BitmapToBitmapFilter, 213
~BoxRegion
gdcm::BoxRegion, 216
~ByteSwapFilter
gdcm::ByteSwapFilter, 224
~ByteValue
gdcm::ByteValue, 227
~CAPICryptographicMessageSyntax
gdcm::CAPICryptographicMessageSyntax, 236
~CSAHeader
gdcm::CSAHeader, 299
~Cleaner
gdcm::Cleaner, 249
~Coder
gdcm::Coder, 258
~Command
gdcm::Command, 267
~CommandDataSet
gdcm::CommandDataSet, 269
~CryptoFactory
gdcm::CryptoFactory, 284
~CryptographicMessageSyntax
gdcm::CryptographicMessageSyntax, 286
~Curve
gdcm::Curve, 315
~DICOMDIRGenerator
gdcm::DICOMDIRGenerator, 368
~DPath
gdcm::DPath, 405
~DataEvent
gdcm::DataEvent, 336
~DataSetEvent
gdcm::DataSetEvent, 353
~Decoder
gdcm::Decoder, 356
~Defs
gdcm::Defs, 360
~DeltaEncodingCodec
gdcm::DeltaEncodingCodec, 364
~DictConverter
gdcm::DictConverter, 377
~DictPrinter
gdcm::DictPrinter, 387
~Dicts
gdcm::Dicts, 389
~DirectionCosines
gdcm::DirectionCosines, 395
~Directory
gdcm::Directory, 399
~Dumper
gdcm::Dumper, 409
~Element
gdcm::Element< TVR, VM::VM1_n >, 419
~EmptyMaskGenerator
gdcm::EmptyMaskGenerator, 439
~Event
gdcm::Event, 450
~Exception
gdcm::Exception, 453
~File
gdcm::File, 464
~FileAnonymizer
gdcm::FileAnonymizer, 468
~FileChangeTransferSyntax
gdcm::FileChangeTransferSyntax, 472
~FileDecompressLookupTable

gdcm::FileDecompressLookupTable, 476
~FileDerivation
gdcm::FileDerivation, 478
~FileExplicitFilter
gdcm::FileExplicitFilter, 482
~FileMetaInformation
gdcm::FileMetaInformation, 487
~FileNameEvent
gdcm::FileNameEvent, 500
~FileStreamer
gdcm::FileStreamer, 510
~FilenameGenerator
gdcm::FilenameGenerator, 504
~Global
gdcm::Global, 527
~GroupDict
gdcm::GroupDict, 531
~IconImageFilter
gdcm::IconImageFilter, 533
~IconImageGenerator
gdcm::IconImageGenerator, 537
~Image
gdcm::Image, 542
~ImageApplyLookupTable
gdcm::ImageApplyLookupTable, 549
~ImageChangePhotometricInterpretation
gdcm::ImageChangePhotometricInterpretation, 552
~ImageChangePlanarConfiguration
gdcm::ImageChangePlanarConfiguration, 556
~ImageChangeTransferSyntax
gdcm::ImageChangeTransferSyntax, 560
~ImageCodec
gdcm::ImageCodec, 566
~ImageConverter
gdcm::ImageConverter, 577
~ImageFragmentSplitter
gdcm::ImageFragmentSplitter, 581
~ImageReader
gdcm::ImageReader, 591
~ImageRegionReader
gdcm::ImageRegionReader, 595
~ImageToImageFilter
gdcm::ImageToImageFilter, 599
~ImageWriter
gdcm::ImageWriter, 602
~JPEG12Codec
gdcm::JPEG12Codec, 633
~JPEG16Codec
gdcm::JPEG16Codec, 636
~JPEG2000Codec
gdcm::JPEG2000Codec, 640
~JPEG8Codec
gdcm::JPEG8Codec, 647
~JPEGCodec
gdcm::JPEGCodec, 651
~JPEGLSCodec
gdcm::JPEGLSCodec, 660
~JSON
gdcm::JSON, 665
~KAKADUCodec
gdcm::KAKADUCodec, 668
~LookupTable
gdcm::LookupTable, 676
~MemberCommand
gdcm::MemberCommand< T >, 706
~MeshPrimitive
gdcm::MeshPrimitive, 712
~ModuleEntry
gdcm::ModuleEntry, 727
~MrProtocol
gdcm::MrProtocol, 739
~Object
gdcm::Object, 769
~OpenSSLCryptographicMessageSyntax
gdcm::OpenSSLCryptographicMessageSyntax, 774
~OpenSSLP7CryptographicMessageSyntax
gdcm::OpenSSLP7CryptographicMessageSyntax, 779
~Orientation
gdcm::Orientation, 782
~Overlay
gdcm::Overlay, 788
~PDBHeader
gdcm::PDBHeader, 810
~PDFCodec
gdcm::PDFCodec, 814
~PGXCodec
gdcm::PGXCodec, 822
~PNMCodec
gdcm::PNMCodec, 857
~PVRGCodec
gdcm::PVRGCodec, 900
~ParseException
gdcm::ParseException, 796
~Parser
gdcm::Parser, 800
~Pixmap
gdcm::Pixmap, 840
~PixmapReader
gdcm::PixmapReader, 846
~PixmapToPixmapFilter
gdcm::PixmapToPixmapFilter, 850
~PixmapWriter
gdcm::PixmapWriter, 853
~Preamble
gdcm::Preamble, 861
~Printer
gdcm::Printer, 884

~PrivateDict
gdcm::PrivateDict, 888
~ProgressEvent
gdcm::ProgressEvent, 897
~PythonFilter
gdcm::PythonFilter, 903
~QueryBase
gdcm::QueryBase, 905
~RAWCodec
gdcm::RAWCodec, 920
~RLECodec
gdcm::RLECodec, 942
~Reader
gdcm::Reader, 926
~Region
gdcm::Region, 933
~Rescaler
gdcm::Rescaler, 937
~SHA1
gdcm::SHA1, 1026
~Scanner
gdcm::Scanner, 952
~Scanner2
gdcm::Scanner2, 963
~Segment
gdcm::Segment, 973
~SegmentReader
gdcm::SegmentReader, 985
~SegmentWriter
gdcm::SegmentWriter, 988
~SegmentedPaletteColorLookupTable
gdcm::SegmentedPaletteColorLookupTable, 981
~SerieHelper
gdcm::SerieHelper, 1010
~ServiceClassUser
gdcm::ServiceClassUser, 1019
~SimpleMemberCommand
gdcm::SimpleMemberCommand< T >, 1030
~SimpleSubjectWatcher
gdcm::SimpleSubjectWatcher, 1033
~SmartPointer
gdcm::SmartPointer< ObjectType >, 1040
~Sorter
gdcm::Sorter, 1049
~Spacing
gdcm::Spacing, 1054
~SplitMosaicFilter
gdcm::SplitMosaicFilter, 1056
~StreamImageReader
gdcm::StreamImageReader, 1062
~StreamImageWriter
gdcm::StreamImageWriter, 1067
~StrictScanner
gdcm::StrictScanner, 1075
~StrictScanner2
gdcm::StrictScanner2, 1085
~StringFilter
gdcm::StringFilter, 1099
~Subject
gdcm::Subject, 1105
~Surface
gdcm::Surface, 1110
~SurfaceReader
gdcm::SurfaceReader, 1127
~SurfaceWriter
gdcm::SurfaceWriter, 1130
~Table
gdcm::Table, 1145
~TableEntry
gdcm::TableEntry, 1147
~TableReader
gdcm::TableReader, 1149
~TableRow
gdcm::network::TableRow, 1153
~TagPath
gdcm::TagPath, 1165
~Testing
gdcm::Testing, 1169
~Trace
gdcm::Trace, 1176
~Transition
gdcm::network::Transition, 1190
~ULAction
gdcm::network::ULAction, 1236
~ULBasicCallback
gdcm::network::ULBasicCallback, 1276
~ULConnection
gdcm::network::ULConnection, 1279
~ULConnectionCallback
gdcm::network::ULConnectionCallback, 1284
~ULConnectionManager
gdcm::network::ULConnectionManager, 1290
~ULEvent
gdcm::network::ULEvent, 1297
~ULWritingCallback
gdcm::network::ULWritingCallback, 1301
~UserInformation
gdcm::network::UserInformation, 1312
~Validate
gdcm::Validate, 1316
~Value
gdcm::Value, 1319
~Version
gdcm::Version, 1323
~Writer
gdcm::Writer, 1470
~XMLDictReader
gdcm::XMLDictReader, 1475

~XMLPrinter
gdcm::XMLPrinter, 1478
~XMLPrivateDictReader
gdcm::XMLPrivateDictReader, 1482
~vtkGDCMImageReader
vtkGDCMImageReader, 1350
~vtkGDCMImageReader2
vtkGDCMImageReader2, 1365
~vtkGDCMImageWriter
vtkGDCMImageWriter, 1379
~vtkGDCMMedicalImageProperties
vtkGDCMMedicalImageProperties, 1387
~vtkGDCMPolyDataReader
vtkGDCMPolyDataReader, 1391
~vtkGDCMPolyDataWriter
vtkGDCMPolyDataWriter, 1396
~vtkGDCMTesting
vtkGDCMTesting, 1401
~vtkGDCMThreadedImageReader
vtkGDCMThreadedImageReader, 1405
~vtkGDCMThreadedImageReader2
vtkGDCMThreadedImageReader2, 1409
~vtkImageColorViewer
vtkImageColorViewer, 1419
~vtkImageMapToColors16
vtkImageMapToColors16, 1432
~vtkImageMapToWindowLevelColors2
vtkImageMapToWindowLevelColors2, 1438
~vtkImagePlanarComponentsToComponents
vtkImagePlanarComponentsToComponents, 1442
~vtkImageRGBToYBR
vtkImageRGBToYBR, 1444
~vtkImageYBRToRGB
vtkImageYBRToRGB, 1447
~vtkLookupTable16
vtkLookupTable16, 1450
~vtkRTStructSetProperties
vtkRTStructSetProperties, 1454

AAbortPDU
gdcm::network::AAbortPDU, 88
AAssociateACPDU
gdcm::network::AAssociateACPDU, 91
gdcm::network::AAssociateRQPDU, 103
AAssociateRJPDU
gdcm::network::AAssociateRJPDU, 95
AAssociateRQPDU
gdcm::network::AAssociateACPDU, 94
gdcm::network::AAssociateRQPDU, 99
AbstractMultiDimensionalImageModel
gdcm::UIDs, 1223
AbstractSyntax
gdcm::network::AbstractSyntax, 105
gdcm::PresentationContext, 867

AcquisitionContextSRStorage
gdcm::UIDs, 1222
ActiveComponent
vtkImageMapToColors16, 1436
Add
gdcm::GroupDict, 531
add1
gdcm, 63
AddAcceptedPresentationContext
gdcm::network::ULConnection, 1279
AddContourReferencedFrameOfReference
vtkRTStructSetProperties, 1454
AddCSAHeaderDictEntry
gdcm::CSAHeaderDict, 304
AddDerivationDescription
gdcm::FileDerivation, 478
AddDictEntry
gdcm::Dict, 372
gdcm::PrivateDict, 888
AddFile
gdcm::FileSet, 507
gdcm::SerieHelper, 1010
AddFileName
gdcm::SerieHelper, 1010
AddFragment
gdcm::SequenceOfFragments, 993
AddFromFile
gdcm::PresentationContextGenerator, 872
AddGroupLength
gdcm::DictConverter, 377
AddImageDirectoryRecord
gdcm::DICOMDIRGenerator, 368
AddInput
vtkImageColorViewer, 1419
AddInputConnection
vtkImageColorViewer, 1419
AddIOD
gdcm::IODs, 619
AddIODEntry
gdcm::IOD, 613
AddItem
gdcm::SequenceOfItems, 1002
AddMacro
gdcm::Macros, 688
gdcm::Module, 723
AddMacroEntry
gdcm::Macro, 685
AddModule
gdcm::Modules, 731
AddModuleEntry
gdcm::Module, 723
gdcm::NestedModuleEntries, 752
AddNewUndefinedLengthItem
gdcm::SequenceOfItems, 1002

AddObserver
gdcm::Subject, 1105
AddPatientDirectoryRecord
gdcm::DICOMDIRGenerator, 368
AddPresentationContext
gdcm::network::AAssociateRQPDU, 99
gdcm::PresentationContextGenerator, 872
AddPresentationContextAC
gdcm::network::AAssociateACPDU, 92
AddPresentationDataValue
gdcm::network::PDataTFPDU, 804
AddPrimitiveData
gdcm::MeshPrimitive, 712
AddPrivateTag
gdcm::Scanner, 952
gdcm::Scanner2, 963
gdcm::StrictScanner, 1075
gdcm::StrictScanner2, 1085
AddPublicTag
gdcm::Scanner2, 963
gdcm::StrictScanner2, 1085
AddPurposeOfReferenceCodeSequence
gdcm::FileDerivation, 479
AddQueryDataSet
gdcm::BaseQuery, 181
AddReference
gdcm::FileDerivation, 479
AddReferencedFrameOfReference
vtkRTStructSetProperties, 1455
AddRestriction
gdcm::SerieHelper, 1010, 1011
AddRoleSelectionSub
gdcm::network::UserInformation, 1313
AddSegment
gdcm::SegmentWriter, 989
AddSelect
gdcm::Sorter, 1049
AddSeriesDirectoryRecord
gdcm::DICOMDIRGenerator, 369
AddSkipTag
gdcm::Scanner, 953
gdcm::Scanner2, 963
gdcm::StrictScanner, 1076
gdcm::StrictScanner2, 1086
AddSOPClassExtendedNegociationSub
gdcm::network::UserInformation, 1313
AddSourceImageSequence
gdcm::FileDerivation, 479
AddStructureSetROI
vtkRTStructSetProperties, 1455
AddStructureSetROIObservation
vtkRTStructSetProperties, 1455
AddStudyDirectoryRecord
gdcm::DICOMDIRGenerator, 369
AddSurface
gdcm::Segment, 973
AddTag
gdcm::Scanner, 953
gdcm::StrictScanner, 1076
AddTransferSyntax
gdcm::network::PresentationContextRQ, 876
gdcm::PresentationContext, 866
AdultMouseAnatomyOntology
gdcm::UIDs, 1220
AdvancedBlendingPresentationStateStorage
gdcm::UIDs, 1221
AE
gdcm::VR, 1336
AEComp
gdcm, 58
AES128_CIPHER
gdcm::CryptographicMessageSyntax, 286
AES192_CIPHER
gdcm::CryptographicMessageSyntax, 286
AES256_CIPHER
gdcm::CryptographicMessageSyntax, 286
AffectedSOPClassUID
gdcm::network::CEchoRQ, 240
ALGOType
gdcm::Segment, 972
ALGOType_END
gdcm::Segment, 973
Allocate
gdcm::LookupTable, 677
AmbulatoryECGWaveformStorage
gdcm::MediaStorage, 697
gdcm::UIDs, 1217
AnatomicRegion
gdcm::Segment, 978
AnatomicRegionModifiers
gdcm::Segment, 978
AnonymizeEvent
gdcm::AnonymizeEvent, 108, 109
Anonymizer
gdcm::Anonymizer, 114
Append
gdcm::ByteValue, 228
gdcm::Global, 527
AppendFrameEncode
gdcm::ImageCodec, 566
gdcm::JPEG2000Codec, 640
gdcm::JPEGCodec, 651
gdcm::JPEGLSCodec, 660
gdcm::RLECodec, 943
AppendImplementationClassUID
gdcm::FileMetaInformation, 488
AppendRowEncode
gdcm::ImageCodec, 567

gdcm::JPEG2000Codec, 640
gdcm::JPEGCodec, 652
gdcm::JPEGLSCodec, 660
gdcm::RLECodec, 943
AppendToDataElement
gdcm::FileStreamer, 510
AppendToGroupDataElement
gdcm::FileStreamer, 511
ApplicationContext
gdcm::network::ApplicationContext, 122
Apply
gdcm::ImageApplyLookupTable, 549
ApplyInverseVideo
vtkGDCMImageReader, 1358
vtkGDCMImageReader2, 1373
ApplyLookupTable
vtkGDCMImageReader, 1359
vtkGDCMImageReader2, 1373
ApplyPlanarConfiguration
vtkGDCMImageReader, 1359
vtkGDCMImageReader2, 1373
ApplyShiftScale
vtkGDCMImageReader, 1359
vtkGDCMImageReader2, 1374
ApplyYBRToRGB
vtkGDCMImageReader, 1359
vtkGDCMImageReader2, 1374
Area
gdcm::BoxRegion, 216
gdcm::Region, 933
AReleaseRPPDU
gdcm::network::AReleaseRPPDU, 127
AReleaseRQPDU
gdcm::network::AReleaseRQPDU, 129
AreOverlaysInPixelData
gdcm::Bitmap, 199
gdcm::Pixmap, 840
ARGB
gdcm::PhotometricInterpretation, 826
ArrayIncludeMacrosType
gdcm::Macro, 685
gdcm::Module, 723
ArrayType
gdcm::Attribute< Group, Element, TVR, TVM >, 138
gdcm::Attribute< Group, Element, TVR, VM::VM1 >, 147
gdcm::Attribute< Group, Element, TVR, VM::VM1_n >, 156
ArterialPulseWaveformStorage
gdcm::UIDs, 1221
ARTIMTimer
gdcm::network::ARTIMTimer, 131
AS
gdcm::VR, 1336
ASComp
gdcm, 58
ASN1
gdcm::ASN1, 133
AsynchronousOperationsWindowSub
gdcm::network::AsynchronousOperationsWindowSub, 135
AT
gdcm::VR, 1336
Attribute
gdcm::Attribute< Group, Element, TVR, VM::VM1_n >, 157
gdcm::terminal, 85
Audio
gdcm::MediaStorage, 699
AudioCodec
gdcm::AudioCodec, 169
AudioSRStorageTrialRetired
gdcm::UIDs, 1218
AUTOMATIC
gdcm::Segment, 973
AutoPixelMinMax
gdcm::IconImageGenerator, 537
AutorefractionMeasurementsStorage
gdcm::UIDs, 1221
AXIAL
gdcm::Orientation, 782

backslash
gdcm, 64
BadBigEndian
gdcm::SwapCode, 1133
BadLittleEndian
gdcm::SwapCode, 1133
BALCPProtect
gdcm::Anonymizer, 114
Base64
gdcm::Base64, 171
BaseQuery
gdcm::BaseQuery, 181
BaseRootQuery
gdcm::BaseRootQuery, 187
BasicAnnotationBoxSOPClass
gdcm::UIDs, 1216
BasicApplicationLevelConfidentialityProfile
gdcm::Anonymizer, 114
BasicCodedEntry
gdcm::SegmentHelper::BasicCodedEntry, 191
BasicCodedEntryVector
gdcm::Segment, 972
BasicColorImageBoxSOPClass
gdcm::UIDs, 1216
BasicColorPrintManagementMetaSOPClass
gdcm::UIDs, 1216

BasicFilmBoxSOPClass
gdcm::UIDs, 1216
BasicFilmSessionSOPClass
gdcm::UIDs, 1216
BasicGrayscaleImageBoxSOPClass
gdcm::UIDs, 1216
BasicGrayscalePrintManagementMetaSOPClass
gdcm::UIDs, 1216
BasicOffsetTable
gdcm::BasicOffsetTable, 194
BasicPrintImageOverlayBoxSOPClassRetired
gdcm::UIDs, 1216
BasicStructuredDisplayStorage
gdcm::UIDs, 1222
BasicStudyContentNotificationSOPClassRetired
gdcm::UIDs, 1215
BasicTextSR
gdcm::MediaStorage, 697
BasicTextSRStorage
gdcm::UIDs, 1218
BasicVoiceAudioWaveformStorage
gdcm::MediaStorage, 697
gdcm::UIDs, 1217
Begin
gdcm::CSAHeaderDict, 304
gdcm::DataSet, 342
gdcm::Dict, 373
gdcm::IODs, 619
gdcm::Scanner, 953
gdcm::Scanner2, 963
gdcm::SequenceOfFragments, 994
gdcm::SequenceOfItems, 1003
gdcm::StrictScanner, 1076
gdcm::StrictScanner2, 1086
BigEndian
gdcm::SwapCode, 1133
Bitmap
gdcm::Bitmap, 199
gdcm::JPEG2000Codec, 645
gdcm::PixelFormat, 837
BitmapToBitmapFilter
gdcm::BitmapToBitmapFilter, 213
BitSample
gdcm::JPEGCodec, 657
gdcm::LookupTable, 681
black
gdcm::terminal, 85
BlendingSoftcopyPresentationStateStorageSOPClass
gdcm::UIDs, 1217
blink
gdcm::terminal, 85
BLUE
gdcm::LookupTable, 676
blue
gdcm::terminal, 85
BOOL_FUNCTION_PFILE_PFILE_POINTER
gdcm, 58
BoundingBox
gdcm::BoxRegion, 216
BoxRegion
gdcm::BoxRegion, 216
BreakConnection
gdcm::network::ULConnectionManager, 1291
BreakConnectionNow
gdcm::network::ULConnectionManager, 1291
BreastImagingRelevantPatientInformationQuery
gdcm::UIDs, 1219
BreastProjectionXRayImageStorageForPresentation
gdcm::MediaStorage, 698
gdcm::UIDs, 1221
BreastProjectionXRayImageStorageForProcessing
gdcm::MediaStorage, 698
gdcm::UIDs, 1221
BreastTomosynthesisImageStorage
gdcm::MediaStorage, 698
gdcm::UIDs, 1220
bright
gdcm::terminal, 85
Build
vtkLookupTable16, 1450
ByteBuffer
gdcm::ByteBuffer, 220
bytes
gdcm::Tag, 1164
ByteSwap
gdcm::ByteSwapFilter, 224
ByteSwapFilter
gdcm::ByteSwapFilter, 223, 224
ByteValue
gdcm::ByteValue, 227

C_CANCEL_RQ
gdcm::network::DIMSE, 393
C_ECHO_RQ
gdcm::network::DIMSE, 393
C_ECHO_RSP
gdcm::network::DIMSE, 393
C_FIND_RQ
gdcm::network::DIMSE, 393
C_FIND_RSP
gdcm::network::DIMSE, 393
C_GET_RQ
gdcm::network::DIMSE, 393
C_GET_RSP
gdcm::network::DIMSE, 393
C_MOVE_RQ
gdcm::network::DIMSE, 393
C_MOVE_RSP

gdcm::network::DIMSE, 393
C_STORE_RQ
gdcm::network::DIMSE, 393
C_STORE_RSP
gdcm::network::DIMSE, 393
CALIBRATED
gdcm::Spacing, 1053
CanCode
gdcm::AudioCodec, 169
gdcm::Coder, 259
gdcm::ImageCodec, 567
gdcm::JPEG2000Codec, 640
gdcm::JPEGCodec, 652
gdcm::JPEGLSCodec, 660
gdcm::KAKADUCodec, 668
gdcm::PDFCodec, 814
gdcm::PGXCodec, 823
gdcm::PNMCodec, 857
gdcm::PVRGCodec, 900
gdcm::RAWCodec, 920
gdcm::RLECodec, 943
CanDecode
gdcm::AudioCodec, 169
gdcm::Decoder, 357
gdcm::DeltaEncodingCodec, 365
gdcm::ImageCodec, 567
gdcm::JPEG2000Codec, 641
gdcm::JPEGCodec, 652
gdcm::JPEGLSCodec, 661
gdcm::KAKADUCodec, 668
gdcm::PDFCodec, 814
gdcm::PGXCodec, 823
gdcm::PNMCodec, 858
gdcm::PVRGCodec, 901
gdcm::RAWCodec, 921
gdcm::RLECodec, 943
CanDisplay
gdcm::VR, 1337
CanEmptyTag
gdcm::Anonymizer, 114
CanRead
gdcm::Reader, 926
CanReadFile
vtkGDCMImageReader, 1350
vtkGDCMImageReader2, 1365
CanReadImage
gdcm::StreamImageReader, 1062
CanStoreLossy
gdcm::TransferSyntax, 1184
CanWriteFile
gdcm::StreamImageWriter, 1067
CAPI
gdcm::CryptoFactory, 283
CAPICryptoFactory
gdcm::CAPICryptoFactory, 234
CAPICryptographicMessageSyntax
gdcm::CAPICryptographicMessageSyntax, 236
CardiacElectrophysiologyWaveformStorage
gdcm::MediaStorage, 697
gdcm::UIDs, 1217
CardiacRelevantPatientInformationQuery
gdcm::UIDs, 1219
CEcho
gdcm::CompositeNetworkFunctions, 274
CFind
gdcm::CompositeNetworkFunctions, 274
Change
gdcm::FileChangeTransferSyntax, 472
gdcm::FileDecompressLookupTable, 476
gdcm::FileExplicitFilter, 482
gdcm::ImageChangePhotometricInterpretation, 552
gdcm::ImageChangePlanarConfiguration, 556
gdcm::ImageChangeTransferSyntax, 561
ChangeFMI
gdcm::FileExplicitFilter, 483
ChangeMonochrome
gdcm::ImageChangePhotometricInterpretation, 553
ChangeRGB2YBR
gdcm::ImageChangePhotometricInterpretation, 553
ChangeYBR2RGB
gdcm::ImageChangePhotometricInterpretation, 553
CharacterDataHandler
gdcm::TableReader, 1149
gdcm::XMLDictReader, 1475
gdcm::XMLPrivateDictReader, 1482
CheckDataElement
gdcm::FileStreamer, 511
CheckEvent
gdcm::AnonymizeEvent, 109
gdcm::DataEvent, 337
gdcm::DataSetEvent, 353
gdcm::Event, 450
gdcm::FileNameEvent, 500
gdcm::ProgressEvent, 897
CheckFileMetaInformationOff
gdcm::Writer, 1470
CheckFileMetaInformationOn
gdcm::Writer, 1470
CheckTemplateFileName
gdcm::FileStreamer, 511
ChestCADSRStorage
gdcm::UIDs, 1218
CipherTypes
gdcm::CryptographicMessageSyntax, 286
Clamp
gdcm, 64
Clean
gdcm::Cleaner, 249

clean
gdcm, 64
Cleaner
gdcm::Cleaner, 249
CleanupUnusedBits
gdcm::ImageCodec, 567
Clear
gdcm::Anonymizer, 115
gdcm::Bitmap, 199
gdcm::ByteValue, 228
gdcm::DataElement, 323
gdcm::DataSet, 342
gdcm::IOD, 613
gdcm::IODs, 619
gdcm::Item, 628
gdcm::LookupTable, 677
gdcm::Macro, 686
gdcm::Macros, 688
gdcm::Module, 723
gdcm::Modules, 731
gdcm::Preamble, 861
gdcm::SequenceOfFragments, 994
gdcm::SequenceOfItems, 1003
gdcm::SerieHelper, 1011
gdcm::Value, 1319
vtkGDCMMedicalImageProperties, 1387
vtkRTStructSetProperties, 1455
ClearInternalUIDs
gdcm::Anonymizer, 115
ClearPrivateTags
gdcm::Scanner2, 964
gdcm::StrictScanner2, 1086
ClearPublicTags
gdcm::Scanner2, 964
gdcm::StrictScanner2, 1086
ClearSkipTags
gdcm::Scanner, 953
gdcm::Scanner2, 964
gdcm::StrictScanner, 1076
gdcm::StrictScanner2, 1086
ClearTags
gdcm::Scanner, 953
gdcm::StrictScanner, 1076
Clone
gdcm::BoxRegion, 217
gdcm::ImageCodec, 568
gdcm::JPEG2000Codec, 641
gdcm::JPEGCodec, 652
gdcm::JPEGLSCodec, 661
gdcm::KAKADUCodec, 669
gdcm::PGXCodec, 823
gdcm::PNMCodec, 858
gdcm::PVRGCodec, 901
gdcm::RAWCodec, 921
gdcm::Region, 934
gdcm::RLECodec, 944
CM
gdcm::SegmentHelper::BasicCodedEntry, 192
cMaxEventID
gdcm::network, 83
cMaxStateID
gdcm::network, 83
CMove
gdcm::CompositeNetworkFunctions, 276
CMYK
gdcm::PhotometricInterpretation, 826
Code
gdcm::Coder, 259
gdcm::JPEG2000Codec, 641
gdcm::JPEGCodec, 653
gdcm::JPEGLSCodec, 661
gdcm::JSON, 665
gdcm::KAKADUCodec, 669
gdcm::PVRGCodec, 901
gdcm::RAWCodec, 921
gdcm::RLECodec, 944
CodeMeaning
gdcm::RealWorldValueMappingContent, 932
CodeString
gdcm::CodeString, 262, 263
CodeValue
gdcm::RealWorldValueMappingContent, 932
ColonCADSRStorage
gdcm::UIDs, 1222
Color
gdcm::terminal, 85
ColorArray
gdcm::SurfaceHelper, 1123
ColorPaletteQueryRetrieveInformationModelFIND
gdcm::UIDs, 1223
ColorPaletteQueryRetrieveInformationModelGET
gdcm::UIDs, 1223
ColorPaletteQueryRetrieveInformationModelMOVE
gdcm::UIDs, 1223
ColorPaletteStorage
gdcm::UIDs, 1223
ColorSoftcopyPresentationStateStorageSOPClass
gdcm::UIDs, 1217
Command
gdcm::Command, 266, 267
CommandDataSet
gdcm::CommandDataSet, 269
CommandTypes
gdcm::network::DIMSE, 393
Compatible
gdcm::VM, 1332
gdcm::VR, 1338
Component

gdcm::PersonName, 820
CompOperators
gdcm, 61
CompositeInstanceRetrieveWithoutBulkDataGET
gdcm::UIDs, 1222
CompositeInstanceRootRetrieveGET
gdcm::UIDs, 1222
CompositeInstanceRootRetrieveMOVE
gdcm::UIDs, 1222
CompositingPlanarMPRVolumetricPresentationStateStorage
gdcm::UIDs, 1221
Comprehensive3DSRStorage
gdcm::UIDs, 1222
ComprehensiveSR
gdcm::MediaStorage, 697
ComprehensiveSRStorage
gdcm::UIDs, 1218
ComprehensiveSRStorageTrialRetired
gdcm::UIDs, 1218
CompressionTypes
vtkGDCMImageWriter, 1379
Compute
gdcm::EquipmentManufacturer, 448
gdcm::MD5, 692
gdcm::SHA1, 1026
ComputeBoundingBox
gdcm::BoxRegion, 217
gdcm::Region, 934
ComputeBufferLength
gdcm::ImageRegionReader, 595
ComputeByteLength
gdcm::SequenceOfFragments, 994
ComputeDataElement
gdcm::DataSet, 342
ComputeDataSetMediaStorageSOPClass
gdcm::FileMetaInformation, 488
ComputeDataSetTransferSyntax
gdcm::FileMetaInformation, 488
ComputeDistAlongNormal
gdcm::DirectionCosines, 395
ComputedRadiographyImageStorage
gdcm::MediaStorage, 696
gdcm::UIDs, 1216
ComputeFile
gdcm::MD5, 692
gdcm::SHA1, 1026
ComputeFileMD5
gdcm::Testing, 1169
ComputeGroupLength
gdcm::DataSet, 342
ComputeInterceptSlopePixelType
gdcm::Rescaler, 937
ComputeLength
gdcm::ByteValue, 228
gdcm::Fragment, 523
gdcm::SequenceOfFragments, 994
gdcm::SequenceOfItems, 1003
ComputeLossyFlag
gdcm::Bitmap, 199
ComputeMD5
gdcm::Testing, 1169
ComputeMediaStorageFromModality
gdcm::ImageHelper, 583
ComputeMOSAICDimensions
gdcm::SplitMosaicFilter, 1056
ComputeMOSAICSliceNormal
gdcm::SplitMosaicFilter, 1056
ComputeMOSAICSlicePosition
gdcm::SplitMosaicFilter, 1057
ComputeNumberOfSurfaces
gdcm::SurfaceWriter, 1130
ComputeOffsetTable
gdcm::JPEGCodec, 653
ComputePixelAspectRatioFromPixelSpacing
gdcm::Spacing, 1054
ComputePixelTypeFromMinMax
gdcm::Rescaler, 937
ComputeSpacingFromImagePositionPatient
gdcm::ImageHelper, 584
ComputeTargetMediaStorage
gdcm::ImageWriter, 602
ComputeVR
gdcm::DataSetHelper, 355
ComputeZSpacing
gdcm::IPPSorter, 625
ConcatenatePDVBlobs
gdcm::network::PresentationDataValue, 879
ConcatenatePDVBlobsAsExplicit
gdcm::network::PresentationDataValue, 879
CONDENSED_STYLE
gdcm::Printer, 884
Conditional
gdcm::Usage, 1309
CONSOLE
gdcm::terminal, 85
const
gdcm::SOPClassUIDToIOD, 1045
const_iterator
gdcm::CodeString, 261
gdcm::LO, 671
gdcm::String< TDelimiter, TMaxLength, TPadChar >, 1094
const_reference
gdcm::CodeString, 261
gdcm::LO, 671
gdcm::String< TDelimiter, TMaxLength, TPadChar >, 1094
const_reverse_iterator

gdcm::CodeString, 261
gdcm::LO, 671
gdcm::String< TDelimiter, TMaxLength, TPadChar >, 1094
ConstCharWrapper
gdcm::ConstCharWrapper, 278
ConstIterator
gdcm::CSAHeaderDict, 303
gdcm::DataSet, 341
gdcm::Dict, 372
gdcm::Scanner, 951
gdcm::SequenceOfFragments, 993
gdcm::SequenceOfItems, 1001
gdcm::StrictScanner, 1074
Construct
gdcm::BaseRootQuery, 187
ConstructAbortPDU
gdcm::network::PDUFactory, 816
ConstructCEchoRQ
gdcm::network::CompositeMessageFactory, 271
ConstructCFindRQ
gdcm::network::CompositeMessageFactory, 271
ConstructCMoveRQ
gdcm::network::CompositeMessageFactory, 271
ConstructCStoreRQ
gdcm::network::CompositeMessageFactory, 272
ConstructCStoreRSP
gdcm::network::CompositeMessageFactory, 272
ConstructFromString
gdcm::DPath, 405
gdcm::TagPath, 1165
ConstructFromTagList
gdcm::TagPath, 1166
ConstructNAction
gdcm::network::NormalizedMessageFactory, 760
ConstructNCreate
gdcm::network::NormalizedMessageFactory, 760
ConstructNDelete
gdcm::network::NormalizedMessageFactory, 760
ConstructNEventReport
gdcm::network::NormalizedMessageFactory, 761
ConstructNGet
gdcm::network::NormalizedMessageFactory, 761
ConstructNSet
gdcm::network::NormalizedMessageFactory, 761
ConstructorType
gdcm::Dicts, 389
ConstructPDU
gdcm::network::PDUFactory, 816
ConstructPDV
gdcm::network::BaseCompositeMessage, 174
gdcm::network::BaseNormalizedMessage, 176
gdcm::network::CEchoRQ, 240
gdcm::network::CFindRQ, 245
gdcm::network::CMoveRQ, 255
gdcm::network::CStoreRQ, 311
gdcm::network::CStoreRSP, 313
gdcm::network::NActionRQ, 742
gdcm::network::NCreateRQ, 745
gdcm::network::NDeleteRQ, 748
gdcm::network::NEventReportRQ, 754
gdcm::network::NGetRQ, 757
gdcm::network::NSetRQ, 765
ConstructPDVByDataSet
gdcm::network::CEchoRSP, 242
gdcm::network::CFindCancelRQ, 243
gdcm::network::CFindRSP, 246
gdcm::network::CMoveCancelRq, 253
gdcm::network::CMoveRSP, 256
gdcm::network::NActionRSP, 743
gdcm::network::NCreateRSP, 746
gdcm::network::NDeleteRSP, 749
gdcm::network::NEventReportRSP, 755
gdcm::network::NGetRSP, 758
gdcm::network::NSetRSP, 767
ConstructQuery
gdcm::CompositeNetworkFunctions, 276, 277
gdcm::NormalizedNetworkFunctions, 762
ConstructReleasePDU
gdcm::network::PDUFactory, 816
ContentAssessmentResultsStorage
gdcm::UIDs, 1222
Convert
gdcm::DictConverter, 377
gdcm::ImageConverter, 578
ConvertRGBToPaletteColor
gdcm::IconImageGenerator, 537
ConvertToCXX
gdcm::DictConverter, 377
ConvertToUNC
gdcm::System, 1137
ConvertToXML
gdcm::DictConverter, 377
CornealTopographyMapStorage
gdcm::UIDs, 1222
CORONAL
gdcm::Orientation, 782
Create
gdcm::Preamble, 861
CreateCEchoPDU
gdcm::network::PDUFactory, 816
CreateCFindPDU
gdcm::network::PDUFactory, 816
CreateCMovePDU
gdcm::network::PDUFactory, 816
CreateCMSProvider
gdcm::CAPICryptoFactory, 235
gdcm::CryptoFactory, 284

gdcm::OpenSSLCryptoFactory, 772
gdcm::OpenSSLP7CryptoFactory, 777
CreateCStoreRQPDU
gdcm::network::PDUFactory, 817
CreateCStoreRSPPDU
gdcm::network::PDUFactory, 817
CreateDefaultUniqueSeriesIdentifier
gdcm::SerieHelper, 1011
CreateNActionPDU
gdcm::network::PDUFactory, 817
CreateNCreatePDU
gdcm::network::PDUFactory, 817
CreateNDeletePDU
gdcm::network::PDUFactory, 817
CreateNEventReportPDU
gdcm::network::PDUFactory, 817
CreateNGetPDU
gdcm::network::PDUFactory, 818
CreateNSetPDU
gdcm::network::PDUFactory, 818
CreateUniqueSeriesIdentifier
gdcm::SerieHelper, 1011
Cross
gdcm::DirectionCosines, 395
CrossDot
gdcm::DirectionCosines, 395
CryptoFactory
gdcm::CryptoFactory, 283
CryptographicMessageSyntax
gdcm::CryptographicMessageSyntax, 286
CryptoLib
gdcm::CryptoFactory, 283
CS
gdcm::VR, 1336
CSAElement
gdcm::CSAElement, 291
CSAHeader
gdcm::CSAHeader, 299
gdcm::DataSet, 351
CSAHeaderDict
gdcm::CSAHeaderDict, 304
CSAHeaderDictEntry
gdcm::CSAHeaderDictEntry, 307
CSAHeaderType
gdcm::CSAHeader, 298
CSANonImageStorage
gdcm::MediaStorage, 697
CSComp
gdcm, 58
CSD
gdcm::SegmentHelper::BasicCodedEntry, 192
CStore
gdcm::CompositeNetworkFunctions, 277
CSV
gdcm::SegmentHelper::BasicCodedEntry, 192
CT_private_ELE
gdcm::TransferSyntax, 1183
CTDefinedProcedureProtocolStorage
gdcm::UIDs, 1222
CTImageStorage
gdcm::MediaStorage, 696
gdcm::UIDs, 1216
CTPerformedProcedureProtocolStorage
gdcm::UIDs, 1222
Curve
gdcm::Curve, 315
vtkGDCMImageReader, 1359
vtkGDCMImageReader2, 1374
Curves
gdcm::Pixmap, 843
CV
gdcm::SegmentHelper::BasicCodedEntry, 193
CXX
gdcm::Printer, 884
cyan
gdcm::terminal, 85

DA
gdcm::VR, 1336
DAComp
gdcm, 59
DataElement
gdcm::DataElement, 323
gdcm::Value, 1320
DataElementSet
gdcm::DataSet, 341
DataElementType
gdcm::ModuleEntry, 729
DataEvent
gdcm::DataEvent, 336, 337
DataField
gdcm::CSAElement, 295
DataPtr
gdcm::CSAElement, 290
DATASET_FORMAT
gdcm::CSAHeader, 299
DataSetEvent
gdcm::DataSetEvent, 353
DataSetHandled
gdcm::network::ULConnectionCallback, 1285
DataSetHandles
gdcm::network::ULConnectionCallback, 1285
DataSetMS
gdcm::FileMetaInformation, 494
DataSetTS
gdcm::FileMetaInformation, 494
DataWasPassed
vtkImageMapToColors16, 1436

dCor
gdcm::MrProtocol::Vector3, 1322
DebugOff
gdcm::Trace, 1176
DebugOn
gdcm::Trace, 1176
Decode
gdcm::AudioCodec, 170
gdcm::Base64, 171
gdcm::Curve, 315
gdcm::Decoder, 357
gdcm::DeltaEncodingCodec, 365
gdcm::ImageCodec, 568
gdcm::JPEG2000Codec, 641
gdcm::JPEGCodec, 653
gdcm::JPEGLSCodec, 661, 662
gdcm::JSON, 665
gdcm::KAKADUCodec, 669
gdcm::LookupTable, 677
gdcm::PDFCodec, 814
gdcm::PVRGCodec, 901
gdcm::RAWCodec, 921
gdcm::RLECodec, 944
Decode8
gdcm::LookupTable, 677
DecodeByStreams
gdcm::Decoder, 357
gdcm::ImageCodec, 568
gdcm::JPEG12Codec, 633
gdcm::JPEG16Codec, 636
gdcm::JPEG2000Codec, 642
gdcm::JPEG8Codec, 647
gdcm::JPEGCodec, 653
gdcm::RAWCodec, 922
gdcm::RLECodec, 944
DecodeBytes
gdcm::RAWCodec, 922
DecodeExtent
gdcm::JPEG2000Codec, 642
gdcm::JPEGCodec, 653
gdcm::JPEGLSCodec, 662
gdcm::RLECodec, 945
Decompress
gdcm::Overlay, 788
Decrypt
gdcm::CAPICryptographicMessageSyntax, 237
gdcm::CryptographicMessageSyntax, 286
gdcm::OpenSSLCryptographicMessageSyntax, 774
gdcm::OpenSSLP7CryptographicMessageSyntax, 779
DeepCopy
vtkRTStructSetProperties, 1455
DEFAULT
gdcm::CryptoFactory, 283
Default
gdcm::FileMetaInformation, 488
DefinedProcedureProtocolInformationModelFIND
gdcm::UIDs, 1222
DefinedProcedureProtocolInformationModelGET
gdcm::UIDs, 1222
DefinedProcedureProtocolInformationModelMOVE
gdcm::UIDs, 1222
DefinedTerms
gdcm::DefinedTerms, 358
DefinePixelExtent
gdcm::StreamImageReader, 1062
gdcm::StreamImageWriter, 1067
DefineProperBufferLength
gdcm::StreamImageReader, 1062
gdcm::StreamImageWriter, 1067
DeflatedExplicitVRLittleEndian
gdcm::TransferSyntax, 1183
gdcm::UIDs, 1214
DeformableSpatialRegistrationStorage
gdcm::UIDs, 1217
Defs
gdcm::Defs, 359, 360
DeleteDirectory
gdcm::System, 1138
DeltaEncodingCodec
gdcm::DeltaEncodingCodec, 364
Derive
gdcm::FileDerivation, 479
DES3_CIPHER
gdcm::CryptographicMessageSyntax, 286
Description
gdcm::ModuleEntry, 727
DescriptionField
gdcm::ModuleEntry, 729
DetachedInterpretationManagementSOPClassRetired
gdcm::UIDs, 1216
DetachedPatientManagementMetaSOPClassRetired
gdcm::UIDs, 1215
DetachedPatientManagementSOPClass
gdcm::MediaStorage, 697
DetachedPatientManagementSOPClassRetired
gdcm::UIDs, 1215
DetachedResultsManagementMetaSOPClassRetired
gdcm::UIDs, 1215
DetachedResultsManagementSOPClassRetired
gdcm::UIDs, 1215
DetachedStudyManagementMetaSOPClassRetired
gdcm::UIDs, 1215
DetachedStudyManagementSOPClass
gdcm::MediaStorage, 697
DetachedStudyManagementSOPClassRetired
gdcm::UIDs, 1215
DetachedVisitManagementSOPClass

gdcm::MediaStorage, 697
DetachedVisitManagementSOPClassRetired
gdcm::UIDs, 1215
DetailSRStorageTrialRetired
gdcm::UIDs, 1218
DETECTOR
gdcm::Spacing, 1053
DetermineEventByPDU
gdcm::network::PDUFactory, 818
dicomAETitle
gdcm::UIDs, 1219
dicomApplicationCluster
gdcm::UIDs, 1219
DICOMApplicationContextName
gdcm::UIDs, 1215
dicomAssociationAcceptor
gdcm::UIDs, 1219
dicomAssociationInitiator
gdcm::UIDs, 1219
dicomAuthorizedNodeCertificateReference
gdcm::UIDs, 1219
dicomConfigurationRoot
gdcm::UIDs, 1220
DICOMContentMappingResource
gdcm::UIDs, 1223
DICOMControlledTerminology
gdcm::UIDs, 1215
dicomDescription
gdcm::UIDs, 1219
dicomDevice
gdcm::UIDs, 1220
dicomDeviceName
gdcm::UIDs, 1219
dicomDeviceSerialNumber
gdcm::UIDs, 1220
dicomDevicesRoot
gdcm::UIDs, 1220
DICOMDIR
gdcm::DICOMDIR, 366
DICOMDIRGenerator
gdcm::DICOMDIRGenerator, 368
dicomHostname
gdcm::UIDs, 1219
dicomInstalled
gdcm::UIDs, 1219
dicomInstitutionAddress
gdcm::UIDs, 1220
dicomInstitutionDepartmentName
gdcm::UIDs, 1220
dicomInstitutionName
gdcm::UIDs, 1220
dicomIssuerOfPatientID
gdcm::UIDs, 1220
dicomManufacturer
gdcm::UIDs, 1219
dicomManufacturerModelName
gdcm::UIDs, 1219
dicomNetworkAE
gdcm::UIDs, 1220
dicomNetworkConnection
gdcm::UIDs, 1220
dicomNetworkConnectionReference
gdcm::UIDs, 1219
dicomPort
gdcm::UIDs, 1219
dicomPreferredCalledAETitle
gdcm::UIDs, 1219
dicomPreferredCallingAETitle
gdcm::UIDs, 1220
dicomPrimaryDeviceType
gdcm::UIDs, 1219
dicomRelatedDeviceReference
gdcm::UIDs, 1219
dicomSoftwareVersion
gdcm::UIDs, 1219
dicomSOPClass
gdcm::UIDs, 1219
dicomStationName
gdcm::UIDs, 1220
dicomSupportedCharacterSet
gdcm::UIDs, 1220
dicomThisNodeCertificateReference
gdcm::UIDs, 1219
dicomTLSCyphersuite
gdcm::UIDs, 1219
dicomTransferCapability
gdcm::UIDs, 1220
dicomTransferRole
gdcm::UIDs, 1219
dicomTransferSyntax
gdcm::UIDs, 1219
DICOMUIDRegistry
gdcm::UIDs, 1215
dicomUniqueAETitle
gdcm::UIDs, 1220
dicomUniqueAETitlesRegistryRoot
gdcm::UIDs, 1220
dicomVendorData
gdcm::UIDs, 1219
DICOS2DAITStorage
gdcm::UIDs, 1222
DICOS3DAITStorage
gdcm::UIDs, 1222
DICOSCTImageStorage
gdcm::UIDs, 1222
DICOSDigitalXRayImageStorageForPresentation
gdcm::UIDs, 1222
DICOSDigitalXRayImageStorageForProcessing

gdcm::UIDs, 1222
DICOSQuadrupoleResonanceQRStorage
gdcm::UIDs, 1222
DICOSThreatDetectionReportStorage
gdcm::UIDs, 1222
Dict
gdcm::Dict, 372
gdcm::DictEntry, 384
DICT_DEBUG
gdcm::DictConverter, 376
DICT_DEFAULT
gdcm::DictConverter, 376
DICT_XML
gdcm::DictConverter, 376
DictConverter
gdcm::DictConverter, 377
DictEntry
gdcm::DictEntry, 381
DictPrinter
gdcm::DictPrinter, 386
Dicts
gdcm::CSAHeaderDict, 305
gdcm::Dict, 374
gdcm::Dicts, 389
gdcm::PrivateDict, 890
difference_type
gdcm::CodeString, 261
gdcm::LO, 671
gdcm::String< TDelimiter, TMaxLength, TPadChar >, 1094
DigitalIntraoralXRayImageStorageForPresentation
gdcm::UIDs, 1216
DigitalIntraoralXrayImageStorageForPresentation
gdcm::MediaStorage, 696
DigitalIntraoralXRayImageStorageForProcessing
gdcm::MediaStorage, 696
gdcm::UIDs, 1216
DigitalMammographyImageStorageForPresentation
gdcm::MediaStorage, 696
DigitalMammographyImageStorageForProcessing
gdcm::MediaStorage, 696
DigitalMammographyXRayImageStorageForPresentation
gdcm::UIDs, 1216
DigitalMammographyXRayImageStorageForProcessing
gdcm::UIDs, 1216
DigitalXRayImageStorageForPresentation
gdcm::MediaStorage, 696
gdcm::UIDs, 1216
DigitalXRayImageStorageForProcessing
gdcm::MediaStorage, 696
gdcm::UIDs, 1216
dim
gdcm::terminal, 85
Dimensions
gdcm::Bitmap, 210
gdcm::ImageCodec, 575
DirCosTolerance
gdcm::IPPSorter, 625
DirectionCosines
gdcm::DirectionCosines, 395
vtkGDCMImageReader, 1359
vtkGDCMImageReader2, 1374
Directory
gdcm::Directory, 399
DisplaySystemSOPClass
gdcm::UIDs, 1221
DisplaySystemSOPInstance
gdcm::UIDs, 1221
DoByteSwap
gdcm::ImageCodec, 568
DoIconImage
gdcm::PixmapWriter, 853
DoInvertMonochrome
gdcm::ImageCodec, 569
DoOverlayCleanup
gdcm::ImageCodec, 569
DoPaddedCompositePixelCode
gdcm::ImageCodec, 569
DoPlanarConfiguration
gdcm::ImageCodec, 569
doround
gdcm, 64
DoSimpleCopy
gdcm::ImageCodec, 569
Dot
gdcm::DirectionCosines, 396
DoYBR
gdcm::ImageCodec, 569
DoYBRFull422
gdcm::ImageCodec, 570
DPath
gdcm::DPath, 405
DropDuplicatePositions
gdcm::IPPSorter, 625
DS
gdcm::VR, 1336
dSag
gdcm::MrProtocol::Vector3, 1322
DT
gdcm::VR, 1336
DTComp
gdcm, 59
dTra
gdcm::MrProtocol::Vector3, 1322
Dumper
gdcm::Dumper, 409
DuplicateAttributeError
gdcm::Parser, 800

eAABORTPDUReceivedOpen
gdcm::network, 82
eAABORTRequest
gdcm::network, 82
eAASSOCIATE_RQPDUreceived
gdcm::network, 82
eAASSOCIATERequestLocalUser
gdcm::network, 82
eAASSOCIATEresponseAccept
gdcm::network, 82
eAASSOCIATEresponseReject
gdcm::network, 82
eArabic
gdcm, 62
eARELEASE_RPPDUReceived
gdcm::network, 82
eARELEASE_RQPDUReceivedOpen
gdcm::network, 82
eARELEASERequest
gdcm::network, 82
eARELEASEResponse
gdcm::network, 82
eARTIMTimerExpired
gdcm::network, 82
eASSOCIATE_ACPDUreceived
gdcm::network, 82
eASSOCIATE_RJPDUreceived
gdcm::network, 82
ECG12leadWaveformStorage
gdcm::UIDs, 1217
ECharSet
gdcm, 61
eCreateMMPS
gdcm, 62
eCyrillic
gdcm, 62
EddyCurrentImageStorage
gdcm::UIDs, 1222
EddyCurrentMultiframeImageStorage
gdcm::UIDs, 1222
EDGE
gdcm::MeshPrimitive, 712
eEventDoesNotExist
gdcm::network, 82
EEventID
gdcm::network, 81
eFind
gdcm, 63
eGB18030
gdcm, 62
eGreek
gdcm, 62
eHebrew
gdcm, 62
eImage
gdcm, 62
eJapanese
gdcm, 62
eJapaneseKanjiMultibyte
gdcm, 62
eJapaneseSupplementaryKanjiMultibyte
gdcm, 62
eKoreanHangulHanjaMultibyte
gdcm, 62
eLatin1
gdcm, 62
eLatin2
gdcm, 62
eLatin3
gdcm, 62
eLatin4
gdcm, 62
eLatin5
gdcm, 62
elem
gdcm::SerieHelper, 1013
Element
gdcm::Element< TVR, VM::VM1_n >, 419
eMove
gdcm, 63
Empty
gdcm::Anonymizer, 115
gdcm::BoxRegion, 217
gdcm::Cleaner, 249, 250
gdcm::DataElement, 323
gdcm::FileAnonymizer, 468
gdcm::Region, 934
EmptyMaskGenerator
gdcm::EmptyMaskGenerator, 439
EncapsulatedCDAStorage
gdcm::MediaStorage, 697
gdcm::UIDs, 1218
EncapsulatedDocument
gdcm::EncapsulatedDocument, 441
EncapsulatedPDFStorage
gdcm::MediaStorage, 697
gdcm::UIDs, 1218
EncapsulatedSTLStorage
gdcm::UIDs, 1222
Encode
gdcm::Base64, 171
EncodeBuffer
gdcm::JPEG12Codec, 633
gdcm::JPEG16Codec, 637
gdcm::JPEG8Codec, 647
gdcm::JPEGCodec, 654
EncodeBytes
gdcm::System, 1138

Encrypt
gdcm::CAPICryptographicMessageSyntax, 237
gdcm::CryptographicMessageSyntax, 287
gdcm::OpenSSLCryptographicMessageSyntax, 774
gdcm::OpenSSLP7CryptographicMessageSyntax, 779
End
gdcm::CSAHeaderDict, 304
gdcm::DataSet, 342, 343
gdcm::Dict, 373
gdcm::IODs, 620
gdcm::Scanner, 953
gdcm::Scanner2, 964
gdcm::SequenceOfFragments, 994, 995
gdcm::SequenceOfItems, 1003
gdcm::StrictScanner, 1076
gdcm::StrictScanner2, 1086
EndElement
gdcm::TableReader, 1149
gdcm::XMLDictReader, 1476
gdcm::XMLPrivateDictReader, 1482
EndElementHandler
gdcm::Parser, 798
EndFilter
gdcm::SimpleSubjectWatcher, 1033
EndWith
gdcm::Filename, 496
EnhancedCTImageStorage
gdcm::MediaStorage, 696
gdcm::UIDs, 1216
EnhancedMRColorImageStorage
gdcm::MediaStorage, 698
gdcm::UIDs, 1223
EnhancedMRImageStorage
gdcm::MediaStorage, 697
gdcm::UIDs, 1216
EnhancedPETImageStorage
gdcm::MediaStorage, 698
gdcm::UIDs, 1222
EnhancedSR
gdcm::MediaStorage, 697
EnhancedSRStorage
gdcm::UIDs, 1218
EnhancedUSVolumeStorage
gdcm::MediaStorage, 698
gdcm::UIDs, 1220
EnhancedXAImageStorage
gdcm::MediaStorage, 698
gdcm::UIDs, 1217
EnhancedXRFImageStorage
gdcm::UIDs, 1217
ENQueryType
gdcm, 62
EnumeratedValues
gdcm::EnumeratedValues, 446
ePatient
gdcm, 62
ePatientRootType
gdcm, 63
ePDATArequest
gdcm::network, 82
ePDATATFPDU
gdcm::network, 82
EQueryLevel
gdcm, 62
EQueryType
gdcm, 62
ERootType
gdcm, 63
ErrorOff
gdcm::Trace, 1177
ErrorOn
gdcm::Trace, 1177
ErrorType
gdcm::Parser, 799
eSeries
gdcm, 62
eSetMMPS
gdcm, 62
eSta10ReleaseCollisionAc
gdcm::network, 83
eSta11ReleaseCollisionRq
gdcm::network, 83
eSta12ReleaseCollisionAcLocal
gdcm::network, 83
eSta13AwaitingClose
gdcm::network, 83
eSta1Idle
gdcm::network, 82
eSta2Open
gdcm::network, 82
eSta3WaitLocalAssoc
gdcm::network, 82
eSta4LocalAssocDone
gdcm::network, 82
eSta5WaitRemoteAssoc
gdcm::network, 82
eSta6TransferReady
gdcm::network, 83
eSta7WaitRelease
gdcm::network, 83
eSta8WaitLocalRelease
gdcm::network, 83
eSta9ReleaseCollisionRqLocal
gdcm::network, 83
EstablishConnection
gdcm::network::ULConnectionManager, 1291
EstablishConnectionMove

gdcm::network::ULConnectionManager, 1291
eStaDoesNotExist
gdcm::network, 82
EStateID
gdcm::network, 82
eStudy
gdcm, 62
eStudyRootType
gdcm, 63
eThai
gdcm, 62
eTransportConnConfirmLocal
gdcm::network, 82
eTransportConnectionClosed
gdcm::network, 82
eTransportConnIndicLocal
gdcm::network, 82
eUnrecognizedPDUReceived
gdcm::network, 82
eUTF8
gdcm, 62
Event
gdcm::Event, 450
eWLMFind
gdcm, 63
Exception
gdcm::Exception, 453
Execute
gdcm::Command, 267
gdcm::EmptyMaskGenerator, 439
gdcm::MemberCommand< T >, 707
gdcm::SimpleMemberCommand< T >, 1030
ExecuteData
vtkGDCMImageReader, 1350
vtkGDCMThreadedImageReader, 1405
ExecuteInformation
vtkGDCMImageReader, 1351
vtkGDCMThreadedImageReader, 1405
ExecuteQuery
gdcm::StringFilter, 1099
Explicit
gdcm::TransferSyntax, 1182
ExplicitVRBigEndian
gdcm::TransferSyntax, 1183
gdcm::UIDs, 1214
ExplicitVRLittleEndian
gdcm::TransferSyntax, 1183
gdcm::UIDs, 1214
Explore
gdcm::Directory, 399
ExtensibleSRStorage
gdcm::UIDs, 1222
Extract
gdcm::IconImageFilter, 534
ExtractIconImages
gdcm::IconImageFilter, 534
ExtractVeproIconImages
gdcm::IconImageFilter, 534

F
gdcm::Printer, 887
gdcm::Reader, 931
gdcm::Validate, 1317
gdcm::XMLPrinter, 1480
FACET
gdcm::MeshPrimitive, 712
FallColorPaletteSOPInstance
gdcm::UIDs, 1220
FD
gdcm::VR, 1336
Fiducials
gdcm::Fiducials, 461
File
gdcm::File, 463
FileAnonymizer
gdcm::FileAnonymizer, 468
FileChangeTransferSyntax
gdcm::FileChangeTransferSyntax, 472
gdcm::ImageCodec, 575
FileDecompressLookupTable
gdcm::FileDecompressLookupTable, 475
FileDerivation
gdcm::FileDerivation, 478
FileExists
gdcm::System, 1138
FileExplicitFilter
gdcm::FileExplicitFilter, 482
FileIsDirectory
gdcm::System, 1138
FileIsSymlink
gdcm::System, 1138
FileList
gdcm, 59
FileMetaInformation
gdcm::FileMetaInformation, 487, 488
FileName
vtkGDCMPolyDataReader, 1394
Filename
gdcm::Filename, 495
filename
gdcm::FileWithName, 515
FileNameEvent
gdcm::FileNameEvent, 500
FilenameGenerator
gdcm::FilenameGenerator, 503
FileNameOrdering
gdcm::SerieHelper, 1011
FileNames

vtkGDCMImageReader, 1359
Filenames
gdcm::Sorter, 1051
FilenamesType
gdcm::DICOMDIRGenerator, 368
gdcm::Directory, 399
gdcm::FilenameGenerator, 503
FilenameType
gdcm::DICOMDIRGenerator, 368
gdcm::Directory, 399
gdcm::FilenameGenerator, 503
FileSet
gdcm::FileSet, 507
FileSize
gdcm::System, 1139
FileStreamer
gdcm::FileStreamer, 510
FilesType
gdcm::FileSet, 506
FileTime
gdcm::System, 1139
FileType
gdcm::FileSet, 507
FileWithName
gdcm::FileWithName, 515
Fill
gdcm::ByteValue, 228
FillFromDataSet
gdcm::FileMetaInformation, 488
FillMedicalImageInformation
vtkGDCMImageReader, 1351
vtkGDCMImageReader2, 1365
vtkGDCMPolyDataReader, 1391
FindContext
gdcm::network::ULConnection, 1279
FindCSAElementByName
gdcm::CSAHeader, 299
FindDataElement
gdcm::DataSet, 343
gdcm::Item, 628
gdcm::SequenceOfItems, 1004
FindDictEntry
gdcm::PrivateDict, 888
FindMacroEntry
gdcm::Macro, 686
FindModuleEntryInMacros
gdcm::Module, 724
FindMrProtocolByName
gdcm::MrProtocol, 739
FindNextDataElement
gdcm::DataSet, 343
FindPatientRootQuery
gdcm::FindPatientRootQuery, 517
FindPDBElementByName
gdcm::PDBHeader, 811
FindStudyRootQuery
gdcm::FindStudyRootQuery, 520
FirstRender
vtkImageColorViewer, 1429
FL
gdcm::VR, 1336
FLOAT16
gdcm::PixelFormat, 831
FLOAT32
gdcm::PixelFormat, 831
FLOAT64
gdcm::PixelFormat, 831
ForceRescale
vtkGDCMImageReader, 1360
vtkGDCMImageReader2, 1374
FormatDateTime
gdcm::System, 1139
Fragment
gdcm::Fragment, 523
FragmentVector
gdcm::SequenceOfFragments, 993
FromString
gdcm::StringFilter, 1099
FUJI
gdcm::EquipmentManufacturer, 447
FujiPrivateCRImageStorage
gdcm::MediaStorage, 698
FujiPrivateMammoCRImageStorage
gdcm::MediaStorage, 698

gdcm, 43
add1, 63
AEComp, 58
ASComp, 58
backslash, 64
BOOL_FUNCTION_PFILE_PFILE_POINTER, 58
Clamp, 64
clean, 64
CompOperators, 61
CSComp, 58
DAComp, 59
doround, 64
DTComp, 59
eArabic, 62
ECharSet, 61
eCreateMMPS, 62
eCyrillic, 62
eFind, 63
eGB18030, 62
eGreek, 62
eHebrew, 62
eImage, 62
eJapanese, 62

eJapaneseKanjiMultibyte, 62
eJapaneseSupplementaryKanjiMultibyte, 62
eKoreanHangulHanjaMultibyte, 62
eLatin1, 62
eLatin2, 62
eLatin3, 62
eLatin4, 62
eLatin5, 62
eMove, 63
ENQueryType, 62
ePatient, 62
ePatientRootType, 63
EQueryLevel, 62
EQueryType, 62
ERootType, 63
eSeries, 62
eSetMMPS, 62
eStudy, 62
eStudyRootType, 63
eThai, 62
eUTF8, 62
eWLMFind, 63
FileList, 59
GDCM_DIFFERENT, 61
GDCM_EQUAL, 61
GDCM_GREATER, 61
GDCM_GREATEROREQUAL, 61
GDCM_LESS, 61
GDCM_LESSOREQUAL, 61
GetVRFromTag, 64
GlobalInstance, 77
IconImage, 59
LD_ALL, 63
LD_NOSEQ, 63
LD_NOSHADOW, 63
LD_NOSHADOWSEQ, 63
LOComp, 59
LodModeType, 63
LTComp, 59
MacroEntry, 59
NestedMacroEntries, 60
operator!=, 65
operator<<, 65–75
operator>>, 75, 76
operator==, 75
PNComp, 60
Round, 76
roundat, 76
SHComp, 60
STComp, 60
TMComp, 60
TYPETOENCODING, 76
UCComp, 60
UIComp, 60
URComp, 61
UTComp, 61
VRBINARY, 77
x16printf, 76
gdcm::AbortEvent, 103
gdcm::AnonymizeEvent, 107
~AnonymizeEvent, 109
AnonymizeEvent, 108, 109
CheckEvent, 109
GetEventName, 109
GetTag, 109
MakeObject, 110
operator=, 110
Self, 108
SetTag, 110
Superclass, 108
gdcm::Anonymizer, 111
~Anonymizer, 114
Anonymizer, 114
BALCPProtect, 114
BasicApplicationLevelConfidentialityProfile, 114
CanEmptyTag, 114
Clear, 115
ClearInternalUIDs, 115
Empty, 115
GetBasicApplicationLevelConfidentialityProfileAttributes, 116
GetCryptographicMessageSyntax, 116
GetFile, 116
New, 116
RecurseDataSet, 117
Remove, 117
RemoveGroupLength, 117
RemovePrivateTags, 117
RemoveRetired, 118
Replace, 118
SetCryptographicMessageSyntax, 119
SetFile, 119
gdcm::AnyEvent, 120
gdcm::ApplicationEntity, 123
Internal, 125
IsValid, 124
MaxLength, 125
MaxNumberOfComponents, 125
Padding, 125
Print, 124
Separator, 125
SetBlob, 124
Squeeze, 125
gdcm::ASN1, 132
~ASN1, 133
ASN1, 133
operator=, 134
ParseDump, 134

ParseDumpFile, 134
TestPBKDF2, 134
gdcm::Attribute< Group, Element, TVR, TVM >, 136
ArrayType, 138
GDCM_STATIC_ASSERT, 139
GetAsDataElement, 139
GetDictVM, 139
GetDictVR, 140
GetNumberOfValues, 140
GetTag, 140
GetValue, 140, 141
GetValues, 141
GetVM, 141
GetVR, 141
Internal, 145
operator!=, 142
operator<, 142
operator==, 142
operator[], 142
Print, 143
Set, 143
SetByteValue, 143
SetByteValueNoSwap, 143
SetFromDataElement, 144
SetFromDataSet, 144
SetValue, 144
SetValues, 145
VMType, 139
gdcm::Attribute< Group, Element, TVR, VM::VM1 >, 146
ArrayType, 147
GDCM_STATIC_ASSERT, 148
GetAsDataElement, 148
GetDictVM, 149
GetDictVR, 149
GetNumberOfValues, 149
GetTag, 149
GetValue, 149
GetValues, 150
GetVM, 150
GetVR, 150
Internal, 152
operator!=, 150
operator<, 150
operator==, 150
Print, 151
Set, 151
SetByteValue, 151
SetByteValueNoSwap, 151
SetFromDataElement, 151
SetFromDataSet, 152
SetValue, 152
VMType, 148
gdcm::Attribute< Group, Element, TVR, VM::VM1_3 >, 153
GetVM, 153
gdcm::Attribute< Group, Element, TVR, VM::VM1_8 >, 154
GetVM, 155
gdcm::Attribute< Group, Element, TVR, VM::VM1_n >, 155
~Attribute, 157
ArrayType, 156
Attribute, 157
GDCM_STATIC_ASSERT, 157
GetAsDataElement, 157
GetDictVM, 158
GetDictVR, 158
GetNumberOfValues, 158
GetTag, 158
GetValue, 158
GetValues, 159
GetVM, 159
GetVR, 159
operator[], 159
Print, 159
Set, 160
SetByteValue, 160
SetFromDataElement, 160
SetFromDataSet, 160
SetNumberOfValues, 160
SetValue, 161
SetValues, 161
gdcm::Attribute< Group, Element, TVR, VM::VM2_2n >, 162
GetVM, 163
gdcm::Attribute< Group, Element, TVR, VM::VM2_n >, 163
GetVM, 164
gdcm::Attribute< Group, Element, TVR, VM::VM3_3n >, 165
GetVM, 166
gdcm::Attribute< Group, Element, TVR, VM::VM3_n >, 166
GetVM, 167
gdcm::AudioCodec, 168
~AudioCodec, 169
AudioCodec, 169
CanCode, 169
CanDecode, 169
Decode, 170
gdcm::Base64, 170
Base64, 171
Decode, 171
Encode, 171
GetDecodeLength, 172
GetEncodeLength, 172
operator=, 172
gdcm::BaseQuery, 179

~BaseQuery, 181
AddQueryDataSet, 181
BaseQuery, 181
GetAbstractSyntaxUID, 182
GetQueryDataSet, 182
GetSOPInstanceUID, 182
mDataSet, 184
mSopInstanceUID, 184
Print, 182
QueryFactory, 184
SetSearchParameter, 182, 183
SetSOPInstanceUID, 183
ValidateQuery, 183
ValidDataSet, 183
WriteHelpFile, 184
WriteQuery, 184
gdcm::BaseRootQuery, 185
~BaseRootQuery, 187
BaseRootQuery, 187
Construct, 187
GetQueryLevelFromQueryRoot, 187
GetQueryLevelFromString, 187
GetQueryLevelString, 187
GetTagListByLevel, 188
InitializeDataSet, 188
mHelpDescription, 189
mImage, 189
mPatient, 189
mRootType, 189
mSeries, 189
mStudy, 189
QueryFactory, 189
ValidateQuery, 188
gdcm::BasicOffsetTable, 193
BasicOffsetTable, 194
operator<<, 195
Read, 195
gdcm::Bitmap, 196
~Bitmap, 199
AreOverlaysInPixelData, 199
Bitmap, 199
Clear, 199
ComputeLossyFlag, 199
Dimensions, 210
GetBuffer, 200
GetBuffer2, 200
GetBufferLength, 200
GetColumns, 200
GetDataElement, 200, 201
GetDimension, 201
GetDimensions, 201
GetLUT, 201
GetNeedByteSwap, 202
GetNumberOfDimensions, 202
GetPhotometricInterpretation, 202
GetPixelFormat, 202, 203
GetPlanarConfiguration, 203
GetRows, 203
GetTransferSyntax, 203
ImageChangeTransferSyntax, 209
IsEmpty, 203
IsLossy, 204
IsTransferSyntaxCompatible, 204
LossyFlag, 210
LUT, 210
LUTPtr, 199
NeedByteSwap, 210
NumberOfDimensions, 210
PF, 210
PI, 210
PixelData, 211
PixmapReader, 209
PlanarConfiguration, 211
Print, 204
SetColumns, 204
SetDataElement, 204
SetDimension, 205
SetDimensions, 205
SetLossyFlag, 205
SetLUT, 205
SetNeedByteSwap, 206
SetNumberOfDimensions, 206
SetPhotometricInterpretation, 206
SetPixelFormat, 206
SetPlanarConfiguration, 207
SetRows, 207
SetTransferSyntax, 207
TryJPEG2000Codec, 207
TryJPEG2000Codec2, 208
TryJPEGCodec, 208
TryJPEGCodec2, 208
TryJPEGLSCodec, 208
TryKAKADUCodec, 208
TryPVRGCodec, 208
TryRAWCodec, 209
TryRLECodec, 209
TS, 211
UnusedBitsPresentInPixelData, 209
gdcm::BitmapToBitmapFilter, 211
~BitmapToBitmapFilter, 213
BitmapToBitmapFilter, 213
GetOutput, 213
GetOutputAsBitmap, 213
Input, 214
Output, 214
SetInput, 213
gdcm::BoxRegion, 214
~BoxRegion, 216

Area, 216
BoundingBox, 216
BoxRegion, 216
Clone, 217
ComputeBoundingBox, 217
Empty, 217
GetXMax, 217
GetXMin, 217
GetYMax, 218
GetYMin, 218
GetZMax, 218
GetZMin, 218
IsValid, 218
operator=, 218
Print, 219
SetDomain, 219
gdcm::ByteBuffer, 219
ByteBuffer, 220
Get, 220
GetStart, 220
ShiftEnd, 220
UpdatePosition, 221
gdcm::ByteSwap< T >, 221
Swap, 222
SwapFromSwapCodeIntoSystem, 222
SwapRange, 222
SwapRangeFromSwapCodeIntoSystem, 222
SystemIsBigEndian, 222
SystemIsLittleEndian, 223
gdcm::ByteSwapFilter, 223
~ByteSwapFilter, 224
ByteSwap, 224
ByteSwapFilter, 223, 224
operator=, 224
SetByteSwapTag, 224
gdcm::ByteValue, 225
~ByteValue, 227
Append, 228
ByteValue, 227
Clear, 228
ComputeLength, 228
Fill, 228
GetBuffer, 228
GetLength, 228
GetPointer, 229
GetVoidPointer, 229
IsEmpty, 230
IsPrintable, 230
operator const std::vector< char > &, 230
operator=, 230
operator==, 230, 231
Print, 231
PrintASCII, 231
PrintASCIIXML, 231
PrintGroupLength, 231
PrintHex, 231
PrintHexXML, 232
PrintPNXML, 232
Read, 232
SetLength, 232
SetLengthOnly, 232
Write, 233
WriteBuffer, 233
gdcm::CAPICryptoFactory, 234
CAPICryptoFactory, 234
CreateCMSProvider, 235
gdcm::CAPICryptographicMessageSyntax, 235
~CAPICryptographicMessageSyntax, 236
CAPICryptographicMessageSyntax, 236
Decrypt, 237
Encrypt, 237
GetCipherType, 237
GetInitialized, 237
ParseCertificateFile, 238
ParseKeyFile, 238
SetCipherType, 238
SetPassword, 238
gdcm::Cleaner, 247
~Cleaner, 249
Clean, 249
Cleaner, 249
Empty, 249, 250
GetFile, 250
New, 250
Preserve, 250
Remove, 250, 251
RemoveAllGroupLength, 251
RemoveAllIllegal, 251
RemoveAllMissingPrivateCreator, 251
RemoveMissingPrivateCreator, 251
Scrub, 252
SetFile, 252
gdcm::Codec, 257
gdcm::Coder, 258
~Coder, 258
CanCode, 259
Code, 259
InternalCode, 259
gdcm::CodeString, 260
CodeString, 262, 263
const_iterator, 261
const_reference, 261
const_reverse_iterator, 261
difference_type, 261
GetAsString, 263
IsValid, 263
iterator, 261
operator!=, 264

operator<<, 264
operator==, 264
pointer, 262
reference, 262
reverse_iterator, 262
Size, 264
size_type, 262
TrimInternal, 264
value_type, 262
gdcm::Command, 265
~Command, 267
Command, 266, 267
Execute, 267
operator=, 267
gdcm::CommandDataSet, 268
~CommandDataSet, 269
CommandDataSet, 269
Insert, 269
operator<<, 270
Read, 270
Replace, 270
Write, 270
gdcm::CompositeNetworkFunctions, 272
CEcho, 274
CFind, 274
CMove, 276
ConstructQuery, 276, 277
CStore, 277
KeyValuePairArrayType, 273
KeyValuePairType, 273
gdcm::ConstCharWrapper, 278
ConstCharWrapper, 278
operator const char *, 279
gdcm::CP246ExplicitDataElement, 279
GetLength, 280
Read, 281
ReadPreValue, 281
ReadValue, 281
ReadWithLength, 281
gdcm::CryptoFactory, 282
~CryptoFactory, 284
CAPI, 283
CreateCMSProvider, 284
CryptoFactory, 283
CryptoLib, 283
DEFAULT, 283
GetFactoryInstance, 284
OPENSSL, 283
OPENSSLP7, 283
gdcm::CryptographicMessageSyntax, 285
~CryptographicMessageSyntax, 286
AES128_CIPHER, 286
AES192_CIPHER, 286
AES256_CIPHER, 286
CipherTypes, 286
CryptographicMessageSyntax, 286
Decrypt, 286
DES3_CIPHER, 286
Encrypt, 287
GetCipherType, 287
operator=, 287
ParseCertificateFile, 287
ParseKeyFile, 288
SetCipherType, 288
SetPassword, 288
gdcm::CSAElement, 289
CSAElement, 291
DataField, 295
DataPtr, 290
GetByteValue, 291
GetKey, 291
GetName, 292
GetNoOfItems, 292
GetSyngoDT, 292
GetValue, 292
GetVM, 293
GetVR, 293
IsEmpty, 293
KeyField, 296
NameField, 296
NoOfItemsField, 296
operator<, 293
operator<<, 295
operator=, 293
operator==, 294
SetByteValue, 294
SetKey, 294
SetName, 294
SetNoOfItems, 294
SetSyngoDT, 294
SetValue, 295
SetVM, 295
SetVR, 295
SyngoDTField, 296
ValueMultiplicityField, 296
VRField, 296
gdcm::CSAHeader, 297
~CSAHeader, 299
CSAHeader, 299
CSAHeaderType, 298
DATASET_FORMAT, 299
FindCSAElementByName, 299
GetCSADataInfo, 299
GetCSAEEnd, 300
GetCSAElementByName, 300
GetCSAImageHeaderInfoTag, 300
GetCSASeriesHeaderInfoTag, 300
GetDataSet, 301

GetFormat, 301
GetInterfile, 301
GetMrProtocol, 301
INTERFILE, 299
LoadFromDataElement, 301
NOMAGIC, 299
operator<<, 302
Print, 302
SV10, 299
UNKNOWN, 299
ZEROED_OUT, 299
gdcm::CSAHeaderDict, 302
AddCSAHeaderDictEntry, 304
Begin, 304
ConstIterator, 303
CSAHeaderDict, 304
Dicts, 305
End, 304
GetCSAHeaderDictEntry, 305
IsEmpty, 305
Iterator, 303
LoadDefault, 305
MapCSAHeaderDictEntry, 304
operator<<, 305
operator=, 305
gdcm::CSAHeaderDictEntry, 306
CSAHeaderDictEntry, 307
GetDescription, 307
GetName, 307
GetVM, 308
GetVR, 308
operator<, 308
operator<<, 309
SetDescription, 308
SetName, 308
SetVM, 309
SetVR, 309
gdcm::CSAHeaderDictException, 310
gdcm::Curve, 313
~Curve, 315
Curve, 315
Decode, 315
GetAsPoints, 316
GetCurveDataDescriptor, 316
GetDataValueRepresentation, 316
GetDimensions, 316
GetGroup, 316
GetNumberOfCurves, 316
GetNumberOfPoints, 316
GetTypeOfData, 317
GetTypeOfDataDescription, 317
IsEmpty, 317
Print, 317
SetCoordinateStartValue, 317
SetCoordinateStepValue, 317
SetCurve, 318
SetCurveDataDescriptor, 318
SetCurveDescription, 318
SetDataValueRepresentation, 318
SetDimensions, 318
SetGroup, 318
SetNumberOfPoints, 319
SetTypeOfData, 319
Update, 319
gdcm::DataElement, 319
Clear, 323
DataElement, 323
Empty, 323
GetByteValue, 324
GetLength, 324
GetSequenceOfFragments, 324
GetTag, 325
GetValue, 325
GetValueAsSQ, 326
GetVL, 326
GetVR, 327
IsEmpty, 327
IsUndefinedLength, 327
operator<, 328
operator<<, 332
operator=, 328
operator==, 328
Read, 328
ReadOrSkip, 328
ReadPreValue, 329
ReadValue, 329
ReadValueWithLength, 329
ReadWithLength, 329
SetByteValue, 329
SetTag, 330
SetValue, 330
SetValueFieldLength, 331
SetVL, 331
SetVLToUndefined, 331
SetVR, 331
TagField, 332
ValueField, 333
ValueLengthField, 333
ValuePtr, 323
VRField, 333
Write, 332
gdcm::DataElementException, 334
gdcm::DataEvent, 334
~DataEvent, 336
CheckEvent, 337
DataEvent, 336, 337
GetData, 337
GetDataLength, 337

GetEventName, 337
MakeObject, 337
operator=, 338
Self, 336
SetData, 338
Superclass, 336
gdcm::DataSet, 338
Begin, 342
Clear, 342
ComputeDataElement, 342
ComputeGroupLength, 342
ConstIterator, 341
CSAHeader, 351
DataElementSet, 341
End, 342, 343
FindDataElement, 343
FindNextDataElement, 343
GetDataElement, 344
GetDEEnd, 344
GetDES, 345
GetLength, 345
GetMediaStorage, 345
GetPrivateCreator, 345
GetPrivateTag, 345
Insert, 346
InsertDataElement, 346
IsEmpty, 346
Iterator, 341
operator<<, 351
operator(), 346
operator=, 347
operator[], 347
Print, 347
Read, 347
ReadNested, 347
ReadSelectedPrivateTags, 348
ReadSelectedPrivateTagsWithLength, 348
ReadSelectedTags, 348
ReadSelectedTagsWithLength, 348
ReadUpToTag, 348
ReadUpToTagWithLength, 349
ReadWithLength, 349
Remove, 349
Replace, 349
ReplaceEmpty, 350
Size, 350
SizeType, 341
Write, 350
gdcm::DataSetEvent, 351
~DataSetEvent, 353
CheckEvent, 353
DataSetEvent, 353
GetDataSet, 354
GetEventName, 354
m_DataSet, 354
MakeObject, 354
operator=, 354
Self, 353
Superclass, 353
gdcm::DataSetHelper, 355
ComputeVR, 355
gdcm::Decoder, 356
~Decoder, 356
CanDecode, 357
Decode, 357
DecodeByStreams, 357
gdcm::DefinedTerms, 358
DefinedTerms, 358
gdcm::Defs, 358
~Defs, 360
Defs, 359, 360
GetIODFromFile, 360
GetIODNameFromMediaStorage, 360
GetIODs, 360
GetMacros, 361
GetModules, 361
GetTypeFromTag, 361
Global, 363
IsEmpty, 362
LoadDefaults, 362
LoadFromFile, 362
operator=, 362
Verify, 362
gdcm::DeltaEncodingCodec, 363
~DeltaEncodingCodec, 364
CanDecode, 365
Decode, 365
DeltaEncodingCodec, 364
gdcm::DICOMDIR, 365
DICOMDIR, 366
gdcm::DICOMDIRGenerator, 366
~DICOMDIRGenerator, 368
AddImageDirectoryRecord, 368
AddPatientDirectoryRecord, 368
AddSeriesDirectoryRecord, 369
AddStudyDirectoryRecord, 369
DICOMDIRGenerator, 368
FilenamesType, 368
FilenameType, 368
Generate, 369
GetFile, 369
GetScanner, 369
SetDescriptor, 369
SetFile, 370
SetFilenames, 370
SetRootDirectory, 370
gdcm::Dict, 371
AddDictEntry, 372

Begin, 373
ConstIterator, 372
Dict, 372
Dicts, 374
End, 373
GetDictEntry, 373
GetDictEntryByKeyword, 373
GetDictEntryByName, 373
GetKeywordFromTag, 374
IsEmpty, 374
Iterator, 372
LoadDefault, 374
MapDictEntry, 372
operator<<, 375
operator=, 374
gdcm::DictConverter, 375
~DictConverter, 377
AddGroupLength, 377
Convert, 377
ConvertToCXX, 377
ConvertToXML, 377
DICT_DEBUG, 376
DICT_DEFAULT, 376
DICT_XML, 376
DictConverter, 377
GetDictName, 378
GetInputFilename, 378
GetOutputFilename, 378
GetOutputType, 378
OutputTypes, 376
Readuint16, 378
ReadVM, 378
ReadVR, 379
SetDictName, 379
SetInputFileName, 379
SetOutputFileName, 379
SetOutputType, 379
WriteFooter, 379
WriteHeader, 380
gdcm::DictEntry, 380
Dict, 384
DictEntry, 381
GetKeyword, 381
GetName, 382
GetRetired, 382
GetVM, 382
GetVR, 382
IsUnique, 383
operator<<, 384
SetElementXX, 383
SetGroupXX, 383
SetKeyword, 383
SetName, 383
SetRetired, 384
SetVM, 384
SetVR, 384
gdcm::DictPrinter, 385
~DictPrinter, 387
DictPrinter, 386
Print, 387
PrintDataElement2, 387
PrintDataSet2, 387
gdcm::Dicts, 388
~Dicts, 389
ConstructorType, 389
Dicts, 389
GEMS, 389
GetConstructorString, 390
GetCSAHeaderDict, 390
GetDictEntry, 390
GetPrivateDict, 390, 391
GetPublicDict, 391
Global, 391
IsEmpty, 391
LoadDefaults, 391
operator<<, 392
operator=, 391
PHILIPS, 389
SIEMENS, 389
gdcm::DirectionCosines, 394
~DirectionCosines, 395
ComputeDistAlongNormal, 395
Cross, 395
CrossDot, 395
DirectionCosines, 395
Dot, 396
IsValid, 396
Normalize, 396
operator const double *, 397
Print, 397
SetFromString, 397
gdcm::Directory, 397
~Directory, 399
Directory, 399
Explore, 399
FilenamesType, 399
FilenameType, 399
GetDirectories, 400
GetFilenames, 400
GetToplevel, 400
Load, 400
operator<<, 401
Print, 401
gdcm::DirectoryHelper, 402
GetCTImageSeriesUIDs, 402
GetFilenamesFromSeriesUIDs, 402
GetFrameOfReference, 403
GetMRImageSeriesUIDs, 403

GetRTStructSeriesUIDs, 403
GetSeriesUIDsBySOPClassUID, 403
GetSOPClassUID, 403
GetStringValueFromTag, 403
LoadImageFromFiles, 404
RetrieveSOPInstanceUIDFromIndex, 404
RetrieveSOPInstanceUIDFromZPosition, 404
gdcm::DPath, 404
~DPath, 405
ConstructFromString, 405
DPath, 405
IsValid, 406
Match, 406
operator<, 406
operator<<, 406
Print, 406
gdcm::DummyValueGenerator, 407
Generate, 407
gdcm::Dumper, 408
~Dumper, 409
Dumper, 409
gdcm::Element< TVR, TVM >, 409
GetAsDataElement, 412
GetLength, 412
GetValue, 412
GetValues, 413
GetVM, 413
GetVR, 413
Internal, 415
operator[], 413
Print, 413
Read, 414
Set, 414
SetFromDataElement, 414
SetNoSwap, 414
SetValue, 415
Type, 412
Write, 415
gdcm::Element< TVR, VM::VM1_2 >, 416
Parent, 417
SetLength, 417
gdcm::Element< TVR, VM::VM1_n >, 417
~Element, 419
Element, 419
GetAsDataElement, 419
GetLength, 419
GetValue, 419, 420
GetVM, 420
GetVR, 420
operator=, 420
operator[], 420
Print, 420
Read, 421
Set, 421
SetArray, 421
SetFromDataElement, 421
SetLength, 421
SetNoSwap, 422
SetValue, 422
Type, 418
Write, 422
WriteASCII, 422
gdcm::Element< TVR, VM::VM2_2n >, 423
Parent, 424
SetLength, 424
gdcm::Element< TVR, VM::VM2_n >, 425
Parent, 426
SetLength, 426
gdcm::Element< TVR, VM::VM3_3n >, 426
Parent, 428
SetLength, 428
gdcm::Element< TVR, VM::VM3_4 >, 428
Parent, 429
SetLength, 430
gdcm::Element< TVR, VM::VM3_n >, 430
Parent, 431
SetLength, 431
gdcm::Element< VR::AS, VM::VM5 >, 432
GetLength, 432
Internal, 432
Print, 432
gdcm::Element< VR::OB, VM::VM1 >, 433
gdcm::Element< VR::OW, VM::VM1 >, 434
gdcm::ElementDisableCombinations< TVR, TVM >, 436
gdcm::ElementDisableCombinations< VR::OB, VM::VM1_n >, 437
gdcm::ElementDisableCombinations< VR::OW, VM::VM1_n >, 437
gdcm::EmptyMaskGenerator, 437
~EmptyMaskGenerator, 439
EmptyMaskGenerator, 439
Execute, 439
SetInputDirectory, 439
SetOutputDirectory, 440
SetSOPClassUIDMode, 440
SOPClassUIDMode, 439
UseGrayscaleSecondaryImageStorage, 439
UseOriginalSOPClassUID, 439
gdcm::EncapsulatedDocument, 440
EncapsulatedDocument, 441
gdcm::EncodingImplementation< T >, 441
gdcm::EncodingImplementation< VR::VRASCII >, 442
Read, 442
ReadComputeLength, 442
ReadNoSwap, 442
Write, 443
gdcm::EncodingImplementation< VR::VRBINARY >, 443
Read, 444

ReadComputeLength, 444
ReadNoSwap, 444
Write, 444
gdcm::EndEvent, 445
gdcm::EnumeratedValues, 446
EnumeratedValues, 446
gdcm::EquipmentManufacturer, 446
Compute, 448
FUJI, 447
GEMS, 447
HITACHI, 447
KODAK, 447
MARCONI, 447
PMS, 447
SIEMENS, 447
TOSHIBA, 447
Type, 447
TypeToString, 448
UNKNOWN, 447
gdcm::Event, 448
∼Event, 450
CheckEvent, 450
Event, 450
GetEventName, 450
MakeObject, 451
operator=, 451
Print, 451
gdcm::Exception, 452
∼Exception, 453
Exception, 453
GetDescription, 453
what, 454
gdcm::ExitEvent, 454
gdcm::ExplicitDataElement, 455
GetLength, 457
Read, 457
ReadPreValue, 457
ReadValue, 457
ReadWithLength, 457
Write, 457
gdcm::ExplicitImplicitDataElement, 458
GetLength, 460
Read, 460
ReadPreValue, 460
ReadValue, 460
ReadWithLength, 460
gdcm::Fiducials, 461
Fiducials, 461
gdcm::File, 461
∼File, 464
File, 463
GetDataSet, 464
GetHeader, 464, 465
operator<<, 466
Read, 465
SetDataSet, 465
SetHeader, 465
Write, 465
gdcm::FileAnonymizer, 466
∼FileAnonymizer, 468
Empty, 468
FileAnonymizer, 468
Remove, 468
Replace, 469
SetInputFileName, 469
SetOutputFileName, 469
Write, 470
gdcm::FileChangeTransferSyntax, 470
∼FileChangeTransferSyntax, 472
Change, 472
FileChangeTransferSyntax, 472
GetCodec, 472
New, 473
SetInputFileName, 473
SetOutputFileName, 473
SetTransferSyntax, 473
gdcm::FileDecompressLookupTable, 474
∼FileDecompressLookupTable, 476
Change, 476
FileDecompressLookupTable, 475
GetFile, 476
GetPixmap, 476
SetFile, 476
SetPixmap, 477
gdcm::FileDerivation, 477
∼FileDerivation, 478
AddDerivationDescription, 478
AddPurposeOfReferenceCodeSequence, 479
AddReference, 479
AddSourceImageSequence, 479
Derive, 479
FileDerivation, 478
GetFile, 479, 480
SetAppendDerivationHistory, 480
SetDerivationCodeSequenceCodeValue, 480
SetDerivationDescription, 480
SetFile, 480
SetPurposeOfReferenceCodeSequenceCodeValue, 481
gdcm::FileExplicitFilter, 481
∼FileExplicitFilter, 482
Change, 482
ChangeFMI, 483
FileExplicitFilter, 482
GetFile, 483
ProcessDataSet, 483
SetChangePrivateTags, 483
SetFile, 483

SetRecomputeItemLength, 484
SetRecomputeSequenceLength, 484
SetUseVRUN, 484
gdcm::FileMetaInformation, 485
~FileMetaInformation, 487
AppendImplementationClassUID, 488
ComputeDataSetMediaStorageSOPClass, 488
ComputeDataSetTransferSyntax, 488
DataSetMS, 494
DataSetTS, 494
Default, 488
FileMetaInformation, 487, 488
FillFromDataSet, 488
GetDataSetTransferSyntax, 489
GetFileMetaInformationVersion, 489
GetFullLength, 489
GetGDCMImplementationClassUID, 489
GetGDCMImplementationVersionName, 489
GetGDCMSourceApplicationEntityTitle, 489
GetImplementationClassUID, 490
GetImplementationVersionName, 490
GetMediaStorage, 490
GetMediaStorageAsString, 490
GetMetaInformationTS, 490
GetPreamble, 490
GetSourceApplicationEntityTitle, 491
Insert, 491
IsValid, 491
MetaInformationTS, 494
operator<<, 493
operator=, 491
Read, 491
ReadCompat, 491
ReadCompatInternal, 492
Replace, 492
SetDataSetTransferSyntax, 492
SetImplementationClassUID, 492
SetImplementationVersionName, 493
SetPreamble, 493
SetSourceApplicationEntityTitle, 493
Write, 493
gdcm::Filename, 494
EndWith, 496
Filename, 495
GetExtension, 496
GetFileName, 496
GetName, 496
GetPath, 496
IsEmpty, 496
IsIdentical, 497
Join, 497
operator const char *, 497
ToUnixSlashes, 497
ToWindowsSlashes, 497
gdcm::FileNameEvent, 498
~FileNameEvent, 500
CheckEvent, 500
FileNameEvent, 500
GetEventName, 501
GetFileName, 501
MakeObject, 501
operator=, 501
Self, 500
SetFileName, 501
Superclass, 500
gdcm::FilenameGenerator, 502
~FilenameGenerator, 504
FilenameGenerator, 503
FilenamesType, 503
FilenameType, 503
Generate, 504
GetFilename, 504
GetFilenames, 504
GetNumberOfFilenames, 504
GetPattern, 505
GetPrefix, 505
SetNumberOfFilenames, 505
SetPattern, 505
SetPrefix, 505
SizeType, 503
gdcm::FileSet, 506
AddFile, 507
FileSet, 507
FilesType, 506
FileType, 507
GetFiles, 507
operator<<, 508
SetFiles, 508
gdcm::FileStreamer, 508
~FileStreamer, 510
AppendToDataElement, 510
AppendToGroupDataElement, 511
CheckDataElement, 511
CheckTemplateFileName, 511
FileStreamer, 510
New, 511
ReserveDataElement, 512
ReserveGroupDataElement, 512
SetOutputFileName, 512
SetTemplateFileName, 512
StartDataElement, 512
StartGroupDataElement, 513
StopDataElement, 513
StopGroupDataElement, 513
gdcm::FileWithName, 514
filename, 515
FileWithName, 515
gdcm::FindPatientRootQuery, 516

FindPatientRootQuery, 517
GetAbstractSyntaxUID, 517
GetTagListByLevel, 517
InitializeDataSet, 517
QueryFactory, 518
ValidateQuery, 518
gdcm::FindStudyRootQuery, 519
FindStudyRootQuery, 520
GetAbstractSyntaxUID, 520
GetTagListByLevel, 520
InitializeDataSet, 520
QueryFactory, 521
ValidateQuery, 521
gdcm::Fragment, 522
ComputeLength, 523
Fragment, 523
GetLength, 524
operator<<, 525
Read, 524
ReadBacktrack, 524
ReadPreValue, 524
ReadValue, 524
Write, 525
gdcm::Global, 525
~Global, 527
Append, 527
GetDefs, 527
GetDicts, 527, 528
GetInstance, 528
Global, 526, 527
LoadResourcesFiles, 528
Locate, 528
operator<<, 529
operator=, 529
Prepend, 529
gdcm::GroupDict, 529
~GroupDict, 531
Add, 531
GetAbbreviation, 531
GetName, 531
GroupDict, 530
GroupStringVector, 530
Insert, 531
operator<<, 532
Size, 531
gdcm::IconImageFilter, 532
~IconImageFilter, 533
Extract, 534
ExtractIconImages, 534
ExtractVeproIconImages, 534
GetFile, 534
GetIconImage, 534
GetNumberOfIconImages, 535
IconImageFilter, 533
SetFile, 535
gdcm::IconImageGenerator, 535
~IconImageGenerator, 537
AutoPixelMinMax, 537
ConvertRGBToPaletteColor, 537
Generate, 537
GetIconImage, 537
GetPixmap, 538
IconImageGenerator, 536
SetOutputDimensions, 538
SetOutsideValuePixel, 538
SetPixelMinMax, 538
SetPixmap, 539
gdcm::ignore_char, 539
ignore_char, 539
m_char, 540
gdcm::Image, 540
~Image, 542
GetDirectionCosines, 543
GetIntercept, 543
GetOrigin, 543
GetSlope, 544
GetSpacing, 544
Image, 542
Print, 544
SetDirectionCosines, 544, 545
SetIntercept, 545
SetOrigin, 545, 546
SetSlope, 546
SetSpacing, 546
gdcm::ImageApplyLookupTable, 547
~ImageApplyLookupTable, 549
Apply, 549
ImageApplyLookupTable, 549
SetRGB8, 549
gdcm::ImageChangePhotometricInterpretation, 550
~ImageChangePhotometricInterpretation, 552
Change, 552
ChangeMonochrome, 553
ChangeRGB2YBR, 553
ChangeYBR2RGB, 553
GetPhotometricInterpretation, 553
ImageChangePhotometricInterpretation, 552
RGB2YBR, 553
SetPhotometricInterpretation, 553
YBR2RGB, 554
gdcm::ImageChangePlanarConfiguration, 554
~ImageChangePlanarConfiguration, 556
Change, 556
GetPlanarConfiguration, 556
ImageChangePlanarConfiguration, 556
RGBPixelsToRGBPlanes, 557
RGBPlanesToRGBPixels, 557
SetPlanarConfiguration, 557

gdcm::ImageChangeTransferSyntax, 558
~ImageChangeTransferSyntax, 560
Change, 561
GetTransferSyntax, 561
ImageChangeTransferSyntax, 560
SetCompressIconImage, 561
SetForce, 561
SetTransferSyntax, 561
SetUserCodec, 562
TryJPEG2000Codec, 562
TryJPEGCodec, 562
TryJPEGLSCodec, 562
TryRAWCodec, 563
TryRLECodec, 563
gdcm::ImageCodec, 563
~ImageCodec, 566
AppendFrameEncode, 566
AppendRowEncode, 567
CanCode, 567
CanDecode, 567
CleanupUnusedBits, 567
Clone, 568
Decode, 568
DecodeByStreams, 568
Dimensions, 575
DoByteSwap, 568
DoInvertMonochrome, 569
DoOverlayCleanup, 569
DoPaddedCompositePixelCode, 569
DoPlanarConfiguration, 569
DoSimpleCopy, 569
DoYBR, 569
DoYBRFull422, 570
FileChangeTransferSyntax, 575
GetDimensions, 570
GetHeaderInfo, 570
GetLossyFlag, 570
GetLUT, 570
GetNeedByteSwap, 570
GetNumberOfDimensions, 571
GetPhotometricInterpretation, 571
GetPixelFormat, 571
GetPlanarConfiguration, 571
ImageChangePhotometricInterpretation, 575
ImageCodec, 566
IsFrameEncoder, 571
IsLossy, 572
IsRowEncoder, 572
IsValid, 572
LossyFlag, 575
LUT, 575
LUTPtr, 566
NeedByteSwap, 575
NeedOverlayCleanup, 576
NumberOfDimensions, 576
PF, 576
PI, 576
PlanarConfiguration, 576
RequestPaddedCompositePixelCode, 576
RequestPlanarConfiguration, 576
SetDimensions, 572
SetLossyFlag, 572
SetLUT, 573
SetNeedByteSwap, 573
SetNeedOverlayCleanup, 573
SetNumberOfDimensions, 573
SetPhotometricInterpretation, 573
SetPixelFormat, 574
SetPlanarConfiguration, 574
StartEncode, 574
StopEncode, 574
gdcm::ImageConverter, 577
~ImageConverter, 577
Convert, 578
GetOuput, 578
ImageConverter, 577
SetInput, 578
gdcm::ImageFragmentSplitter, 579
~ImageFragmentSplitter, 581
GetFragmentSizeMax, 581
ImageFragmentSplitter, 581
SetForce, 581
SetFragmentSizeMax, 581
Split, 582
gdcm::ImageHelper, 582
ComputeMediaStorageFromModality, 583
ComputeSpacingFromImagePositionPatient, 584
GetDimensionsValue, 584
GetDirectionCosinesFromDataSet, 584
GetDirectionCosinesValue, 584
GetForcePixelSpacing, 585
GetForceRescaleInterceptSlope, 585
GetLUT, 585
GetOriginValue, 585
GetPhotometricInterpretationValue, 585
GetPixelFormatValue, 585
GetPlanarConfigurationValue, 586
GetPMSRescaleInterceptSlope, 586
GetPointerFromElement, 586
GetRealWorldValueMappingContent, 586
GetRescaleInterceptSlopeValue, 586
GetSpacingTagFromMediaStorage, 586
GetSpacingValue, 587
GetZSpacingTagFromMediaStorage, 587
SetDimensionsValue, 587
SetDirectionCosinesValue, 587
SetForcePixelSpacing, 587
SetForceRescaleInterceptSlope, 588

SetOriginValue, 588
SetPMSRescaleInterceptSlope, 588
SetRescaleInterceptSlopeValue, 588
SetSpacingValue, 588
gdcm::ImageReader, 589
~ImageReader, 591
GetImage, 591
ImageReader, 591
Read, 592
ReadACRNEMAImage, 592
ReadImage, 592
gdcm::ImageRegionReader, 593
~ImageRegionReader, 595
ComputeBufferLength, 595
GetRegion, 596
ImageRegionReader, 595
Read, 596
ReadInformation, 596
ReadIntoBuffer, 596
SetRegion, 597
gdcm::ImageToImageFilter, 597
~ImageToImageFilter, 599
GetInput, 599
GetOutput, 599
ImageToImageFilter, 599
gdcm::ImageWriter, 600
~ImageWriter, 602
ComputeTargetMediaStorage, 602
GetImage, 603
ImageWriter, 602
Write, 603
gdcm::ImplicitDataElement, 608
GetLength, 609
Read, 609
ReadPreValue, 609
ReadValue, 610
ReadValueWithLength, 610
ReadWithLength, 610
Write, 610
gdcm::InitializeEvent, 611
gdcm::IOD, 612
AddIODEntry, 613
Clear, 613
GetIODEntry, 613
GetNumberOfIODs, 613
GetTypeFromTag, 614
IOD, 613
MapIODEntry, 612
operator<<, 614
SizeType, 613
gdcm::IODEntry, 614
GetIE, 616
GetName, 616
GetRef, 616
GetUsage, 616
GetUsageType, 616
IODEntry, 615
operator<<, 617
SetIE, 616
SetName, 616
SetRef, 617
SetUsage, 617
gdcm::IODs, 617
AddIOD, 619
Begin, 619
Clear, 619
End, 620
GetIOD, 620
IODMapType, 618
IODMapTypeConstIterator, 618
IODName, 619
IODs, 619
operator<<, 620
gdcm::IPPSorter, 621
ComputeZSpacing, 625
DirCosTolerance, 625
DropDuplicatePositions, 625
GetDirectionCosinesTolerance, 623
GetZSpacing, 623
GetZSpacingTolerance, 623
IPPSorter, 622
SetComputeZSpacing, 623
SetDirectionCosinesTolerance, 623
SetDropDuplicatePositions, 624
SetZSpacingTolerance, 624
Sort, 624
ZSpacing, 625
ZTolerance, 625
gdcm::Item, 626
Clear, 628
FindDataElement, 628
GetDataElement, 628
GetLength, 628
GetNestedDataSet, 628, 629
InsertDataElement, 629
Item, 627, 628
operator<<, 630
Read, 629
SetNestedDataSet, 629
Write, 629
gdcm::IterationEvent, 630
gdcm::JPEG12Codec, 631
~JPEG12Codec, 633
DecodeByStreams, 633
EncodeBuffer, 633
GetHeaderInfo, 634
InternalCode, 634
IsStateSuspension, 634

JPEG12Codec, 633
gdcm::JPEG16Codec, 635
~JPEG16Codec, 636
DecodeByStreams, 636
EncodeBuffer, 637
GetHeaderInfo, 637
InternalCode, 637
IsStateSuspension, 637
JPEG16Codec, 636
gdcm::JPEG2000Codec, 638
~JPEG2000Codec, 640
AppendFrameEncode, 640
AppendRowEncode, 640
Bitmap, 645
CanCode, 640
CanDecode, 641
Clone, 641
Code, 641
Decode, 641
DecodeByStreams, 642
DecodeExtent, 642
GetHeaderInfo, 642
GetQuality, 642
GetRate, 643
ImageRegionReader, 645
IsFrameEncoder, 643
IsRowEncoder, 643
JPEG2000Codec, 640
SetMCT, 643
SetNumberOfResolutions, 643
SetNumberOfThreadsForDecompression, 643
SetQuality, 644
SetRate, 644
SetReversible, 644
SetTileSize, 644
StartEncode, 644
StopEncode, 644
gdcm::JPEG8Codec, 645
~JPEG8Codec, 647
DecodeByStreams, 647
EncodeBuffer, 647
GetHeaderInfo, 648
InternalCode, 648
IsStateSuspension, 648
JPEG8Codec, 647
gdcm::JPEGCodec, 649
~JPEGCodec, 651
AppendFrameEncode, 651
AppendRowEncode, 652
BitSample, 657
CanCode, 652
CanDecode, 652
Clone, 652
Code, 653
ComputeOffsetTable, 653
Decode, 653
DecodeByStreams, 653
DecodeExtent, 653
EncodeBuffer, 654
GetHeaderInfo, 654
GetLossless, 654
GetQuality, 654
ImageRegionReader, 657
IsFrameEncoder, 655
IsRowEncoder, 655
IsStateSuspension, 655
IsValid, 655
JPEGCodec, 651
Quality, 657
SetBitSample, 655
SetLossless, 655
SetPixelFormat, 656
SetQuality, 656
StartEncode, 656
StopEncode, 656
gdcm::JPEGLSCodec, 658
~JPEGLSCodec, 660
AppendFrameEncode, 660
AppendRowEncode, 660
CanCode, 660
CanDecode, 661
Clone, 661
Code, 661
Decode, 661, 662
DecodeExtent, 662
GetBufferLength, 662
GetHeaderInfo, 662
GetLossless, 663
ImageRegionReader, 664
IsFrameEncoder, 663
IsRowEncoder, 663
JPEGLSCodec, 660
SetBufferLength, 663
SetLossless, 663
SetLossyError, 663
StartEncode, 664
StopEncode, 664
gdcm::JSON, 664
~JSON, 665
Code, 665
Decode, 665
GetPrettyPrint, 666
JSON, 665
PrettyPrintOff, 666
PrettyPrintOn, 666
SetPrettyPrint, 666
gdcm::KAKADUCodec, 667
~KAKADUCodec, 668

CanCode, 668
CanDecode, 668
Clone, 669
Code, 669
Decode, 669
KAKADUCodec, 668
gdcm::LO, 670
const_iterator, 671
const_reference, 671
const_reverse_iterator, 671
difference_type, 671
IsValid, 673
iterator, 671
LO, 673
pointer, 672
reference, 672
reverse_iterator, 672
size_type, 672
Superclass, 672
value_type, 672
gdcm::LookupTable, 674
~LookupTable, 676
Allocate, 677
BitSample, 681
BLUE, 676
Clear, 677
Decode, 677
Decode8, 677
GetBitSample, 678
GetBufferAsRGBA, 678
GetLUT, 678
GetLUTDescriptor, 678
GetLUTLength, 678
GetPointer, 679
GRAY, 676
GREEN, 676
IncompleteLUT, 682
InitializeBlueLUT, 679
Initialized, 679
InitializeGreenLUT, 679
InitializeLUT, 679
InitializeRedLUT, 680
Internal, 682
IsRGB8, 680
LookupTable, 676
LookupTableType, 676
Print, 680
RED, 676
SetBlueLUT, 680
SetGreenLUT, 681
SetLUT, 681
SetRedLUT, 681
UNKNOWN, 676
WriteBufferAsRGBA, 681
gdcm::Macro, 684
AddMacroEntry, 685
ArrayIncludeMacrosType, 685
Clear, 686
FindMacroEntry, 686
GetMacroEntry, 686
GetName, 686
Macro, 685
MapModuleEntry, 685
operator<<, 687
SetName, 686
Verify, 686
gdcm::Macros, 687
AddMacro, 688
Clear, 688
GetMacro, 689
IsEmpty, 689
Macros, 688
ModuleMapType, 688
operator<<, 689
gdcm::MD5, 691
Compute, 692
ComputeFile, 692
gdcm::MediaStorage, 692
AmbulatoryECGWaveformStorage, 697
Audio, 699
BasicTextSR, 697
BasicVoiceAudioWaveformStorage, 697
BreastProjectionXRayImageStorageForPresentation, 698
BreastProjectionXRayImageStorageForProcessing, 698
BreastTomosynthesisImageStorage, 698
CardiacElectrophysiologyWaveformStorage, 697
ComprehensiveSR, 697
ComputedRadiographyImageStorage, 696
CSANonImageStorage, 697
CTImageStorage, 696
DetachedPatientManagementSOPClass, 697
DetachedStudyManagementSOPClass, 697
DetachedVisitManagementSOPClass, 697
DigitalIntraoralXrayImageStorageForPresentation, 696
DigitalIntraoralXRayImageStorageForProcessing, 696
DigitalMammographyImageStorageForPresentation, 696
DigitalMammographyImageStorageForProcessing, 696
DigitalXRayImageStorageForPresentation, 696
DigitalXRayImageStorageForProcessing, 696
EncapsulatedCDAStorage, 697
EncapsulatedPDFStorage, 697
EnhancedCTImageStorage, 696

EnhancedMRColorImageStorage, 698
EnhancedMRImageStorage, 697
EnhancedPETImageStorage, 698
EnhancedSR, 697
EnhancedUSVolumeStorage, 698
EnhancedXAImageStorage, 698
FujiPrivateCRImageStorage, 698
FujiPrivateMammoCRImageStorage, 698
GeneralECGWaveformStorage, 697
GeneralElectricMagneticResonanceImageStorage, 698
GEPrivate3DModelStorage, 698
GetModality, 699
GetModalityDimension, 699
GetMSString, 700
GetMSType, 700
GetNumberOfModality, 700
GetNumberOfMSString, 700
GetNumberOfMSType, 700
GetString, 701
GrayscaleSoftcopyPresentationStateStorageSOPClass, 697
GuessFromModality, 701
HangingProtocolStorage, 698
HardcopyColorImageStorage, 698
HardcopyGrayscaleImageStorage, 697
HemodynamicWaveformStorage, 697
IsImage, 701
IsUndefined, 701
IVOCTForPresentation, 698
IVOCTForProcessing, 698
KeyObjectSelectionDocument, 698
LeadECGWaveformStorage, 697
LegacyConvertedEnhancedCTImageStorage, 698
LegacyConvertedEnhancedMRImageStorage, 698
LegacyConvertedEnhancedPETImageStorage, 698
MammographyCADSR, 698
MediaStorage, 699
MediaStorageDirectoryStorage, 696
ModalityPerformedProcedureStepSOPClass, 698
MRImageStorage, 696
MRSpectroscopyStorage, 697
MS_END, 698
MSType, 696
MultiframeGrayscaleByteSecondaryCaptureImageStorage, 697
MultiframeGrayscaleWordSecondaryCaptureImageStorage, 697
MultiframeSingleBitSecondaryCaptureImageStorage, 697
MultiframeTrueColorSecondaryCaptureImageStorage, 697
NoObject, 699
NuclearMedicineImageStorage, 697
NuclearMedicineImageStorageRetired, 697
ObjectEnd, 699
ObjectType, 699
operator MSType, 702
operator<<, 703
OphthalmicPhotography16BitImageStorage, 698
OphthalmicPhotography8BitImageStorage, 698
OphthalmicTomographyImageStorage, 698
PDF, 699
PETImageStorage, 697
Philips3D, 697
PhilipsPrivateMRSyntheticImageStorage, 698
RawDataStorage, 697
RTDoseStorage, 697
RTImageStorage, 697
RTIonBeamsTreatmentRecordStorage, 698
RTIonPlanStorage, 698
RTPlanStorage, 697
RTStructureSetStorage, 697
RTTreatmentSummaryRecordStorage, 698
SecondaryCaptureImageStorage, 697
Segmentation, 699
SegmentationStorage, 698
SetFromDataSet, 702
SetFromFile, 702
SetFromHeader, 702
SetFromModality, 702
SetFromSourceImageSequence, 703
SpacialFiducialsStorage, 697
SpacialRegistrationStorage, 697
StandaloneCurveStorage, 697
StandaloneModalityLUTStorage, 697
StandaloneOverlayStorage, 697
StandaloneVOILUTStorage, 697
StudyComponentManagementSOPClass, 697
SurfaceSegmentationStorage, 698
ToshibaPrivateDataStorage, 698
UltrasoundImageStorage, 696
UltrasoundImageStorageRetired, 696
UltrasoundMultiFrameImageStorage, 696
UltrasoundMultiFrameImageStorageRetired, 696
URI, 699
Video, 699
VideoEndoscopicImageStorage, 698
VideoMicroscopicImageStorage, 698
VideoPhotographicImageStorage, 698
VLEndoscopicImageStorage, 698
VLMicroscopicImageStorage, 698
VLPhotographicImageStorage, 698
VLWholeSlideMicroscopyImageStorage, 698
Waveform, 699
XRay3DAngiographicImageStorage, 698
XRay3DCraniofacialImageStorage, 698
XRayAngiographicBiPlaneImageStorageRetired, 697

XRayAngiographicImageStorage, 697
XRayRadiationDoseSR, 698
XRayRadiofluoroscopingImageStorage, 697
gdcm::MemberCommand< T >, 703
~MemberCommand, 706
Execute, 707
m_ConstMemberFunction, 708
m_MemberFunction, 708
m_This, 709
MemberCommand, 706
New, 707
operator=, 707
Self, 705
SetCallbackFunction, 708
TConstMemberFunctionPointer, 706
TMemberFunctionPointer, 706
gdcm::MeshPrimitive, 709
~MeshPrimitive, 712
AddPrimitiveData, 712
EDGE, 712
FACET, 712
GetMPType, 712
GetMPTypeString, 712
GetNumberOfPrimitivesData, 713
GetPrimitiveData, 713
GetPrimitivesData, 713
GetPrimitiveType, 714
LINE, 712
MeshPrimitive, 712
MPType, 711
MPType_END, 712
PrimitiveData, 714
PrimitivesData, 711
PrimitiveType, 715
SetPrimitiveData, 714
SetPrimitivesData, 714
SetPrimitiveType, 714
TRIANGLE, 712
TRIANGLE_FAN, 712
TRIANGLE_STRIP, 712
VERTEX, 712
gdcm::ModalityPerformedProcedureStepCreateQuery, 715
GetAbstractSyntaxUID, 717
GetRequiredDataSet, 717
ModalityPerformedProcedureStepCreateQuery, 717
QueryFactory, 717
ValidateQuery, 717
gdcm::ModalityPerformedProcedureStepSetQuery, 718
GetAbstractSyntaxUID, 720
GetRequiredDataSet, 720
ModalityPerformedProcedureStepSetQuery, 720
QueryFactory, 720
ValidateQuery, 720
gdcm::ModifiedEvent, 721
gdcm::Module, 722
AddMacro, 723
AddModuleEntry, 723
ArrayIncludeMacrosType, 723
Clear, 723
FindModuleEntryInMacros, 724
GetModuleEntryInMacros, 724
GetName, 724
MapModuleEntry, 723
Module, 723
operator<<, 725
SetName, 724
Verify, 724
gdcm::ModuleEntry, 725
~ModuleEntry, 727
DataElementType, 729
Description, 727
DescriptionField, 729
GetDescription, 728
GetName, 728
GetType, 728
ModuleEntry, 727
Name, 729
operator<<, 729
SetDescription, 728
SetName, 728
SetType, 728
gdcm::Modules, 729
AddModule, 731
Clear, 731
GetModule, 731
IsEmpty, 731
ModuleMapType, 730
Modules, 730
operator<<, 731
gdcm::MovePatientRootQuery, 732
GetAbstractSyntaxUID, 734
GetTagListByLevel, 734
InitializeDataSet, 734
MovePatientRootQuery, 733
QueryFactory, 735
ValidateQuery, 734
gdcm::MoveStudyRootQuery, 735
GetAbstractSyntaxUID, 737
GetTagListByLevel, 737
InitializeDataSet, 737
MoveStudyRootQuery, 736
QueryFactory, 738
ValidateQuery, 737
gdcm::MrProtocol, 738
~MrProtocol, 739
FindMrProtocolByName, 739
GetMrProtocolByName, 739

GetSliceArray, 739
GetVersion, 740
Load, 740
MrProtocol, 739
operator<<, 740
Print, 740
gdcm::MrProtocol::Slice, 1036
Normal, 1036
Position, 1036
gdcm::MrProtocol::SliceArray, 1037
Slices, 1037
gdcm::MrProtocol::Vector3, 1322
dCor, 1322
dSag, 1322
dTra, 1322
gdcm::NestedModuleEntries, 750
AddModuleEntry, 752
GetModuleEntry, 752
GetNumberOfModuleEntries, 752
NestedModuleEntries, 751
operator<<, 752
SizeType, 751
gdcm::network, 77
cMaxEventID, 83
cMaxStateID, 83
eAABORTPDUReceivedOpen, 82
eAABORTRequest, 82
eAASSOCIATE_RQPDUreceived, 82
eAASSOCIATERequestLocalUser, 82
eAASSOCIATEresponseAccept, 82
eAASSOCIATEresponseReject, 82
eARELEASE_RPPDUReceived, 82
eARELEASE_RQPDUReceivedOpen, 82
eARELEASERequest, 82
eARELEASEResponse, 82
eARTIMTimerExpired, 82
eASSOCIATE_ACPDUreceived, 82
eASSOCIATE_RJPDUreceived, 82
eEventDoesNotExist, 82
EEventID, 81
ePDATArequest, 82
ePDATATFPDU, 82
eSta10ReleaseCollisionAc, 83
eSta11ReleaseCollisionRq, 83
eSta12ReleaseCollisionAcLocal, 83
eSta13AwaitingClose, 83
eSta1Idle, 82
eSta2Open, 82
eSta3WaitLocalAssoc, 82
eSta4LocalAssocDone, 82
eSta5WaitRemoteAssoc, 82
eSta6TransferReady, 83
eSta7WaitRelease, 83
eSta8WaitLocalRelease, 83
eSta9ReleaseCollisionRqLocal, 83
eStaDoesNotExist, 82
EStateID, 82
eTransportConnConfirmLocal, 82
eTransportConnectionClosed, 82
eTransportConnIndicLocal, 82
eUnrecognizedPDUReceived, 82
GetStateIndex, 83
gdcm::network::AAbortPDU, 87
AAbortPDU, 88
IsLastFragment, 88
Print, 88
Read, 88
SetReason, 89
SetSource, 89
Size, 89
Write, 89
gdcm::network::AAssociateACPDU, 90
AAssociateACPDU, 91
AAssociateRQPDU, 94
AddPresentationContextAC, 92
GetNumberOfPresentationContextAC, 92
GetPresentationContextAC, 92
GetUserInformation, 92
InitFromRQ, 92
IsLastFragment, 92
Print, 93
Read, 93
SetCalledAETitle, 93
SetCallingAETitle, 93
Size, 93
SizeType, 91
Write, 93
gdcm::network::AAssociateRJPDU, 94
AAssociateRJPDU, 95
IsLastFragment, 95
Print, 96
Read, 96
Size, 96
Write, 96
gdcm::network::AAssociateRQPDU, 97
AAssociateACPDU, 103
AAssociateRQPDU, 99
AddPresentationContext, 99
GetCalledAETitle, 99
GetCallingAETitle, 99
GetNumberOfPresentationContext, 100
GetPresentationContext, 100
GetPresentationContextByAbstractSyntax, 100
GetPresentationContextByID, 100
GetPresentationContexts, 100
GetReserved43_74, 100
GetUserInformation, 101
IsAETitleValid, 101

IsLastFragment, 101
PresentationContextArrayType, 98
Print, 101
Read, 101
SetCalledAETitle, 102
SetCallingAETitle, 102
SetUserInformation, 102
Size, 102
SizeType, 99
Write, 102
gdcm::network::AbstractSyntax, 104
AbstractSyntax, 105
GetAsDataElement, 105
GetName, 105
operator==, 105
Print, 105
Read, 105
SetName, 106
SetNameFromUID, 106
Size, 106
Write, 106
gdcm::network::ApplicationContext, 121
ApplicationContext, 122
GetName, 122
Print, 122
Read, 122
SetName, 122
Size, 122
Write, 123
gdcm::network::AReleaseRPPDU, 126
AReleaseRPPDU, 127
IsLastFragment, 127
Print, 127
Read, 127
Size, 128
Write, 128
gdcm::network::AReleaseRQPDU, 128
AReleaseRQPDU, 129
IsLastFragment, 129
Print, 130
Read, 130
Size, 130
Write, 130
gdcm::network::ARTIMTimer, 131
ARTIMTimer, 131
GetElapsedTime, 131
GetHasExpired, 132
GetTimeout, 132
SetTimeout, 132
Start, 132
Stop, 132
gdcm::network::AsynchronousOperationsWindowSub, 134
AsynchronousOperationsWindowSub, 135
Print, 135
Read, 135
Size, 135
Write, 136
gdcm::network::BaseCompositeMessage, 173
~BaseCompositeMessage, 174
ConstructPDV, 174
gdcm::network::BaseNormalizedMessage, 175
~BaseNormalizedMessage, 176
ConstructPDV, 176
gdcm::network::BasePDU, 177
~BasePDU, 178
IsLastFragment, 178
Print, 178
Read, 178
Size, 179
Write, 179
gdcm::network::CEchoRQ, 239
AffectedSOPClassUID, 240
ConstructPDV, 240
MessageID, 240
gdcm::network::CEchoRSP, 241
ConstructPDVByDataSet, 242
gdcm::network::CFind, 242
gdcm::network::CFindCancelRQ, 242
ConstructPDVByDataSet, 243
gdcm::network::CFindRQ, 244
ConstructPDV, 245
gdcm::network::CFindRSP, 245
ConstructPDVByDataSet, 246
gdcm::network::CMoveCancelRq, 253
ConstructPDVByDataSet, 253
gdcm::network::CMoveRQ, 254
ConstructPDV, 255
gdcm::network::CMoveRSP, 255
ConstructPDVByDataSet, 256
gdcm::network::CompositeMessageFactory, 271
ConstructCEchoRQ, 271
ConstructCFindRQ, 271
ConstructCMoveRQ, 271
ConstructCStoreRQ, 272
ConstructCStoreRSP, 272
gdcm::network::CStoreRQ, 310
ConstructPDV, 311
gdcm::network::CStoreRSP, 312
ConstructPDV, 313
gdcm::network::DIMSE, 392
C_CANCEL_RQ, 393
C_ECHO_RQ, 393
C_ECHO_RSP, 393
C_FIND_RQ, 393
C_FIND_RSP, 393
C_GET_RQ, 393
C_GET_RSP, 393

C_MOVE_RQ, 393
C_MOVE_RSP, 393
C_STORE_RQ, 393
C_STORE_RSP, 393
CommandTypes, 393
N_ACTION_RQ, 393
N_ACTION_RSP, 393
N_CREATE_RQ, 393
N_CREATE_RSP, 393
N_DELETE_RQ, 393
N_DELETE_RSP, 393
N_EVENT_REPORT_RQ, 393
N_EVENT_REPORT_RSP, 393
N_GET_RQ, 393
N_GET_RSP, 393
N_SET_RQ, 393
N_SET_RSP, 393
gdcm::network::ImplementationClassUIDSub, 604
ImplementationClassUIDSub, 604
Print, 604
Read, 604
Size, 605
Write, 605
gdcm::network::ImplementationUIDSub, 605
ImplementationUIDSub, 605
Write, 606
gdcm::network::ImplementationVersionNameSub, 606
ImplementationVersionNameSub, 606
Print, 607
Read, 607
Size, 607
Write, 607
gdcm::network::MaximumLengthSub, 689
GetMaximumLength, 690
MaximumLengthSub, 690
Print, 690
Read, 690
SetMaximumLength, 691
Size, 691
Write, 691
gdcm::network::NActionRQ, 741
ConstructPDV, 742
gdcm::network::NActionRSP, 742
ConstructPDVByDataSet, 743
gdcm::network::NCreateRQ, 744
ConstructPDV, 745
gdcm::network::NCreateRSP, 745
ConstructPDVByDataSet, 746
gdcm::network::NDeleteRQ, 747
ConstructPDV, 748
gdcm::network::NDeleteRSP, 748
ConstructPDVByDataSet, 749
gdcm::network::NEventReportRQ, 753
ConstructPDV, 754
gdcm::network::NEventReportRSP, 754
ConstructPDVByDataSet, 755
gdcm::network::NGetRQ, 756
ConstructPDV, 757
gdcm::network::NGetRSP, 757
ConstructPDVByDataSet, 758
gdcm::network::NormalizedMessageFactory, 760
ConstructNAction, 760
ConstructNCreate, 760
ConstructNDelete, 760
ConstructNEventReport, 761
ConstructNGet, 761
ConstructNSet, 761
gdcm::network::NSetRQ, 764
ConstructPDV, 765
gdcm::network::NSetRSP, 766
ConstructPDVByDataSet, 767
gdcm::network::PDataTFPDU, 803
AddPresentationDataValue, 804
GetNumberOfPresentationDataValues, 804
GetPresentationDataValue, 805
IsLastFragment, 805
PDataTFPDU, 804
Print, 805
Read, 805
ReadInto, 805
Size, 805
SizeType, 804
Write, 806
gdcm::network::PDUFactory, 815
ConstructAbortPDU, 816
ConstructPDU, 816
ConstructReleasePDU, 816
CreateCEchoPDU, 816
CreateCFindPDU, 816
CreateCMovePDU, 816
CreateCStoreRQPDU, 817
CreateCStoreRSPPDU, 817
CreateNActionPDU, 817
CreateNCreatePDU, 817
CreateNDeletePDU, 817
CreateNEventReportPDU, 817
CreateNGetPDU, 818
CreateNSetPDU, 818
DetermineEventByPDU, 818
GetPDVs, 818
gdcm::network::PresentationContextAC, 868
GetPresentationContextID, 869
GetReason, 869
GetTransferSyntax, 869
PresentationContextAC, 868
Print, 869
Read, 869
SetPresentationContextID, 869

SetReason, 870
SetTransferSyntax, 870
Size, 870
Write, 870
gdcm::network::PresentationContextRQ, 874
AddTransferSyntax, 876
GetAbstractSyntax, 876
GetNumberOfTransferSyntaxes, 876
GetPresentationContextID, 876
GetTransferSyntax, 876
GetTransferSyntaxes, 877
operator==, 877
PresentationContextRQ, 875
Print, 877
Read, 877
SetAbstractSyntax, 877
SetPresentationContextID, 877
Size, 878
SizeType, 875
Write, 878
gdcm::network::PresentationDataValue, 878
ConcatenatePDVBlobs, 879
ConcatenatePDVBlobsAsExplicit, 879
GetBlob, 879
GetIsCommand, 880
GetIsLastFragment, 880
GetMessageHeader, 880
GetPresentationContextID, 880
PresentationDataValue, 879
Print, 880
Read, 880
ReadInto, 880
SetBlob, 881
SetCommand, 881
SetDataSet, 881
SetLastFragment, 881
SetMessageHeader, 881
SetPresentationContextID, 882
Size, 882
Write, 882
gdcm::network::RoleSelectionSub, 947
Print, 947
Read, 948
RoleSelectionSub, 947
SetTuple, 948
Size, 948
Write, 948
gdcm::network::ServiceClassApplicationInformation, 1015
Print, 1015
Read, 1016
ServiceClassApplicationInformation, 1015
SetTuple, 1016
Size, 1016
Write, 1016
gdcm::network::SOPClassExtendedNegociationSub, 1042
Print, 1043
Read, 1043
SetTuple, 1043
Size, 1043
SOPClassExtendedNegociationSub, 1043
Write, 1044
gdcm::network::TableRow, 1152
∼TableRow, 1153
TableRow, 1153
transitions, 1153
gdcm::network::TransferSyntaxSub, 1187
GetName, 1187
operator==, 1188
Print, 1188
Read, 1188
SetName, 1188
SetNameFromUID, 1188
Size, 1188
TransferSyntaxSub, 1187
Write, 1189
gdcm::network::Transition, 1189
∼Transition, 1190
mAction, 1191
MakeNew, 1190
mEnd, 1191
Transition, 1190
gdcm::network::ULAction, 1235
∼ULAction, 1236
operator=, 1237
PerformAction, 1237
ULAction, 1236, 1237
gdcm::network::ULActionAA1, 1238
PerformAction, 1238
gdcm::network::ULActionAA2, 1239
PerformAction, 1240
gdcm::network::ULActionAA3, 1240
PerformAction, 1241
gdcm::network::ULActionAA4, 1242
PerformAction, 1242
gdcm::network::ULActionAA5, 1243
PerformAction, 1244
gdcm::network::ULActionAA6, 1244
PerformAction, 1245
gdcm::network::ULActionAA7, 1246
PerformAction, 1246
gdcm::network::ULActionAA8, 1247
PerformAction, 1248
gdcm::network::ULActionAE1, 1248
PerformAction, 1249
gdcm::network::ULActionAE2, 1250
PerformAction, 1250
gdcm::network::ULActionAE3, 1251

PerformAction, 1252
gdcm::network::ULActionAE4, 1252
PerformAction, 1253
gdcm::network::ULActionAE5, 1254
PerformAction, 1254
gdcm::network::ULActionAE6, 1255
PerformAction, 1256
gdcm::network::ULActionAE7, 1256
PerformAction, 1257
gdcm::network::ULActionAE8, 1258
PerformAction, 1258
gdcm::network::ULActionAR1, 1259
PerformAction, 1260
gdcm::network::ULActionAR10, 1260
PerformAction, 1261
gdcm::network::ULActionAR2, 1262
PerformAction, 1262
gdcm::network::ULActionAR3, 1263
PerformAction, 1264
gdcm::network::ULActionAR4, 1264
PerformAction, 1265
gdcm::network::ULActionAR5, 1266
PerformAction, 1266
gdcm::network::ULActionAR6, 1267
PerformAction, 1268
gdcm::network::ULActionAR7, 1268
PerformAction, 1269
gdcm::network::ULActionAR8, 1270
PerformAction, 1270
gdcm::network::ULActionAR9, 1271
PerformAction, 1272
gdcm::network::ULActionDT1, 1272
PerformAction, 1273
gdcm::network::ULActionDT2, 1274
PerformAction, 1274
gdcm::network::ULBasicCallback, 1275
~ULBasicCallback, 1276
GetDataSets, 1276
GetResponses, 1276
HandleDataSet, 1277
HandleResponse, 1277
ULBasicCallback, 1276
gdcm::network::ULConnection, 1277
~ULConnection, 1279
AddAcceptedPresentationContext, 1279
FindContext, 1279
GetAcceptedPresentationContexts, 1279, 1280
GetConnectionInfo, 1280
GetMaxPDUSize, 1280
GetPresentationContextACByID, 1280
GetPresentationContextIDFromPresentationContext, 1280
GetPresentationContextRQByID, 1280
GetPresentationContexts, 1281
GetProtocol, 1281
GetState, 1281
GetTimer, 1281
InitializeConnection, 1281
InitializeIncomingConnection, 1281
operator=, 1282
SetMaxPDUSize, 1282
SetPresentationContexts, 1282
SetState, 1282
StopProtocol, 1282
ULActionAE6, 1283
ULConnection, 1279
ULConnectionManager, 1283
gdcm::network::ULConnectionCallback, 1283
~ULConnectionCallback, 1284
DataSetHandled, 1285
DataSetHandles, 1285
HandleDataSet, 1285
HandleResponse, 1285
mImplicit, 1286
ResetHandledDataSet, 1285
SetImplicitFlag, 1285
ULConnectionCallback, 1284
gdcm::network::ULConnectionInfo, 1286
GetCalledAETitle, 1287
GetCalledComputerName, 1287
GetCalledIPAddress, 1287
GetCalledIPPort, 1287
GetCallingAETitle, 1287
GetMaxPDULength, 1287
Initialize, 1287
SetMaxPDULength, 1288
ULConnectionInfo, 1286
gdcm::network::ULConnectionManager, 1288
~ULConnectionManager, 1290
BreakConnection, 1291
BreakConnectionNow, 1291
EstablishConnection, 1291
EstablishConnectionMove, 1291
mConnection, 1295
mSecondaryConnection, 1295
mTransitions, 1295
RunEventLoop, 1291
RunMoveEventLoop, 1292
SendEcho, 1292
SendFind, 1292
SendMove, 1292
SendNAction, 1293
SendNCreate, 1293
SendNDelete, 1293
SendNEventReport, 1294
SendNGet, 1294
SendNSet, 1294
SendStore, 1295

ULConnectionManager, 1290
gdcm::network::ULEvent, 1296
~ULEvent, 1297
GetDataSetPos, 1297
GetEvent, 1297
GetIStream, 1297
GetPDUs, 1297
SetEvent, 1298
SetPDU, 1298
ULEvent, 1296, 1297
gdcm::network::ULTransitionTable, 1298
HandleEvent, 1299
PrintTable, 1299
ULTransitionTable, 1299
gdcm::network::ULWritingCallback, 1300
~ULWritingCallback, 1301
HandleDataSet, 1301
HandleResponse, 1301
SetDirectory, 1301
ULWritingCallback, 1301
gdcm::network::UserInformation, 1312
~UserInformation, 1312
AddRoleSelectionSub, 1313
AddSOPClassExtendedNegociationSub, 1313
GetMaximumLengthSub, 1313
operator=, 1313
Print, 1313
Read, 1314
Size, 1314
UserInformation, 1312
Write, 1314
gdcm::NoEvent, 759
gdcm::NormalizedNetworkFunctions, 761
ConstructQuery, 762
NAction, 762
NCreate, 763
NDelete, 763
NEventReport, 763
NGet, 763
NSet, 764
gdcm::Object, 767
~Object, 769
Object, 769
operator<<, 770
operator=, 770
Print, 770
Register, 770
SmartPointer, 771
UnRegister, 770
gdcm::OpenSSLCryptoFactory, 771
CreateCMSProvider, 772
InitOpenSSL, 772
OpenSSLCryptoFactory, 772
gdcm::OpenSSLCryptographicMessageSyntax, 773
~OpenSSLCryptographicMessageSyntax, 774
Decrypt, 774
Encrypt, 774
GetCipherType, 775
OpenSSLCryptographicMessageSyntax, 774
ParseCertificateFile, 775
ParseKeyFile, 775
SetCipherType, 775
SetPassword, 775
gdcm::OpenSSLP7CryptoFactory, 776
CreateCMSProvider, 777
OpenSSLP7CryptoFactory, 777
gdcm::OpenSSLP7CryptographicMessageSyntax, 778
~OpenSSLP7CryptographicMessageSyntax, 779
Decrypt, 779
Encrypt, 779
GetCipherType, 780
OpenSSLP7CryptographicMessageSyntax, 779
ParseCertificateFile, 780
ParseKeyFile, 780
SetCipherType, 780
SetPassword, 780
gdcm::Orientation, 781
~Orientation, 782
AXIAL, 782
CORONAL, 782
GetLabel, 783
GetMajorAxisFromPatientRelativeDirectionCosine, 783
GetObliquityThresholdCosineValue, 783
GetType, 783
OBLIQUE, 782
operator<<, 784
Orientation, 782
OrientationType, 782
Print, 783
SAGITTAL, 782
SetObliquityThresholdCosineValue, 784
UNKNOWN, 782
gdcm::Overlay, 784
~Overlay, 788
Decompress, 788
GetBitPosition, 788
GetBitsAllocated, 788
GetColumns, 789
GetDescription, 789
GetGroup, 789
GetOrigin, 789
GetOverlayData, 789
GetOverlayTypeAsString, 789
GetOverlayTypeFromString, 790
GetRows, 790
GetType, 790
GetTypeAsEnum, 790

GetUnpackBuffer, 790
GetUnpackBufferLength, 790
GrabOverlayFromPixelData, 791
Graphics, 787
Invalid, 787
IsEmpty, 791
IsInPixelData, 791
IsZero, 791
operator=, 791
Overlay, 788
OverlayType, 787
Print, 792
ROI, 787
SetBitPosition, 792
SetBitsAllocated, 792
SetColumns, 792
SetDescription, 792
SetFrameOrigin, 793
SetGroup, 793
SetNumberOfFrames, 793
SetOrigin, 793
SetOverlay, 793
SetRows, 794
SetType, 794
Update, 794
gdcm::ParseException, 795
~ParseException, 796
GetLastElement, 796
operator=, 796
ParseException, 796
SetLastElement, 797
gdcm::Parser, 797
~Parser, 800
DuplicateAttributeError, 800
EndElementHandler, 798
ErrorType, 799
GetBuffer, 800
GetCurrentByteIndex, 800
GetErrorCode, 800
GetErrorString, 801
GetUserData, 801
JunkAfterDocElementError, 800
NoElementsError, 800
NoError, 800
NoMemoryError, 800
Parse, 801
ParseBuffer, 801
Parser, 800
Process, 801
SetElementHandler, 801
SetUserData, 802
StartElementHandler, 798
SyntaxError, 800
TagMismatchError, 800
UndefinedEntityError, 800
UnexpectedStateError, 800
gdcm::Patient, 802
Patient, 802
gdcm::PDBElement, 806
GetName, 807
GetValue, 808
NameField, 809
operator<<, 808
operator==, 808
PDBElement, 807
SetName, 808
SetValue, 808
ValueField, 809
gdcm::PDBHeader, 809
~PDBHeader, 810
FindPDBElementByName, 811
GetPDBEEnd, 811
GetPDBElementByName, 811
GetPDBInfoTag, 811
LoadFromDataElement, 811
operator<<, 812
PDBHeader, 810
Print, 812
gdcm::PDFCodec, 812
~PDFCodec, 814
CanCode, 814
CanDecode, 814
Decode, 814
PDFCodec, 814
gdcm::PersonName, 818
Component, 820
GetMaxLength, 819
GetNumberOfComponents, 819
MaxLength, 820
MaxNumberOfComponents, 820
Padding, 821
Print, 819
Separator, 821
SetBlob, 820
SetComponents, 820
gdcm::PGXCodec, 821
~PGXCodec, 822
CanCode, 823
CanDecode, 823
Clone, 823
GetHeaderInfo, 823
PGXCodec, 822
Read, 823
Write, 824
gdcm::PhotometricInterpretation, 824
ARGB, 826
CMYK, 826
GetPIString, 826

GetPIType, 827
GetSamplesPerPixel, 827
GetString, 827
GetType, 827
HSV, 826
IsLossless, 827
IsLossy, 827
IsRetired, 827
IsSameColorSpace, 828
MONOCHROME1, 826
MONOCHROME2, 826
operator PIType, 828
operator<<, 828
PALETTE_COLOR, 826
PhotometricInterpretation, 826
PI_END, 826
PIType, 825
RGB, 826
UNKNOWN, 826
YBR_FULL, 826
YBR_FULL_422, 826
YBR_ICT, 826
YBR_PARTIAL_420, 826
YBR_PARTIAL_422, 826
YBR_RCT, 826
gdcm::PixelFormat, 828
Bitmap, 837
FLOAT16, 831
FLOAT32, 831
FLOAT64, 831
GetBitsAllocated, 832
GetBitsStored, 832
GetHighBit, 832
GetMax, 832
GetMin, 833
GetPixelRepresentation, 833
GetPixelSize, 833
GetSamplesPerPixel, 833
GetScalarType, 834
GetScalarTypeAsString, 834
INT12, 831
INT16, 831
INT32, 831
INT64, 831
INT8, 831
IsCompatible, 834
IsValid, 834
operator ScalarType, 835
operator!=, 835
operator<<, 837
operator==, 835
PixelFormat, 831
Print, 835
ScalarType, 830
SetBitsAllocated, 836
SetBitsStored, 836
SetHighBit, 836
SetPixelRepresentation, 836
SetSamplesPerPixel, 836
SetScalarType, 837
SINGLEBIT, 831
UINT12, 831
UINT16, 831
UINT32, 831
UINT64, 831
UINT8, 831
UNKNOWN, 831
Validate, 837
gdcm::Pixmap, 838
~Pixmap, 840
AreOverlaysInPixelData, 840
Curves, 843
GetCurve, 840, 841
GetIconImage, 841
GetNumberOfCurves, 841
GetNumberOfOverlays, 841
GetOverlay, 841, 842
Icon, 843
Overlays, 843
Pixmap, 840
Print, 842
RemoveOverlay, 842
SetIconImage, 842
SetNumberOfCurves, 842
SetNumberOfOverlays, 842
UnusedBitsPresentInPixelData, 843
gdcm::PixmapReader, 844
~PixmapReader, 846
GetPixmap, 846, 847
PixelData, 848
PixmapReader, 846
Read, 847
ReadACRNEMAImage, 847
ReadImage, 847
ReadImageInternal, 848
gdcm::PixmapToPixmapFilter, 848
~PixmapToPixmapFilter, 850
GetInput, 850
GetOutput, 850
GetOutputAsPixmap, 850
PixmapToPixmapFilter, 850
gdcm::PixmapWriter, 851
~PixmapWriter, 853
DoIconImage, 853
GetImage, 854
GetPixmap, 854
PixelData, 855
PixmapWriter, 853

PrepareWrite, 854
SetImage, 854
SetPixmap, 855
Write, 855
gdcm::PNMCodec, 856
~PNMCodec, 857
CanCode, 857
CanDecode, 858
Clone, 858
GetBufferLength, 858
GetHeaderInfo, 858
PNMCodec, 857
Read, 858
SetBufferLength, 859
Write, 859
gdcm::Preamble, 859
~Preamble, 861
Clear, 861
Create, 861
GetInternal, 861
GetLength, 861
IsEmpty, 862
IsValid, 862
operator<<, 863
operator=, 862
Preamble, 860, 861
Print, 862
Read, 862
Remove, 862
Valid, 863
Write, 863
gdcm::PresentationContext, 864
AbstractSyntax, 867
AddTransferSyntax, 866
GetAbstractSyntax, 866
GetNumberOfTransferSyntaxes, 866
GetPresentationContextID, 866
GetTransferSyntax, 866
ID, 867
operator==, 866
PresentationContext, 865
Print, 867
SetAbstractSyntax, 867
SetPresentationContextID, 867
SizeType, 865
TransferSyntaxArrayType, 865
TransferSyntaxes, 867
gdcm::PresentationContextGenerator, 870
AddFromFile, 872
AddPresentationContext, 872
GenerateFromFilenames, 872
GenerateFromUID, 873
GetDefaultTransferSyntax, 873
GetPresentationContexts, 873
PresentationContextArrayType, 872
PresentationContextGenerator, 872
SetDefaultTransferSyntax, 873
SetMergeModeToAbstractSyntax, 873
SetMergeModeToTransferSyntax, 874
SizeType, 872
gdcm::Printer, 882
~Printer, 884
CONDENSED_STYLE, 884
CXX, 884
F, 887
GetPrintStyle, 885
MaxPrintLength, 887
Print, 885
PrintDataElement, 885
PrintDataSet, 885
Printer, 884
PrintSQ, 886
PrintStyle, 887
PrintStyles, 884
SetColor, 886
SetFile, 886
SetStyle, 886
VERBOSE_STYLE, 884
XML, 884
gdcm::PrivateDict, 887
~PrivateDict, 888
AddDictEntry, 888
Dicts, 890
FindDictEntry, 888
GetDictEntry, 889
IsEmpty, 889
LoadDefault, 889
operator<<, 890
PrintXML, 889
PrivateDict, 888
RemoveDictEntry, 889
gdcm::PrivateTag, 890
GetAsDataElement, 892
GetOwner, 892
operator!=, 892, 893
operator<, 893
operator<<, 894
operator=, 893
operator==, 893
PrivateTag, 892
ReadFromCommaSeparatedString, 894
SetOwner, 894
gdcm::ProgressEvent, 895
~ProgressEvent, 897
CheckEvent, 897
GetEventName, 897
GetProgress, 897
MakeObject, 898

operator=, 898
ProgressEvent, 896, 897
Self, 896
SetProgress, 898
Superclass, 896
gdcm::PVRGCodec, 899
~PVRGCodec, 900
CanCode, 900
CanDecode, 901
Clone, 901
Code, 901
Decode, 901
PVRGCodec, 900
SetLossyFlag, 902
gdcm::PythonFilter, 902
~PythonFilter, 903
GetFile, 903
PythonFilter, 903
SetDicts, 903
SetFile, 903
ToPyObject, 903
UseDictAlways, 904
gdcm::QueryBase, 904
~QueryBase, 905
GetAllRequiredTags, 905
GetAllTags, 905
GetHierachicalSearchTags, 906
GetName, 906
GetOptionalTags, 906
GetQueryLevel, 906
GetRequiredTags, 906
GetUniqueTags, 907
gdcm::QueryFactory, 907
GetCharacterFromCurrentLocale, 908
ListCharSets, 908
ProduceCharacterSetDataElement, 908
ProduceQuery, 908
gdcm::QueryImage, 909
GetHierachicalSearchTags, 910
GetName, 910
GetOptionalTags, 910
GetQueryLevel, 910
GetRequiredTags, 911
GetUniqueTags, 911
gdcm::QueryPatient, 911
GetHierachicalSearchTags, 912
GetName, 912
GetOptionalTags, 913
GetQueryLevel, 913
GetRequiredTags, 913
GetUniqueTags, 913
gdcm::QuerySeries, 914
GetHierachicalSearchTags, 915
GetName, 915
GetOptionalTags, 915
GetQueryLevel, 915
GetRequiredTags, 915
GetUniqueTags, 916
gdcm::QueryStudy, 916
GetHierachicalSearchTags, 917
GetName, 917
GetOptionalTags, 918
GetQueryLevel, 918
GetRequiredTags, 918
GetUniqueTags, 918
gdcm::RAWCodec, 919
~RAWCodec, 920
CanCode, 920
CanDecode, 921
Clone, 921
Code, 921
Decode, 921
DecodeByStreams, 922
DecodeBytes, 922
GetHeaderInfo, 922
RAWCodec, 920
gdcm::Reader, 923
~Reader, 926
CanRead, 926
F, 931
GetFile, 926
GetStreamCurrentPosition, 927
GetStreamPtr, 927
Read, 927
ReadDataSet, 928
Reader, 926
ReadMetaInformation, 928
ReadPreamble, 928
ReadSelectedPrivateTags, 928
ReadSelectedTags, 929
ReadUpToTag, 929
SetFile, 929
SetFileName, 929
SetStream, 930
StreamImageReader, 930
gdcm::RealWorldValueMappingContent, 931
CodeMeaning, 932
CodeValue, 932
RealWorldValueIntercept, 932
RealWorldValueSlope, 932
gdcm::Region, 932
~Region, 933
Area, 933
Clone, 934
ComputeBoundingBox, 934
Empty, 934
IsValid, 934
Print, 934

Region, 933
gdcm::Rescaler, 935
~Rescaler, 937
ComputeInterceptSlopePixelType, 937
ComputePixelTypeFromMinMax, 937
GetIntercept, 937
GetSlope, 937
InverseRescale, 938
InverseRescaleFunctionIntoBestFit, 938
Rescale, 938
RescaleFunctionIntoBestFit, 938
Rescaler, 937
SetIntercept, 939
SetMinMaxForPixelType, 939
SetPixelFormat, 939
SetSlope, 939
SetTargetPixelType, 940
SetUseTargetPixelType, 940
gdcm::RLECodec, 940
~RLECodec, 942
AppendFrameEncode, 943
AppendRowEncode, 943
CanCode, 943
CanDecode, 943
Clone, 944
Code, 944
Decode, 944
DecodeByStreams, 944
DecodeExtent, 945
GetBufferLength, 945
GetHeaderInfo, 945
ImageRegionReader, 946
IsFrameEncoder, 945
IsRowEncoder, 945
RLECodec, 942
SetBufferLength, 946
SetLength, 946
StartEncode, 946
StopEncode, 946
gdcm::Scanner, 949
~Scanner, 952
AddPrivateTag, 952
AddSkipTag, 953
AddTag, 953
Begin, 953
ClearSkipTags, 953
ClearTags, 953
ConstIterator, 951
End, 953
GetAllFilenamesFromTagToValue, 954
GetFilenameFromTagToValue, 954
GetFilenames, 954
GetKeys, 954
GetMapping, 954
GetMappingFromTagToValue, 955
GetMappings, 955
GetOrderedValues, 955
GetValue, 955
GetValues, 955, 956
IsKey, 956
MappingType, 951
New, 956
operator<<, 957
Print, 956
PrintTable, 957
ProcessPublicTag, 957
Scan, 957
Scanner, 952
TagToValue, 952
TagToValueValueType, 952
ValuesType, 952
gdcm::Scanner2, 958
~Scanner2, 963
AddPrivateTag, 963
AddPublicTag, 963
AddSkipTag, 963
Begin, 963
ClearPrivateTags, 964
ClearPublicTags, 964
ClearSkipTags, 964
End, 964
GetAllFilenamesFromPrivateTagToValue, 964
GetAllFilenamesFromPublicTagToValue, 964
GetFilenameFromPrivateTagToValue, 964
GetFilenameFromPublicTagToValue, 965
GetFilenames, 965
GetKeys, 965
GetMappingFromPrivateTagToValue, 965
GetMappingFromPublicTagToValue, 965
GetPrivateMapping, 965
GetPrivateMappings, 966
GetPrivateOrderedValues, 966
GetPrivateValue, 966
GetPrivateValues, 966
GetPublicMapping, 966
GetPublicMappings, 966
GetPublicOrderedValues, 967
GetPublicValue, 967
GetPublicValues, 967
GetValues, 967
IsKey, 967
New, 968
operator<<, 969
Print, 968
PrintTable, 968
PrivateBegin, 968
PrivateConstIterator, 961
PrivateEnd, 968

PrivateMappingType, 961
PrivateTagToValue, 961
PrivateTagToValueValueType, 962
ProcessPrivateTag, 969
ProcessPublicTag, 969
PublicConstIterator, 962
PublicMappingType, 962
PublicTagToValue, 962
PublicTagToValueValueType, 962
Scan, 969
Scanner2, 963
ValuesType, 962
gdcm::Scanner2::ltstr, 682
operator(), 682
gdcm::Scanner::ltstr, 683
operator(), 683
gdcm::Segment, 970
~Segment, 973
AddSurface, 973
ALGOType, 972
ALGOType_END, 973
AnatomicRegion, 978
AnatomicRegionModifiers, 978
AUTOMATIC, 973
BasicCodedEntryVector, 972
GetALGOType, 973
GetALGOTypeString, 973
GetAnatomicRegion, 974
GetAnatomicRegionModifiers, 974
GetPropertyCategory, 974
GetPropertyType, 974, 975
GetPropertyTypeModifiers, 975
GetSegmentAlgorithmName, 975
GetSegmentAlgorithmType, 975
GetSegmentDescription, 975
GetSegmentLabel, 975
GetSegmentNumber, 976
GetSurface, 976
GetSurfaceCount, 976
GetSurfaces, 976
MANUAL, 973
PropertyCategory, 979
PropertyType, 979
PropertyTypeModifiers, 979
Segment, 973
SegmentAlgorithmName, 979
SegmentAlgorithmType, 979
SegmentDescription, 979
SegmentLabel, 979
SegmentNumber, 980
SEMIAUTOMATIC, 973
SetAnatomicRegion, 976
SetAnatomicRegionModifiers, 976
SetPropertyCategory, 977
SetPropertyType, 977
SetPropertyTypeModifiers, 977
SetSegmentAlgorithmName, 977
SetSegmentAlgorithmType, 977
SetSegmentDescription, 978
SetSegmentLabel, 978
SetSegmentNumber, 978
SetSurfaceCount, 978
SurfaceCount, 980
Surfaces, 980
SurfaceVector, 972
gdcm::SegmentedPaletteColorLookupTable, 980
~SegmentedPaletteColorLookupTable, 981
Print, 982
SegmentedPaletteColorLookupTable, 981
SetLUT, 982
gdcm::SegmentHelper, 84
gdcm::SegmentHelper::BasicCodedEntry, 190
BasicCodedEntry, 191
CM, 192
CSD, 192
CSV, 192
CV, 193
IsEmpty, 192
gdcm::SegmentReader, 983
~SegmentReader, 985
GetSegments, 985
Read, 985
ReadSegment, 986
ReadSegments, 986
SegmentMap, 984
SegmentReader, 985
Segments, 986
SegmentVector, 985
gdcm::SegmentWriter, 987
~SegmentWriter, 988
AddSegment, 989
GetNumberOfSegments, 989
GetSegment, 989
GetSegments, 989
PrepareWrite, 989
Segments, 990
SegmentVector, 988
SegmentWriter, 988
SetNumberOfSegments, 989
SetSegments, 990
Write, 990
gdcm::SequenceOfFragments, 991
AddFragment, 993
Begin, 994
Clear, 994
ComputeByteLength, 994
ComputeLength, 994
ConstIterator, 993

End, 994, 995
FragmentVector, 993
GetBuffer, 995
GetFragBuffer, 995
GetFragment, 995
GetLength, 995
GetNumberOfFragments, 996
GetTable, 996
Iterator, 993
New, 996
operator==, 996
Print, 997
Read, 997
ReadPreValue, 997
ReadValue, 997
SequenceOfFragments, 993
SetLength, 997
SizeType, 993
Write, 998
WriteBuffer, 998
gdcm::SequenceOfItems, 999
AddItem, 1002
AddNewUndefinedLengthItem, 1002
Begin, 1003
Clear, 1003
ComputeLength, 1003
ConstIterator, 1001
End, 1003
FindDataElement, 1004
GetItem, 1004
GetLength, 1004
GetNumberOfItems, 1004
IsEmpty, 1005
IsUndefinedLength, 1005
Items, 1007
ItemVector, 1001
Iterator, 1002
New, 1005
operator=, 1005
operator==, 1005
Print, 1006
Read, 1006
RemoveItemByIndex, 1006
SequenceLengthField, 1007
SequenceOfItems, 1002
SetLength, 1006
SetLengthToUndefined, 1006
SetNumberOfItems, 1007
SizeType, 1002
Write, 1007
gdcm::SerieHelper, 1008
~SerieHelper, 1010
AddFile, 1010
AddFileName, 1010
AddRestriction, 1010, 1011
Clear, 1011
CreateDefaultUniqueSeriesIdentifier, 1011
CreateUniqueSeriesIdentifier, 1011
elem, 1013
FileNameOrdering, 1011
GetFirstSingleSerieUIDFileSet, 1012
GetNextSingleSerieUIDFileSet, 1012
ImageNumberOrdering, 1012
ImagePositionPatientOrdering, 1012
ItFileSetHt, 1013
op, 1013
OrderFileList, 1012
Rule, 1009
SerieHelper, 1010
SerieRestrictions, 1009
SetDirectory, 1012
SetLoadMode, 1013
SetUseSeriesDetails, 1013
SingleSerieUIDFileSetHT, 1014
SingleSerieUIDFileSetmap, 1010
UserOrdering, 1013
value, 1014
gdcm::Series, 1014
Series, 1014
gdcm::ServiceClassUser, 1017
~ServiceClassUser, 1019
GetAETitle, 1020
GetCalledAETitle, 1020
GetTimeout, 1020
InitializeConnection, 1020
IsPresentationContextAccepted, 1020
New, 1020
operator=, 1021
SendEcho, 1021
SendFind, 1021
SendMove, 1021, 1022
SendStore, 1022
ServiceClassUser, 1019
SetAETitle, 1022
SetCalledAETitle, 1023
SetHostname, 1023
SetPort, 1023
SetPortSCP, 1023
SetPresentationContexts, 1024
SetTimeout, 1024
StartAssociation, 1024
StopAssociation, 1024
gdcm::SHA1, 1025
~SHA1, 1026
Compute, 1026
ComputeFile, 1026
operator=, 1026
SHA1, 1026

gdcm::SimpleMemberCommand< T >, 1027
~SimpleMemberCommand, 1030
Execute, 1030
m_MemberFunction, 1031
m_This, 1031
New, 1030
operator=, 1031
Self, 1029
SetCallbackFunction, 1031
SimpleMemberCommand, 1029
TMemberFunctionPointer, 1029
gdcm::SimpleSubjectWatcher, 1032
~SimpleSubjectWatcher, 1033
EndFilter, 1033
operator=, 1033
ShowAbort, 1033
ShowAnonymization, 1034
ShowData, 1034
ShowDataSet, 1034
ShowFileName, 1034
ShowIteration, 1034
ShowProgress, 1035
SimpleSubjectWatcher, 1033
StartFilter, 1035
TestAbortOff, 1035
TestAbortOn, 1035
gdcm::SmartPointer< ObjectType >, 1038
~SmartPointer, 1040
GetPointer, 1040
operator ObjectType *, 1041
operator*, 1041
operator->, 1041
operator=, 1041, 1042
SmartPointer, 1039, 1040
gdcm::SOPClassUIDToIOD, 1044
const, 1045
GetIOD, 1045
GetIODFromSOPClassUID, 1045
GetNumberOfSOPClassToIOD, 1045
GetSOPClassUIDFromIOD, 1045
GetSOPClassUIDToIOD, 1046
GetSOPClassUIDToIODs, 1046
gdcm::Sorter, 1046
~Sorter, 1049
AddSelect, 1049
Filenames, 1051
GetFilenames, 1049
operator<<, 1050
Print, 1049
Selection, 1051
SelectionMap, 1048
SetSortFunction, 1049
SetTagsToRead, 1050
Sort, 1050
Sorter, 1048
SortFunc, 1051
SortFunction, 1048
StableSort, 1050
TagsToRead, 1051
gdcm::Spacing, 1051
~Spacing, 1054
CALIBRATED, 1053
ComputePixelAspectRatioFromPixelSpacing, 1054
DETECTOR, 1053
MAGNIFIED, 1053
Spacing, 1053
SpacingType, 1053
UNKNOWN, 1053
gdcm::Spectroscopy, 1054
Spectroscopy, 1054
gdcm::SplitMosaicFilter, 1055
~SplitMosaicFilter, 1056
ComputeMOSAICDimensions, 1056
ComputeMOSAICSliceNormal, 1056
ComputeMOSAICSlicePosition, 1057
GetAcquisitionSize, 1057
GetFile, 1057
GetImage, 1057
GetNumberOfImagesInMosaic, 1058
SetFile, 1058
SetImage, 1058
Split, 1058
SplitMosaicFilter, 1056
gdcm::StartEvent, 1059
gdcm::static_assert_test< x >, 1060
gdcm::STATIC_ASSERTION_FAILURE< true >, 1060
value, 1060
gdcm::STATIC_ASSERTION_FAILURE< x >, 1060
gdcm::StreamImageReader, 1061
~StreamImageReader, 1062
CanReadImage, 1062
DefinePixelExtent, 1062
DefineProperBufferLength, 1062
GetDimensionsValueForResolution, 1063
GetFile, 1063
Read, 1063
ReadImageInformation, 1063
SetFileName, 1064
SetStream, 1064
StreamImageReader, 1061
gdcm::StreamImageWriter, 1065
~StreamImageWriter, 1067
CanWriteFile, 1067
DefinePixelExtent, 1067
DefineProperBufferLength, 1067
mElementOffsets, 1070
mElementOffsets1, 1070
mspFile, 1070

mWriter, 1070
mXMax, 1070
mXMin, 1071
mYMax, 1071
mYMin, 1071
mZMax, 1071
mZMin, 1071
SetFile, 1068
SetFileName, 1068
SetStream, 1068
StreamImageWriter, 1066
Write, 1068
WriteImageInformation, 1069
WriteImageSubregionRAW, 1069
WriteRawHeader, 1069
gdcm::StrictScanner, 1072
~StrictScanner, 1075
AddPrivateTag, 1075
AddSkipTag, 1076
AddTag, 1076
Begin, 1076
ClearSkipTags, 1076
ClearTags, 1076
ConstIterator, 1074
End, 1076
GetAllFilenamesFromTagToValue, 1077
GetFilenameFromTagToValue, 1077
GetFilenames, 1077
GetKeys, 1077
GetMapping, 1077
GetMappingFromTagToValue, 1077
GetMappings, 1078
GetOrderedValues, 1078
GetValue, 1078
GetValues, 1078
IsKey, 1079
MappingType, 1074
New, 1079
operator<<, 1080
Print, 1079
PrintTable, 1079
ProcessPublicTag, 1080
Scan, 1080
StrictScanner, 1075
TagToValue, 1075
TagToValueValueType, 1075
ValuesType, 1075
gdcm::StrictScanner2, 1081
~StrictScanner2, 1085
AddPrivateTag, 1085
AddPublicTag, 1085
AddSkipTag, 1086
Begin, 1086
ClearPrivateTags, 1086
ClearPublicTags, 1086
ClearSkipTags, 1086
End, 1086
GetAllFilenamesFromPrivateTagToValue, 1087
GetAllFilenamesFromPublicTagToValue, 1087
GetFilenameFromPrivateTagToValue, 1087
GetFilenameFromPublicTagToValue, 1087
GetFilenames, 1087
GetKeys, 1087
GetMappingFromPrivateTagToValue, 1088
GetMappingFromPublicTagToValue, 1088
GetPrivateMapping, 1088
GetPrivateMappings, 1088
GetPrivateOrderedValues, 1088
GetPrivateValue, 1088
GetPrivateValues, 1089
GetPublicMapping, 1089
GetPublicMappings, 1089
GetPublicOrderedValues, 1089
GetPublicValue, 1089
GetPublicValues, 1089
GetValues, 1090
IsKey, 1090
New, 1090
operator<<, 1092
Print, 1090
PrintTable, 1090
PrivateBegin, 1091
PrivateConstIterator, 1084
PrivateEnd, 1091
PrivateMappingType, 1084
PrivateTagToValue, 1084
PrivateTagToValueValueType, 1084
ProcessPrivateTag, 1091
ProcessPublicTag, 1091
PublicConstIterator, 1084
PublicMappingType, 1084
PublicTagToValue, 1084
PublicTagToValueValueType, 1085
Scan, 1091
StrictScanner2, 1085
ValuesType, 1085
gdcm::StrictScanner2::ltstr, 683
operator(), 683
gdcm::StrictScanner::ltstr, 684
operator(), 684
gdcm::String< TDelimiter, TMaxLength, TPadChar >, 1092
const_iterator, 1094
const_reference, 1094
const_reverse_iterator, 1094
difference_type, 1094
IsValid, 1097
iterator, 1095

operator const char *, 1097
pointer, 1095
reference, 1095
reverse_iterator, 1095
size_type, 1095
String, 1096
Trim, 1097
Truncate, 1097
value_type, 1095
gdcm::StringFilter, 1098
~StringFilter, 1099
ExecuteQuery, 1099
FromString, 1099
GetFile, 1100
SetDicts, 1100
SetFile, 1100
StringFilter, 1099
ToString, 1100, 1101
ToStringPair, 1101, 1102
UseDictAlways, 1102
gdcm::Study, 1102
Study, 1102
gdcm::Subject, 1103
~Subject, 1105
AddObserver, 1105
GetCommand, 1105
HasObserver, 1105
InvokeEvent, 1106
RemoveAllObservers, 1106
RemoveObserver, 1106
Subject, 1104
gdcm::Surface, 1107
~Surface, 1110
GetAlgorithmFamily, 1111
GetAlgorithmName, 1111
GetAlgorithmVersion, 1111
GetAxisOfRotation, 1111
GetCenterOfRotation, 1111
GetFiniteVolume, 1112
GetManifold, 1112
GetMaximumPointDistance, 1112
GetMeanPointDistance, 1112
GetMeshPrimitive, 1112
GetNumberOfSurfacePoints, 1113
GetNumberOfVectors, 1113
GetPointCoordinatesData, 1113
GetPointPositionAccuracy, 1113
GetPointsBoundingBoxCoordinates, 1113
GetProcessingAlgorithm, 1114
GetRecommendedDisplayCIELabValue, 1114
GetRecommendedDisplayGrayscaleValue, 1114
GetRecommendedPresentationOpacity, 1114
GetRecommendedPresentationType, 1115
GetSTATES, 1115
GetSTATESString, 1115
GetSurfaceComments, 1115
GetSurfaceNumber, 1115
GetSurfaceProcessing, 1115
GetSurfaceProcessingDescription, 1115
GetSurfaceProcessingRatio, 1116
GetVectorAccuracy, 1116
GetVectorCoordinateData, 1116
GetVectorDimensionality, 1116
GetVIEWType, 1116
GetVIEWTypeString, 1116
NO, 1110
POINTS, 1110
SetAlgorithmFamily, 1117
SetAlgorithmName, 1117
SetAlgorithmVersion, 1117
SetAxisOfRotation, 1117
SetCenterOfRotation, 1117
SetFiniteVolume, 1117
SetManifold, 1118
SetMaximumPointDistance, 1118
SetMeanPointDistance, 1118
SetMeshPrimitive, 1118
SetNumberOfSurfacePoints, 1118
SetNumberOfVectors, 1118
SetPointCoordinatesData, 1119
SetPointPositionAccuracy, 1119
SetPointsBoundingBoxCoordinates, 1119
SetProcessingAlgorithm, 1119
SetRecommendedDisplayCIELabValue, 1119, 1120
SetRecommendedDisplayGrayscaleValue, 1120
SetRecommendedPresentationOpacity, 1120
SetRecommendedPresentationType, 1120
SetSurfaceComments, 1120
SetSurfaceNumber, 1120
SetSurfaceProcessing, 1121
SetSurfaceProcessingDescription, 1121
SetSurfaceProcessingRatio, 1121
SetVectorAccuracy, 1121
SetVectorCoordinateData, 1121
SetVectorDimensionality, 1121
STATES, 1109
STATES_END, 1110
SURFACE, 1110
Surface, 1110
UNKNOWN, 1110
VIEWType, 1110
VIEWType_END, 1110
WIREFRAME, 1110
YES, 1110
gdcm::SurfaceHelper, 1122
ColorArray, 1123
RecommendedDisplayCIELabToRGB, 1123
RGBToRecommendedDisplayCIELab, 1124

RGBToRecommendedDisplayGrayscale, 1124
gdcm::SurfaceReader, 1125
~SurfaceReader, 1127
GetNumberOfSurfaces, 1127
Read, 1128
ReadPointMacro, 1128
ReadSurface, 1128
ReadSurfaces, 1128
SurfaceReader, 1127
gdcm::SurfaceWriter, 1129
~SurfaceWriter, 1130
ComputeNumberOfSurfaces, 1130
GetNumberOfSurfaces, 1130
NumberOfSurfaces, 1131
PrepareWrite, 1131
PrepareWritePointMacro, 1131
SetNumberOfSurfaces, 1131
SurfaceWriter, 1130
Write, 1131
gdcm::SwapCode, 1132
BadBigEndian, 1133
BadLittleEndian, 1133
BigEndian, 1133
GetIndex, 1133
GetSwapCodeString, 1133
LittleEndian, 1133
operator SwapCode::SwapCodeType, 1134
operator<<, 1134
SwapCode, 1133
SwapCodeType, 1133
Unknown, 1133
gdcm::SwapperDoOp, 1134
Swap, 1134
SwapArray, 1135
gdcm::SwapperNoOp, 1135
Swap, 1135
SwapArray, 1136
gdcm::System, 1136
ConvertToUNC, 1137
DeleteDirectory, 1138
EncodeBytes, 1138
FileExists, 1138
FileIsDirectory, 1138
FileIsSymlink, 1138
FileSize, 1139
FileTime, 1139
FormatDateTime, 1139
GetCurrentDateTime, 1139
GetCurrentModuleFileName, 1140
GetCurrentProcessFileName, 1140
GetCurrentResourcesDirectory, 1140
GetCWD, 1140
GetHostName, 1140
GetLastSystemError, 1140
GetLocaleCharset, 1141
GetPermissions, 1141
GetTimezoneOffsetFromUTC, 1141
MakeDirectory, 1141
ParseDateTime, 1141, 1142
RemoveFile, 1142
SetPermissions, 1142
StrCaseCmp, 1142
StrNCaseCmp, 1143
StrSep, 1143
StrTokR, 1143
gdcm::Table, 1144
~Table, 1145
GetTableEntry, 1145
InsertEntry, 1146
MapTableEntry, 1145
operator<<, 1146
operator=, 1146
Table, 1145
TableInternal, 1146
gdcm::TableEntry, 1147
~TableEntry, 1147
TableEntry, 1147
gdcm::TableReader, 1148
~TableReader, 1149
CharacterDataHandler, 1149
EndElement, 1149
GetDefs, 1149
GetFilename, 1150
HandleIOD, 1150
HandleIODEntry, 1150
HandleMacro, 1150
HandleMacroEntry, 1150
HandleMacroEntryDescription, 1150
HandleModule, 1150
HandleModuleEntry, 1151
HandleModuleEntryDescription, 1151
HandleModuleInclude, 1151
Read, 1151
SetFilename, 1151
StartElement, 1151
TableReader, 1149
gdcm::Tag, 1153
bytes, 1164
GetElement, 1156
GetElementTag, 1156
GetGroup, 1157
GetLength, 1157
GetPrivateCreator, 1157
IsGroupLength, 1157
IsGroupXX, 1158
IsIllegal, 1158
IsPrivate, 1158
IsPrivateCreator, 1158

IsPublic, 1159
operator!=, 1159
operator<, 1159
operator<<, 1163
operator<=, 1159
operator>>, 1163
operator=, 1159
operator==, 1160
operator[], 1160
PrintAsContinuousString, 1160
PrintAsContinuousUpperCaseString, 1160
PrintAsPipeSeparatedString, 1161
Read, 1161
ReadFromCommaSeparatedString, 1161
ReadFromContinuousString, 1161
ReadFromPipeSeparatedString, 1161
SetElement, 1162
SetElementTag, 1162
SetGroup, 1162
SetPrivateCreator, 1163
Tag, 1155, 1156
tag, 1164
tags, 1164
Write, 1163
gdcm::TagPath, 1164
~TagPath, 1165
ConstructFromString, 1165
ConstructFromTagList, 1166
IsValid, 1166
Print, 1166
Push, 1166
TagPath, 1165
gdcm::terminal, 84
Attribute, 85
black, 85
blink, 85
blue, 85
bright, 85
Color, 85
CONSOLE, 85
cyan, 85
dim, 85
green, 85
hidden, 85
magenta, 85
Mode, 85
red, 85
reset, 85
reverse, 85
setattribute, 86
setbgcolor, 86
setfgcolor, 86
setmode, 86
underline, 85
VT100, 85
white, 85
yellow, 85
gdcm::Testing, 1167
~Testing, 1169
ComputeFileMD5, 1169
ComputeMD5, 1169
GetDataExtraRoot, 1169
GetDataRoot, 1170
GetFileName, 1170
GetFileNames, 1170
GetLossyFlagFromFile, 1170
GetMD5DataImage, 1171
GetMD5DataImages, 1171
GetMD5FromBrokenFile, 1171
GetMD5FromFile, 1171
GetMediaStorageDataFile, 1171
GetMediaStorageDataFiles, 1171
GetMediaStorageFromFile, 1172
GetNumberOfFileNames, 1172
GetNumberOfMD5DataImages, 1172
GetNumberOfMediaStorageDataFiles, 1172
GetPixelSpacingDataRoot, 1172
GetSelectedPrivateGroupOffsetFromFile, 1172
GetSelectedTagsOffsetFromFile, 1173
GetSourceDirectory, 1173
GetStreamOffsetFromFile, 1173
GetTempDirectory, 1173
GetTempDirectoryW, 1173
GetTempFilename, 1174
GetTempFilenameW, 1174
MD5DataImagesType, 1168
MediaStorageDataFilesType, 1168
Print, 1174
Testing, 1169
gdcm::Trace, 1175
~Trace, 1176
DebugOff, 1176
DebugOn, 1176
ErrorOff, 1177
ErrorOn, 1177
GetDebugFlag, 1177
GetDebugStream, 1177
GetErrorFlag, 1177
GetErrorStream, 1177
GetStream, 1178
GetWarningFlag, 1178
GetWarningStream, 1178
SetDebug, 1178
SetDebugStream, 1178
SetError, 1178
SetErrorStream, 1179
SetStream, 1179
SetStreamToFile, 1179

SetWarning, 1179
SetWarningStream, 1179
Trace, 1176
WarningOff, 1180
WarningOn, 1180
gdcm::TransferSyntax, 1180
CanStoreLossy, 1184
CT_private_ELE, 1183
DeflatedExplicitVRLittleEndian, 1183
Explicit, 1182
ExplicitVRBigEndian, 1183
ExplicitVRLittleEndian, 1183
GetNegociatedType, 1184
GetString, 1184
GetSwapCode, 1184
GetTSString, 1184
GetTSType, 1185
Implicit, 1182
ImplicitVRBigEndianACRNEMA, 1183
ImplicitVRBigEndianPrivateGE, 1183
ImplicitVRLittleEndian, 1183
IsEncapsulated, 1185
IsEncoded, 1185
IsExplicit, 1185
IsImplicit, 1185
IsLossless, 1186
IsLossy, 1186
IsValid, 1186
JPEG2000, 1183
JPEG2000Lossless, 1183
JPEG2000Part2, 1183
JPEG2000Part2Lossless, 1183
JPEGBaselineProcess1, 1183
JPEGExtendedProcess2_4, 1183
JPEGExtendedProcess3_5, 1183
JPEGFullProgressionProcess10_12, 1183
JPEGLosslessProcess14, 1183
JPEGLosslessProcess14_1, 1183
JPEGLSLossless, 1183
JPEGLSNearLossless, 1183
JPEGSpectralSelectionProcess6_8, 1183
JPIPReferenced, 1183
MPEG2MainProfile, 1183
MPEG2MainProfileHighLevel, 1183
MPEG4AVCH264BDcompatibleHighProfileLevel4_1, 1183
MPEG4AVCH264HighProfileLevel4_1, 1183
NegociatedType, 1182
operator TSType, 1186
operator<<, 1186
RLELossless, 1183
TransferSyntax, 1183
TS_END, 1183
TSType, 1183
Unknown, 1182
WeirdPapryus, 1183
gdcm::Type, 1191
GetTypeString, 1193
GetTypeType, 1193
operator TypeType, 1193
operator<<, 1193
T1, 1192
T1C, 1192
T2, 1192
T2C, 1192
T3, 1192
Type, 1193
TypeType, 1192
UNKNOWN, 1192
gdcm::UI, 1194
Internal, 1194
operator<<, 1194
gdcm::UIDGenerator, 1195
Generate, 1196
GenerateUUID, 1196
GetGDCMUID, 1196
GetRoot, 1196
IsValid, 1197
SetRoot, 1197
UIDGenerator, 1195
gdcm::UIDs, 1197
AbstractMultiDimensionalImageModel, 1223
AcquisitionContextSRStorage, 1222
AdultMouseAnatomyOntology, 1220
AdvancedBlendingPresentationStateStorage, 1221
AmbulatoryECGWaveformStorage, 1217
ArterialPulseWaveformStorage, 1221
AudioSRStorageTrialRetired, 1218
AutorefractionMeasurementsStorage, 1221
BasicAnnotationBoxSOPClass, 1216
BasicColorImageBoxSOPClass, 1216
BasicColorPrintManagementMetaSOPClass, 1216
BasicFilmBoxSOPClass, 1216
BasicFilmSessionSOPClass, 1216
BasicGrayscaleImageBoxSOPClass, 1216
BasicGrayscalePrintManagementMetaSOPClass, 1216
BasicPrintImageOverlayBoxSOPClassRetired, 1216
BasicStructuredDisplayStorage, 1222
BasicStudyContentNotificationSOPClassRetired, 1215
BasicTextSRStorage, 1218
BasicVoiceAudioWaveformStorage, 1217
BlendingSoftcopyPresentationStateStorageSOP-Class, 1217
BreastImagingRelevantPatientInformationQuery, 1219

BreastProjectionXRayImageStorageForPresentation, 1221
BreastProjectionXRayImageStorageForProcessing, 1221
BreastTomosynthesisImageStorage, 1220
CardiacElectrophysiologyWaveformStorage, 1217
CardiacRelevantPatientInformationQuery, 1219
ChestCADSRStorage, 1218
ColonCADSRStorage, 1222
ColorPaletteQueryRetrieveInformationModelFIND, 1223
ColorPaletteQueryRetrieveInformationModelGET, 1223
ColorPaletteQueryRetrieveInformationModelMOVE, 1223
ColorPaletteStorage, 1223
ColorSoftcopyPresentationStateStorageSOPClass, 1217
CompositeInstanceRetrieveWithoutBulkDataGET, 1222
CompositeInstanceRootRetrieveGET, 1222
CompositeInstanceRootRetrieveMOVE, 1222
CompositingPlanarMPRVolumetricPresentationStateStorage, 1221
Comprehensive3DSRStorage, 1222
ComprehensiveSRStorage, 1218
ComprehensiveSRStorageTrialRetired, 1218
ComputedRadiographyImageStorage, 1216
ContentAssessmentResultsStorage, 1222
CornealTopographyMapStorage, 1222
CTDefinedProcedureProtocolStorage, 1222
CTImageStorage, 1216
CTPerformedProcedureProtocolStorage, 1222
DefinedProcedureProtocolInformationModelFIND, 1222
DefinedProcedureProtocolInformationModelGET, 1222
DefinedProcedureProtocolInformationModelMOVE, 1222
DeflatedExplicitVRLittleEndian, 1214
DeformableSpatialRegistrationStorage, 1217
DetachedInterpretationManagementSOPClassRetired, 1216
DetachedPatientManagementMetaSOPClassRetired, 1215
DetachedPatientManagementSOPClassRetired, 1215
DetachedResultsManagementMetaSOPClassRetired, 1215
DetachedResultsManagementSOPClassRetired, 1215
DetachedStudyManagementMetaSOPClassRetired, 1215
DetachedStudyManagementSOPClassRetired, 1215
DetachedVisitManagementSOPClassRetired, 1215
DetailSRStorageTrialRetired, 1218
dicomAETitle, 1219
dicomApplicationCluster, 1219
DICOMApplicationContextName, 1215
dicomAssociationAcceptor, 1219
dicomAssociationInitiator, 1219
dicomAuthorizedNodeCertificateReference, 1219
dicomConfigurationRoot, 1220
DICOMContentMappingResource, 1223
DICOMControlledTerminology, 1215
dicomDescription, 1219
dicomDevice, 1220
dicomDeviceName, 1219
dicomDeviceSerialNumber, 1220
dicomDevicesRoot, 1220
dicomHostname, 1219
dicomInstalled, 1219
dicomInstitutionAddress, 1220
dicomInstitutionDepartmentName, 1220
dicomInstitutionName, 1220
dicomIssuerOfPatientID, 1220
dicomManufacturer, 1219
dicomManufacturerModelName, 1219
dicomNetworkAE, 1220
dicomNetworkConnection, 1220
dicomNetworkConnectionReference, 1219
dicomPort, 1219
dicomPreferredCalledAETitle, 1219
dicomPreferredCallingAETitle, 1220
dicomPrimaryDeviceType, 1219
dicomRelatedDeviceReference, 1219
dicomSoftwareVersion, 1219
dicomSOPClass, 1219
dicomStationName, 1220
dicomSupportedCharacterSet, 1220
dicomThisNodeCertificateReference, 1219
dicomTLSCyphersuite, 1219
dicomTransferCapability, 1220
dicomTransferRole, 1219
dicomTransferSyntax, 1219
DICOMUIDRegistry, 1215
dicomUniqueAETitle, 1220
dicomUniqueAETitlesRegistryRoot, 1220
dicomVendorData, 1219
DICOS2DAITStorage, 1222
DICOS3DAITStorage, 1222
DICOSCTImageStorage, 1222
DICOSDigitalXRayImageStorageForPresentation, 1222
DICOSDigitalXRayImageStorageForProcessing, 1222
DICOSQuadrupoleResonanceQRStorage, 1222
DICOSThreatDetectionReportStorage, 1222

DigitalIntraoralXRayImageStorageForPresentation, 1216
DigitalIntraoralXRayImageStorageForProcessing, 1216
DigitalMammographyXRayImageStorageForPresentation, 1216
DigitalMammographyXRayImageStorageForProcessing, 1216
DigitalXRayImageStorageForPresentation, 1216
DigitalXRayImageStorageForProcessing, 1216
DisplaySystemSOPClass, 1221
DisplaySystemSOPInstance, 1221
ECG12leadWaveformStorage, 1217
EddyCurrentImageStorage, 1222
EddyCurrentMultiframeImageStorage, 1222
EncapsulatedCDAStorage, 1218
EncapsulatedPDFStorage, 1218
EncapsulatedSTLStorage, 1222
EnhancedCTImageStorage, 1216
EnhancedMRColorImageStorage, 1223
EnhancedMRImageStorage, 1216
EnhancedPETImageStorage, 1222
EnhancedSRStorage, 1218
EnhancedUSVolumeStorage, 1220
EnhancedXAImageStorage, 1217
EnhancedXRFImageStorage, 1217
ExplicitVRBigEndian, 1214
ExplicitVRLittleEndian, 1214
ExtensibleSRStorage, 1222
FallColorPaletteSOPInstance, 1220
GeneralAudioWaveformStorage, 1221
GeneralECGWaveformStorage, 1217
GeneralPurposePerformedProcedureStepSOPClass, 1219
GeneralPurposeScheduledProcedureStepSOPClass, 1219
GeneralPurposeWorklistInformationModelFIND, 1218
GeneralPurposeWorklistManagementMetaSOPClass, 1219
GeneralRelevantPatientInformationQuery, 1219
GenericImplantTemplateInformationModelFIND, 1223
GenericImplantTemplateInformationModelGET, 1223
GenericImplantTemplateInformationModelMOVE, 1223
GenericImplantTemplateStorage, 1223
GetName, 1233
GetNumberOfTransferSyntaxStrings, 1233
GetString, 1233
GetTransferSyntaxString, 1233
GetTransferSyntaxStrings, 1233
GetUIDName, 1234
GetUIDString, 1234
GrayscalePlanarMPRVolumetricPresentationStateStorage, 1221
GrayscaleSoftcopyPresentationStateStorageSOPClass, 1217
HangingProtocolInformationModelFIND, 1219
HangingProtocolInformationModelGET, 1223
HangingProtocolInformationModelMOVE, 1219
HangingProtocolStorage, 1219
HardcopyColorImageStorageSOPClassRetired, 1216
HardcopyGrayscaleImageStorageSOPClassRetired, 1216
HemodynamicWaveformStorage, 1217
HEVCH_265Main10ProfileLevel5_1, 1221
HEVCH_265MainProfileLevel5_1, 1221
HotIronColorPaletteSOPInstance, 1221
HotMetalBlueColorPaletteSOPInstance, 1220
ICBM452T1FrameofReference, 1215
ICBMSingleSubjectMRIFrameofReference, 1215
ICD11, 1220
ImageBiomarkerStandardisationInitiative, 1220
ImageOverlayBoxSOPClassRetired, 1216
ImplantAssemblyTemplateInformationModelFIND, 1223
ImplantAssemblyTemplateInformationModelGET, 1223
ImplantAssemblyTemplateInformationModelMOVE, 1223
ImplantAssemblyTemplateStorage, 1223
ImplantationPlanSRStorage, 1222
ImplantTemplateGroupInformationModelFIND, 1223
ImplantTemplateGroupInformationModelGET, 1223
ImplantTemplateGroupInformationModelMOVE, 1223
ImplantTemplateGroupStorage, 1223
ImplicitVRLittleEndianDefaultTransferSyntaxforDICOM, 1214
InstanceAvailabilityNotificationSOPClass, 1219
IntegratedTaxonomicInformationSystemITISTaxonomicSerialNumberTSN, 1220
IntraocularLensCalculationsStorage, 1221
IntravascularOpticalCoherenceTomographyImageStorageForPresentation, 1221
IntravascularOpticalCoherenceTomographyImageStorageForProcessing, 1221
JPEG2000ImageCompression, 1214
JPEG2000ImageCompressionLosslessOnly, 1214
JPEG2000Part2MulticomponentImageCompression, 1214
JPEG2000Part2MulticomponentImageCompressionLosslessOnly, 1214
JPEGBaselineProcess1DefaultTransferSyntaxforLossyJPEG8BitImage 1214
JPEGExtendedHierarchicalProcess1618Retired, 1214

JPEGExtendedHierarchicalProcess1719Retired, 1214
JPEGExtendedProcess24DefaultTransferSyntaxforLossyJPEG8BitImageCompressionRetired, 1214
JPEGExtendedProcess35Retired, 1214
JPEGFullProgressionHierarchicalProcess2426Retired, 1214
JPEGFullProgressionHierarchicalProcess2527Retired, 1214
JPEGFullProgressionNonHierarchicalProcess1012Retired, 1214
JPEGFullProgressionNonHierarchicalProcess1113Retired, 1214
JPEGLosslessHierarchicalProcess28Retired, 1214
JPEGLosslessHierarchicalProcess29Retired, 1214
JPEGLosslessNonHierarchicalFirstOrderPrediction-Process14SelectionValue1DefaultTransferSyntaxforLossless8BitImageCompression, 1214
JPEGLosslessNonHierarchicalProcess14, 1214
JPEGLosslessNonHierarchicalProcess15Retired, 1214
JPEGLSLosslessImageCompression, 1214
JPEGLSLossyNearLosslessImageCompression, 1214
JPEGSpectralSelectionHierarchicalProcess2022Retired, 1214
JPEGSpectralSelectionHierarchicalProcess2123Retired, 1214
JPEGSpectralSelectionNonHierarchicalProcess68Retired, 1214
JPEGSpectralSelectionNonHierarchicalProcess79Retired, 1214
JPIPReferenced, 1214
JPIPReferencedDeflate, 1214
KeratometryMeasurementsStorage, 1221
KeyObjectSelectionDocumentStorage, 1218
LegacyConvertedEnhancedCTImageStorage, 1220
LegacyConvertedEnhancedMRImageStorage, 1220
LegacyConvertedEnhancedPETImageStorage, 1220
LensometryMeasurementsStorage, 1221
MacularGridThicknessandVolumeReportStorage, 1221
MammographyCADSRStorage, 1218
MayoClinicNonradiologicalImagesSBSAnatomical-SurfaceRegionGuide, 1220
MediaCreationManagementSOPClassUID, 1216
MediaStorageDirectoryStorage, 1215
ModalityPerformedProcedureStepNotificationSOP-Class, 1215
ModalityPerformedProcedureStepRetrieveSOP-Class, 1215
ModalityPerformedProcedureStepSOPClass, 1215
ModalityWorklistInformationModelFIND, 1218
MouseGenomeInitiativeMGI, 1220
MPEG2MainProfileHighLevel, 1220
MPEG2MainProfileMainLevel, 1214
MPEG4AVCH_264BDCompatibleHighProfileLevel4_1, 1220
MPEG4AVCH_264HighProfileLevel4_1, 1220
MPEG4AVCH_264HighProfileLevel4_2For2DVideo, 1221
MPEG4AVCH_264HighProfileLevel4_2For3DVideo, 1221
MPEG4AVCH_264StereoHighProfileLevel4_2, 1221
MRImageStorage, 1216
MRSpectroscopyStorage, 1216
MultiframeGrayscaleByteSecondaryCaptureIm-ageStorage, 1217
MultiframeGrayscaleWordSecondaryCaptureIm-ageStorage, 1217
MultiframeSingleBitSecondaryCaptureImageStor-age, 1217
MultiframeTrueColorSecondaryCaptureImageStor-age, 1217
MultipleVolumeRenderingVolumetricPresentation-StateStorage, 1221
NativeDICOMModel, 1223
NewYorkUniversityMelanomaClinicalCooperative-Group, 1220
NuclearMedicineImageStorage, 1217
NuclearMedicineImageStorageRetired, 1217
Null0, 1221
Null1, 1221
operator TSType, 1234
OphthalmicAxialMeasurementsStorage, 1221
OphthalmicOpticalCoherenceTomographyBscanVol-umeAnalysisStorage, 1221
OphthalmicOpticalCoherenceTomographyEnFaceIm-ageStorage, 1221
OphthalmicPhotography16BitImageStorage, 1218
OphthalmicPhotography8BitImageStorage, 1218
OphthalmicThicknessMapStorage, 1221
OphthalmicTomographyImageStorage, 1218
OphthalmicVisualFieldStaticPerimetryMeasure-mentsStorage, 1221
Papyrus3ImplicitVRLittleEndian, 1220
ParametricMapStorage, 1221
PatientRadiationDoseSRStorage, 1222
PatientRootQueryRetrieveInformationModelFIND, 1218
PatientRootQueryRetrieveInformationModelGET, 1218
PatientRootQueryRetrieveInformationModelMOVE, 1218
PatientStudyOnlyQueryRetrieveInformationMod-elFINDRetired, 1218
PatientStudyOnlyQueryRetrieveInformationModel-GETRetired, 1218

PatientStudyOnlyQueryRetrieveInformationModel-MOVERetired, 1218
PerformedImagingAgentAdministrationSRStorage, 1222
PET20StepColorPaletteSOPInstance, 1220
PETColorPaletteSOPInstance, 1220
PlannedImagingAgentAdministrationSRStorage, 1222
PositronEmissionTomographyImageStorage, 1218
PresentationLUTSOPClass, 1216
PrinterConfigurationRetrievalSOPClass, 1216
PrinterConfigurationRetrievalSOPInstance, 1216
PrinterSOPClass, 1216
PrinterSOPInstance, 1216
PrintJobSOPClass, 1216
PrintQueueManagementSOPClassRetired, 1216
PrintQueueSOPInstanceRetired, 1216
ProceduralEventLoggingSOPClass, 1215
ProceduralEventLoggingSOPInstance, 1215
ProcedureLogStorage, 1218
ProductCharacteristicsQuerySOPClass, 1219
ProtocolApprovalInformationModelFIND, 1222
ProtocolApprovalInformationModelGET, 1222
ProtocolApprovalInformationModelMOVE, 1222
ProtocolApprovalStorage, 1222
PseudoColorSoftcopyPresentationStateStorage-SOPClass, 1217
PubChemCompoundCID, 1220
PullPrintRequestSOPClassRetired, 1216
PullStoredPrintManagementMetaSOPClassRetired, 1216
RadiomicsOntology, 1220
RadiopharmaceuticalRadiationDoseSRStorage, 1222
RawDataStorage, 1217
RealWorldValueMappingStorage, 1217
ReferencedColorPrintManagementMetaSOPClass-Retired, 1216
ReferencedGrayscalePrintManagementMetaSOP-ClassRetired, 1216
ReferencedImageBoxSOPClassRetired, 1216
RespiratoryWaveformStorage, 1221
RFC2557MIMEencapsulation, 1214
RLELossless, 1214
RTBeamsDeliveryInstructionStorage, 1223
RTBeamsDeliveryInstructionStorageSupplement74FrozenDraft, 1219
RTBeamsTreatmentRecordStorage, 1218
RTBrachyApplicationSetupDeliveryInstructionStor-age, 1223
RTBrachyTreatmentRecordStorage, 1218
RTConventionalMachineVerification, 1223
RTConventionalMachineVerificationSupplement74FrozenDraft, 1219
RTDoseStorage, 1218
RTImageStorage, 1218
RTIonBeamsTreatmentRecordStorage, 1218
RTIonMachineVerification, 1223
RTIonMachineVerificationSupplement74FrozenDraft, 1219
RTIonPlanStorage, 1218
RTPhysicianIntentStorage, 1222
RTPlanStorage, 1218
RTSegmentAnnotationStorage, 1222
RTStructureSetStorage, 1218
RTTreatmentSummaryRecordStorage, 1218
SecondaryCaptureImageStorage, 1217
SegmentationStorage, 1217
SegmentedVolumeRenderingVolumetricPresenta-tionStateStorage, 1221
SetFromUID, 1234
SimplifiedAdultEchoSRStorage, 1222
SpatialFiducialsStorage, 1217
SpatialRegistrationStorage, 1217
SpectaclePrescriptionReportStorage, 1221
SPM2AVG152PDFrameofReference, 1215
SPM2AVG152T1FrameofReference, 1215
SPM2AVG152T2FrameofReference, 1215
SPM2AVG305T1FrameofReference, 1215
SPM2BRAINMASKFrameofReference, 1215
SPM2CSFFrameofReference, 1215
SPM2EPIFrameofReference, 1215
SPM2FILT1FrameofReference, 1215
SPM2GRAYFrameofReference, 1215
SPM2PDFrameofReference, 1215
SPM2PETFrameofReference, 1215
SPM2SINGLESUBJT1FrameofReference, 1215
SPM2SPECTFrameofReference, 1215
SPM2T1FrameofReference, 1215
SPM2T2FrameofReference, 1215
SPM2TRANSMFrameofReference, 1215
SPM2WHITEFrameofReference, 1215
SpringColorPaletteSOPInstance, 1220
StandaloneCurveStorageRetired, 1217
StandaloneModalityLUTStorageRetired, 1217
StandaloneOverlayStorageRetired, 1217
StandalonePETCurveStorageRetired, 1218
StandaloneVOILUTStorageRetired, 1217
StereometricRelationshipStorage, 1218
StorageCommitmentPullModelSOPClassRetired, 1215
StorageCommitmentPullModelSOPInstanceRetired, 1215
StorageCommitmentPushModelSOPClass, 1215
StorageCommitmentPushModelSOPInstance, 1215
StorageServiceClass, 1216
StoredPrintStorageSOPClassRetired, 1216

StudyComponentManagementSOPClassRetired, 1215
StudyRootQueryRetrieveInformationModelFIND, 1218
StudyRootQueryRetrieveInformationModelGET, 1218
StudyRootQueryRetrieveInformationModelMOVE, 1218
SubjectiveRefractionMeasurementsStorage, 1221
SubstanceAdministrationLoggingSOPClass, 1215
SubstanceAdministrationLoggingSOPInstance, 1215
SubstanceApprovalQuerySOPClass, 1219
SummerColorPaletteSOPInstance, 1220
SurfaceScanMeshStorage, 1221
SurfaceScanPointCloudStorage, 1221
SurfaceSegmentationStorage, 1220
TalairachBrainAtlasFrameofReference, 1215
TextSRStorageTrialRetired, 1218
TractographyResultsStorage, 1221
TransferSyntaxStringsType, 1213
TSName, 1214
TSType, 1223
UberonOntology, 1220
uid_1_2_840_10008_15_0_3_1, 1228
uid_1_2_840_10008_15_0_3_10, 1229
uid_1_2_840_10008_15_0_3_11, 1229
uid_1_2_840_10008_15_0_3_12, 1229
uid_1_2_840_10008_15_0_3_13, 1229
uid_1_2_840_10008_15_0_3_14, 1229
uid_1_2_840_10008_15_0_3_15, 1229
uid_1_2_840_10008_15_0_3_16, 1229
uid_1_2_840_10008_15_0_3_17, 1229
uid_1_2_840_10008_15_0_3_18, 1229
uid_1_2_840_10008_15_0_3_19, 1229
uid_1_2_840_10008_15_0_3_2, 1228
uid_1_2_840_10008_15_0_3_20, 1229
uid_1_2_840_10008_15_0_3_21, 1229
uid_1_2_840_10008_15_0_3_22, 1229
uid_1_2_840_10008_15_0_3_23, 1229
uid_1_2_840_10008_15_0_3_24, 1229
uid_1_2_840_10008_15_0_3_25, 1229
uid_1_2_840_10008_15_0_3_26, 1229
uid_1_2_840_10008_15_0_3_27, 1229
uid_1_2_840_10008_15_0_3_28, 1229
uid_1_2_840_10008_15_0_3_29, 1229
uid_1_2_840_10008_15_0_3_3, 1229
uid_1_2_840_10008_15_0_3_30, 1229
uid_1_2_840_10008_15_0_3_31, 1229
uid_1_2_840_10008_15_0_3_4, 1229
uid_1_2_840_10008_15_0_3_5, 1229
uid_1_2_840_10008_15_0_3_6, 1229
uid_1_2_840_10008_15_0_3_7, 1229
uid_1_2_840_10008_15_0_3_8, 1229
uid_1_2_840_10008_15_0_3_9, 1229
uid_1_2_840_10008_15_0_4_1, 1229
uid_1_2_840_10008_15_0_4_2, 1229
uid_1_2_840_10008_15_0_4_3, 1229
uid_1_2_840_10008_15_0_4_4, 1229
uid_1_2_840_10008_15_0_4_5, 1229
uid_1_2_840_10008_15_0_4_6, 1229
uid_1_2_840_10008_15_0_4_7, 1229
uid_1_2_840_10008_15_0_4_8, 1229
uid_1_2_840_10008_15_1_1, 1232
uid_1_2_840_10008_1_1, 1223
uid_1_2_840_10008_1_2, 1223
uid_1_2_840_10008_1_20, 1230
uid_1_2_840_10008_1_20_1, 1225
uid_1_2_840_10008_1_20_1_1, 1225
uid_1_2_840_10008_1_20_2, 1225
uid_1_2_840_10008_1_20_2_1, 1225
uid_1_2_840_10008_1_2_1, 1223
uid_1_2_840_10008_1_2_1_99, 1223
uid_1_2_840_10008_1_2_2, 1223
uid_1_2_840_10008_1_2_4_100, 1224
uid_1_2_840_10008_1_2_4_101, 1229
uid_1_2_840_10008_1_2_4_102, 1229
uid_1_2_840_10008_1_2_4_103, 1230
uid_1_2_840_10008_1_2_4_104, 1230
uid_1_2_840_10008_1_2_4_105, 1230
uid_1_2_840_10008_1_2_4_106, 1230
uid_1_2_840_10008_1_2_4_107, 1230
uid_1_2_840_10008_1_2_4_108, 1230
uid_1_2_840_10008_1_2_4_50, 1223
uid_1_2_840_10008_1_2_4_51, 1223
uid_1_2_840_10008_1_2_4_52, 1224
uid_1_2_840_10008_1_2_4_53, 1224
uid_1_2_840_10008_1_2_4_54, 1224
uid_1_2_840_10008_1_2_4_55, 1224
uid_1_2_840_10008_1_2_4_56, 1224
uid_1_2_840_10008_1_2_4_57, 1224
uid_1_2_840_10008_1_2_4_58, 1224
uid_1_2_840_10008_1_2_4_59, 1224
uid_1_2_840_10008_1_2_4_60, 1224
uid_1_2_840_10008_1_2_4_61, 1224
uid_1_2_840_10008_1_2_4_62, 1224
uid_1_2_840_10008_1_2_4_63, 1224
uid_1_2_840_10008_1_2_4_64, 1224
uid_1_2_840_10008_1_2_4_65, 1224
uid_1_2_840_10008_1_2_4_66, 1224
uid_1_2_840_10008_1_2_4_70, 1224
uid_1_2_840_10008_1_2_4_80, 1224
uid_1_2_840_10008_1_2_4_81, 1224
uid_1_2_840_10008_1_2_4_90, 1224
uid_1_2_840_10008_1_2_4_91, 1224
uid_1_2_840_10008_1_2_4_92, 1224
uid_1_2_840_10008_1_2_4_93, 1224
uid_1_2_840_10008_1_2_4_94, 1224
uid_1_2_840_10008_1_2_4_95, 1224

uid_1_2_840_10008_1_2_5, 1224
uid_1_2_840_10008_1_2_6_1, 1224
uid_1_2_840_10008_1_2_6_2, 1224
uid_1_2_840_10008_1_3_10, 1224
uid_1_2_840_10008_1_40, 1225
uid_1_2_840_10008_1_40_1, 1225
uid_1_2_840_10008_1_42, 1225
uid_1_2_840_10008_1_42_1, 1225
uid_1_2_840_10008_1_4_1_1, 1224
uid_1_2_840_10008_1_4_1_10, 1224
uid_1_2_840_10008_1_4_1_11, 1224
uid_1_2_840_10008_1_4_1_12, 1224
uid_1_2_840_10008_1_4_1_13, 1224
uid_1_2_840_10008_1_4_1_14, 1224
uid_1_2_840_10008_1_4_1_15, 1224
uid_1_2_840_10008_1_4_1_16, 1224
uid_1_2_840_10008_1_4_1_17, 1224
uid_1_2_840_10008_1_4_1_18, 1225
uid_1_2_840_10008_1_4_1_2, 1224
uid_1_2_840_10008_1_4_1_3, 1224
uid_1_2_840_10008_1_4_1_4, 1224
uid_1_2_840_10008_1_4_1_5, 1224
uid_1_2_840_10008_1_4_1_6, 1224
uid_1_2_840_10008_1_4_1_7, 1224
uid_1_2_840_10008_1_4_1_8, 1224
uid_1_2_840_10008_1_4_1_9, 1224
uid_1_2_840_10008_1_4_2_1, 1225
uid_1_2_840_10008_1_4_2_2, 1225
uid_1_2_840_10008_1_5_1, 1230
uid_1_2_840_10008_1_5_2, 1230
uid_1_2_840_10008_1_5_3, 1230
uid_1_2_840_10008_1_5_4, 1230
uid_1_2_840_10008_1_5_5, 1230
uid_1_2_840_10008_1_5_6, 1230
uid_1_2_840_10008_1_5_7, 1230
uid_1_2_840_10008_1_5_8, 1230
uid_1_2_840_10008_1_9, 1225
uid_1_2_840_10008_2_16_10, 1230
uid_1_2_840_10008_2_16_11, 1230
uid_1_2_840_10008_2_16_12, 1230
uid_1_2_840_10008_2_16_13, 1230
uid_1_2_840_10008_2_16_14, 1230
uid_1_2_840_10008_2_16_4, 1225
uid_1_2_840_10008_2_16_5, 1230
uid_1_2_840_10008_2_16_6, 1230
uid_1_2_840_10008_2_16_7, 1230
uid_1_2_840_10008_2_16_8, 1230
uid_1_2_840_10008_2_16_9, 1230
uid_1_2_840_10008_2_6_1, 1225
uid_1_2_840_10008_3_1_1_1, 1225
uid_1_2_840_10008_3_1_2_1_1, 1225
uid_1_2_840_10008_3_1_2_1_4, 1225
uid_1_2_840_10008_3_1_2_2_1, 1225
uid_1_2_840_10008_3_1_2_3_1, 1225
uid_1_2_840_10008_3_1_2_3_2, 1225
uid_1_2_840_10008_3_1_2_3_3, 1225
uid_1_2_840_10008_3_1_2_3_4, 1225
uid_1_2_840_10008_3_1_2_3_5, 1225
uid_1_2_840_10008_3_1_2_5_1, 1225
uid_1_2_840_10008_3_1_2_5_4, 1225
uid_1_2_840_10008_3_1_2_5_5, 1225
uid_1_2_840_10008_3_1_2_6_1, 1225
uid_1_2_840_10008_4_2, 1225
uid_1_2_840_10008_5_1_1_1, 1225
uid_1_2_840_10008_5_1_1_14, 1225
uid_1_2_840_10008_5_1_1_15, 1225
uid_1_2_840_10008_5_1_1_16, 1225
uid_1_2_840_10008_5_1_1_16_376, 1225
uid_1_2_840_10008_5_1_1_17, 1225
uid_1_2_840_10008_5_1_1_17_376, 1225
uid_1_2_840_10008_5_1_1_18, 1225
uid_1_2_840_10008_5_1_1_18_1, 1225
uid_1_2_840_10008_5_1_1_2, 1225
uid_1_2_840_10008_5_1_1_22, 1225
uid_1_2_840_10008_5_1_1_23, 1225
uid_1_2_840_10008_5_1_1_24, 1225
uid_1_2_840_10008_5_1_1_24_1, 1226
uid_1_2_840_10008_5_1_1_25, 1226
uid_1_2_840_10008_5_1_1_26, 1226
uid_1_2_840_10008_5_1_1_27, 1226
uid_1_2_840_10008_5_1_1_29, 1226
uid_1_2_840_10008_5_1_1_30, 1226
uid_1_2_840_10008_5_1_1_31, 1226
uid_1_2_840_10008_5_1_1_32, 1226
uid_1_2_840_10008_5_1_1_33, 1226
uid_1_2_840_10008_5_1_1_4, 1225
uid_1_2_840_10008_5_1_1_40, 1230
uid_1_2_840_10008_5_1_1_40_1, 1230
uid_1_2_840_10008_5_1_1_4_1, 1225
uid_1_2_840_10008_5_1_1_4_2, 1225
uid_1_2_840_10008_5_1_1_9, 1225
uid_1_2_840_10008_5_1_1_9_1, 1225
uid_1_2_840_10008_5_1_4_1_1_1, 1226
uid_1_2_840_10008_5_1_4_1_1_10, 1226
uid_1_2_840_10008_5_1_4_1_1_104_1, 1227
uid_1_2_840_10008_5_1_4_1_1_104_2, 1228
uid_1_2_840_10008_5_1_4_1_1_104_3, 1231
uid_1_2_840_10008_5_1_4_1_1_11, 1226
uid_1_2_840_10008_5_1_4_1_1_11_1, 1226
uid_1_2_840_10008_5_1_4_1_1_11_10, 1230
uid_1_2_840_10008_5_1_4_1_1_11_11, 1230
uid_1_2_840_10008_5_1_4_1_1_11_2, 1226
uid_1_2_840_10008_5_1_4_1_1_11_3, 1226
uid_1_2_840_10008_5_1_4_1_1_11_4, 1226
uid_1_2_840_10008_5_1_4_1_1_11_5, 1230
uid_1_2_840_10008_5_1_4_1_1_11_6, 1230
uid_1_2_840_10008_5_1_4_1_1_11_7, 1230
uid_1_2_840_10008_5_1_4_1_1_11_8, 1230

uid_1_2_840_10008_5_1_4_1_1_11_9, 1230
uid_1_2_840_10008_5_1_4_1_1_128, 1228
uid_1_2_840_10008_5_1_4_1_1_128_1, 1229
uid_1_2_840_10008_5_1_4_1_1_129, 1228
uid_1_2_840_10008_5_1_4_1_1_12_1, 1227
uid_1_2_840_10008_5_1_4_1_1_12_1_1, 1227
uid_1_2_840_10008_5_1_4_1_1_12_2, 1227
uid_1_2_840_10008_5_1_4_1_1_12_2_1, 1227
uid_1_2_840_10008_5_1_4_1_1_12_3, 1227
uid_1_2_840_10008_5_1_4_1_1_12_77, 1230
uid_1_2_840_10008_5_1_4_1_1_130, 1231
uid_1_2_840_10008_5_1_4_1_1_131, 1231
uid_1_2_840_10008_5_1_4_1_1_13_1_1, 1227
uid_1_2_840_10008_5_1_4_1_1_13_1_2, 1227
uid_1_2_840_10008_5_1_4_1_1_13_1_3, 1229
uid_1_2_840_10008_5_1_4_1_1_13_1_4, 1230
uid_1_2_840_10008_5_1_4_1_1_13_1_5, 1230
uid_1_2_840_10008_5_1_4_1_1_14_1, 1230
uid_1_2_840_10008_5_1_4_1_1_14_2, 1230
uid_1_2_840_10008_5_1_4_1_1_1_1, 1226
uid_1_2_840_10008_5_1_4_1_1_1_1_1, 1226
uid_1_2_840_10008_5_1_4_1_1_1_2, 1226
uid_1_2_840_10008_5_1_4_1_1_1_2_1, 1226
uid_1_2_840_10008_5_1_4_1_1_1_3, 1226
uid_1_2_840_10008_5_1_4_1_1_1_3_1, 1226
uid_1_2_840_10008_5_1_4_1_1_2, 1226
uid_1_2_840_10008_5_1_4_1_1_20, 1227
uid_1_2_840_10008_5_1_4_1_1_200_1, 1231
uid_1_2_840_10008_5_1_4_1_1_200_2, 1231
uid_1_2_840_10008_5_1_4_1_1_200_3, 1231
uid_1_2_840_10008_5_1_4_1_1_200_4, 1231
uid_1_2_840_10008_5_1_4_1_1_200_5, 1231
uid_1_2_840_10008_5_1_4_1_1_200_6, 1231
uid_1_2_840_10008_5_1_4_1_1_2_1, 1226
uid_1_2_840_10008_5_1_4_1_1_2_2, 1229
uid_1_2_840_10008_5_1_4_1_1_3, 1226
uid_1_2_840_10008_5_1_4_1_1_30, 1230
uid_1_2_840_10008_5_1_4_1_1_3_1, 1226
uid_1_2_840_10008_5_1_4_1_1_4, 1226
uid_1_2_840_10008_5_1_4_1_1_40, 1230
uid_1_2_840_10008_5_1_4_1_1_481_1, 1228
uid_1_2_840_10008_5_1_4_1_1_481_10, 1231
uid_1_2_840_10008_5_1_4_1_1_481_11, 1231
uid_1_2_840_10008_5_1_4_1_1_481_2, 1228
uid_1_2_840_10008_5_1_4_1_1_481_3, 1228
uid_1_2_840_10008_5_1_4_1_1_481_4, 1228
uid_1_2_840_10008_5_1_4_1_1_481_5, 1228
uid_1_2_840_10008_5_1_4_1_1_481_6, 1228
uid_1_2_840_10008_5_1_4_1_1_481_7, 1228
uid_1_2_840_10008_5_1_4_1_1_481_8, 1228
uid_1_2_840_10008_5_1_4_1_1_481_9, 1228
uid_1_2_840_10008_5_1_4_1_1_4_1, 1226
uid_1_2_840_10008_5_1_4_1_1_4_2, 1226
uid_1_2_840_10008_5_1_4_1_1_4_3, 1232
uid_1_2_840_10008_5_1_4_1_1_4_4, 1229
uid_1_2_840_10008_5_1_4_1_1_5, 1226
uid_1_2_840_10008_5_1_4_1_1_501_1, 1231
uid_1_2_840_10008_5_1_4_1_1_501_2_1, 1231
uid_1_2_840_10008_5_1_4_1_1_501_2_2, 1231
uid_1_2_840_10008_5_1_4_1_1_501_3, 1231
uid_1_2_840_10008_5_1_4_1_1_501_4, 1232
uid_1_2_840_10008_5_1_4_1_1_501_5, 1232
uid_1_2_840_10008_5_1_4_1_1_501_6, 1232
uid_1_2_840_10008_5_1_4_1_1_6, 1226
uid_1_2_840_10008_5_1_4_1_1_601_1, 1232
uid_1_2_840_10008_5_1_4_1_1_601_2, 1232
uid_1_2_840_10008_5_1_4_1_1_66, 1227
uid_1_2_840_10008_5_1_4_1_1_66_1, 1227
uid_1_2_840_10008_5_1_4_1_1_66_2, 1227
uid_1_2_840_10008_5_1_4_1_1_66_3, 1227
uid_1_2_840_10008_5_1_4_1_1_66_4, 1227
uid_1_2_840_10008_5_1_4_1_1_66_5, 1229
uid_1_2_840_10008_5_1_4_1_1_66_6, 1230
uid_1_2_840_10008_5_1_4_1_1_67, 1227
uid_1_2_840_10008_5_1_4_1_1_68_1, 1230
uid_1_2_840_10008_5_1_4_1_1_68_2, 1231
uid_1_2_840_10008_5_1_4_1_1_6_1, 1226
uid_1_2_840_10008_5_1_4_1_1_6_2, 1229
uid_1_2_840_10008_5_1_4_1_1_7, 1226
uid_1_2_840_10008_5_1_4_1_1_77_1, 1227
uid_1_2_840_10008_5_1_4_1_1_77_1_1, 1227
uid_1_2_840_10008_5_1_4_1_1_77_1_1_1, 1227
uid_1_2_840_10008_5_1_4_1_1_77_1_2, 1227
uid_1_2_840_10008_5_1_4_1_1_77_1_2_1, 1227
uid_1_2_840_10008_5_1_4_1_1_77_1_3, 1227
uid_1_2_840_10008_5_1_4_1_1_77_1_4, 1227
uid_1_2_840_10008_5_1_4_1_1_77_1_4_1, 1227
uid_1_2_840_10008_5_1_4_1_1_77_1_5_1, 1227
uid_1_2_840_10008_5_1_4_1_1_77_1_5_2, 1227
uid_1_2_840_10008_5_1_4_1_1_77_1_5_3, 1227
uid_1_2_840_10008_5_1_4_1_1_77_1_5_4, 1227
uid_1_2_840_10008_5_1_4_1_1_77_1_5_5, 1231
uid_1_2_840_10008_5_1_4_1_1_77_1_5_6, 1231
uid_1_2_840_10008_5_1_4_1_1_77_1_5_7, 1231
uid_1_2_840_10008_5_1_4_1_1_77_1_5_8, 1231
uid_1_2_840_10008_5_1_4_1_1_77_1_6, 1229
uid_1_2_840_10008_5_1_4_1_1_77_2, 1227
uid_1_2_840_10008_5_1_4_1_1_78_1, 1231
uid_1_2_840_10008_5_1_4_1_1_78_2, 1231
uid_1_2_840_10008_5_1_4_1_1_78_3, 1231
uid_1_2_840_10008_5_1_4_1_1_78_4, 1231
uid_1_2_840_10008_5_1_4_1_1_78_5, 1231
uid_1_2_840_10008_5_1_4_1_1_78_6, 1231
uid_1_2_840_10008_5_1_4_1_1_78_7, 1231
uid_1_2_840_10008_5_1_4_1_1_78_8, 1231
uid_1_2_840_10008_5_1_4_1_1_79_1, 1231
uid_1_2_840_10008_5_1_4_1_1_7_1, 1226
uid_1_2_840_10008_5_1_4_1_1_7_2, 1226

uid_1_2_840_10008_5_1_4_1_1_7_3, 1226
uid_1_2_840_10008_5_1_4_1_1_7_4, 1226
uid_1_2_840_10008_5_1_4_1_1_8, 1226
uid_1_2_840_10008_5_1_4_1_1_80_1, 1231
uid_1_2_840_10008_5_1_4_1_1_81_1, 1231
uid_1_2_840_10008_5_1_4_1_1_82_1, 1231
uid_1_2_840_10008_5_1_4_1_1_88_1, 1227
uid_1_2_840_10008_5_1_4_1_1_88_11, 1227
uid_1_2_840_10008_5_1_4_1_1_88_2, 1227
uid_1_2_840_10008_5_1_4_1_1_88_22, 1227
uid_1_2_840_10008_5_1_4_1_1_88_3, 1227
uid_1_2_840_10008_5_1_4_1_1_88_33, 1227
uid_1_2_840_10008_5_1_4_1_1_88_34, 1231
uid_1_2_840_10008_5_1_4_1_1_88_35, 1231
uid_1_2_840_10008_5_1_4_1_1_88_4, 1227
uid_1_2_840_10008_5_1_4_1_1_88_40, 1227
uid_1_2_840_10008_5_1_4_1_1_88_50, 1227
uid_1_2_840_10008_5_1_4_1_1_88_59, 1227
uid_1_2_840_10008_5_1_4_1_1_88_65, 1227
uid_1_2_840_10008_5_1_4_1_1_88_67, 1227
uid_1_2_840_10008_5_1_4_1_1_88_68, 1231
uid_1_2_840_10008_5_1_4_1_1_88_69, 1231
uid_1_2_840_10008_5_1_4_1_1_88_70, 1231
uid_1_2_840_10008_5_1_4_1_1_88_71, 1231
uid_1_2_840_10008_5_1_4_1_1_88_72, 1231
uid_1_2_840_10008_5_1_4_1_1_88_73, 1231
uid_1_2_840_10008_5_1_4_1_1_88_74, 1231
uid_1_2_840_10008_5_1_4_1_1_88_75, 1231
uid_1_2_840_10008_5_1_4_1_1_9, 1226
uid_1_2_840_10008_5_1_4_1_1_90_1, 1231
uid_1_2_840_10008_5_1_4_1_1_9_1, 1226
uid_1_2_840_10008_5_1_4_1_1_9_1_1, 1226
uid_1_2_840_10008_5_1_4_1_1_9_1_2, 1226
uid_1_2_840_10008_5_1_4_1_1_9_1_3, 1226
uid_1_2_840_10008_5_1_4_1_1_9_2_1, 1226
uid_1_2_840_10008_5_1_4_1_1_9_3_1, 1226
uid_1_2_840_10008_5_1_4_1_1_9_4_1, 1226
uid_1_2_840_10008_5_1_4_1_1_9_4_2, 1230
uid_1_2_840_10008_5_1_4_1_1_9_5_1, 1230
uid_1_2_840_10008_5_1_4_1_1_9_6_1, 1230
uid_1_2_840_10008_5_1_4_1_2_1_1, 1228
uid_1_2_840_10008_5_1_4_1_2_1_2, 1228
uid_1_2_840_10008_5_1_4_1_2_1_3, 1228
uid_1_2_840_10008_5_1_4_1_2_2_1, 1228
uid_1_2_840_10008_5_1_4_1_2_2_2, 1228
uid_1_2_840_10008_5_1_4_1_2_2_3, 1228
uid_1_2_840_10008_5_1_4_1_2_3_1, 1228
uid_1_2_840_10008_5_1_4_1_2_3_2, 1228
uid_1_2_840_10008_5_1_4_1_2_3_3, 1228
uid_1_2_840_10008_5_1_4_1_2_4_2, 1232
uid_1_2_840_10008_5_1_4_1_2_4_3, 1232
uid_1_2_840_10008_5_1_4_1_2_5_3, 1232
uid_1_2_840_10008_5_1_4_20_1, 1232
uid_1_2_840_10008_5_1_4_20_2, 1232
uid_1_2_840_10008_5_1_4_20_3, 1232
uid_1_2_840_10008_5_1_4_31, 1228
uid_1_2_840_10008_5_1_4_32, 1228
uid_1_2_840_10008_5_1_4_32_1, 1228
uid_1_2_840_10008_5_1_4_32_2, 1228
uid_1_2_840_10008_5_1_4_32_3, 1228
uid_1_2_840_10008_5_1_4_33, 1228
uid_1_2_840_10008_5_1_4_34_1, 1228
uid_1_2_840_10008_5_1_4_34_10, 1232
uid_1_2_840_10008_5_1_4_34_2, 1228
uid_1_2_840_10008_5_1_4_34_3, 1228
uid_1_2_840_10008_5_1_4_34_4, 1228
uid_1_2_840_10008_5_1_4_34_4_1, 1228
uid_1_2_840_10008_5_1_4_34_4_2, 1228
uid_1_2_840_10008_5_1_4_34_4_3, 1228
uid_1_2_840_10008_5_1_4_34_4_4, 1228
uid_1_2_840_10008_5_1_4_34_5, 1228
uid_1_2_840_10008_5_1_4_34_5_1, 1232
uid_1_2_840_10008_5_1_4_34_6, 1232
uid_1_2_840_10008_5_1_4_34_6_1, 1232
uid_1_2_840_10008_5_1_4_34_6_2, 1232
uid_1_2_840_10008_5_1_4_34_6_3, 1232
uid_1_2_840_10008_5_1_4_34_6_4, 1232
uid_1_2_840_10008_5_1_4_34_7, 1232
uid_1_2_840_10008_5_1_4_34_8, 1232
uid_1_2_840_10008_5_1_4_34_9, 1232
uid_1_2_840_10008_5_1_4_37_1, 1228
uid_1_2_840_10008_5_1_4_37_2, 1228
uid_1_2_840_10008_5_1_4_37_3, 1228
uid_1_2_840_10008_5_1_4_38_1, 1228
uid_1_2_840_10008_5_1_4_38_2, 1228
uid_1_2_840_10008_5_1_4_38_3, 1228
uid_1_2_840_10008_5_1_4_38_4, 1232
uid_1_2_840_10008_5_1_4_39_1, 1232
uid_1_2_840_10008_5_1_4_39_2, 1232
uid_1_2_840_10008_5_1_4_39_3, 1232
uid_1_2_840_10008_5_1_4_39_4, 1232
uid_1_2_840_10008_5_1_4_41, 1228
uid_1_2_840_10008_5_1_4_42, 1228
uid_1_2_840_10008_5_1_4_43_1, 1232
uid_1_2_840_10008_5_1_4_43_2, 1232
uid_1_2_840_10008_5_1_4_43_3, 1232
uid_1_2_840_10008_5_1_4_43_4, 1232
uid_1_2_840_10008_5_1_4_44_1, 1232
uid_1_2_840_10008_5_1_4_44_2, 1232
uid_1_2_840_10008_5_1_4_44_3, 1232
uid_1_2_840_10008_5_1_4_44_4, 1232
uid_1_2_840_10008_5_1_4_45_1, 1232
uid_1_2_840_10008_5_1_4_45_2, 1232
uid_1_2_840_10008_5_1_4_45_3, 1232
uid_1_2_840_10008_5_1_4_45_4, 1232
uid_1_2_840_10008_7_1_1, 1232
uid_1_2_840_10008_7_1_2, 1232
uid_1_2_840_10008_8_1_1, 1232

UltrasoundImageStorage, 1217
UltrasoundImageStorageRetired, 1217
UltrasoundMultiframeImageStorage, 1216
UltrasoundMultiframeImageStorageRetired, 1216
UnifiedProcedureStepEventSOPClass, 1219
UnifiedProcedureStepEventSOPClass1, 1223
UnifiedProcedureStepPullSOPClass, 1219
UnifiedProcedureStepPullSOPClass1, 1222
UnifiedProcedureStepPushSOPClass, 1219
UnifiedProcedureStepPushSOPClass1, 1222
UnifiedProcedureStepWatchSOPClass, 1219
UnifiedProcedureStepWatchSOPClass1, 1222
UnifiedWorklistandProcedureStepServiceClass, 1219
UnifiedWorklistandProcedureStepServiceClass1, 1222
UnifiedWorklistandProcedureStepSOPInstance, 1219
UniversalCoordinatedTime, 1223
UPSFilteredGlobalSubscriptionSOPInstance, 1222
VerificationSOPClass, 1214
VideoEndoscopicImageStorage, 1217
VideoMicroscopicImageStorage, 1217
VideoPhotographicImageStorage, 1218
VisualAcuityMeasurementsStorage, 1221
VLEndoscopicImageStorage, 1217
VLImageStorageTrialRetired, 1217
VLMicroscopicImageStorage, 1217
VLMultiframeImageStorageTrialRetired, 1217
VLPhotographicImageStorage, 1218
VLSlideCoordinatesMicroscopicImageStorage, 1218
VLWholeSlideMicroscopyImageStorage, 1220
VOILUTBoxSOPClass, 1216
VolumeRenderingVolumetricPresentationStateStorage, 1221
WaveformStorageTrialRetired, 1217
WideFieldOphthalmicPhotography3DCoordinatesImageStorage, 1221
WideFieldOphthalmicPhotographyStereographicProjectionImageStorage, 1221
WinterColorPaletteSOPInstance, 1220
XAXRFGrayscaleSoftcopyPresentationStateStorage, 1221
XMLEncoding, 1214
XRay3DAngiographicImageStorage, 1217
XRay3DCraniofacialImageStorage, 1217
XRayAngiographicBiPlaneImageStorageRetired, 1217
XRayAngiographicImageStorage, 1217
XRayRadiationDoseSRStorage, 1218
XRayRadiofluoroscopicImageStorage, 1217
gdcm::UNExplicitDataElement, 1302
GetLength, 1303
Read, 1304
ReadPreValue, 1304
ReadValue, 1304
ReadWithLength, 1304
gdcm::UNExplicitImplicitDataElement, 1305
GetLength, 1306
Read, 1306
ReadPreValue, 1306
ReadValue, 1306
gdcm::Unpacker12Bits, 1307
Pack, 1307
Unpack, 1308
gdcm::Usage, 1308
Conditional, 1309
GetUsageString, 1310
GetUsageType, 1310
Invalid, 1309
Mandatory, 1309
operator UsageType, 1310
operator<<, 1310
Usage, 1310
UsageType, 1309
UserOption, 1309
gdcm::UserEvent, 1311
gdcm::UUIDGenerator, 1314
Generate, 1315
IsValid, 1315
gdcm::Validate, 1315
~Validate, 1316
F, 1317
GetValidatedFile, 1316
SetFile, 1316
V, 1317
Validate, 1316
Validation, 1317
gdcm::Value, 1318
~Value, 1319
Clear, 1319
DataElement, 1320
GetLength, 1319
operator==, 1320
SetLength, 1320
SetLengthOnly, 1320
Value, 1319
gdcm::ValueIO< TDE, TSwap, TType >, 1321
Read, 1321
Write, 1321
gdcm::Version, 1322
~Version, 1323
GetBuildVersion, 1323
GetMajorVersion, 1324
GetMinorVersion, 1324
GetVersion, 1324
operator<<, 1324
Print, 1324

Version, 1323
gdcm::VL, 1325
GetLength, 1326
GetVL16Max, 1326
GetVL32Max, 1327
IsOdd, 1327
IsUndefined, 1327
operator uint32_t, 1327
operator<<, 1329
operator++, 1327
operator+=, 1327
Read, 1328
Read16, 1328
SetToUndefined, 1328
Type, 1326
VL, 1326
Write, 1328
Write16, 1328
gdcm::VM, 1329
Compatible, 1332
GetIndex, 1332
GetLength, 1332
GetNumberOfElementsFromArray, 1332
GetVMString, 1332
GetVMType, 1333
GetVMTypeFromLength, 1333
IsValid, 1333
operator VMType, 1333
operator<<, 1333
VM, 1332
VM0, 1331
VM1, 1331
VM10, 1331
VM12, 1331
VM16, 1331
VM18, 1331
VM1_2, 1331
VM1_3, 1331
VM1_32, 1331
VM1_4, 1331
VM1_5, 1331
VM1_8, 1331
VM1_99, 1331
VM1_n, 1331
VM2, 1331
VM24, 1331
VM256, 1331
VM28, 1331
VM2_2n, 1331
VM2_n, 1331
VM3, 1331
VM30_30n, 1331
VM32, 1331
VM35, 1331
VM3_3n, 1331
VM3_4, 1331
VM3_n, 1331
VM4, 1331
VM47_47n, 1331
VM4_4n, 1331
VM5, 1331
VM6, 1331
VM6_6n, 1331
VM6_n, 1331
VM7_7n, 1331
VM8, 1331
VM9, 1331
VM99, 1331
VM_END, 1331
VMType, 1331
gdcm::VMToLength< T >, 1334
gdcm::VR, 1334
AE, 1336
AS, 1336
AT, 1336
CanDisplay, 1337
Compatible, 1338
CS, 1336
DA, 1336
DS, 1336
DT, 1336
FD, 1336
FL, 1336
GetLength, 1338
GetSize, 1338
GetSizeof, 1338
GetVRString, 1339
GetVRStringFromFile, 1339
GetVRType, 1339
GetVRTypeFromFile, 1339
INVALID, 1336
IS, 1336
IsASCII, 1339
IsASCII2, 1339
IsBinary, 1340
IsBinary2, 1340
IsDual, 1340
IsSwap, 1340
IsValid, 1340
IsVRFile, 1341
LO, 1336
LT, 1336
OB, 1336
OB_OW, 1337
OD, 1336
OF, 1336
OL, 1336
operator VRType, 1341

operator<<, 1341
OV, 1336
OW, 1336
PN, 1336
Read, 1341
SH, 1336
SL, 1337
SQ, 1337
SS, 1337
ST, 1337
SV, 1337
TM, 1337
UC, 1337
UI, 1337
UL, 1337
UN, 1337
UR, 1337
US, 1337
US_OW, 1337
US_SS, 1337
US_SS_OW, 1337
UT, 1337
UV, 1337
VL16, 1337
VL32, 1337
VR, 1337
VR_END, 1337
VR_VM1, 1337
VRALL, 1337
VRASCII, 1337
VRBINARY, 1337
VRType, 1336
Write, 1341
gdcm::VR16ExplicitDataElement, 1342
GetLength, 1343
Read, 1344
ReadPreValue, 1344
ReadValue, 1344
ReadWithLength, 1344
gdcm::VRToEncoding< T >, 1345
gdcm::VRToType< T >, 1345
gdcm::VRVLSize< 0 >, 1346
Read, 1346
Write, 1346
gdcm::VRVLSize< 1 >, 1346
Read, 1346
Write, 1347
gdcm::VRVLSize< T >, 1345
gdcm::Waveform, 1463
Waveform, 1464
gdcm::WLMFindQuery, 1464
GetAbstractSyntaxUID, 1466
GetTagListByLevel, 1466
GetValidDataSet, 1466
InitializeDataSet, 1466
QueryFactory, 1467
ValidateQuery, 1466
WLMFindQuery, 1465
gdcm::Writer, 1467
~Writer, 1470
CheckFileMetaInformationOff, 1470
CheckFileMetaInformationOn, 1470
GetCheckFileMetaInformation, 1471
GetFile, 1471
GetStreamPtr, 1471
Ofstream, 1473
SetCheckFileMetaInformation, 1471
SetFile, 1471
SetFileName, 1472
SetStream, 1472
SetWriteDataSetOnly, 1472
Stream, 1473
StreamImageWriter, 1473
Write, 1473
Writer, 1470
gdcm::XMLDictReader, 1474
~XMLDictReader, 1475
CharacterDataHandler, 1475
EndElement, 1476
GetDict, 1476
HandleDescription, 1476
HandleEntry, 1476
StartElement, 1476
XMLDictReader, 1475
gdcm::XMLPrinter, 1477
~XMLPrinter, 1478
F, 1480
GetPrintStyle, 1478
HandleBulkData, 1478
LOADBULKDATA, 1478
OnlyUUID, 1478
Print, 1479
PrintDataElement, 1479
PrintDataSet, 1479
PrintSQ, 1479
PrintStyle, 1480
PrintStyles, 1478
SetFile, 1479
SetStyle, 1480
XMLPrinter, 1478
gdcm::XMLPrivateDictReader, 1481
~XMLPrivateDictReader, 1482
CharacterDataHandler, 1482
EndElement, 1482
GetPrivateDict, 1483
HandleDescription, 1483
HandleEntry, 1483
StartElement, 1483

XMLPrivateDictReader, 1482
GDCM_DIFFERENT
gdcm, 61
GDCM_DO_JOIN
gdcmStaticAssert.h, 1537
GDCM_DO_JOIN2
gdcmStaticAssert.h, 1537
GDCM_EQUAL
gdcm, 61
GDCM_EXPORT
gdcmWin32.h, 1564
GDCM_FUNCTION
gdcmTrace.h, 1556
GDCM_GREATER
gdcm, 61
GDCM_GREATEROREQUAL
gdcm, 61
GDCM_JOIN
gdcmStaticAssert.h, 1538
GDCM_LEGACY
gdcmLegacyMacro.h, 1517
GDCM_LEGACY_BODY
gdcmLegacyMacro.h, 1517
GDCM_LEGACY_REPLACED_BODY
gdcmLegacyMacro.h, 1517
GDCM_LESS
gdcm, 61
GDCM_LESSOREQUAL
gdcm, 61
GDCM_NOOP_STATEMENT
gdcmLegacyMacro.h, 1518
GDCM_STATIC_ASSERT
gdcm::Attribute< Group, Element, TVR, TVM >, 139
gdcm::Attribute< Group, Element, TVR, VM::VM1 >, 148
gdcm::Attribute< Group, Element, TVR, VM::VM1_n >, 157
gdcmStaticAssert.h, 1538
gdcmAAbortPDU.h, 1981, 1982
gdcmAAssociateACPDU.h, 1982, 1983
gdcmAAssociateRJPDU.h, 1985
gdcmAAssociateRQPDU.h, 1986, 1987
gdcmAbstractSyntax.h, 1989, 1990
gdcmAnonymizeEvent.h, 1790, 1792
gdcmAnonymizer.h, 1792, 1793
gdcmApplicationContext.h, 1991, 1992
gdcmApplicationEntity.h, 1794, 1795
gdcmAReleaseRPPDU.h, 1992, 1993
gdcmAReleaseRQPDU.h, 1994, 1995
gdcmARTIMTimer.h, 1995, 1996
gdcmASN1.h, 1485, 1486
gdcmAssertAlwaysMacro
gdcmTrace.h, 1556
gdcmAssertMacro
gdcmTrace.h, 1556
gdcmAsynchronousOperationsWindowSub.h, 1997
gdcmAttribute.h, 1603, 1604
gdcmAudioCodec.h, 1796, 1797
gdcmBase64.h, 1487
gdcmBaseCompositeMessage.h, 1998, 1999
gdcmBaseNormalizedMessage.h, 2000, 2001
gdcmBasePDU.h, 2001, 2002
gdcmBaseQuery.h, 2003, 2004
gdcmBaseRootQuery.h, 2005, 2006
gdcmBasicOffsetTable.h, 1617, 1618
gdcmBitmap.h, 1797, 1798
gdcmBitmapToBitmapFilter.h, 1801
gdcmBoxRegion.h, 1488, 1489
gdcmByteBuffer.h, 1620, 1621
gdcmByteSwap.h, 1489, 1490
gdcmByteSwapFilter.h, 1622, 1623
gdcmByteValue.h, 1623, 1624
gdcmCAPICryptoFactory.h, 1491, 1492
gdcmCAPICryptographicMessageSyntax.h, 1492, 1493
gdcmCEchoMessages.h, 2007, 2008
gdcmCFindMessages.h, 2008, 2009
gdcmCleaner.h, 1802, 1803
gdcmCMoveMessages.h, 2010, 2011
gdcmCodec.h, 1804, 1805
gdcmCoder.h, 1805, 1806
gdcmCodeString.h, 1628
gdcmCommand.h, 1494, 1495
gdcmCommandDataSet.h, 2012
gdcmCompositeMessageFactory.h, 2013, 2014
gdcmCompositeNetworkFunctions.h, 2015
gdcmConstCharWrapper.h, 1807
gdcmCP246ExplicitDataElement.h, 1630
gdcmCryptoFactory.h, 1497, 1498
gdcmCryptographicMessageSyntax.h, 1499, 1500
gdcmCSAElement.h, 1631, 1632
gdcmCSAHeader.h, 1634, 1635
gdcmCSAHeaderDict.h, 1566, 1567
gdcmCSAHeaderDictEntry.h, 1569, 1570
gdcmCStoreMessages.h, 2016, 2017
gdcmCurve.h, 1808, 1809
gdcmDataElement.h, 1637, 1638
gdcmDataEvent.h, 1501, 1502
gdcmDataSet.h, 1640, 1642
gdcmDataSetEvent.h, 1645, 1646
gdcmDataSetHelper.h, 1810
gdcmDebugMacro
gdcmTrace.h, 1556
gdcmDecoder.h, 1811, 1812
gdcmDefinedTerms.h, 1743
gdcmDeflateStream.h, 1503
gdcmDefs.h, 1744, 1745
gdcmDeltaEncodingCodec.h, 1813
gdcmDICOMDIR.h, 1814, 1815

gdcmDICOMDIRGenerator.h, 1815, 1816
gdcmDict.h, 1572, 1573
gdcmDictConverter.h, 1577, 1578
gdcmDictEntry.h, 1579, 1580
gdcmDictPrinter.h, 1817, 1818
gdcmDicts.h, 1582, 1583
gdcmDIMSE.h, 2018
gdcmDirectionCosines.h, 1818, 1819
gdcmDirectory.h, 1503, 1504
gdcmDirectoryHelper.h, 1820
gdcmDPath.h, 1821, 1822
gdcmDummyValueGenerator.h, 1506
gdcmDumper.h, 1823, 1824
gdcmElement.h, 1647, 1648
gdcmEmptyMaskGenerator.h, 1825
gdcmEncapsulatedDocument.h, 1826, 1827
gdcmEnumeratedValues.h, 1747
gdcmEquipmentManufacturer.h, 1827, 1828
gdcmErrorMacro
    gdcmTrace.h, 1557
gdcmEvent.h, 1507, 1509
    gdcmEventMacro, 1508
gdcmEventMacro
    gdcmEvent.h, 1508
gdcmException.h, 1510, 1511
gdcmExplicitDataElement.h, 1659, 1660
gdcmExplicitImplicitDataElement.h, 1661, 1662
gdcmFiducials.h, 1829
gdcmFile.h, 1662, 1663
gdcmFileAnonymizer.h, 1830, 1831
gdcmFileChangeTransferSyntax.h, 1831, 1832
gdcmFileDecompressLookupTable.h, 1833, 1834
gdcmFileDerivation.h, 1835
gdcmFileExplicitFilter.h, 1836, 1837
gdcmFileMetaInformation.h, 1664, 1665
gdcmFilename.h, 1512, 1513
gdcmFileNameEvent.h, 1513, 1514
gdcmFilenameGenerator.h, 1515, 1516
gdcmFileSet.h, 1667, 1669
gdcmFileStreamer.h, 1838
gdcmFindPatientRootQuery.h, 2020, 2021
gdcmFindStudyRootQuery.h, 2022
gdcmFragment.h, 1669, 1671
gdcmGlobal.h, 1584, 1585
gdcmGroupDict.h, 1586, 1587
gdcmIconImage.h, 1839, 1840
gdcmIconImageFilter.h, 1841, 1842
gdcmIconImageGenerator.h, 1843, 1844
gdcmImage.h, 1844, 1846
gdcmImageApplyLookupTable.h, 1847
gdcmImageChangePhotometricInterpretation.h, 1848, 1849
gdcmImageChangePlanarConfiguration.h, 1851
gdcmImageChangeTransferSyntax.h, 1852, 1853
gdcmImageCodec.h, 1854, 1855
gdcmImageConverter.h, 1857, 1858
gdcmImageFragmentSplitter.h, 1859
gdcmImageHelper.h, 1860, 1861
gdcmImageReader.h, 1862, 1863
gdcmImageRegionReader.h, 1864, 1865
gdcmImageToImageFilter.h, 1866
gdcmImageWriter.h, 1867, 1868
gdcmImplementationClassUIDSub.h, 2023, 2024
gdcmImplementationUIDSub.h, 2025
gdcmImplementationVersionNameSub.h, 2026, 2027
gdcmImplicitDataElement.h, 1674
gdcmIOD.h, 1748, 1749
gdcmIODEntry.h, 1750, 1752
gdcmIODs.h, 1753, 1754
gdcmIPPSorter.h, 1868, 1869
gdcmItem.h, 1675, 1676
gdcmJPEG12Codec.h, 1870, 1871
gdcmJPEG16Codec.h, 1872
gdcmJPEG2000Codec.h, 1873, 1874
gdcmJPEG8Codec.h, 1875
gdcmJPEGCodec.h, 1876, 1877
gdcmJPEGLSCodec.h, 1879
gdcmJSON.h, 1880, 1881
gdcmKAKADUCodec.h, 1882, 1883
gdcmLegacyMacro.h, 1516, 1518
    GDCM_LEGACY, 1517
    GDCM_LEGACY_BODY, 1517
    GDCM_LEGACY_REPLACED_BODY, 1517
    GDCM_NOOP_STATEMENT, 1518
gdcmLO.h, 1681
gdcmLookupTable.h, 1883, 1884
gdcmMacro.h, 1755, 1757
gdcmMacroEntry.h, 1758, 1760
    GDCMMACROENTRY_H, 1759
GDCMMACROENTRY_H
    gdcmMacroEntry.h, 1759
gdcmMacros.h, 1761, 1762
gdcmMaximumLengthSub.h, 2028, 2029
gdcmMD5.h, 1519, 1520
gdcmMediaStorage.h, 1682, 1683
gdcmMeshPrimitive.h, 1886, 1887
gdcmModalityPerformedProcedureStepCreateQuery.h, 2030
gdcmModalityPerformedProcedureStepSetQuery.h, 2031, 2032
gdcmModule.h, 1763, 1765
gdcmModuleEntry.h, 1766, 1768
gdcmModules.h, 1769, 1770
gdcmMovePatientRootQuery.h, 2032, 2033
gdcmMoveStudyRootQuery.h, 2034
gdcmMrProtocol.h, 1686, 1687
gdcmNActionMessages.h, 2035, 2036
gdcmNCreateMessages.h, 2036, 2037

gdcmNDeleteMessages.h, 2038
gdcmNestedModuleEntries.h, 1771, 1772
gdcmNetworkEvents.h, 2039, 2040
gdcmNetworkStateID.h, 2041, 2042
gdcmNEventReportMessages.h, 2043, 2044
gdcmNGetMessages.h, 2044, 2045
gdcmNormalizedMessageFactory.h, 2045, 2046
gdcmNormalizedNetworkFunctions.h, 2047, 2048
gdcmNSetMessages.h, 2049
gdcmObject.h, 1520, 1521
gdcmOpenSSLCryptoFactory.h, 1523
gdcmOpenSSLCryptographicMessageSyntax.h, 1524, 1525
gdcmOpenSSLP7CryptoFactory.h, 1526, 1527
gdcmOpenSSLP7CryptographicMessageSyntax.h, 1527, 1529
gdcmOrientation.h, 1889
gdcmOverlay.h, 1890, 1891
gdcmParseException.h, 1688, 1689
gdcmParser.h, 1690, 1691
gdcmPatient.h, 1773
gdcmPDataTFPDU.h, 2050, 2051
gdcmPDBElement.h, 1693, 1694
gdcmPDBHeader.h, 1695
gdcmPDFCodec.h, 1893
gdcmPDUFactory.h, 2052
gdcmPersonName.h, 1894, 1895
gdcmPGXCodec.h, 1896
gdcmPhotometricInterpretation.h, 1897, 1898
gdcmPixelFormat.h, 1899, 1901
gdcmPixmap.h, 1903, 1904
gdcmPixmapReader.h, 1905, 1907
gdcmPixmapToPixmapFilter.h, 1908
gdcmPixmapWriter.h, 1909, 1910
gdcmPNMCodec.h, 1911, 1912
gdcmPreamble.h, 1696, 1698
gdcmPresentationContext.h, 2053, 2054
gdcmPresentationContextAC.h, 2055, 2057
gdcmPresentationContextGenerator.h, 2057, 2058
gdcmPresentationContextRQ.h, 2059, 2060
gdcmPresentationDataValue.h, 2061, 2062
gdcmPrinter.h, 1912, 1914
gdcmPrivateTag.h, 1699, 1700
gdcmProgressEvent.h, 1529, 1530
gdcmPVRGCodec.h, 1915, 1916
gdcmPythonFilter.h, 2157, 2158
gdcmQueryBase.h, 2063, 2065
gdcmQueryFactory.h, 2066, 2067
gdcmQueryImage.h, 2067, 2068
gdcmQueryPatient.h, 2069, 2070
gdcmQuerySeries.h, 2071
gdcmQueryStudy.h, 2072, 2073
gdcmRAWCodec.h, 1916, 1917
gdcmReader.h, 1701, 1702
gdcmRegion.h, 1531, 1532
gdcmRescaler.h, 1918
gdcmRLECodec.h, 1920
gdcmRoleSelectionSub.h, 2074
gdcmScanner.h, 1921, 1922
gdcmScanner2.h, 1924, 1925
gdcmSegment.h, 1927, 1929
gdcmSegmentedPaletteColorLookupTable.h, 1931
gdcmSegmentHelper.h, 1932, 1933
gdcmSegmentReader.h, 1934, 1936
gdcmSegmentWriter.h, 1936, 1938
gdcmSequenceOfFragments.h, 1703, 1704
gdcmSequenceOfItems.h, 1708, 1709
gdcmSerieHelper.h, 1938, 1940
gdcmSeries.h, 1774, 1775
gdcmServiceClassApplicationInformation.h, 2075, 2076
gdcmServiceClassUser.h, 2077, 2078
gdcmSHA1.h, 1533, 1534
gdcmSimpleSubjectWatcher.h, 1941, 1942
gdcmSmartPointer.h, 1535
gdcmSOPClassExtendedNegociationSub.h, 2079, 2080
gdcmSOPClassUIDToIOD.h, 1588
gdcmSorter.h, 1943, 1945
gdcmSpacing.h, 1946
gdcmSpectroscopy.h, 1947, 1948
gdcmSplitMosaicFilter.h, 1948, 1949
gdcmStaticAssert.h, 1537, 1538
  GDCM_DO_JOIN, 1537
  GDCM_DO_JOIN2, 1537
  GDCM_JOIN, 1538
  GDCM_STATIC_ASSERT, 1538
gdcmStreamImageReader.h, 1950, 1951
gdcmStreamImageWriter.h, 1952, 1953
gdcmStrictScanner.h, 1954, 1955
gdcmStrictScanner2.h, 1956, 1957
gdcmString.h, 1539, 1540
gdcmStringFilter.h, 1959, 1960
gdcmStudy.h, 1776, 1777
gdcmSubject.h, 1542
gdcmSurface.h, 1961, 1962
gdcmSurfaceHelper.h, 1965, 1966
gdcmSurfaceReader.h, 1968, 1969
gdcmSurfaceWriter.h, 1970, 1971
gdcmSwapCode.h, 1543, 1544
gdcmSwapper.h, 1545, 1546
gdcmSystem.h, 1548
gdcmTable.h, 1777, 1778
gdcmTableEntry.h, 1779, 1780
gdcmTableReader.h, 1781, 1782
gdcmTag.h, 1712, 1713
gdcmTagPath.h, 1971, 1972
gdcmTagToVR.h, 1717
gdcmTerminal.h, 1550, 1551
gdcmTestDriver.h, 1552

gdcmTesting.h, 1553
gdcmTrace.h, 1554, 1558
GDCM_FUNCTION, 1556
gdcmAssertAlwaysMacro, 1556
gdcmAssertMacro, 1556
gdcmDebugMacro, 1556
gdcmErrorMacro, 1557
gdcmWarningMacro, 1557
gdcmTransferSyntax.h, 1718, 1719
gdcmTransferSyntaxSub.h, 2080, 2082
gdcmType.h, 1783, 1784
gdcmTypes.h, 1560, 1561
gdcmUIDGenerator.h, 1973, 1974
gdcmUIDs.h, 1589, 1590
gdcmULAction.h, 2082, 2083
gdcmULActionAA.h, 2084, 2085
gdcmULActionAE.h, 2086, 2087
gdcmULActionAR.h, 2088, 2089
gdcmULActionDT.h, 2091
gdcmULBasicCallback.h, 2092, 2093
gdcmULConnection.h, 2093, 2094
gdcmULConnectionCallback.h, 2096, 2097
gdcmULConnectionInfo.h, 2097, 2099
gdcmULConnectionManager.h, 2099, 2100
gdcmULEvent.h, 2102, 2103
gdcmULTransitionTable.h, 2104, 2105
gdcmULWritingCallback.h, 2107
gdcmUNExplicitDataElement.h, 1720, 1721
gdcmUNExplicitImplicitDataElement.h, 1722, 1723
gdcmUnpacker12Bits.h, 1562
gdcmUsage.h, 1785, 1787
gdcmUserInformation.h, 2108, 2109
gdcmUUIDGenerator.h, 1975
gdcmValidate.h, 1976, 1977
gdcmValue.h, 1723, 1724
gdcmValueIO.h, 1725, 1726
gdcmVersion.h, 1563, 1564
gdcmVL.h, 1726, 1727
gdcmVM.h, 1729, 1730
TYPETOLENGTH, 1730
gdcmVR.h, 1732, 1734
TYPETOENCODING, 1734
VRTypeTemplateCase, 1734
gdcmVR16ExplicitDataElement.h, 1739, 1740
gdcmWarningMacro
gdcmTrace.h, 1557
gdcmWaveform.h, 1977, 1978
gdcmWin32.h, 1564, 1565
GDCM_EXPORT, 1564
gdcmWLMFindQuery.h, 2110, 2111
gdcmWriter.h, 1741, 1742
gdcmXMLDictReader.h, 1788
gdcmXMLPrinter.h, 1978, 1979
gdcmXMLPrivateDictReader.h, 1789, 1790
GEMS
gdcm::Dicts, 389
gdcm::EquipmentManufacturer, 447
GeneralAudioWaveformStorage
gdcm::UIDs, 1221
GeneralECGWaveformStorage
gdcm::MediaStorage, 697
gdcm::UIDs, 1217
GeneralElectricMagneticResonanceImageStorage
gdcm::MediaStorage, 698
GeneralPurposePerformedProcedureStepSOPClass
gdcm::UIDs, 1219
GeneralPurposeScheduledProcedureStepSOPClass
gdcm::UIDs, 1219
GeneralPurposeWorklistInformationModelFIND
gdcm::UIDs, 1218
GeneralPurposeWorklistManagementMetaSOPClass
gdcm::UIDs, 1219
GeneralRelevantPatientInformationQuery
gdcm::UIDs, 1219
Generate
gdcm::DICOMDIRGenerator, 369
gdcm::DummyValueGenerator, 407
gdcm::FilenameGenerator, 504
gdcm::IconImageGenerator, 537
gdcm::UIDGenerator, 1196
gdcm::UUIDGenerator, 1315
GenerateFromFilenames
gdcm::PresentationContextGenerator, 872
GenerateFromUID
gdcm::PresentationContextGenerator, 873
GenerateUUID
gdcm::UIDGenerator, 1196
GenericImplantTemplateInformationModelFIND
gdcm::UIDs, 1223
GenericImplantTemplateInformationModelGET
gdcm::UIDs, 1223
GenericImplantTemplateInformationModelMOVE
gdcm::UIDs, 1223
GenericImplantTemplateStorage
gdcm::UIDs, 1223
GEPrivate3DModelStorage
gdcm::MediaStorage, 698
Get
gdcm::ByteBuffer, 220
GetAbbreviation
gdcm::GroupDict, 531
GetAbstractSyntax
gdcm::network::PresentationContextRQ, 876
gdcm::PresentationContext, 866
GetAbstractSyntaxUID
gdcm::BaseQuery, 182
gdcm::FindPatientRootQuery, 517
gdcm::FindStudyRootQuery, 520

gdcm::ModalityPerformedProcedureStepCreateQuery, 717
gdcm::ModalityPerformedProcedureStepSetQuery, 720
gdcm::MovePatientRootQuery, 734
gdcm::MoveStudyRootQuery, 737
gdcm::WLMFindQuery, 1466
GetAcceptedPresentationContexts
gdcm::network::ULConnection, 1279, 1280
GetAcquisitionSize
gdcm::SplitMosaicFilter, 1057
GetAETitle
gdcm::ServiceClassUser, 1020
GetAlgorithmFamily
gdcm::Surface, 1111
GetAlgorithmName
gdcm::Surface, 1111
GetAlgorithmVersion
gdcm::Surface, 1111
GetALGOType
gdcm::Segment, 973
GetALGOTypeString
gdcm::Segment, 973
GetAllFilenamesFromPrivateTagToValue
gdcm::Scanner2, 964
gdcm::StrictScanner2, 1087
GetAllFilenamesFromPublicTagToValue
gdcm::Scanner2, 964
gdcm::StrictScanner2, 1087
GetAllFilenamesFromTagToValue
gdcm::Scanner, 954
gdcm::StrictScanner, 1077
GetAllRequiredTags
gdcm::QueryBase, 905
GetAllTags
gdcm::QueryBase, 905
GetAnatomicRegion
gdcm::Segment, 974
GetAnatomicRegionModifiers
gdcm::Segment, 974
GetAsDataElement
gdcm::Attribute< Group, Element, TVR, TVM >, 139
gdcm::Attribute< Group, Element, TVR, VM::VM1 >, 148
gdcm::Attribute< Group, Element, TVR, VM::VM1_n >, 157
gdcm::Element< TVR, TVM >, 412
gdcm::Element< TVR, VM::VM1_n >, 419
gdcm::network::AbstractSyntax, 105
gdcm::PrivateTag, 892
GetAsPoints
gdcm::Curve, 316
GetAsString
gdcm::CodeString, 263
GetAxisOfRotation
gdcm::Surface, 1111
GetBasicApplicationLevelConfidentialityProfileAttributes
gdcm::Anonymizer, 116
GetBitPosition
gdcm::Overlay, 788
GetBitsAllocated
gdcm::Overlay, 788
gdcm::PixelFormat, 832
GetBitSample
gdcm::LookupTable, 678
GetBitsStored
gdcm::PixelFormat, 832
GetBlob
gdcm::network::PresentationDataValue, 879
GetBuffer
gdcm::Bitmap, 200
gdcm::ByteValue, 228
gdcm::Parser, 800
gdcm::SequenceOfFragments, 995
GetBuffer2
gdcm::Bitmap, 200
GetBufferAsRGBA
gdcm::LookupTable, 678
GetBufferLength
gdcm::Bitmap, 200
gdcm::JPEGLSCodec, 662
gdcm::PNMCodec, 858
gdcm::RLECodec, 945
GetBuildVersion
gdcm::Version, 1323
GetByteValue
gdcm::CSAElement, 291
gdcm::DataElement, 324
GetCalledAETitle
gdcm::network::AAssociateRQPDU, 99
gdcm::network::ULConnectionInfo, 1287
gdcm::ServiceClassUser, 1020
GetCalledComputerName
gdcm::network::ULConnectionInfo, 1287
GetCalledIPAddress
gdcm::network::ULConnectionInfo, 1287
GetCalledIPPort
gdcm::network::ULConnectionInfo, 1287
GetCallingAETitle
gdcm::network::AAssociateRQPDU, 99
gdcm::network::ULConnectionInfo, 1287
GetCenterOfRotation
gdcm::Surface, 1111
GetCharacterFromCurrentLocale
gdcm::QueryFactory, 908
GetCheckFileMetaInformation
gdcm::Writer, 1471
GetCipherType

gdcm::CAPICryptographicMessageSyntax, 237
gdcm::CryptographicMessageSyntax, 287
gdcm::OpenSSLCryptographicMessageSyntax, 775
gdcm::OpenSSLP7CryptographicMessageSyntax, 780
GetCodec
gdcm::FileChangeTransferSyntax, 472
GetColorLevel
vtkImageColorViewer, 1419
GetColorWindow
vtkImageColorViewer, 1420
GetColumns
gdcm::Bitmap, 200
gdcm::Overlay, 789
GetCommand
gdcm::Subject, 1105
GetConnectionInfo
gdcm::network::ULConnection, 1280
GetConstructorString
gdcm::Dicts, 390
GetContourReferencedFrameOfReferenceClassUID
vtkRTStructSetProperties, 1456
GetContourReferencedFrameOfReferenceInstanceUID
vtkRTStructSetProperties, 1456
GetCryptographicMessageSyntax
gdcm::Anonymizer, 116
GetCSADataInfo
gdcm::CSAHeader, 299
GetCSAEEnd
gdcm::CSAHeader, 300
GetCSAElementByName
gdcm::CSAHeader, 300
GetCSAHeaderDict
gdcm::Dicts, 390
GetCSAHeaderDictEntry
gdcm::CSAHeaderDict, 305
GetCSAImageHeaderInfoTag
gdcm::CSAHeader, 300
GetCSASeriesHeaderInfoTag
gdcm::CSAHeader, 300
GetCTImageSeriesUIDs
gdcm::DirectoryHelper, 402
GetCurrentByteIndex
gdcm::Parser, 800
GetCurrentDateTime
gdcm::System, 1139
GetCurrentModuleFileName
gdcm::System, 1140
GetCurrentProcessFileName
gdcm::System, 1140
GetCurrentResourcesDirectory
gdcm::System, 1140
GetCurve
gdcm::Pixmap, 840, 841
GetCurveDataDescriptor
gdcm::Curve, 316
GetCWD
gdcm::System, 1140
GetData
gdcm::DataEvent, 337
GetDataElement
gdcm::Bitmap, 200, 201
gdcm::DataSet, 344
gdcm::Item, 628
GetDataExtraRoot
gdcm::Testing, 1169
GetDataLength
gdcm::DataEvent, 337
GetDataRoot
gdcm::Testing, 1170
GetDataSet
gdcm::CSAHeader, 301
gdcm::DataSetEvent, 354
gdcm::File, 464
GetDataSetPos
gdcm::network::ULEvent, 1297
GetDataSets
gdcm::network::ULBasicCallback, 1276
GetDataSetTransferSyntax
gdcm::FileMetaInformation, 489
GetDataValueRepresentation
gdcm::Curve, 316
GetDebugFlag
gdcm::Trace, 1177
GetDebugStream
gdcm::Trace, 1177
GetDecodeLength
gdcm::Base64, 172
GetDEEnd
gdcm::DataSet, 344
GetDefaultTransferSyntax
gdcm::PresentationContextGenerator, 873
GetDefs
gdcm::Global, 527
gdcm::TableReader, 1149
GetDES
gdcm::DataSet, 345
GetDescription
gdcm::CSAHeaderDictEntry, 307
gdcm::Exception, 453
gdcm::ModuleEntry, 728
gdcm::Overlay, 789
GetDescriptiveName
vtkGDCMImageReader, 1351
vtkGDCMImageReader2, 1365
vtkGDCMImageWriter, 1379
GetDict
gdcm::XMLDictReader, 1476

GetDictEntry
gdcm::Dict, 373
gdcm::Dicts, 390
gdcm::PrivateDict, 889
GetDictEntryByKeyword
gdcm::Dict, 373
GetDictEntryByName
gdcm::Dict, 373
GetDictName
gdcm::DictConverter, 378
GetDicts
gdcm::Global, 527, 528
GetDictVM
gdcm::Attribute< Group, Element, TVR, TVM >, 139
gdcm::Attribute< Group, Element, TVR, VM::VM1 >, 149
gdcm::Attribute< Group, Element, TVR, VM::VM1_n >, 158
GetDictVR
gdcm::Attribute< Group, Element, TVR, TVM >, 140
gdcm::Attribute< Group, Element, TVR, VM::VM1 >, 149
gdcm::Attribute< Group, Element, TVR, VM::VM1_n >, 158
GetDimension
gdcm::Bitmap, 201
GetDimensions
gdcm::Bitmap, 201
gdcm::Curve, 316
gdcm::ImageCodec, 570
GetDimensionsValue
gdcm::ImageHelper, 584
GetDimensionsValueForResolution
gdcm::StreamImageReader, 1063
GetDirectionCosines
gdcm::Image, 543
GetDirectionCosinesFromDataSet
gdcm::ImageHelper, 584
GetDirectionCosinesTolerance
gdcm::IPPSorter, 623
GetDirectionCosinesValue
gdcm::ImageHelper, 584
GetDirectories
gdcm::Directory, 400
GetElapsedTime
gdcm::network::ARTIMTimer, 131
GetElement
gdcm::Tag, 1156
GetElementTag
gdcm::Tag, 1156
GetEncodeLength
gdcm::Base64, 172
GetErrorCode
gdcm::Parser, 800
GetErrorFlag
gdcm::Trace, 1177
GetErrorStream
gdcm::Trace, 1177
GetErrorString
gdcm::Parser, 801
GetEvent
gdcm::network::ULEvent, 1297
GetEventName
gdcm::AnonymizeEvent, 109
gdcm::DataEvent, 337
gdcm::DataSetEvent, 354
gdcm::Event, 450
gdcm::FileNameEvent, 501
gdcm::ProgressEvent, 897
GetExtension
gdcm::Filename, 496
GetFactoryInstance
gdcm::CryptoFactory, 284
GetFile
gdcm::Anonymizer, 116
gdcm::Cleaner, 250
gdcm::DICOMDIRGenerator, 369
gdcm::FileDecompressLookupTable, 476
gdcm::FileDerivation, 479, 480
gdcm::FileExplicitFilter, 483
gdcm::IconImageFilter, 534
gdcm::PythonFilter, 903
gdcm::Reader, 926
gdcm::SplitMosaicFilter, 1057
gdcm::StreamImageReader, 1063
gdcm::StringFilter, 1100
gdcm::Writer, 1471
vtkGDCMMedicalImageProperties, 1388
GetFileExtensions
vtkGDCMImageReader, 1351
vtkGDCMImageReader2, 1365
vtkGDCMImageWriter, 1380
GetFileMetaInformationVersion
gdcm::FileMetaInformation, 489
GetFileName
gdcm::Filename, 496
gdcm::FileNameEvent, 501
gdcm::Testing, 1170
vtkGDCMImageWriter, 1380
vtkGDCMThreadedImageReader2, 1409
GetFilename
gdcm::FilenameGenerator, 504
gdcm::TableReader, 1150
GetFilenameFromPrivateTagToValue
gdcm::Scanner2, 964
gdcm::StrictScanner2, 1087
GetFilenameFromPublicTagToValue
gdcm::Scanner2, 965

gdcm::StrictScanner2, 1087
GetFilenameFromTagToValue
gdcm::Scanner, 954
gdcm::StrictScanner, 1077
GetFileNames
gdcm::Testing, 1170
GetFilenames
gdcm::Directory, 400
gdcm::FilenameGenerator, 504
gdcm::Scanner, 954
gdcm::Scanner2, 965
gdcm::Sorter, 1049
gdcm::StrictScanner, 1077
gdcm::StrictScanner2, 1087
GetFilenamesFromSeriesUIDs
gdcm::DirectoryHelper, 402
GetFiles
gdcm::FileSet, 507
GetFiniteVolume
gdcm::Surface, 1112
GetFirstSingleSerieUIDFileSet
gdcm::SerieHelper, 1012
GetForcePixelSpacing
gdcm::ImageHelper, 585
GetForceRescaleInterceptSlope
gdcm::ImageHelper, 585
GetFormat
gdcm::CSAHeader, 301
GetFragBuffer
gdcm::SequenceOfFragments, 995
GetFragment
gdcm::SequenceOfFragments, 995
GetFragmentSizeMax
gdcm::ImageFragmentSplitter, 581
GetFrameOfReference
gdcm::DirectoryHelper, 403
GetFullLength
gdcm::FileMetaInformation, 489
GetGDCMDataRoot
vtkGDCMTesting, 1401
GetGDCMImplementationClassUID
gdcm::FileMetaInformation, 489
GetGDCMImplementationVersionName
gdcm::FileMetaInformation, 489
GetGDCMSourceApplicationEntityTitle
gdcm::FileMetaInformation, 489
GetGDCMUID
gdcm::UIDGenerator, 1196
GetGroup
gdcm::Curve, 316
gdcm::Overlay, 789
gdcm::Tag, 1157
GetHasExpired
gdcm::network::ARTIMTimer, 132
GetHeader
gdcm::File, 464, 465
GetHeaderInfo
gdcm::ImageCodec, 570
gdcm::JPEG12Codec, 634
gdcm::JPEG16Codec, 637
gdcm::JPEG2000Codec, 642
gdcm::JPEG8Codec, 648
gdcm::JPEGCodec, 654
gdcm::JPEGLSCodec, 662
gdcm::PGXCodec, 823
gdcm::PNMCodec, 858
gdcm::RAWCodec, 922
gdcm::RLECodec, 945
GetHierachicalSearchTags
gdcm::QueryBase, 906
gdcm::QueryImage, 910
gdcm::QueryPatient, 912
gdcm::QuerySeries, 915
gdcm::QueryStudy, 917
GetHighBit
gdcm::PixelFormat, 832
GetHostName
gdcm::System, 1140
GetIconImage
gdcm::IconImageFilter, 534
gdcm::IconImageGenerator, 537
gdcm::Pixmap, 841
vtkGDCMImageReader, 1351
vtkGDCMImageReader2, 1366
GetIconImagePort
vtkGDCMImageReader2, 1366
GetIE
gdcm::IODEntry, 616
GetImage
gdcm::ImageReader, 591
gdcm::ImageWriter, 603
gdcm::PixmapWriter, 854
gdcm::SplitMosaicFilter, 1057
GetImplementationClassUID
gdcm::FileMetaInformation, 490
GetImplementationVersionName
gdcm::FileMetaInformation, 490
GetIndex
gdcm::SwapCode, 1133
gdcm::VM, 1332
GetInitialized
gdcm::CAPICryptographicMessageSyntax, 237
GetInput
gdcm::ImageToImageFilter, 599
gdcm::PixmapToPixmapFilter, 850
vtkImageColorViewer, 1420
GetInputFilename
gdcm::DictConverter, 378

GetInstance
gdcm::Global, 528
GetIntercept
gdcm::Image, 543
gdcm::Rescaler, 937
GetInterfile
gdcm::CSAHeader, 301
GetInternal
gdcm::Preamble, 861
GetIOD
gdcm::IODs, 620
gdcm::SOPClassUIDToIOD, 1045
GetIODEntry
gdcm::IOD, 613
GetIODFromFile
gdcm::Defs, 360
GetIODFromSOPClassUID
gdcm::SOPClassUIDToIOD, 1045
GetIODNameFromMediaStorage
gdcm::Defs, 360
GetIODs
gdcm::Defs, 360
GetIsCommand
gdcm::network::PresentationDataValue, 880
GetIsLastFragment
gdcm::network::PresentationDataValue, 880
GetIStream
gdcm::network::ULEvent, 1297
GetItem
gdcm::SequenceOfItems, 1004
GetKey
gdcm::CSAElement, 291
GetKeys
gdcm::Scanner, 954
gdcm::Scanner2, 965
gdcm::StrictScanner, 1077
gdcm::StrictScanner2, 1087
GetKeyword
gdcm::DictEntry, 381
GetKeywordFromTag
gdcm::Dict, 374
GetLabel
gdcm::Orientation, 783
GetLastElement
gdcm::ParseException, 796
GetLastSystemError
gdcm::System, 1140
GetLength
gdcm::ByteValue, 228
gdcm::CP246ExplicitDataElement, 280
gdcm::DataElement, 324
gdcm::DataSet, 345
gdcm::Element< TVR, TVM >, 412
gdcm::Element< TVR, VM::VM1_n >, 419
gdcm::Element< VR::AS, VM::VM5 >, 432
gdcm::ExplicitDataElement, 457
gdcm::ExplicitImplicitDataElement, 460
gdcm::Fragment, 524
gdcm::ImplicitDataElement, 609
gdcm::Item, 628
gdcm::Preamble, 861
gdcm::SequenceOfFragments, 995
gdcm::SequenceOfItems, 1004
gdcm::Tag, 1157
gdcm::UNExplicitDataElement, 1303
gdcm::UNExplicitImplicitDataElement, 1306
gdcm::Value, 1319
gdcm::VL, 1326
gdcm::VM, 1332
gdcm::VR, 1338
gdcm::VR16ExplicitDataElement, 1343
GetLocaleCharset
gdcm::System, 1141
GetLossless
gdcm::JPEGCodec, 654
gdcm::JPEGLSCodec, 663
GetLossyFlag
gdcm::ImageCodec, 570
GetLossyFlagFromFile
gdcm::Testing, 1170
GetLUT
gdcm::Bitmap, 201
gdcm::ImageCodec, 570
gdcm::ImageHelper, 585
gdcm::LookupTable, 678
GetLUTDescriptor
gdcm::LookupTable, 678
GetLUTLength
gdcm::LookupTable, 678
GetMacro
gdcm::Macros, 689
GetMacroEntry
gdcm::Macro, 686
GetMacros
gdcm::Defs, 361
GetMajorAxisFromPatientRelativeDirectionCosine
gdcm::Orientation, 783
GetMajorVersion
gdcm::Version, 1324
GetManifold
gdcm::Surface, 1112
GetMapping
gdcm::Scanner, 954
gdcm::StrictScanner, 1077
GetMappingFromPrivateTagToValue
gdcm::Scanner2, 965
gdcm::StrictScanner2, 1088
GetMappingFromPublicTagToValue

gdcm::Scanner2, 965
gdcm::StrictScanner2, 1088
GetMappingFromTagToValue
gdcm::Scanner, 955
gdcm::StrictScanner, 1077
GetMappings
gdcm::Scanner, 955
gdcm::StrictScanner, 1078
GetMax
gdcm::PixelFormat, 832
GetMaximumLength
gdcm::network::MaximumLengthSub, 690
GetMaximumLengthSub
gdcm::network::UserInformation, 1313
GetMaximumPointDistance
gdcm::Surface, 1112
GetMaxLength
gdcm::PersonName, 819
GetMaxPDULength
gdcm::network::ULConnectionInfo, 1287
GetMaxPDUSize
gdcm::network::ULConnection, 1280
GetMD5DataImage
gdcm::Testing, 1171
GetMD5DataImages
gdcm::Testing, 1171
GetMD5FromBrokenFile
gdcm::Testing, 1171
GetMD5FromFile
gdcm::Testing, 1171
GetMD5MetaImage
vtkGDCMTesting, 1401
GetMeanPointDistance
gdcm::Surface, 1112
GetMediaStorage
gdcm::DataSet, 345
gdcm::FileMetaInformation, 490
GetMediaStorageAsString
gdcm::FileMetaInformation, 490
GetMediaStorageDataFile
gdcm::Testing, 1171
GetMediaStorageDataFiles
gdcm::Testing, 1171
GetMediaStorageFromFile
gdcm::Testing, 1172
GetMeshPrimitive
gdcm::Surface, 1112
GetMessageHeader
gdcm::network::PresentationDataValue, 880
GetMetaInformationTS
gdcm::FileMetaInformation, 490
GetMHDMD5FromFile
vtkGDCMTesting, 1401
GetMin
gdcm::PixelFormat, 833
GetMinorVersion
gdcm::Version, 1324
GetModality
gdcm::MediaStorage, 699
GetModalityDimension
gdcm::MediaStorage, 699
GetModule
gdcm::Modules, 731
GetModuleEntry
gdcm::NestedModuleEntries, 752
GetModuleEntryInMacros
gdcm::Module, 724
GetModules
gdcm::Defs, 361
GetMPType
gdcm::MeshPrimitive, 712
GetMPTypeString
gdcm::MeshPrimitive, 712
GetMRImageSeriesUIDs
gdcm::DirectoryHelper, 403
GetMrProtocol
gdcm::CSAHeader, 301
GetMrProtocolByName
gdcm::MrProtocol, 739
GetMSString
gdcm::MediaStorage, 700
GetMSType
gdcm::MediaStorage, 700
GetMTime
vtkImageMapToColors16, 1433
GetName
gdcm::CSAElement, 292
gdcm::CSAHeaderDictEntry, 307
gdcm::DictEntry, 382
gdcm::Filename, 496
gdcm::GroupDict, 531
gdcm::IODEntry, 616
gdcm::Macro, 686
gdcm::Module, 724
gdcm::ModuleEntry, 728
gdcm::network::AbstractSyntax, 105
gdcm::network::ApplicationContext, 122
gdcm::network::TransferSyntaxSub, 1187
gdcm::PDBElement, 807
gdcm::QueryBase, 906
gdcm::QueryImage, 910
gdcm::QueryPatient, 912
gdcm::QuerySeries, 915
gdcm::QueryStudy, 917
gdcm::UIDs, 1233
GetNeedByteSwap
gdcm::Bitmap, 202
gdcm::ImageCodec, 570

GetNegociatedType
gdcm::TransferSyntax, 1184
GetNestedDataSet
gdcm::Item, 628, 629
GetNextSingleSerieUIDFileSet
gdcm::SerieHelper, 1012
GetNoOfItems
gdcm::CSAElement, 292
GetNumberOfComponents
gdcm::PersonName, 819
GetNumberOfContourReferencedFrameOfReferences
vtkRTStructSetProperties, 1456
GetNumberOfCurves
gdcm::Curve, 316
gdcm::Pixmap, 841
GetNumberOfDimensions
gdcm::Bitmap, 202
gdcm::ImageCodec, 571
GetNumberOfElementsFromArray
gdcm::VM, 1332
GetNumberOfFileNames
gdcm::Testing, 1172
GetNumberOfFilenames
gdcm::FilenameGenerator, 504
GetNumberOfFragments
gdcm::SequenceOfFragments, 996
GetNumberOfIconImages
gdcm::IconImageFilter, 535
GetNumberOfImagesInMosaic
gdcm::SplitMosaicFilter, 1058
GetNumberOfIODs
gdcm::IOD, 613
GetNumberOfItems
gdcm::SequenceOfItems, 1004
GetNumberOfMD5DataImages
gdcm::Testing, 1172
GetNumberOfMD5MetaImages
vtkGDCMTesting, 1401
GetNumberOfMediaStorageDataFiles
gdcm::Testing, 1172
GetNumberOfModality
gdcm::MediaStorage, 700
GetNumberOfModuleEntries
gdcm::NestedModuleEntries, 752
GetNumberOfMSString
gdcm::MediaStorage, 700
GetNumberOfMSType
gdcm::MediaStorage, 700
GetNumberOfOverlays
gdcm::Pixmap, 841
GetNumberOfPoints
gdcm::Curve, 316
GetNumberOfPresentationContext
gdcm::network::AAssociateRQPDU, 100
GetNumberOfPresentationContextAC
gdcm::network::AAssociateACPDU, 92
GetNumberOfPresentationDataValues
gdcm::network::PDataTFPDU, 804
GetNumberOfPrimitivesData
gdcm::MeshPrimitive, 713
GetNumberOfReferencedFrameOfReferences
vtkRTStructSetProperties, 1456
GetNumberOfSegments
gdcm::SegmentWriter, 989
GetNumberOfSOPClassToIOD
gdcm::SOPClassUIDToIOD, 1045
GetNumberOfStructureSetROIs
vtkRTStructSetProperties, 1456
GetNumberOfSurfacePoints
gdcm::Surface, 1113
GetNumberOfSurfaces
gdcm::SurfaceReader, 1127
gdcm::SurfaceWriter, 1130
GetNumberOfTransferSyntaxes
gdcm::network::PresentationContextRQ, 876
gdcm::PresentationContext, 866
GetNumberOfTransferSyntaxStrings
gdcm::UIDs, 1233
GetNumberOfValues
gdcm::Attribute< Group, Element, TVR, TVM >, 140
gdcm::Attribute< Group, Element, TVR, VM::VM1 >, 149
gdcm::Attribute< Group, Element, TVR, VM::VM1_n >, 158
GetNumberOfVectors
gdcm::Surface, 1113
GetObliquityThresholdCosineValue
gdcm::Orientation, 783
GetOffScreenRendering
vtkImageColorViewer, 1420
GetOptionalTags
gdcm::QueryBase, 906
gdcm::QueryImage, 910
gdcm::QueryPatient, 913
gdcm::QuerySeries, 915
gdcm::QueryStudy, 918
GetOrderedValues
gdcm::Scanner, 955
gdcm::StrictScanner, 1078
GetOrigin
gdcm::Image, 543
gdcm::Overlay, 789
GetOriginValue
gdcm::ImageHelper, 585
GetOuput
gdcm::ImageConverter, 578
GetOutput
gdcm::BitmapToBitmapFilter, 213

gdcm::ImageToImageFilter, 599
gdcm::PixmapToPixmapFilter, 850
GetOutputAsBitmap
gdcm::BitmapToBitmapFilter, 213
GetOutputAsPixmap
gdcm::PixmapToPixmapFilter, 850
GetOutputFilename
gdcm::DictConverter, 378
GetOutputType
gdcm::DictConverter, 378
GetOverlay
gdcm::Pixmap, 841, 842
vtkGDCMImageReader, 1351
vtkGDCMImageReader2, 1366
GetOverlayData
gdcm::Overlay, 789
GetOverlayPort
vtkGDCMImageReader2, 1366
GetOverlayTypeAsString
gdcm::Overlay, 789
GetOverlayTypeFromString
gdcm::Overlay, 790
GetOverlayVisibility
vtkImageColorViewer, 1420
GetOwner
gdcm::PrivateTag, 892
GetPath
gdcm::Filename, 496
GetPattern
gdcm::FilenameGenerator, 505
GetPDBEEnd
gdcm::PDBHeader, 811
GetPDBElementByName
gdcm::PDBHeader, 811
GetPDBInfoTag
gdcm::PDBHeader, 811
GetPDUs
gdcm::network::ULEvent, 1297
GetPDVs
gdcm::network::PDUFactory, 818
GetPermissions
gdcm::System, 1141
GetPhotometricInterpretation
gdcm::Bitmap, 202
gdcm::ImageChangePhotometricInterpretation, 553
gdcm::ImageCodec, 571
GetPhotometricInterpretationValue
gdcm::ImageHelper, 585
GetPIString
gdcm::PhotometricInterpretation, 826
GetPIType
gdcm::PhotometricInterpretation, 827
GetPixelFormat
gdcm::Bitmap, 202, 203
gdcm::ImageCodec, 571
GetPixelFormatValue
gdcm::ImageHelper, 585
GetPixelRepresentation
gdcm::PixelFormat, 833
GetPixelSize
gdcm::PixelFormat, 833
GetPixelSpacingDataRoot
gdcm::Testing, 1172
GetPixmap
gdcm::FileDecompressLookupTable, 476
gdcm::IconImageGenerator, 538
gdcm::PixmapReader, 846, 847
gdcm::PixmapWriter, 854
GetPlanarConfiguration
gdcm::Bitmap, 203
gdcm::ImageChangePlanarConfiguration, 556
gdcm::ImageCodec, 571
GetPlanarConfigurationValue
gdcm::ImageHelper, 586
GetPMSRescaleInterceptSlope
gdcm::ImageHelper, 586
GetPointCoordinatesData
gdcm::Surface, 1113
GetPointer
gdcm::ByteValue, 229
gdcm::LookupTable, 679
gdcm::SmartPointer< ObjectType >, 1040
vtkLookupTable16, 1450
GetPointerFromElement
gdcm::ImageHelper, 586
GetPointPositionAccuracy
gdcm::Surface, 1113
GetPointsBoundingBoxCoordinates
gdcm::Surface, 1113
GetPosition
vtkImageColorViewer, 1420
GetPreamble
gdcm::FileMetaInformation, 490
GetPrefix
gdcm::FilenameGenerator, 505
GetPresentationContext
gdcm::network::AAssociateRQPDU, 100
GetPresentationContextAC
gdcm::network::AAssociateACPDU, 92
GetPresentationContextACByID
gdcm::network::ULConnection, 1280
GetPresentationContextByAbstractSyntax
gdcm::network::AAssociateRQPDU, 100
GetPresentationContextByID
gdcm::network::AAssociateRQPDU, 100
GetPresentationContextID
gdcm::network::PresentationContextAC, 869
gdcm::network::PresentationContextRQ, 876

gdcm::network::PresentationDataValue, 880
gdcm::PresentationContext, 866
GetPresentationContextIDFromPresentationContext
gdcm::network::ULConnection, 1280
GetPresentationContextRQByID
gdcm::network::ULConnection, 1280
GetPresentationContexts
gdcm::network::AAssociateRQPDU, 100
gdcm::network::ULConnection, 1281
gdcm::PresentationContextGenerator, 873
GetPresentationDataValue
gdcm::network::PDataTFPDU, 805
GetPrettyPrint
gdcm::JSON, 666
GetPrimitiveData
gdcm::MeshPrimitive, 713
GetPrimitivesData
gdcm::MeshPrimitive, 713
GetPrimitiveType
gdcm::MeshPrimitive, 714
GetPrintStyle
gdcm::Printer, 885
gdcm::XMLPrinter, 1478
GetPrivateCreator
gdcm::DataSet, 345
gdcm::Tag, 1157
GetPrivateDict
gdcm::Dicts, 390, 391
gdcm::XMLPrivateDictReader, 1483
GetPrivateMapping
gdcm::Scanner2, 965
gdcm::StrictScanner2, 1088
GetPrivateMappings
gdcm::Scanner2, 966
gdcm::StrictScanner2, 1088
GetPrivateOrderedValues
gdcm::Scanner2, 966
gdcm::StrictScanner2, 1088
GetPrivateTag
gdcm::DataSet, 345
GetPrivateValue
gdcm::Scanner2, 966
gdcm::StrictScanner2, 1088
GetPrivateValues
gdcm::Scanner2, 966
gdcm::StrictScanner2, 1089
GetProcessingAlgorithm
gdcm::Surface, 1114
GetProgress
gdcm::ProgressEvent, 897
GetPropertyCategory
gdcm::Segment, 974
GetPropertyType
gdcm::Segment, 974, 975
GetPropertyTypeModifiers
gdcm::Segment, 975
GetProtocol
gdcm::network::ULConnection, 1281
GetPublicDict
gdcm::Dicts, 391
GetPublicMapping
gdcm::Scanner2, 966
gdcm::StrictScanner2, 1089
GetPublicMappings
gdcm::Scanner2, 966
gdcm::StrictScanner2, 1089
GetPublicOrderedValues
gdcm::Scanner2, 967
gdcm::StrictScanner2, 1089
GetPublicValue
gdcm::Scanner2, 967
gdcm::StrictScanner2, 1089
GetPublicValues
gdcm::Scanner2, 967
gdcm::StrictScanner2, 1089
GetQuality
gdcm::JPEG2000Codec, 642
gdcm::JPEGCodec, 654
GetQueryDataSet
gdcm::BaseQuery, 182
GetQueryLevel
gdcm::QueryBase, 906
gdcm::QueryImage, 910
gdcm::QueryPatient, 913
gdcm::QuerySeries, 915
gdcm::QueryStudy, 918
GetQueryLevelFromQueryRoot
gdcm::BaseRootQuery, 187
GetQueryLevelFromString
gdcm::BaseRootQuery, 187
GetQueryLevelString
gdcm::BaseRootQuery, 187
GetRate
gdcm::JPEG2000Codec, 643
GetRAWMD5FromFile
vtkGDCMTesting, 1402
GetRealWorldValueMappingContent
gdcm::ImageHelper, 586
GetReason
gdcm::network::PresentationContextAC, 869
GetRecommendedDisplayCIELabValue
gdcm::Surface, 1114
GetRecommendedDisplayGrayscaleValue
gdcm::Surface, 1114
GetRecommendedPresentationOpacity
gdcm::Surface, 1114
GetRecommendedPresentationType
gdcm::Surface, 1115

GetRef
gdcm::IODEntry, 616
GetReferencedFrameOfReferenceClassUID
vtkRTStructSetProperties, 1457
GetReferencedFrameOfReferenceInstanceUID
vtkRTStructSetProperties, 1457
GetRegion
gdcm::ImageRegionReader, 596
GetRequiredDataSet
gdcm::ModalityPerformedProcedureStepCreateQuery, 717
gdcm::ModalityPerformedProcedureStepSetQuery, 720
GetRequiredTags
gdcm::QueryBase, 906
gdcm::QueryImage, 911
gdcm::QueryPatient, 913
gdcm::QuerySeries, 915
gdcm::QueryStudy, 918
GetRescaleInterceptSlopeValue
gdcm::ImageHelper, 586
GetReserved43_74
gdcm::network::AAssociateRQPDU, 100
GetResponses
gdcm::network::ULBasicCallback, 1276
GetRetired
gdcm::DictEntry, 382
GetRoot
gdcm::UIDGenerator, 1196
GetRows
gdcm::Bitmap, 203
gdcm::Overlay, 790
GetRTStructSeriesUIDs
gdcm::DirectoryHelper, 403
GetSamplesPerPixel
gdcm::PhotometricInterpretation, 827
gdcm::PixelFormat, 833
GetScalarType
gdcm::PixelFormat, 834
GetScalarTypeAsString
gdcm::PixelFormat, 834
GetScanner
gdcm::DICOMDIRGenerator, 369
GetSegment
gdcm::SegmentWriter, 989
GetSegmentAlgorithmName
gdcm::Segment, 975
GetSegmentAlgorithmType
gdcm::Segment, 975
GetSegmentDescription
gdcm::Segment, 975
GetSegmentLabel
gdcm::Segment, 975
GetSegmentNumber
gdcm::Segment, 976
GetSegments
gdcm::SegmentReader, 985
gdcm::SegmentWriter, 989
GetSelectedPrivateGroupOffsetFromFile
gdcm::Testing, 1172
GetSelectedTagsOffsetFromFile
gdcm::Testing, 1173
GetSequenceOfFragments
gdcm::DataElement, 324
GetSeriesUIDsBySOPClassUID
gdcm::DirectoryHelper, 403
GetSize
gdcm::VR, 1338
vtkImageColorViewer, 1420
GetSizeof
gdcm::VR, 1338
GetSliceArray
gdcm::MrProtocol, 739
GetSliceMax
vtkImageColorViewer, 1420
GetSliceMin
vtkImageColorViewer, 1421
GetSliceRange
vtkImageColorViewer, 1421
GetSlope
gdcm::Image, 544
gdcm::Rescaler, 937
GetSOPClassUID
gdcm::DirectoryHelper, 403
GetSOPClassUIDFromIOD
gdcm::SOPClassUIDToIOD, 1045
GetSOPClassUIDToIOD
gdcm::SOPClassUIDToIOD, 1046
GetSOPClassUIDToIODs
gdcm::SOPClassUIDToIOD, 1046
GetSOPInstanceUID
gdcm::BaseQuery, 182
GetSourceApplicationEntityTitle
gdcm::FileMetaInformation, 491
GetSourceDirectory
gdcm::Testing, 1173
GetSpacing
gdcm::Image, 544
GetSpacingTagFromMediaStorage
gdcm::ImageHelper, 586
GetSpacingValue
gdcm::ImageHelper, 587
GetStart
gdcm::ByteBuffer, 220
GetState
gdcm::network::ULConnection, 1281
GetStateIndex
gdcm::network, 83

GetSTATES
gdcm::Surface, 1115
GetSTATESString
gdcm::Surface, 1115
GetStream
gdcm::Trace, 1178
GetStreamCurrentPosition
gdcm::Reader, 927
GetStreamOffsetFromFile
gdcm::Testing, 1173
GetStreamPtr
gdcm::Reader, 927
gdcm::Writer, 1471
GetString
gdcm::MediaStorage, 701
gdcm::PhotometricInterpretation, 827
gdcm::TransferSyntax, 1184
gdcm::UIDs, 1233
GetStringValueFromTag
gdcm::DirectoryHelper, 403
GetStructureSetObservationNumber
vtkRTStructSetProperties, 1457
GetStructureSetROIDescription
vtkRTStructSetProperties, 1457
GetStructureSetROIGenerationAlgorithm
vtkRTStructSetProperties, 1457
GetStructureSetROIName
vtkRTStructSetProperties, 1457
GetStructureSetROINumber
vtkRTStructSetProperties, 1458
GetStructureSetROIObservationLabel
vtkRTStructSetProperties, 1458
GetStructureSetROIRefFrameRefUID
vtkRTStructSetProperties, 1458
GetStructureSetRTROIInterpretedType
vtkRTStructSetProperties, 1458
GetSurface
gdcm::Segment, 976
GetSurfaceComments
gdcm::Surface, 1115
GetSurfaceCount
gdcm::Segment, 976
GetSurfaceNumber
gdcm::Surface, 1115
GetSurfaceProcessing
gdcm::Surface, 1115
GetSurfaceProcessingDescription
gdcm::Surface, 1115
GetSurfaceProcessingRatio
gdcm::Surface, 1116
GetSurfaces
gdcm::Segment, 976
GetSwapCode
gdcm::TransferSyntax, 1184
GetSwapCodeString
gdcm::SwapCode, 1133
GetSyngoDT
gdcm::CSAElement, 292
GetTable
gdcm::SequenceOfFragments, 996
GetTableEntry
gdcm::Table, 1145
GetTag
gdcm::AnonymizeEvent, 109
gdcm::Attribute< Group, Element, TVR, TVM >, 140
gdcm::Attribute< Group, Element, TVR, VM::VM1 >, 149
gdcm::Attribute< Group, Element, TVR, VM::VM1_n >, 158
gdcm::DataElement, 325
GetTagListByLevel
gdcm::BaseRootQuery, 188
gdcm::FindPatientRootQuery, 517
gdcm::FindStudyRootQuery, 520
gdcm::MovePatientRootQuery, 734
gdcm::MoveStudyRootQuery, 737
gdcm::WLMFindQuery, 1466
GetTempDirectory
gdcm::Testing, 1173
GetTempDirectoryW
gdcm::Testing, 1173
GetTempFilename
gdcm::Testing, 1174
GetTempFilenameW
gdcm::Testing, 1174
GetTimeout
gdcm::network::ARTIMTimer, 132
gdcm::ServiceClassUser, 1020
GetTimer
gdcm::network::ULConnection, 1281
GetTimezoneOffsetFromUTC
gdcm::System, 1141
GetToplevel
gdcm::Directory, 400
GetTransferSyntax
gdcm::Bitmap, 203
gdcm::ImageChangeTransferSyntax, 561
gdcm::network::PresentationContextAC, 869
gdcm::network::PresentationContextRQ, 876
gdcm::PresentationContext, 866
GetTransferSyntaxes
gdcm::network::PresentationContextRQ, 877
GetTransferSyntaxString
gdcm::UIDs, 1233
GetTransferSyntaxStrings
gdcm::UIDs, 1233
GetTSString
gdcm::TransferSyntax, 1184

GetTSType
gdcm::TransferSyntax, 1185
GetType
gdcm::ModuleEntry, 728
gdcm::Orientation, 783
gdcm::Overlay, 790
gdcm::PhotometricInterpretation, 827
GetTypeAsEnum
gdcm::Overlay, 790
GetTypeFromTag
gdcm::Defs, 361
gdcm::IOD, 614
GetTypeOfData
gdcm::Curve, 317
GetTypeOfDataDescription
gdcm::Curve, 317
GetTypeString
gdcm::Type, 1193
GetTypeType
gdcm::Type, 1193
GetUIDName
gdcm::UIDs, 1234
GetUIDString
gdcm::UIDs, 1234
GetUniqueTags
gdcm::QueryBase, 907
gdcm::QueryImage, 911
gdcm::QueryPatient, 913
gdcm::QuerySeries, 916
gdcm::QueryStudy, 918
GetUnpackBuffer
gdcm::Overlay, 790
GetUnpackBufferLength
gdcm::Overlay, 790
GetUsage
gdcm::IODEntry, 616
GetUsageString
gdcm::Usage, 1310
GetUsageType
gdcm::IODEntry, 616
gdcm::Usage, 1310
GetUserData
gdcm::Parser, 801
GetUserInformation
gdcm::network::AAssociateACPDU, 92
gdcm::network::AAssociateRQPDU, 101
GetValidatedFile
gdcm::Validate, 1316
GetValidDataSet
gdcm::WLMFindQuery, 1466
GetValue
gdcm::Attribute< Group, Element, TVR, TVM >, 140, 141
gdcm::Attribute< Group, Element, TVR, VM::VM1 >, 149
gdcm::Attribute< Group, Element, TVR, VM::VM1_n >, 158
gdcm::CSAElement, 292
gdcm::DataElement, 325
gdcm::Element< TVR, TVM >, 412
gdcm::Element< TVR, VM::VM1_n >, 419, 420
gdcm::PDBElement, 808
gdcm::Scanner, 955
gdcm::StrictScanner, 1078
GetValueAsSQ
gdcm::DataElement, 326
GetValues
gdcm::Attribute< Group, Element, TVR, TVM >, 141
gdcm::Attribute< Group, Element, TVR, VM::VM1 >, 150
gdcm::Attribute< Group, Element, TVR, VM::VM1_n >, 159
gdcm::Element< TVR, TVM >, 413
gdcm::Scanner, 955, 956
gdcm::Scanner2, 967
gdcm::StrictScanner, 1078
gdcm::StrictScanner2, 1090
GetVectorAccuracy
gdcm::Surface, 1116
GetVectorCoordinateData
gdcm::Surface, 1116
GetVectorDimensionality
gdcm::Surface, 1116
GetVersion
gdcm::MrProtocol, 740
gdcm::Version, 1324
GetVIEWType
gdcm::Surface, 1116
GetVIEWTypeString
gdcm::Surface, 1116
GetVL
gdcm::DataElement, 326
GetVL16Max
gdcm::VL, 1326
GetVL32Max
gdcm::VL, 1327
GetVM
gdcm::Attribute< Group, Element, TVR, TVM >, 141
gdcm::Attribute< Group, Element, TVR, VM::VM1 >, 150
gdcm::Attribute< Group, Element, TVR, VM::VM1_3 >, 153
gdcm::Attribute< Group, Element, TVR, VM::VM1_8 >, 155
gdcm::Attribute< Group, Element, TVR, VM::VM1_n >, 159

gdcm::Attribute< Group, Element, TVR, VM::VM2_2n >, 163
gdcm::Attribute< Group, Element, TVR, VM::VM2_n >, 164
gdcm::Attribute< Group, Element, TVR, VM::VM3_3n >, 166
gdcm::Attribute< Group, Element, TVR, VM::VM3_n >, 167
gdcm::CSAElement, 293
gdcm::CSAHeaderDictEntry, 308
gdcm::DictEntry, 382
gdcm::Element< TVR, TVM >, 413
gdcm::Element< TVR, VM::VM1_n >, 420
GetVMString
gdcm::VM, 1332
GetVMType
gdcm::VM, 1333
GetVMTypeFromLength
gdcm::VM, 1333
GetVoidPointer
gdcm::ByteValue, 229
GetVR
gdcm::Attribute< Group, Element, TVR, TVM >, 141
gdcm::Attribute< Group, Element, TVR, VM::VM1 >, 150
gdcm::Attribute< Group, Element, TVR, VM::VM1_n >, 159
gdcm::CSAElement, 293
gdcm::CSAHeaderDictEntry, 308
gdcm::DataElement, 327
gdcm::DictEntry, 382
gdcm::Element< TVR, TVM >, 413
gdcm::Element< TVR, VM::VM1_n >, 420
GetVRFromTag
gdcm, 64
GetVRString
gdcm::VR, 1339
GetVRStringFromFile
gdcm::VR, 1339
GetVRType
gdcm::VR, 1339
GetVRTypeFromFile
gdcm::VR, 1339
GetVTKDataRoot
vtkGDCMTesting, 1402
GetWarningFlag
gdcm::Trace, 1178
GetWarningStream
gdcm::Trace, 1178
GetWindowName
vtkImageColorViewer, 1421
GetXMax
gdcm::BoxRegion, 217
GetXMin
gdcm::BoxRegion, 217
GetYMax
gdcm::BoxRegion, 218
GetYMin
gdcm::BoxRegion, 218
GetZMax
gdcm::BoxRegion, 218
GetZMin
gdcm::BoxRegion, 218
GetZSpacing
gdcm::IPPSorter, 623
GetZSpacingTagFromMediaStorage
gdcm::ImageHelper, 587
GetZSpacingTolerance
gdcm::IPPSorter, 623
Global
gdcm::Defs, 363
gdcm::Dicts, 391
gdcm::Global, 526, 527
GlobalInstance
gdcm, 77
GrabOverlayFromPixelData
gdcm::Overlay, 791
Graphics
gdcm::Overlay, 787
GRAY
gdcm::LookupTable, 676
GrayscalePlanarMPRVolumetricPresentationStateStorage
gdcm::UIDs, 1221
GrayscaleSoftcopyPresentationStateStorageSOPClass
gdcm::MediaStorage, 697
gdcm::UIDs, 1217
GREEN
gdcm::LookupTable, 676
green
gdcm::terminal, 85
GroupDict
gdcm::GroupDict, 530
GroupStringVector
gdcm::GroupDict, 530
GuessFromModality
gdcm::MediaStorage, 701

HandleBulkData
gdcm::XMLPrinter, 1478
HandleDataSet
gdcm::network::ULBasicCallback, 1277
gdcm::network::ULConnectionCallback, 1285
gdcm::network::ULWritingCallback, 1301
HandleDescription
gdcm::XMLDictReader, 1476
gdcm::XMLPrivateDictReader, 1483
HandleEntry
gdcm::XMLDictReader, 1476

gdcm::XMLPrivateDictReader, 1483
HandleEvent
gdcm::network::ULTransitionTable, 1299
HandleIOD
gdcm::TableReader, 1150
HandleIODEntry
gdcm::TableReader, 1150
HandleMacro
gdcm::TableReader, 1150
HandleMacroEntry
gdcm::TableReader, 1150
HandleMacroEntryDescription
gdcm::TableReader, 1150
HandleModule
gdcm::TableReader, 1150
HandleModuleEntry
gdcm::TableReader, 1151
HandleModuleEntryDescription
gdcm::TableReader, 1151
HandleModuleInclude
gdcm::TableReader, 1151
HandleResponse
gdcm::network::ULBasicCallback, 1277
gdcm::network::ULConnectionCallback, 1285
gdcm::network::ULWritingCallback, 1301
HangingProtocolInformationModelFIND
gdcm::UIDs, 1219
HangingProtocolInformationModelGET
gdcm::UIDs, 1223
HangingProtocolInformationModelMOVE
gdcm::UIDs, 1219
HangingProtocolStorage
gdcm::MediaStorage, 698
gdcm::UIDs, 1219
HardcopyColorImageStorage
gdcm::MediaStorage, 698
HardcopyColorImageStorageSOPClassRetired
gdcm::UIDs, 1216
HardcopyGrayscaleImageStorage
gdcm::MediaStorage, 697
HardcopyGrayscaleImageStorageSOPClassRetired
gdcm::UIDs, 1216
HasObserver
gdcm::Subject, 1105
HemodynamicWaveformStorage
gdcm::MediaStorage, 697
gdcm::UIDs, 1217
HEVCH_265Main10ProfileLevel5_1
gdcm::UIDs, 1221
HEVCH_265MainProfileLevel5_1
gdcm::UIDs, 1221
hidden
gdcm::terminal, 85
HITACHI
gdcm::EquipmentManufacturer, 447
HotIronColorPaletteSOPInstance
gdcm::UIDs, 1221
HotMetalBlueColorPaletteSOPInstance
gdcm::UIDs, 1220
HSV
gdcm::PhotometricInterpretation, 826

ICBM452T1FrameofReference
gdcm::UIDs, 1215
ICBMSingleSubjectMRIFrameofReference
gdcm::UIDs, 1215
ICD11
gdcm::UIDs, 1220
Icon
gdcm::Pixmap, 843
IconDataScalarType
vtkGDCMImageReader, 1360
vtkGDCMImageReader2, 1374
IconImage
gdcm, 59
IconImageDataExtent
vtkGDCMImageReader, 1360
vtkGDCMImageReader2, 1374
IconImageFilter
gdcm::IconImageFilter, 533
IconImageGenerator
gdcm::IconImageGenerator, 536
IconNumberOfScalarComponents
vtkGDCMImageReader, 1360
vtkGDCMImageReader2, 1375
ID
gdcm::PresentationContext, 867
ignore_char
gdcm::ignore_char, 539
Image
gdcm::Image, 542
ImageActor
vtkImageColorViewer, 1429
ImageApplyLookupTable
gdcm::ImageApplyLookupTable, 549
ImageBiomarkerStandardisationInitiative
gdcm::UIDs, 1220
ImageChangePhotometricInterpretation
gdcm::ImageChangePhotometricInterpretation, 552
gdcm::ImageCodec, 575
ImageChangePlanarConfiguration
gdcm::ImageChangePlanarConfiguration, 556
ImageChangeTransferSyntax
gdcm::Bitmap, 209
gdcm::ImageChangeTransferSyntax, 560
ImageCodec
gdcm::ImageCodec, 566
ImageConverter

gdcm::ImageConverter, 577
ImageFormat
vtkGDCMImageReader, 1360
vtkGDCMImageReader2, 1375
ImageFragmentSplitter
gdcm::ImageFragmentSplitter, 581
ImageNumberOrdering
gdcm::SerieHelper, 1012
ImageOrientationPatient
vtkGDCMImageReader, 1360
vtkGDCMImageReader2, 1375
ImageOverlayBoxSOPClassRetired
gdcm::UIDs, 1216
ImagePositionPatient
vtkGDCMImageReader, 1360
vtkGDCMImageReader2, 1375
ImagePositionPatientOrdering
gdcm::SerieHelper, 1012
ImageReader
gdcm::ImageReader, 591
ImageRegionReader
gdcm::ImageRegionReader, 595
gdcm::JPEG2000Codec, 645
gdcm::JPEGCodec, 657
gdcm::JPEGLSCodec, 664
gdcm::RLECodec, 946
ImageToImageFilter
gdcm::ImageToImageFilter, 599
ImageWriter
gdcm::ImageWriter, 602
ImplantAssemblyTemplateInformationModelFIND
gdcm::UIDs, 1223
ImplantAssemblyTemplateInformationModelGET
gdcm::UIDs, 1223
ImplantAssemblyTemplateInformationModelMOVE
gdcm::UIDs, 1223
ImplantAssemblyTemplateStorage
gdcm::UIDs, 1223
ImplantationPlanSRStorage
gdcm::UIDs, 1222
ImplantTemplateGroupInformationModelFIND
gdcm::UIDs, 1223
ImplantTemplateGroupInformationModelGET
gdcm::UIDs, 1223
ImplantTemplateGroupInformationModelMOVE
gdcm::UIDs, 1223
ImplantTemplateGroupStorage
gdcm::UIDs, 1223
ImplementationClassUIDSub
gdcm::network::ImplementationClassUIDSub, 604
ImplementationUIDSub
gdcm::network::ImplementationUIDSub, 605
ImplementationVersionNameSub
gdcm::network::ImplementationVersionNameSub, 606
Implicit
gdcm::TransferSyntax, 1182
ImplicitVRBigEndianACRNEMA
gdcm::TransferSyntax, 1183
ImplicitVRBigEndianPrivateGE
gdcm::TransferSyntax, 1183
ImplicitVRLittleEndian
gdcm::TransferSyntax, 1183
ImplicitVRLittleEndianDefaultTransferSyntaxforDICOM
gdcm::UIDs, 1214
IncompleteLUT
gdcm::LookupTable, 682
InitFromRQ
gdcm::network::AAssociateACPDU, 92
Initialize
gdcm::network::ULConnectionInfo, 1287
InitializeBlueLUT
gdcm::LookupTable, 679
InitializeConnection
gdcm::network::ULConnection, 1281
gdcm::ServiceClassUser, 1020
Initialized
gdcm::LookupTable, 679
InitializeDataSet
gdcm::BaseRootQuery, 188
gdcm::FindPatientRootQuery, 517
gdcm::FindStudyRootQuery, 520
gdcm::MovePatientRootQuery, 734
gdcm::MoveStudyRootQuery, 737
gdcm::WLMFindQuery, 1466
InitializeGreenLUT
gdcm::LookupTable, 679
InitializeIncomingConnection
gdcm::network::ULConnection, 1281
InitializeLUT
gdcm::LookupTable, 679
InitializeRedLUT
gdcm::LookupTable, 680
InitializeRTStructSet
vtkGDCMPolyDataWriter, 1396
InitOpenSSL
gdcm::OpenSSLCryptoFactory, 772
Input
gdcm::BitmapToBitmapFilter, 214
Insert
gdcm::CommandDataSet, 269
gdcm::DataSet, 346
gdcm::FileMetaInformation, 491
gdcm::GroupDict, 531
InsertDataElement
gdcm::DataSet, 346
gdcm::Item, 629

InsertEntry
gdcm::Table, 1146
InstallPipeline
vtkImageColorViewer, 1421
InstanceAvailabilityNotificationSOPClass
gdcm::UIDs, 1219
INT12
gdcm::PixelFormat, 831
INT16
gdcm::PixelFormat, 831
INT32
gdcm::PixelFormat, 831
INT64
gdcm::PixelFormat, 831
INT8
gdcm::PixelFormat, 831
IntegratedTaxonomicInformationSystemITISTaxonomicSerialNumber15N
gdcm::UIDs, 1220
Interactor
vtkImageColorViewer, 1429
InteractorStyle
vtkImageColorViewer, 1429
INTERFILE
gdcm::CSAHeader, 299
Internal
gdcm::ApplicationEntity, 125
gdcm::Attribute< Group, Element, TVR, TVM >, 145
gdcm::Attribute< Group, Element, TVR, VM::VM1 >, 152
gdcm::Element< TVR, TVM >, 415
gdcm::Element< VR::AS, VM::VM5 >, 432
gdcm::LookupTable, 682
gdcm::UI, 1194
InternalCode
gdcm::Coder, 259
gdcm::JPEG12Codec, 634
gdcm::JPEG16Codec, 637
gdcm::JPEG8Codec, 648
Internals
vtkRTStructSetProperties, 1462
IntraocularLensCalculationsStorage
gdcm::UIDs, 1221
IntravascularOpticalCoherenceTomographyImageStorageForPresentation
gdcm::UIDs, 1221
IntravascularOpticalCoherenceTomographyImageStorageForProcessing
gdcm::UIDs, 1221
INVALID
gdcm::VR, 1336
Invalid
gdcm::Overlay, 787
gdcm::Usage, 1309
InverseRescale
gdcm::Rescaler, 938
InverseRescaleFunctionIntoBestFit
gdcm::Rescaler, 938
InvokeEvent
gdcm::Subject, 1106
IOD
gdcm::IOD, 613
IODEntry
gdcm::IODEntry, 615
IODMapType
gdcm::IODs, 618
IODMapTypeConstIterator
gdcm::IODs, 618
IODName
gdcm::IODs, 619
IODs
gdcm::IODs, 619
IPPSorter
gdcm::IPPSorter, 622
IS
gdcm::VR, 1336
IsAETitleValid
gdcm::network::AAssociateRQPDU, 101
IsASCII
gdcm::VR, 1339
IsASCII2
gdcm::VR, 1339
IsBinary
gdcm::VR, 1340
IsBinary2
gdcm::VR, 1340
IsCompatible
gdcm::PixelFormat, 834
IsDual
gdcm::VR, 1340
IsEmpty
gdcm::Bitmap, 203
gdcm::ByteValue, 230
gdcm::CSAElement, 293
gdcm::CSAHeaderDict, 305
gdcm::Curve, 317
gdcm::DataElement, 327
gdcm::DataSet, 346
gdcm::Defs, 362
gdcm::Dict, 374
gdcm::Dicts, 391
gdcm::Filename, 496
gdcm::Macros, 689
gdcm::Modules, 731
gdcm::Overlay, 791
gdcm::Preamble, 862
gdcm::PrivateDict, 889
gdcm::SegmentHelper::BasicCodedEntry, 192
gdcm::SequenceOfItems, 1005
IsEncapsulated
gdcm::TransferSyntax, 1185

IsEncoded
gdcm::TransferSyntax, 1185
IsExplicit
gdcm::TransferSyntax, 1185
IsFrameEncoder
gdcm::ImageCodec, 571
gdcm::JPEG2000Codec, 643
gdcm::JPEGCodec, 655
gdcm::JPEGLSCodec, 663
gdcm::RLECodec, 945
IsGroupLength
gdcm::Tag, 1157
IsGroupXX
gdcm::Tag, 1158
IsIdentical
gdcm::Filename, 497
IsIllegal
gdcm::Tag, 1158
IsImage
gdcm::MediaStorage, 701
IsImplicit
gdcm::TransferSyntax, 1185
IsInPixelData
gdcm::Overlay, 791
IsKey
gdcm::Scanner, 956
gdcm::Scanner2, 967
gdcm::StrictScanner, 1079
gdcm::StrictScanner2, 1090
IsLastFragment
gdcm::network::AAbortPDU, 88
gdcm::network::AAssociateACPDU, 92
gdcm::network::AAssociateRJPDU, 95
gdcm::network::AAssociateRQPDU, 101
gdcm::network::AReleaseRPPDU, 127
gdcm::network::AReleaseRQPDU, 129
gdcm::network::BasePDU, 178
gdcm::network::PDataTFPDU, 805
IsLossless
gdcm::PhotometricInterpretation, 827
gdcm::TransferSyntax, 1186
IsLossy
gdcm::Bitmap, 204
gdcm::ImageCodec, 572
gdcm::PhotometricInterpretation, 827
gdcm::TransferSyntax, 1186
IsOdd
gdcm::VL, 1327
IsPresentationContextAccepted
gdcm::ServiceClassUser, 1020
IsPrintable
gdcm::ByteValue, 230
IsPrivate
gdcm::Tag, 1158
IsPrivateCreator
gdcm::Tag, 1158
IsPublic
gdcm::Tag, 1159
IsRetired
gdcm::PhotometricInterpretation, 827
IsRGB8
gdcm::LookupTable, 680
IsRowEncoder
gdcm::ImageCodec, 572
gdcm::JPEG2000Codec, 643
gdcm::JPEGCodec, 655
gdcm::JPEGLSCodec, 663
gdcm::RLECodec, 945
IsSameColorSpace
gdcm::PhotometricInterpretation, 828
IsStateSuspension
gdcm::JPEG12Codec, 634
gdcm::JPEG16Codec, 637
gdcm::JPEG8Codec, 648
gdcm::JPEGCodec, 655
IsSwap
gdcm::VR, 1340
IsTransferSyntaxCompatible
gdcm::Bitmap, 204
IsUndefined
gdcm::MediaStorage, 701
gdcm::VL, 1327
IsUndefinedLength
gdcm::DataElement, 327
gdcm::SequenceOfItems, 1005
IsUnique
gdcm::DictEntry, 383
IsValid
gdcm::ApplicationEntity, 124
gdcm::BoxRegion, 218
gdcm::CodeString, 263
gdcm::DirectionCosines, 396
gdcm::DPath, 406
gdcm::FileMetaInformation, 491
gdcm::ImageCodec, 572
gdcm::JPEGCodec, 655
gdcm::LO, 673
gdcm::PixelFormat, 834
gdcm::Preamble, 862
gdcm::Region, 934
gdcm::String< TDelimiter, TMaxLength, TPadChar >, 1097
gdcm::TagPath, 1166
gdcm::TransferSyntax, 1186
gdcm::UIDGenerator, 1197
gdcm::UUIDGenerator, 1315
gdcm::VM, 1333
gdcm::VR, 1340

IsVRFile
gdcm::VR, 1341
IsZero
gdcm::Overlay, 791
Item
gdcm::Item, 627, 628
Items
gdcm::SequenceOfItems, 1007
ItemVector
gdcm::SequenceOfItems, 1001
Iterator
gdcm::CSAHeaderDict, 303
gdcm::DataSet, 341
gdcm::Dict, 372
gdcm::SequenceOfFragments, 993
gdcm::SequenceOfItems, 1002
iterator
gdcm::CodeString, 261
gdcm::LO, 671
gdcm::String< TDelimiter, TMaxLength, TPadChar >, 1095
ItFileSetHt
gdcm::SerieHelper, 1013
IVOCTForPresentation
gdcm::MediaStorage, 698
IVOCTForProcessing
gdcm::MediaStorage, 698

Join
gdcm::Filename, 497
JPEG12Codec
gdcm::JPEG12Codec, 633
JPEG16Codec
gdcm::JPEG16Codec, 636
JPEG2000
gdcm::TransferSyntax, 1183
JPEG2000_COMPRESSION
vtkGDCMImageWriter, 1379
JPEG2000Codec
gdcm::JPEG2000Codec, 640
JPEG2000ImageCompression
gdcm::UIDs, 1214
JPEG2000ImageCompressionLosslessOnly
gdcm::UIDs, 1214
JPEG2000Lossless
gdcm::TransferSyntax, 1183
JPEG2000Part2
gdcm::TransferSyntax, 1183
JPEG2000Part2Lossless
gdcm::TransferSyntax, 1183
JPEG2000Part2MulticomponentImageCompression
gdcm::UIDs, 1214
JPEG2000Part2MulticomponentImageCompressionLosslessOnly
gdcm::UIDs, 1214
JPEG8Codec
gdcm::JPEG8Codec, 647
JPEG_COMPRESSION
vtkGDCMImageWriter, 1379
JPEGBaselineProcess1
gdcm::TransferSyntax, 1183
JPEGBaselineProcess1DefaultTransferSyntaxforLossyJPEG8BitImageComp
gdcm::UIDs, 1214
JPEGCodec
gdcm::JPEGCodec, 651
JPEGExtendedHierarchicalProcess1618Retired
gdcm::UIDs, 1214
JPEGExtendedHierarchicalProcess1719Retired
gdcm::UIDs, 1214
JPEGExtendedProcess24DefaultTransferSyntaxforLossyJPEG12BitImageC
gdcm::UIDs, 1214
JPEGExtendedProcess2_4
gdcm::TransferSyntax, 1183
JPEGExtendedProcess35Retired
gdcm::UIDs, 1214
JPEGExtendedProcess3_5
gdcm::TransferSyntax, 1183
JPEGFullProgressionHierarchicalProcess2426Retired
gdcm::UIDs, 1214
JPEGFullProgressionHierarchicalProcess2527Retired
gdcm::UIDs, 1214
JPEGFullProgressionNonHierarchicalProcess1012Retired
gdcm::UIDs, 1214
JPEGFullProgressionNonHierarchicalProcess1113Retired
gdcm::UIDs, 1214
JPEGFullProgressionProcess10_12
gdcm::TransferSyntax, 1183
JPEGLosslessHierarchicalProcess28Retired
gdcm::UIDs, 1214
JPEGLosslessHierarchicalProcess29Retired
gdcm::UIDs, 1214
JPEGLosslessNonHierarchicalFirstOrderPredictionProcess14SelectionValue
gdcm::UIDs, 1214
JPEGLosslessNonHierarchicalProcess14
gdcm::UIDs, 1214
JPEGLosslessNonHierarchicalProcess15Retired
gdcm::UIDs, 1214
JPEGLosslessProcess14
gdcm::TransferSyntax, 1183
JPEGLosslessProcess14_1
gdcm::TransferSyntax, 1183
JPEGLS_COMPRESSION
vtkGDCMImageWriter, 1379
JPEGLSCodec
gdcm::JPEGLSCodec, 660
JPEGLSLossless
gdcm::TransferSyntax, 1183
JPEGLSLosslessImageCompression
gdcm::UIDs, 1214

JPEGLSLossyNearLosslessImageCompression
gdcm::UIDs, 1214
JPEGLSNearLossless
gdcm::TransferSyntax, 1183
JPEGSpectralSelectionHierarchicalProcess2022Retired
gdcm::UIDs, 1214
JPEGSpectralSelectionHierarchicalProcess2123Retired
gdcm::UIDs, 1214
JPEGSpectralSelectionNonHierarchicalProcess68Retired
gdcm::UIDs, 1214
JPEGSpectralSelectionNonHierarchicalProcess79Retired
gdcm::UIDs, 1214
JPEGSpectralSelectionProcess6_8
gdcm::TransferSyntax, 1183
JPIPReferenced
gdcm::TransferSyntax, 1183
gdcm::UIDs, 1214
JPIPReferencedDeflate
gdcm::UIDs, 1214
JSON
gdcm::JSON, 665
JunkAfterDocElementError
gdcm::Parser, 800

KAKADUCodec
gdcm::KAKADUCodec, 668
KeratometryMeasurementsStorage
gdcm::UIDs, 1221
KeyField
gdcm::CSAElement, 296
KeyObjectSelectionDocument
gdcm::MediaStorage, 698
KeyObjectSelectionDocumentStorage
gdcm::UIDs, 1218
KeyValuePairArrayType
gdcm::CompositeNetworkFunctions, 273
KeyValuePairType
gdcm::CompositeNetworkFunctions, 273
KODAK
gdcm::EquipmentManufacturer, 447

LD_ALL
gdcm, 63
LD_NOSEQ
gdcm, 63
LD_NOSHADOW
gdcm, 63
LD_NOSHADOWSEQ
gdcm, 63
LeadECGWaveformStorage
gdcm::MediaStorage, 697
LegacyConvertedEnhancedCTImageStorage
gdcm::MediaStorage, 698
gdcm::UIDs, 1220
LegacyConvertedEnhancedMRImageStorage
gdcm::MediaStorage, 698
gdcm::UIDs, 1220
LegacyConvertedEnhancedPETImageStorage
gdcm::MediaStorage, 698
gdcm::UIDs, 1220
LensometryMeasurementsStorage
gdcm::UIDs, 1221
Level
vtkImageMapToWindowLevelColors2, 1440
LINE
gdcm::MeshPrimitive, 712
ListCharSets
gdcm::QueryFactory, 908
LittleEndian
gdcm::SwapCode, 1133
LO
gdcm::LO, 673
gdcm::VR, 1336
Load
gdcm::Directory, 400
gdcm::MrProtocol, 740
LOADBULKDATA
gdcm::XMLPrinter, 1478
LoadDefault
gdcm::CSAHeaderDict, 305
gdcm::Dict, 374
gdcm::PrivateDict, 889
LoadDefaults
gdcm::Defs, 362
gdcm::Dicts, 391
LoadFromDataElement
gdcm::CSAHeader, 301
gdcm::PDBHeader, 811
LoadFromFile
gdcm::Defs, 362
LoadIconImage
vtkGDCMImageReader, 1361
vtkGDCMImageReader2, 1375
LoadImageFromFiles
gdcm::DirectoryHelper, 404
LoadOverlays
vtkGDCMImageReader, 1361
vtkGDCMImageReader2, 1375
LoadResourcesFiles
gdcm::Global, 528
LoadSingleFile
vtkGDCMImageReader, 1351
vtkGDCMImageReader2, 1366
Locate
gdcm::Global, 528
LOComp
gdcm, 59
LodModeType
gdcm, 63

LookupTable
gdcm::LookupTable, 676
vtkImageMapToColors16, 1436
LookupTableType
gdcm::LookupTable, 676
LossyFlag
gdcm::Bitmap, 210
gdcm::ImageCodec, 575
vtkGDCMImageReader, 1361
vtkGDCMImageReader2, 1375
LT
gdcm::VR, 1336
LTComp
gdcm, 59
LUT
gdcm::Bitmap, 210
gdcm::ImageCodec, 575
LUTPtr
gdcm::Bitmap, 199
gdcm::ImageCodec, 566

m_char
gdcm::ignore_char, 540
m_ConstMemberFunction
gdcm::MemberCommand< T >, 708
m_DataSet
gdcm::DataSetEvent, 354
m_MemberFunction
gdcm::MemberCommand< T >, 708
gdcm::SimpleMemberCommand< T >, 1031
m_This
gdcm::MemberCommand< T >, 709
gdcm::SimpleMemberCommand< T >, 1031
Macro
gdcm::Macro, 685
MacroEntry
gdcm, 59
Macros
gdcm::Macros, 688
mAction
gdcm::network::Transition, 1191
MacularGridThicknessandVolumeReportStorage
gdcm::UIDs, 1221
magenta
gdcm::terminal, 85
MAGNIFIED
gdcm::Spacing, 1053
MakeDirectory
gdcm::System, 1141
MakeNew
gdcm::network::Transition, 1190
MakeObject
gdcm::AnonymizeEvent, 110
gdcm::DataEvent, 337
gdcm::DataSetEvent, 354
gdcm::Event, 451
gdcm::FileNameEvent, 501
gdcm::ProgressEvent, 898
MammographyCADSR
gdcm::MediaStorage, 698
MammographyCADSRStorage
gdcm::UIDs, 1218
Mandatory
gdcm::Usage, 1309
MANUAL
gdcm::Segment, 973
MapCSAHeaderDictEntry
gdcm::CSAHeaderDict, 304
MapDictEntry
gdcm::Dict, 372
MapIODEntry
gdcm::IOD, 612
MapModuleEntry
gdcm::Macro, 685
gdcm::Module, 723
MappingType
gdcm::Scanner, 951
gdcm::StrictScanner, 1074
MapScalarsThroughTable2
vtkLookupTable16, 1450
MapTableEntry
gdcm::Table, 1145
MARCONI
gdcm::EquipmentManufacturer, 447
Match
gdcm::DPath, 406
MaximumLengthSub
gdcm::network::MaximumLengthSub, 690
MaxLength
gdcm::ApplicationEntity, 125
gdcm::PersonName, 820
MaxNumberOfComponents
gdcm::ApplicationEntity, 125
gdcm::PersonName, 820
MaxPrintLength
gdcm::Printer, 887
MayoClinicNonradiologicalImagesSBSAnatomicalSurfaceRegionGuide
gdcm::UIDs, 1220
mConnection
gdcm::network::ULConnectionManager, 1295
MD5DataImagesType
gdcm::Testing, 1168
MD5MetaImagesType
vtkGDCMTesting, 1400
mDataSet
gdcm::BaseQuery, 184
MediaCreationManagementSOPClassUID
gdcm::UIDs, 1216

MediaStorage
gdcm::MediaStorage, 699
MediaStorageDataFilesType
gdcm::Testing, 1168
MediaStorageDirectoryStorage
gdcm::MediaStorage, 696
gdcm::UIDs, 1215
MedicalImageProperties
vtkGDCMImageReader, 1361
vtkGDCMPolyDataReader, 1394
vtkGDCMPolyDataWriter, 1398
mElementOffsets
gdcm::StreamImageWriter, 1070
mElementOffsets1
gdcm::StreamImageWriter, 1070
MemberCommand
gdcm::MemberCommand< T >, 706
mEnd
gdcm::network::Transition, 1191
MeshPrimitive
gdcm::MeshPrimitive, 712
MessageID
gdcm::network::CEchoRQ, 240
MetaInformationTS
gdcm::FileMetaInformation, 494
mHelpDescription
gdcm::BaseRootQuery, 189
mImage
gdcm::BaseRootQuery, 189
mImplicit
gdcm::network::ULConnectionCallback, 1286
ModalityPerformedProcedureStepCreateQuery
gdcm::ModalityPerformedProcedureStepCreateQuery, 717
ModalityPerformedProcedureStepNotificationSOPClass
gdcm::UIDs, 1215
ModalityPerformedProcedureStepRetrieveSOPClass
gdcm::UIDs, 1215
ModalityPerformedProcedureStepSetQuery
gdcm::ModalityPerformedProcedureStepSetQuery, 720
ModalityPerformedProcedureStepSOPClass
gdcm::MediaStorage, 698
gdcm::UIDs, 1215
ModalityWorklistInformationModelFIND
gdcm::UIDs, 1218
Mode
gdcm::terminal, 85
Module
gdcm::Module, 723
ModuleEntry
gdcm::ModuleEntry, 727
ModuleMapType
gdcm::Macros, 688
gdcm::Modules, 730
Modules
gdcm::Modules, 730
MONOCHROME1
gdcm::PhotometricInterpretation, 826
MONOCHROME2
gdcm::PhotometricInterpretation, 826
MouseGenomeInitiativeMGI
gdcm::UIDs, 1220
MovePatientRootQuery
gdcm::MovePatientRootQuery, 733
MoveStudyRootQuery
gdcm::MoveStudyRootQuery, 736
mPatient
gdcm::BaseRootQuery, 189
MPEG2MainProfile
gdcm::TransferSyntax, 1183
MPEG2MainProfileHighLevel
gdcm::TransferSyntax, 1183
gdcm::UIDs, 1220
MPEG2MainProfileMainLevel
gdcm::UIDs, 1214
MPEG4AVCH264BDcompatibleHighProfileLevel4_1
gdcm::TransferSyntax, 1183
MPEG4AVCH264HighProfileLevel4_1
gdcm::TransferSyntax, 1183
MPEG4AVCH_264BDcompatibleHighProfileLevel4_1
gdcm::UIDs, 1220
MPEG4AVCH_264HighProfileLevel4_1
gdcm::UIDs, 1220
MPEG4AVCH_264HighProfileLevel4_2For2DVideo
gdcm::UIDs, 1221
MPEG4AVCH_264HighProfileLevel4_2For3DVideo
gdcm::UIDs, 1221
MPEG4AVCH_264StereoHighProfileLevel4_2
gdcm::UIDs, 1221
MPType
gdcm::MeshPrimitive, 711
MPType_END
gdcm::MeshPrimitive, 712
MRImageStorage
gdcm::MediaStorage, 696
gdcm::UIDs, 1216
mRootType
gdcm::BaseRootQuery, 189
MrProtocol
gdcm::MrProtocol, 739
MRSpectroscopyStorage
gdcm::MediaStorage, 697
gdcm::UIDs, 1216
MS_END
gdcm::MediaStorage, 698
mSecondaryConnection
gdcm::network::ULConnectionManager, 1295

mSeries
gdcm::BaseRootQuery, 189
mSopInstanceUID
gdcm::BaseQuery, 184
mspFile
gdcm::StreamImageWriter, 1070
mStudy
gdcm::BaseRootQuery, 189
MSType
gdcm::MediaStorage, 696
mTransitions
gdcm::network::ULConnectionManager, 1295
MultiframeGrayscaleByteSecondaryCaptureImageStorage
gdcm::MediaStorage, 697
gdcm::UIDs, 1217
MultiframeGrayscaleWordSecondaryCaptureImageStorage
gdcm::MediaStorage, 697
gdcm::UIDs, 1217
MultiframeSingleBitSecondaryCaptureImageStorage
gdcm::MediaStorage, 697
gdcm::UIDs, 1217
MultiframeTrueColorSecondaryCaptureImageStorage
gdcm::MediaStorage, 697
gdcm::UIDs, 1217
MultipleVolumeRenderingVolumetricPresentationStateStorage
gdcm::UIDs, 1221
mWriter
gdcm::StreamImageWriter, 1070
mXMax
gdcm::StreamImageWriter, 1070
mXMin
gdcm::StreamImageWriter, 1071
mYMax
gdcm::StreamImageWriter, 1071
mYMin
gdcm::StreamImageWriter, 1071
mZMax
gdcm::StreamImageWriter, 1071
mZMin
gdcm::StreamImageWriter, 1071

N_ACTION_RQ
gdcm::network::DIMSE, 393
N_ACTION_RSP
gdcm::network::DIMSE, 393
N_CREATE_RQ
gdcm::network::DIMSE, 393
N_CREATE_RSP
gdcm::network::DIMSE, 393
N_DELETE_RQ
gdcm::network::DIMSE, 393
N_DELETE_RSP
gdcm::network::DIMSE, 393
N_EVENT_REPORT_RQ
gdcm::network::DIMSE, 393
N_EVENT_REPORT_RSP
gdcm::network::DIMSE, 393
N_GET_RQ
gdcm::network::DIMSE, 393
N_GET_RSP
gdcm::network::DIMSE, 393
N_SET_RQ
gdcm::network::DIMSE, 393
N_SET_RSP
gdcm::network::DIMSE, 393
NAction
gdcm::NormalizedNetworkFunctions, 762
Name
gdcm::ModuleEntry, 729
NameField
gdcm::CSAElement, 296
gdcm::PDBElement, 809
NativeDICOMModel
gdcm::UIDs, 1223
NCreate
gdcm::NormalizedNetworkFunctions, 763
NDelete
gdcm::NormalizedNetworkFunctions, 763
NeedByteSwap
gdcm::Bitmap, 210
gdcm::ImageCodec, 575
NeedOverlayCleanup
gdcm::ImageCodec, 576
NegociatedType
gdcm::TransferSyntax, 1182
NestedMacroEntries
gdcm, 60
NestedModuleEntries
gdcm::NestedModuleEntries, 751
NEventReport
gdcm::NormalizedNetworkFunctions, 763
New
gdcm::Anonymizer, 116
gdcm::Cleaner, 250
gdcm::FileChangeTransferSyntax, 473
gdcm::FileStreamer, 511
gdcm::MemberCommand< T >, 707
gdcm::Scanner, 956
gdcm::Scanner2, 968
gdcm::SequenceOfFragments, 996
gdcm::SequenceOfItems, 1005
gdcm::ServiceClassUser, 1020
gdcm::SimpleMemberCommand< T >, 1030
gdcm::StrictScanner, 1079
gdcm::StrictScanner2, 1090
vtkGDCMImageReader, 1352
vtkGDCMImageReader2, 1366
vtkGDCMImageWriter, 1380

vtkGDCMMedicalImageProperties, 1388
vtkGDCMPolyDataReader, 1391
vtkGDCMPolyDataWriter, 1396
vtkGDCMTesting, 1402
vtkGDCMThreadedImageReader, 1405
vtkGDCMThreadedImageReader2, 1409
vtkImageColorViewer, 1421
vtkImageMapToColors16, 1433
vtkImageMapToWindowLevelColors2, 1439
vtkImagePlanarComponentsToComponents, 1442
vtkImageRGBToYBR, 1445
vtkImageYBRToRGB, 1447
vtkLookupTable16, 1450
vtkRTStructSetProperties, 1458
NewYorkUniversityMelanomaClinicalCooperativeGroup
gdcm::UIDs, 1220
NGet
gdcm::NormalizedNetworkFunctions, 763
NO
gdcm::Surface, 1110
NO_COMPRESSION
vtkGDCMImageWriter, 1379
NoElementsError
gdcm::Parser, 800
NoError
gdcm::Parser, 800
NOMAGIC
gdcm::CSAHeader, 299
NoMemoryError
gdcm::Parser, 800
NoObject
gdcm::MediaStorage, 699
NoOfItemsField
gdcm::CSAElement, 296
Normal
gdcm::MrProtocol::Slice, 1036
Normalize
gdcm::DirectionCosines, 396
NSet
gdcm::NormalizedNetworkFunctions, 764
NuclearMedicineImageStorage
gdcm::MediaStorage, 697
gdcm::UIDs, 1217
NuclearMedicineImageStorageRetired
gdcm::MediaStorage, 697
gdcm::UIDs, 1217
Null0
gdcm::UIDs, 1221
Null1
gdcm::UIDs, 1221
NumberOfDimensions
gdcm::Bitmap, 210
gdcm::ImageCodec, 576
NumberOfIconImages
vtkGDCMImageReader, 1361
vtkGDCMImageReader2, 1376
NumberOfOverlays
vtkGDCMImageReader, 1361
vtkGDCMImageReader2, 1376
NumberOfSurfaces
gdcm::SurfaceWriter, 1131

OB
gdcm::VR, 1336
OB_OW
gdcm::VR, 1337
Object
gdcm::Object, 769
ObjectEnd
gdcm::MediaStorage, 699
ObjectType
gdcm::MediaStorage, 699
OBLIQUE
gdcm::Orientation, 782
OD
gdcm::VR, 1336
OF
gdcm::VR, 1336
Ofstream
gdcm::Writer, 1473
OL
gdcm::VR, 1336
OnlyUUID
gdcm::XMLPrinter, 1478
op
gdcm::SerieHelper, 1013
OPENSSL
gdcm::CryptoFactory, 283
OpenSSLCryptoFactory
gdcm::OpenSSLCryptoFactory, 772
OpenSSLCryptographicMessageSyntax
gdcm::OpenSSLCryptographicMessageSyntax, 774
OPENSSLP7
gdcm::CryptoFactory, 283
OpenSSLP7CryptoFactory
gdcm::OpenSSLP7CryptoFactory, 777
OpenSSLP7CryptographicMessageSyntax
gdcm::OpenSSLP7CryptographicMessageSyntax, 779
operator const char *
gdcm::ConstCharWrapper, 279
gdcm::Filename, 497
gdcm::String< TDelimiter, TMaxLength, TPadChar >, 1097
operator const double *
gdcm::DirectionCosines, 397
operator const std::vector< char > &
gdcm::ByteValue, 230

operator MSType
gdcm::MediaStorage, 702
operator ObjectType ∗
gdcm::SmartPointer< ObjectType >, 1041
operator PIType
gdcm::PhotometricInterpretation, 828
operator ScalarType
gdcm::PixelFormat, 835
operator SwapCode::SwapCodeType
gdcm::SwapCode, 1134
operator TSType
gdcm::TransferSyntax, 1186
gdcm::UIDs, 1234
operator TypeType
gdcm::Type, 1193
operator uint32_t
gdcm::VL, 1327
operator UsageType
gdcm::Usage, 1310
operator VMType
gdcm::VM, 1333
operator VRType
gdcm::VR, 1341
operator!=
gdcm, 65
gdcm::Attribute< Group, Element, TVR, TVM >, 142
gdcm::Attribute< Group, Element, TVR, VM::VM1 >, 150
gdcm::CodeString, 264
gdcm::PixelFormat, 835
gdcm::PrivateTag, 892, 893
gdcm::Tag, 1159
operator<
gdcm::Attribute< Group, Element, TVR, TVM >, 142
gdcm::Attribute< Group, Element, TVR, VM::VM1 >, 150
gdcm::CSAElement, 293
gdcm::CSAHeaderDictEntry, 308
gdcm::DataElement, 328
gdcm::DPath, 406
gdcm::PrivateTag, 893
gdcm::Tag, 1159
operator<<
gdcm, 65–75
gdcm::BasicOffsetTable, 195
gdcm::CodeString, 264
gdcm::CommandDataSet, 270
gdcm::CSAElement, 295
gdcm::CSAHeader, 302
gdcm::CSAHeaderDict, 305
gdcm::CSAHeaderDictEntry, 309
gdcm::DataElement, 332
gdcm::DataSet, 351
gdcm::Dict, 375
gdcm::DictEntry, 384
gdcm::Dicts, 392
gdcm::Directory, 401
gdcm::DPath, 406
gdcm::File, 466
gdcm::FileMetaInformation, 493
gdcm::FileSet, 508
gdcm::Fragment, 525
gdcm::Global, 529
gdcm::GroupDict, 532
gdcm::IOD, 614
gdcm::IODEntry, 617
gdcm::IODs, 620
gdcm::Item, 630
gdcm::Macro, 687
gdcm::Macros, 689
gdcm::MediaStorage, 703
gdcm::Module, 725
gdcm::ModuleEntry, 729
gdcm::Modules, 731
gdcm::MrProtocol, 740
gdcm::NestedModuleEntries, 752
gdcm::Object, 770
gdcm::Orientation, 784
gdcm::PDBElement, 808
gdcm::PDBHeader, 812
gdcm::PhotometricInterpretation, 828
gdcm::PixelFormat, 837
gdcm::Preamble, 863
gdcm::PrivateDict, 890
gdcm::PrivateTag, 894
gdcm::Scanner, 957
gdcm::Scanner2, 969
gdcm::Sorter, 1050
gdcm::StrictScanner, 1080
gdcm::StrictScanner2, 1092
gdcm::SwapCode, 1134
gdcm::Table, 1146
gdcm::Tag, 1163
gdcm::TransferSyntax, 1186
gdcm::Type, 1193
gdcm::UI, 1194
gdcm::Usage, 1310
gdcm::Version, 1324
gdcm::VL, 1329
gdcm::VM, 1333
gdcm::VR, 1341
operator<=
gdcm::Tag, 1159
operator>>
gdcm, 75, 76
gdcm::Tag, 1163
operator∗
gdcm::SmartPointer< ObjectType >, 1041

operator()
gdcm::DataSet, 346
gdcm::Scanner2::Itstr, 682
gdcm::Scanner::Itstr, 683
gdcm::StrictScanner2::Itstr, 683
gdcm::StrictScanner::Itstr, 684
operator++
gdcm::VL, 1327
operator+=
gdcm::VL, 1327
operator->
gdcm::SmartPointer< ObjectType >, 1041
operator=
gdcm::AnonymizeEvent, 110
gdcm::ASN1, 134
gdcm::Base64, 172
gdcm::BoxRegion, 218
gdcm::ByteSwapFilter, 224
gdcm::ByteValue, 230
gdcm::Command, 267
gdcm::CryptographicMessageSyntax, 287
gdcm::CSAElement, 293
gdcm::CSAHeaderDict, 305
gdcm::DataElement, 328
gdcm::DataEvent, 338
gdcm::DataSet, 347
gdcm::DataSetEvent, 354
gdcm::Defs, 362
gdcm::Dict, 374
gdcm::Dicts, 391
gdcm::Element< TVR, VM::VM1_n >, 420
gdcm::Event, 451
gdcm::FileMetaInformation, 491
gdcm::FileNameEvent, 501
gdcm::Global, 529
gdcm::MemberCommand< T >, 707
gdcm::network::ULAction, 1237
gdcm::network::ULConnection, 1282
gdcm::network::UserInformation, 1313
gdcm::Object, 770
gdcm::Overlay, 791
gdcm::ParseException, 796
gdcm::Preamble, 862
gdcm::PrivateTag, 893
gdcm::ProgressEvent, 898
gdcm::SequenceOfItems, 1005
gdcm::ServiceClassUser, 1021
gdcm::SHA1, 1026
gdcm::SimpleMemberCommand< T >, 1031
gdcm::SimpleSubjectWatcher, 1033
gdcm::SmartPointer< ObjectType >, 1041, 1042
gdcm::Table, 1146
gdcm::Tag, 1159
operator==
gdcm, 75
gdcm::Attribute< Group, Element, TVR, TVM >, 142
gdcm::Attribute< Group, Element, TVR, VM::VM1 >, 150
gdcm::ByteValue, 230, 231
gdcm::CodeString, 264
gdcm::CSAElement, 294
gdcm::DataElement, 328
gdcm::network::AbstractSyntax, 105
gdcm::network::PresentationContextRQ, 877
gdcm::network::TransferSyntaxSub, 1188
gdcm::PDBElement, 808
gdcm::PixelFormat, 835
gdcm::PresentationContext, 866
gdcm::PrivateTag, 893
gdcm::SequenceOfFragments, 996
gdcm::SequenceOfItems, 1005
gdcm::Tag, 1160
gdcm::Value, 1320
operator[]
gdcm::Attribute< Group, Element, TVR, TVM >, 142
gdcm::Attribute< Group, Element, TVR, VM::VM1_n >, 159
gdcm::DataSet, 347
gdcm::Element< TVR, TVM >, 413
gdcm::Element< TVR, VM::VM1_n >, 420
gdcm::Tag, 1160
OphthalmicAxialMeasurementsStorage
gdcm::UIDs, 1221
OphthalmicOpticalCoherenceTomographyBscanVolumeAnalysisStorage
gdcm::UIDs, 1221
OphthalmicOpticalCoherenceTomographyEnFaceImageStorage
gdcm::UIDs, 1221
OphthalmicPhotography16BitImageStorage
gdcm::MediaStorage, 698
gdcm::UIDs, 1218
OphthalmicPhotography8BitImageStorage
gdcm::MediaStorage, 698
gdcm::UIDs, 1218
OphthalmicThicknessMapStorage
gdcm::UIDs, 1221
OphthalmicTomographyImageStorage
gdcm::MediaStorage, 698
gdcm::UIDs, 1218
OphthalmicVisualFieldStaticPerimetryMeasurementsStorage
gdcm::UIDs, 1221
OrderFileList
gdcm::SerieHelper, 1012
Orientation
gdcm::Orientation, 782
OrientationType
gdcm::Orientation, 782
Output
gdcm::BitmapToBitmapFilter, 214

OutputFormat
vtkImageMapToColors16, 1436
OutputTypes
gdcm::DictConverter, 376
OV
gdcm::VR, 1336
Overlay
gdcm::Overlay, 788
OverlayImageActor
vtkImageColorViewer, 1429
Overlays
gdcm::Pixmap, 843
OverlayType
gdcm::Overlay, 787
OW
gdcm::VR, 1336

Pack
gdcm::Unpacker12Bits, 1307
Padding
gdcm::ApplicationEntity, 125
gdcm::PersonName, 821
PALETTE_COLOR
gdcm::PhotometricInterpretation, 826
Papyrus3ImplicitVRLittleEndian
gdcm::UIDs, 1220
ParametricMapStorage
gdcm::UIDs, 1221
Parent
gdcm::Element< TVR, VM::VM1_2 >, 417
gdcm::Element< TVR, VM::VM2_2n >, 424
gdcm::Element< TVR, VM::VM2_n >, 426
gdcm::Element< TVR, VM::VM3_3n >, 428
gdcm::Element< TVR, VM::VM3_4 >, 429
gdcm::Element< TVR, VM::VM3_n >, 431
Parse
gdcm::Parser, 801
ParseBuffer
gdcm::Parser, 801
ParseCertificateFile
gdcm::CAPICryptographicMessageSyntax, 238
gdcm::CryptographicMessageSyntax, 287
gdcm::OpenSSLCryptographicMessageSyntax, 775
gdcm::OpenSSLP7CryptographicMessageSyntax, 780
ParseDateTime
gdcm::System, 1141, 1142
ParseDump
gdcm::ASN1, 134
ParseDumpFile
gdcm::ASN1, 134
ParseException
gdcm::ParseException, 796
ParseKeyFile
gdcm::CAPICryptographicMessageSyntax, 238
gdcm::CryptographicMessageSyntax, 288
gdcm::OpenSSLCryptographicMessageSyntax, 775
gdcm::OpenSSLP7CryptographicMessageSyntax, 780
Parser
gdcm::Parser, 800
PassAlphaToOutput
vtkImageMapToColors16, 1437
Patient
gdcm::Patient, 802
PatientRadiationDoseSRStorage
gdcm::UIDs, 1222
PatientRootQueryRetrieveInformationModelFIND
gdcm::UIDs, 1218
PatientRootQueryRetrieveInformationModelGET
gdcm::UIDs, 1218
PatientRootQueryRetrieveInformationModelMOVE
gdcm::UIDs, 1218
PatientStudyOnlyQueryRetrieveInformationModelFINDRetired
gdcm::UIDs, 1218
PatientStudyOnlyQueryRetrieveInformationModelGETRetired
gdcm::UIDs, 1218
PatientStudyOnlyQueryRetrieveInformationModelMOVERetired
gdcm::UIDs, 1218
PDataTFPDU
gdcm::network::PDataTFPDU, 804
PDBElement
gdcm::PDBElement, 807
PDBHeader
gdcm::PDBHeader, 810
PDF
gdcm::MediaStorage, 699
PDFCodec
gdcm::PDFCodec, 814
PerformAction
gdcm::network::ULAction, 1237
gdcm::network::ULActionAA1, 1238
gdcm::network::ULActionAA2, 1240
gdcm::network::ULActionAA3, 1241
gdcm::network::ULActionAA4, 1242
gdcm::network::ULActionAA5, 1244
gdcm::network::ULActionAA6, 1245
gdcm::network::ULActionAA7, 1246
gdcm::network::ULActionAA8, 1248
gdcm::network::ULActionAE1, 1249
gdcm::network::ULActionAE2, 1250
gdcm::network::ULActionAE3, 1252
gdcm::network::ULActionAE4, 1253
gdcm::network::ULActionAE5, 1254
gdcm::network::ULActionAE6, 1256
gdcm::network::ULActionAE7, 1257
gdcm::network::ULActionAE8, 1258
gdcm::network::ULActionAR1, 1260

gdcm::network::ULActionAR10, 1261
gdcm::network::ULActionAR2, 1262
gdcm::network::ULActionAR3, 1264
gdcm::network::ULActionAR4, 1265
gdcm::network::ULActionAR5, 1266
gdcm::network::ULActionAR6, 1268
gdcm::network::ULActionAR7, 1269
gdcm::network::ULActionAR8, 1270
gdcm::network::ULActionAR9, 1272
gdcm::network::ULActionDT1, 1273
gdcm::network::ULActionDT2, 1274
PerformedImagingAgentAdministrationSRStorage
gdcm::UIDs, 1222
PET20StepColorPaletteSOPInstance
gdcm::UIDs, 1220
PETColorPaletteSOPInstance
gdcm::UIDs, 1220
PETImageStorage
gdcm::MediaStorage, 697
PF
gdcm::Bitmap, 210
gdcm::ImageCodec, 576
PGXCodec
gdcm::PGXCodec, 822
PHILIPS
gdcm::Dicts, 389
Philips3D
gdcm::MediaStorage, 697
PhilipsPrivateMRSyntheticImageStorage
gdcm::MediaStorage, 698
PhotometricInterpretation
gdcm::PhotometricInterpretation, 826
PI
gdcm::Bitmap, 210
gdcm::ImageCodec, 576
PI_END
gdcm::PhotometricInterpretation, 826
PIType
gdcm::PhotometricInterpretation, 825
PixelData
gdcm::Bitmap, 211
gdcm::PixmapReader, 848
gdcm::PixmapWriter, 855
PixelFormat
gdcm::PixelFormat, 831
Pixmap
gdcm::Pixmap, 840
PixmapReader
gdcm::Bitmap, 209
gdcm::PixmapReader, 846
PixmapToPixmapFilter
gdcm::PixmapToPixmapFilter, 850
PixmapWriter
gdcm::PixmapWriter, 853
PlanarConfiguration
gdcm::Bitmap, 211
gdcm::ImageCodec, 576
vtkGDCMImageReader, 1361
vtkGDCMImageReader2, 1376
PlannedImagingAgentAdministrationSRStorage
gdcm::UIDs, 1222
PMS
gdcm::EquipmentManufacturer, 447
PN
gdcm::VR, 1336
PNComp
gdcm, 60
PNMCodec
gdcm::PNMCodec, 857
pointer
gdcm::CodeString, 262
gdcm::LO, 672
gdcm::String< TDelimiter, TMaxLength, TPadChar >, 1095
POINTS
gdcm::Surface, 1110
Position
gdcm::MrProtocol::Slice, 1036
PositronEmissionTomographyImageStorage
gdcm::UIDs, 1218
Preamble
gdcm::Preamble, 860, 861
PrepareWrite
gdcm::PixmapWriter, 854
gdcm::SegmentWriter, 989
gdcm::SurfaceWriter, 1131
PrepareWritePointMacro
gdcm::SurfaceWriter, 1131
Prepend
gdcm::Global, 529
PresentationContext
gdcm::PresentationContext, 865
PresentationContextAC
gdcm::network::PresentationContextAC, 868
PresentationContextArrayType
gdcm::network::AAssociateRQPDU, 98
gdcm::PresentationContextGenerator, 872
PresentationContextGenerator
gdcm::PresentationContextGenerator, 872
PresentationContextRQ
gdcm::network::PresentationContextRQ, 875
PresentationDataValue
gdcm::network::PresentationDataValue, 879
PresentationLUTSOPClass
gdcm::UIDs, 1216
Preserve
gdcm::Cleaner, 250
PrettyPrintOff

gdcm::JSON, 666
PrettyPrintOn
gdcm::JSON, 666
PrimitiveData
gdcm::MeshPrimitive, 714
PrimitivesData
gdcm::MeshPrimitive, 711
PrimitiveType
gdcm::MeshPrimitive, 715
Print
gdcm::ApplicationEntity, 124
gdcm::Attribute< Group, Element, TVR, TVM >, 143
gdcm::Attribute< Group, Element, TVR, VM::VM1 >, 151
gdcm::Attribute< Group, Element, TVR, VM::VM1_n >, 159
gdcm::BaseQuery, 182
gdcm::Bitmap, 204
gdcm::BoxRegion, 219
gdcm::ByteValue, 231
gdcm::CSAHeader, 302
gdcm::Curve, 317
gdcm::DataSet, 347
gdcm::DictPrinter, 387
gdcm::DirectionCosines, 397
gdcm::Directory, 401
gdcm::DPath, 406
gdcm::Element< TVR, TVM >, 413
gdcm::Element< TVR, VM::VM1_n >, 420
gdcm::Element< VR::AS, VM::VM5 >, 432
gdcm::Event, 451
gdcm::Image, 544
gdcm::LookupTable, 680
gdcm::MrProtocol, 740
gdcm::network::AAbortPDU, 88
gdcm::network::AAssociateACPDU, 93
gdcm::network::AAssociateRJPDU, 96
gdcm::network::AAssociateRQPDU, 101
gdcm::network::AbstractSyntax, 105
gdcm::network::ApplicationContext, 122
gdcm::network::AReleaseRPPDU, 127
gdcm::network::AReleaseRQPDU, 130
gdcm::network::AsynchronousOperationsWindowSub, 135
gdcm::network::BasePDU, 178
gdcm::network::ImplementationClassUIDSub, 604
gdcm::network::ImplementationVersionNameSub, 607
gdcm::network::MaximumLengthSub, 690
gdcm::network::PDataTFPDU, 805
gdcm::network::PresentationContextAC, 869
gdcm::network::PresentationContextRQ, 877
gdcm::network::PresentationDataValue, 880
gdcm::network::RoleSelectionSub, 947
gdcm::network::ServiceClassApplicationInformation, 1015
gdcm::network::SOPClassExtendedNegociationSub, 1043
gdcm::network::TransferSyntaxSub, 1188
gdcm::network::UserInformation, 1313
gdcm::Object, 770
gdcm::Orientation, 783
gdcm::Overlay, 792
gdcm::PDBHeader, 812
gdcm::PersonName, 819
gdcm::PixelFormat, 835
gdcm::Pixmap, 842
gdcm::Preamble, 862
gdcm::PresentationContext, 867
gdcm::Printer, 885
gdcm::Region, 934
gdcm::Scanner, 956
gdcm::Scanner2, 968
gdcm::SegmentedPaletteColorLookupTable, 982
gdcm::SequenceOfFragments, 997
gdcm::SequenceOfItems, 1006
gdcm::Sorter, 1049
gdcm::StrictScanner, 1079
gdcm::StrictScanner2, 1090
gdcm::TagPath, 1166
gdcm::Testing, 1174
gdcm::Version, 1324
gdcm::XMLPrinter, 1479
PrintASCII
gdcm::ByteValue, 231
PrintASCIIXML
gdcm::ByteValue, 231
PrintAsContinuousString
gdcm::Tag, 1160
PrintAsContinuousUpperCaseString
gdcm::Tag, 1160
PrintAsPipeSeparatedString
gdcm::Tag, 1161
PrintDataElement
gdcm::Printer, 885
gdcm::XMLPrinter, 1479
PrintDataElement2
gdcm::DictPrinter, 387
PrintDataSet
gdcm::Printer, 885
gdcm::XMLPrinter, 1479
PrintDataSet2
gdcm::DictPrinter, 387
Printer
gdcm::Printer, 884
PrinterConfigurationRetrievalSOPClass
gdcm::UIDs, 1216
PrinterConfigurationRetrievalSOPInstance

gdcm::UIDs, 1216
PrinterSOPClass
gdcm::UIDs, 1216
PrinterSOPInstance
gdcm::UIDs, 1216
PrintGroupLength
gdcm::ByteValue, 231
PrintHex
gdcm::ByteValue, 231
PrintHexXML
gdcm::ByteValue, 232
PrintJobSOPClass
gdcm::UIDs, 1216
PrintPNXML
gdcm::ByteValue, 232
PrintQueueManagementSOPClassRetired
gdcm::UIDs, 1216
PrintQueueSOPInstanceRetired
gdcm::UIDs, 1216
PrintSelf
vtkGDCMImageReader, 1352
vtkGDCMImageReader2, 1367
vtkGDCMImageWriter, 1380
vtkGDCMMedicalImageProperties, 1388
vtkGDCMPolyDataReader, 1391
vtkGDCMPolyDataWriter, 1397
vtkGDCMTesting, 1402
vtkGDCMThreadedImageReader, 1405
vtkGDCMThreadedImageReader2, 1409
vtkImageColorViewer, 1422
vtkImageMapToColors16, 1433
vtkImageMapToWindowLevelColors2, 1439
vtkImagePlanarComponentsToComponents, 1442
vtkImageRGBToYBR, 1445
vtkImageYBRToRGB, 1447
vtkLookupTable16, 1451
vtkRTStructSetProperties, 1458
PrintSQ
gdcm::Printer, 886
gdcm::XMLPrinter, 1479
PrintStyle
gdcm::Printer, 887
gdcm::XMLPrinter, 1480
PrintStyles
gdcm::Printer, 884
gdcm::XMLPrinter, 1478
PrintTable
gdcm::network::ULTransitionTable, 1299
gdcm::Scanner, 957
gdcm::Scanner2, 968
gdcm::StrictScanner, 1079
gdcm::StrictScanner2, 1090
PrintXML
gdcm::PrivateDict, 889
PrivateBegin
gdcm::Scanner2, 968
gdcm::StrictScanner2, 1091
PrivateConstIterator
gdcm::Scanner2, 961
gdcm::StrictScanner2, 1084
PrivateDict
gdcm::PrivateDict, 888
PrivateEnd
gdcm::Scanner2, 968
gdcm::StrictScanner2, 1091
PrivateMappingType
gdcm::Scanner2, 961
gdcm::StrictScanner2, 1084
PrivateTag
gdcm::PrivateTag, 892
PrivateTagToValue
gdcm::Scanner2, 961
gdcm::StrictScanner2, 1084
PrivateTagToValueValueType
gdcm::Scanner2, 962
gdcm::StrictScanner2, 1084
ProceduralEventLoggingSOPClass
gdcm::UIDs, 1215
ProceduralEventLoggingSOPInstance
gdcm::UIDs, 1215
ProcedureLogStorage
gdcm::UIDs, 1218
Process
gdcm::Parser, 801
ProcessDataSet
gdcm::FileExplicitFilter, 483
ProcessPrivateTag
gdcm::Scanner2, 969
gdcm::StrictScanner2, 1091
ProcessPublicTag
gdcm::Scanner, 957
gdcm::Scanner2, 969
gdcm::StrictScanner, 1080
gdcm::StrictScanner2, 1091
ProcessRequest
vtkGDCMImageReader2, 1367
ProduceCharacterSetDataElement
gdcm::QueryFactory, 908
ProduceQuery
gdcm::QueryFactory, 908
ProductCharacteristicsQuerySOPClass
gdcm::UIDs, 1219
ProgressEvent
gdcm::ProgressEvent, 896, 897
PropertyCategory
gdcm::Segment, 979
PropertyType
gdcm::Segment, 979

PropertyTypeModifiers
gdcm::Segment, 979
ProtocolApprovalInformationModelFIND
gdcm::UIDs, 1222
ProtocolApprovalInformationModelGET
gdcm::UIDs, 1222
ProtocolApprovalInformationModelMOVE
gdcm::UIDs, 1222
ProtocolApprovalStorage
gdcm::UIDs, 1222
PseudoColorSoftcopyPresentationStateStorageSOPClass
gdcm::UIDs, 1217
PubChemCompoundCID
gdcm::UIDs, 1220
PublicConstIterator
gdcm::Scanner2, 962
gdcm::StrictScanner2, 1084
PublicMappingType
gdcm::Scanner2, 962
gdcm::StrictScanner2, 1084
PublicTagToValue
gdcm::Scanner2, 962
gdcm::StrictScanner2, 1084
PublicTagToValueValueType
gdcm::Scanner2, 962
gdcm::StrictScanner2, 1085
PullPrintRequestSOPClassRetired
gdcm::UIDs, 1216
PullStoredPrintManagementMetaSOPClassRetired
gdcm::UIDs, 1216
Push
gdcm::TagPath, 1166
PushBackFile
vtkGDCMMedicalImageProperties, 1388
PVRGCodec
gdcm::PVRGCodec, 900
PythonFilter
gdcm::PythonFilter, 903

Quality
gdcm::JPEGCodec, 657
QueryFactory
gdcm::BaseQuery, 184
gdcm::BaseRootQuery, 189
gdcm::FindPatientRootQuery, 518
gdcm::FindStudyRootQuery, 521
gdcm::ModalityPerformedProcedureStepCreateQuery, 717
gdcm::ModalityPerformedProcedureStepSetQuery, 720
gdcm::MovePatientRootQuery, 735
gdcm::MoveStudyRootQuery, 738
gdcm::WLMFindQuery, 1467

RadiomicsOntology
gdcm::UIDs, 1220
RadiopharmaceuticalRadiationDoseSRStorage
gdcm::UIDs, 1222
RAWCodec
gdcm::RAWCodec, 920
RawDataStorage
gdcm::MediaStorage, 697
gdcm::UIDs, 1217
Read
gdcm::BasicOffsetTable, 195
gdcm::ByteValue, 232
gdcm::CommandDataSet, 270
gdcm::CP246ExplicitDataElement, 281
gdcm::DataElement, 328
gdcm::DataSet, 347
gdcm::Element< TVR, TVM >, 414
gdcm::Element< TVR, VM::VM1_n >, 421
gdcm::EncodingImplementation< VR::VRASCII >, 442
gdcm::EncodingImplementation< VR::VRBINARY >, 444
gdcm::ExplicitDataElement, 457
gdcm::ExplicitImplicitDataElement, 460
gdcm::File, 465
gdcm::FileMetaInformation, 491
gdcm::Fragment, 524
gdcm::ImageReader, 592
gdcm::ImageRegionReader, 596
gdcm::ImplicitDataElement, 609
gdcm::Item, 629
gdcm::network::AAbortPDU, 88
gdcm::network::AAssociateACPDU, 93
gdcm::network::AAssociateRJPDU, 96
gdcm::network::AAssociateRQPDU, 101
gdcm::network::AbstractSyntax, 105
gdcm::network::ApplicationContext, 122
gdcm::network::AReleaseRPPDU, 127
gdcm::network::AReleaseRQPDU, 130
gdcm::network::AsynchronousOperationsWindowSub, 135
gdcm::network::BasePDU, 178
gdcm::network::ImplementationClassUIDSub, 604
gdcm::network::ImplementationVersionNameSub, 607
gdcm::network::MaximumLengthSub, 690
gdcm::network::PDataTFPDU, 805
gdcm::network::PresentationContextAC, 869
gdcm::network::PresentationContextRQ, 877
gdcm::network::PresentationDataValue, 880
gdcm::network::RoleSelectionSub, 948
gdcm::network::ServiceClassApplicationInformation, 1016
gdcm::network::SOPClassExtendedNegociationSub, 1043

gdcm::network::TransferSyntaxSub, 1188
gdcm::network::UserInformation, 1314
gdcm::PGXCodec, 823
gdcm::PixmapReader, 847
gdcm::PNMCodec, 858
gdcm::Preamble, 862
gdcm::Reader, 927
gdcm::SegmentReader, 985
gdcm::SequenceOfFragments, 997
gdcm::SequenceOfItems, 1006
gdcm::StreamImageReader, 1063
gdcm::SurfaceReader, 1128
gdcm::TableReader, 1151
gdcm::Tag, 1161
gdcm::UNExplicitDataElement, 1304
gdcm::UNExplicitImplicitDataElement, 1306
gdcm::ValueIO< TDE, TSwap, TType >, 1321
gdcm::VL, 1328
gdcm::VR, 1341
gdcm::VR16ExplicitDataElement, 1344
gdcm::VRVLSize< 0 >, 1346
gdcm::VRVLSize< 1 >, 1346
Read16
gdcm::VL, 1328
ReadACRNEMAImage
gdcm::ImageReader, 592
gdcm::PixmapReader, 847
ReadBacktrack
gdcm::Fragment, 524
ReadCompat
gdcm::FileMetaInformation, 491
ReadCompatInternal
gdcm::FileMetaInformation, 492
ReadComputeLength
gdcm::EncodingImplementation< VR::VRASCII >, 442
gdcm::EncodingImplementation< VR::VRBINARY >, 444
ReadDataSet
gdcm::Reader, 928
Reader
gdcm::Reader, 926
ReadFiles
vtkGDCMThreadedImageReader, 1405
ReadFromCommaSeparatedString
gdcm::PrivateTag, 894
gdcm::Tag, 1161
ReadFromContinuousString
gdcm::Tag, 1161
ReadFromPipeSeparatedString
gdcm::Tag, 1161
ReadImage
gdcm::ImageReader, 592
gdcm::PixmapReader, 847
ReadImageInformation
gdcm::StreamImageReader, 1063
ReadImageInternal
gdcm::PixmapReader, 848
ReadInformation
gdcm::ImageRegionReader, 596
ReadInto
gdcm::network::PDataTFPDU, 805
gdcm::network::PresentationDataValue, 880
ReadIntoBuffer
gdcm::ImageRegionReader, 596
README.txt, 1485
ReadMetaInformation
gdcm::Reader, 928
ReadNested
gdcm::DataSet, 347
ReadNoSwap
gdcm::EncodingImplementation< VR::VRASCII >, 442
gdcm::EncodingImplementation< VR::VRBINARY >, 444
ReadOrSkip
gdcm::DataElement, 328
ReadPointMacro
gdcm::SurfaceReader, 1128
ReadPreamble
gdcm::Reader, 928
ReadPreValue
gdcm::CP246ExplicitDataElement, 281
gdcm::DataElement, 329
gdcm::ExplicitDataElement, 457
gdcm::ExplicitImplicitDataElement, 460
gdcm::Fragment, 524
gdcm::ImplicitDataElement, 609
gdcm::SequenceOfFragments, 997
gdcm::UNExplicitDataElement, 1304
gdcm::UNExplicitImplicitDataElement, 1306
gdcm::VR16ExplicitDataElement, 1344
ReadSegment
gdcm::SegmentReader, 986
ReadSegments
gdcm::SegmentReader, 986
ReadSelectedPrivateTags
gdcm::DataSet, 348
gdcm::Reader, 928
ReadSelectedPrivateTagsWithLength
gdcm::DataSet, 348
ReadSelectedTags
gdcm::DataSet, 348
gdcm::Reader, 929
ReadSelectedTagsWithLength
gdcm::DataSet, 348
ReadSurface
gdcm::SurfaceReader, 1128

ReadSurfaces
gdcm::SurfaceReader, 1128
Readuint16
gdcm::DictConverter, 378
ReadUpToTag
gdcm::DataSet, 348
gdcm::Reader, 929
ReadUpToTagWithLength
gdcm::DataSet, 349
ReadValue
gdcm::CP246ExplicitDataElement, 281
gdcm::DataElement, 329
gdcm::ExplicitDataElement, 457
gdcm::ExplicitImplicitDataElement, 460
gdcm::Fragment, 524
gdcm::ImplicitDataElement, 610
gdcm::SequenceOfFragments, 997
gdcm::UNExplicitDataElement, 1304
gdcm::UNExplicitImplicitDataElement, 1306
gdcm::VR16ExplicitDataElement, 1344
ReadValueWithLength
gdcm::DataElement, 329
gdcm::ImplicitDataElement, 610
ReadVM
gdcm::DictConverter, 378
ReadVR
gdcm::DictConverter, 379
ReadWithLength
gdcm::CP246ExplicitDataElement, 281
gdcm::DataElement, 329
gdcm::DataSet, 349
gdcm::ExplicitDataElement, 457
gdcm::ExplicitImplicitDataElement, 460
gdcm::ImplicitDataElement, 610
gdcm::UNExplicitDataElement, 1304
gdcm::VR16ExplicitDataElement, 1344
RealWorldValueIntercept
gdcm::RealWorldValueMappingContent, 932
RealWorldValueMappingStorage
gdcm::UIDs, 1217
RealWorldValueSlope
gdcm::RealWorldValueMappingContent, 932
RecommendedDisplayCIELabToRGB
gdcm::SurfaceHelper, 1123
RecurseDataSet
gdcm::Anonymizer, 117
RED
gdcm::LookupTable, 676
red
gdcm::terminal, 85
reference
gdcm::CodeString, 262
gdcm::LO, 672
gdcm::String< TDelimiter, TMaxLength, TPadChar >, 1095
ReferencedColorPrintManagementMetaSOPClassRetired
gdcm::UIDs, 1216
ReferencedGrayscalePrintManagementMetaSOPClassRetired
gdcm::UIDs, 1216
ReferencedImageBoxSOPClassRetired
gdcm::UIDs, 1216
ReferenceFrameOfReferenceUID
vtkRTStructSetProperties, 1462
ReferenceSeriesInstanceUID
vtkRTStructSetProperties, 1462
Region
gdcm::Region, 933
Register
gdcm::Object, 770
Remove
gdcm::Anonymizer, 117
gdcm::Cleaner, 250, 251
gdcm::DataSet, 349
gdcm::FileAnonymizer, 468
gdcm::Preamble, 862
RemoveAllGroupLength
gdcm::Cleaner, 251
RemoveAllIllegal
gdcm::Cleaner, 251
RemoveAllMissingPrivateCreator
gdcm::Cleaner, 251
RemoveAllObservers
gdcm::Subject, 1106
RemoveDictEntry
gdcm::PrivateDict, 889
RemoveFile
gdcm::System, 1142
RemoveGroupLength
gdcm::Anonymizer, 117
RemoveItemByIndex
gdcm::SequenceOfItems, 1006
RemoveMissingPrivateCreator
gdcm::Cleaner, 251
RemoveObserver
gdcm::Subject, 1106
RemoveOverlay
gdcm::Pixmap, 842
RemovePrivateTags
gdcm::Anonymizer, 117
RemoveRetired
gdcm::Anonymizer, 118
Render
vtkImageColorViewer, 1422
Renderer
vtkImageColorViewer, 1429
RenderWindow
vtkImageColorViewer, 1430

Replace
gdcm::Anonymizer, 118
gdcm::CommandDataSet, 270
gdcm::DataSet, 349
gdcm::FileAnonymizer, 469
gdcm::FileMetaInformation, 492
ReplaceEmpty
gdcm::DataSet, 350
RequestData
vtkGDCMImageReader2, 1367
vtkGDCMPolyDataReader, 1392
vtkImageMapToColors16, 1433
vtkImageMapToWindowLevelColors2, 1439
vtkImagePlanarComponentsToComponents, 1443
RequestData_HemodynamicWaveformStorage
vtkGDCMPolyDataReader, 1392
RequestData_RTStructureSetStorage
vtkGDCMPolyDataReader, 1392
RequestDataCompat
vtkGDCMImageReader, 1352
vtkGDCMImageReader2, 1367
vtkGDCMThreadedImageReader, 1406
RequestInformation
vtkGDCMImageReader2, 1367
vtkGDCMPolyDataReader, 1392
vtkGDCMThreadedImageReader2, 1410
vtkImageMapToColors16, 1433
vtkImageMapToWindowLevelColors2, 1439
RequestInformation_HemodynamicWaveformStorage
vtkGDCMPolyDataReader, 1392
RequestInformation_RTStructureSetStorage
vtkGDCMPolyDataReader, 1392
RequestInformationCompat
vtkGDCMImageReader, 1352
vtkGDCMImageReader2, 1368
RequestPaddedCompositePixelCode
gdcm::ImageCodec, 576
RequestPlanarConfiguration
gdcm::ImageCodec, 576
Rescale
gdcm::Rescaler, 938
RescaleFunctionIntoBestFit
gdcm::Rescaler, 938
Rescaler
gdcm::Rescaler, 937
ReserveDataElement
gdcm::FileStreamer, 512
ReserveGroupDataElement
gdcm::FileStreamer, 512
reset
gdcm::terminal, 85
ResetHandledDataSet
gdcm::network::ULConnectionCallback, 1285
RespiratoryWaveformStorage
gdcm::UIDs, 1221
RetrieveSOPInstanceUIDFromIndex
gdcm::DirectoryHelper, 404
RetrieveSOPInstanceUIDFromZPosition
gdcm::DirectoryHelper, 404
reverse
gdcm::terminal, 85
reverse_iterator
gdcm::CodeString, 262
gdcm::LO, 672
gdcm::String< TDelimiter, TMaxLength, TPadChar >, 1095
RFC2557MIMEencapsulation
gdcm::UIDs, 1214
RGB
gdcm::PhotometricInterpretation, 826
RGB2YBR
gdcm::ImageChangePhotometricInterpretation, 553
RGBPixelsToRGBPlanes
gdcm::ImageChangePlanarConfiguration, 557
RGBPlanesToRGBPixels
gdcm::ImageChangePlanarConfiguration, 557
RGBToRecommendedDisplayCIELab
gdcm::SurfaceHelper, 1124
RGBToRecommendedDisplayGrayscale
gdcm::SurfaceHelper, 1124
RLE_COMPRESSION
vtkGDCMImageWriter, 1379
RLECodec
gdcm::RLECodec, 942
RLELossless
gdcm::TransferSyntax, 1183
gdcm::UIDs, 1214
ROI
gdcm::Overlay, 787
RoleSelectionSub
gdcm::network::RoleSelectionSub, 947
Round
gdcm, 76
roundat
gdcm, 76
RTBeamsDeliveryInstructionStorage
gdcm::UIDs, 1223
RTBeamsDeliveryInstructionStorageSupplement74FrozenDraft
gdcm::UIDs, 1219
RTBeamsTreatmentRecordStorage
gdcm::UIDs, 1218
RTBrachyApplicationSetupDeliveryInstructionStorage
gdcm::UIDs, 1223
RTBrachyTreatmentRecordStorage
gdcm::UIDs, 1218
RTConventionalMachineVerification
gdcm::UIDs, 1223
RTConventionalMachineVerificationSupplement74FrozenDraft

gdcm::UIDs, 1219
RTDoseStorage
gdcm::MediaStorage, 697
gdcm::UIDs, 1218
RTImageStorage
gdcm::MediaStorage, 697
gdcm::UIDs, 1218
RTIonBeamsTreatmentRecordStorage
gdcm::MediaStorage, 698
gdcm::UIDs, 1218
RTIonMachineVerification
gdcm::UIDs, 1223
RTIonMachineVerificationSupplement74FrozenDraft
gdcm::UIDs, 1219
RTIonPlanStorage
gdcm::MediaStorage, 698
gdcm::UIDs, 1218
RTPhysicianIntentStorage
gdcm::UIDs, 1222
RTPlanStorage
gdcm::MediaStorage, 697
gdcm::UIDs, 1218
RTSegmentAnnotationStorage
gdcm::UIDs, 1222
RTStructSetProperties
vtkGDCMPolyDataReader, 1394
vtkGDCMPolyDataWriter, 1399
RTStructureSetStorage
gdcm::MediaStorage, 697
gdcm::UIDs, 1218
RTTreatmentSummaryRecordStorage
gdcm::MediaStorage, 698
gdcm::UIDs, 1218
Rule
gdcm::SerieHelper, 1009
RunEventLoop
gdcm::network::ULConnectionManager, 1291
RunMoveEventLoop
gdcm::network::ULConnectionManager, 1292

SAGITTAL
gdcm::Orientation, 782
ScalarType
gdcm::PixelFormat, 830
Scale
vtkGDCMImageReader, 1362
vtkGDCMImageReader2, 1376
Scan
gdcm::Scanner, 957
gdcm::Scanner2, 969
gdcm::StrictScanner, 1080
gdcm::StrictScanner2, 1091
Scanner
gdcm::Scanner, 952
Scanner2
gdcm::Scanner2, 963
Scrub
gdcm::Cleaner, 252
SecondaryCaptureImageStorage
gdcm::MediaStorage, 697
gdcm::UIDs, 1217
Segment
gdcm::Segment, 973
SegmentAlgorithmName
gdcm::Segment, 979
SegmentAlgorithmType
gdcm::Segment, 979
Segmentation
gdcm::MediaStorage, 699
SegmentationStorage
gdcm::MediaStorage, 698
gdcm::UIDs, 1217
SegmentDescription
gdcm::Segment, 979
SegmentedPaletteColorLookupTable
gdcm::SegmentedPaletteColorLookupTable, 981
SegmentedVolumeRenderingVolumetricPresentationStateStorage
gdcm::UIDs, 1221
SegmentLabel
gdcm::Segment, 979
SegmentMap
gdcm::SegmentReader, 984
SegmentNumber
gdcm::Segment, 980
SegmentReader
gdcm::SegmentReader, 985
Segments
gdcm::SegmentReader, 986
gdcm::SegmentWriter, 990
SegmentVector
gdcm::SegmentReader, 985
gdcm::SegmentWriter, 988
SegmentWriter
gdcm::SegmentWriter, 988
Selection
gdcm::Sorter, 1051
SelectionMap
gdcm::Sorter, 1048
Self
gdcm::AnonymizeEvent, 108
gdcm::DataEvent, 336
gdcm::DataSetEvent, 353
gdcm::FileNameEvent, 500
gdcm::MemberCommand< T >, 705
gdcm::ProgressEvent, 896
gdcm::SimpleMemberCommand< T >, 1029
SEMIAUTOMATIC
gdcm::Segment, 973

SendEcho
gdcm::network::ULConnectionManager, 1292
gdcm::ServiceClassUser, 1021
SendFind
gdcm::network::ULConnectionManager, 1292
gdcm::ServiceClassUser, 1021
SendMove
gdcm::network::ULConnectionManager, 1292
gdcm::ServiceClassUser, 1021, 1022
SendNAction
gdcm::network::ULConnectionManager, 1293
SendNCreate
gdcm::network::ULConnectionManager, 1293
SendNDelete
gdcm::network::ULConnectionManager, 1293
SendNEventReport
gdcm::network::ULConnectionManager, 1294
SendNGet
gdcm::network::ULConnectionManager, 1294
SendNSet
gdcm::network::ULConnectionManager, 1294
SendStore
gdcm::network::ULConnectionManager, 1295
gdcm::ServiceClassUser, 1022
Separator
gdcm::ApplicationEntity, 125
gdcm::PersonName, 821
SequenceLengthField
gdcm::SequenceOfItems, 1007
SequenceOfFragments
gdcm::SequenceOfFragments, 993
SequenceOfItems
gdcm::SequenceOfItems, 1002
SerieHelper
gdcm::SerieHelper, 1010
SerieRestrictions
gdcm::SerieHelper, 1009
Series
gdcm::Series, 1014
SeriesInstanceUID
vtkRTStructSetProperties, 1462
ServiceClassApplicationInformation
gdcm::network::ServiceClassApplicationInformation, 1015
ServiceClassUser
gdcm::ServiceClassUser, 1019
Set
gdcm::Attribute< Group, Element, TVR, TVM >, 143
gdcm::Attribute< Group, Element, TVR, VM::VM1 >, 151
gdcm::Attribute< Group, Element, TVR, VM::VM1_n >, 160
gdcm::Element< TVR, TVM >, 414
gdcm::Element< TVR, VM::VM1_n >, 421
SetAbstractSyntax
gdcm::network::PresentationContextRQ, 877
gdcm::PresentationContext, 867
SetAETitle
gdcm::ServiceClassUser, 1022
SetAlgorithmFamily
gdcm::Surface, 1117
SetAlgorithmName
gdcm::Surface, 1117
SetAlgorithmVersion
gdcm::Surface, 1117
SetAnatomicRegion
gdcm::Segment, 976
SetAnatomicRegionModifiers
gdcm::Segment, 976
SetAppendDerivationHistory
gdcm::FileDerivation, 480
SetArray
gdcm::Element< TVR, VM::VM1_n >, 421
setattribute
gdcm::terminal, 86
SetAxisOfRotation
gdcm::Surface, 1117
setbgcolor
gdcm::terminal, 86
SetBitPosition
gdcm::Overlay, 792
SetBitsAllocated
gdcm::Overlay, 792
gdcm::PixelFormat, 836
SetBitSample
gdcm::JPEGCodec, 655
SetBitsStored
gdcm::PixelFormat, 836
SetBlob
gdcm::ApplicationEntity, 124
gdcm::network::PresentationDataValue, 881
gdcm::PersonName, 820
SetBlueLUT
gdcm::LookupTable, 680
SetBufferLength
gdcm::JPEGLSCodec, 663
gdcm::PNMCodec, 859
gdcm::RLECodec, 946
SetByteSwapTag
gdcm::ByteSwapFilter, 224
SetByteValue
gdcm::Attribute< Group, Element, TVR, TVM >, 143
gdcm::Attribute< Group, Element, TVR, VM::VM1 >, 151
gdcm::Attribute< Group, Element, TVR, VM::VM1_n >, 160
gdcm::CSAElement, 294
gdcm::DataElement, 329

SetByteValueNoSwap
gdcm::Attribute< Group, Element, TVR, TVM >, 143
gdcm::Attribute< Group, Element, TVR, VM::VM1 >, 151
SetCallbackFunction
gdcm::MemberCommand< T >, 708
gdcm::SimpleMemberCommand< T >, 1031
SetCalledAETitle
gdcm::network::AAssociateACPDU, 93
gdcm::network::AAssociateRQPDU, 102
gdcm::ServiceClassUser, 1023
SetCallingAETitle
gdcm::network::AAssociateACPDU, 93
gdcm::network::AAssociateRQPDU, 102
SetCenterOfRotation
gdcm::Surface, 1117
SetChangePrivateTags
gdcm::FileExplicitFilter, 483
SetCheckFileMetaInformation
gdcm::Writer, 1471
SetCipherType
gdcm::CAPICryptographicMessageSyntax, 238
gdcm::CryptographicMessageSyntax, 288
gdcm::OpenSSLCryptographicMessageSyntax, 775
gdcm::OpenSSLP7CryptographicMessageSyntax, 780
SetColor
gdcm::Printer, 886
SetColorLevel
vtkImageColorViewer, 1422
SetColorWindow
vtkImageColorViewer, 1422
SetColumns
gdcm::Bitmap, 204
gdcm::Overlay, 792
SetCommand
gdcm::network::PresentationDataValue, 881
SetComponents
gdcm::PersonName, 820
SetCompressIconImage
gdcm::ImageChangeTransferSyntax, 561
SetComputeZSpacing
gdcm::IPPSorter, 623
SetCoordinateStartValue
gdcm::Curve, 317
SetCoordinateStepValue
gdcm::Curve, 317
SetCryptographicMessageSyntax
gdcm::Anonymizer, 119
SetCurve
gdcm::Curve, 318
vtkGDCMImageReader, 1352
vtkGDCMImageReader2, 1368
SetCurveDataDescriptor
gdcm::Curve, 318
SetCurveDescription
gdcm::Curve, 318
SetData
gdcm::DataEvent, 338
SetDataElement
gdcm::Bitmap, 204
SetDataSet
gdcm::File, 465
gdcm::network::PresentationDataValue, 881
SetDataSetTransferSyntax
gdcm::FileMetaInformation, 492
SetDataValueRepresentation
gdcm::Curve, 318
SetDebug
gdcm::Trace, 1178
SetDebugStream
gdcm::Trace, 1178
SetDefaultTransferSyntax
gdcm::PresentationContextGenerator, 873
SetDerivationCodeSequenceCodeValue
gdcm::FileDerivation, 480
SetDerivationDescription
gdcm::FileDerivation, 480
SetDescription
gdcm::CSAHeaderDictEntry, 308
gdcm::ModuleEntry, 728
gdcm::Overlay, 792
SetDescriptor
gdcm::DICOMDIRGenerator, 369
SetDictName
gdcm::DictConverter, 379
SetDicts
gdcm::PythonFilter, 903
gdcm::StringFilter, 1100
SetDimension
gdcm::Bitmap, 205
SetDimensions
gdcm::Bitmap, 205
gdcm::Curve, 318
gdcm::ImageCodec, 572
SetDimensionsValue
gdcm::ImageHelper, 587
SetDirectionCosines
gdcm::Image, 544, 545
vtkGDCMImageWriter, 1380
SetDirectionCosinesFromImageOrientationPatient
vtkGDCMImageWriter, 1380
SetDirectionCosinesTolerance
gdcm::IPPSorter, 623
SetDirectionCosinesValue
gdcm::ImageHelper, 587
SetDirectory
gdcm::network::ULWritingCallback, 1301

gdcm::SerieHelper, 1012
SetDisplayId
vtkImageColorViewer, 1422
SetDomain
gdcm::BoxRegion, 219
SetDropDuplicatePositions
gdcm::IPPSorter, 624
SetElement
gdcm::Tag, 1162
SetElementHandler
gdcm::Parser, 801
SetElementTag
gdcm::Tag, 1162
SetElementXX
gdcm::DictEntry, 383
SetError
gdcm::Trace, 1178
SetErrorStream
gdcm::Trace, 1179
SetEvent
gdcm::network::ULEvent, 1298
setfgcolor
gdcm::terminal, 86
SetFile
gdcm::Anonymizer, 119
gdcm::Cleaner, 252
gdcm::DICOMDIRGenerator, 370
gdcm::FileDecompressLookupTable, 476
gdcm::FileDerivation, 480
gdcm::FileExplicitFilter, 483
gdcm::IconImageFilter, 535
gdcm::Printer, 886
gdcm::PythonFilter, 903
gdcm::Reader, 929
gdcm::SplitMosaicFilter, 1058
gdcm::StreamImageWriter, 1068
gdcm::StringFilter, 1100
gdcm::Validate, 1316
gdcm::Writer, 1471
gdcm::XMLPrinter, 1479
SetFileName
gdcm::FileNameEvent, 501
gdcm::Reader, 929
gdcm::StreamImageReader, 1064
gdcm::StreamImageWriter, 1068
gdcm::Writer, 1472
vtkGDCMThreadedImageReader2, 1410
SetFilename
gdcm::TableReader, 1151
SetFileNames
vtkGDCMImageReader, 1353
vtkGDCMImageWriter, 1381
vtkGDCMThreadedImageReader2, 1410
SetFilenames
gdcm::DICOMDIRGenerator, 370
SetFilePattern
vtkGDCMImageReader, 1353
vtkGDCMImageReader2, 1368
SetFilePrefix
vtkGDCMImageReader, 1353
vtkGDCMImageReader2, 1368
SetFiles
gdcm::FileSet, 508
SetFiniteVolume
gdcm::Surface, 1117
SetForce
gdcm::ImageChangeTransferSyntax, 561
gdcm::ImageFragmentSplitter, 581
SetForcePixelSpacing
gdcm::ImageHelper, 587
SetForceRescaleInterceptSlope
gdcm::ImageHelper, 588
SetFragmentSizeMax
gdcm::ImageFragmentSplitter, 581
SetFrameOrigin
gdcm::Overlay, 793
SetFromDataElement
gdcm::Attribute< Group, Element, TVR, TVM >, 144
gdcm::Attribute< Group, Element, TVR, VM::VM1 >, 151
gdcm::Attribute< Group, Element, TVR, VM::VM1_n >, 160
gdcm::Element< TVR, TVM >, 414
gdcm::Element< TVR, VM::VM1_n >, 421
SetFromDataSet
gdcm::Attribute< Group, Element, TVR, TVM >, 144
gdcm::Attribute< Group, Element, TVR, VM::VM1 >, 152
gdcm::Attribute< Group, Element, TVR, VM::VM1_n >, 160
gdcm::MediaStorage, 702
SetFromFile
gdcm::MediaStorage, 702
SetFromHeader
gdcm::MediaStorage, 702
SetFromModality
gdcm::MediaStorage, 702
SetFromSourceImageSequence
gdcm::MediaStorage, 703
SetFromString
gdcm::DirectionCosines, 397
SetFromUID
gdcm::UIDs, 1234
SetGreenLUT
gdcm::LookupTable, 681
SetGroup
gdcm::Curve, 318
gdcm::Overlay, 793

gdcm::Tag, 1162
SetGroupXX
gdcm::DictEntry, 383
SetHeader
gdcm::File, 465
SetHighBit
gdcm::PixelFormat, 836
SetHostname
gdcm::ServiceClassUser, 1023
SetIconImage
gdcm::Pixmap, 842
SetIE
gdcm::IODEntry, 616
SetImage
gdcm::PixmapWriter, 854
gdcm::SplitMosaicFilter, 1058
SetImplementationClassUID
gdcm::FileMetaInformation, 492
SetImplementationVersionName
gdcm::FileMetaInformation, 493
SetImplicitFlag
gdcm::network::ULConnectionCallback, 1285
SetInput
gdcm::BitmapToBitmapFilter, 213
gdcm::ImageConverter, 578
vtkImageColorViewer, 1423
SetInputConnection
vtkImageColorViewer, 1423
SetInputDirectory
gdcm::EmptyMaskGenerator, 439
SetInputFileName
gdcm::DictConverter, 379
gdcm::FileAnonymizer, 469
gdcm::FileChangeTransferSyntax, 473
SetIntercept
gdcm::Image, 545
gdcm::Rescaler, 939
SetKey
gdcm::CSAElement, 294
SetKeyword
gdcm::DictEntry, 383
SetLastElement
gdcm::ParseException, 797
SetLastFragment
gdcm::network::PresentationDataValue, 881
SetLength
gdcm::ByteValue, 232
gdcm::Element< TVR, VM::VM1_2 >, 417
gdcm::Element< TVR, VM::VM1_n >, 421
gdcm::Element< TVR, VM::VM2_2n >, 424
gdcm::Element< TVR, VM::VM2_n >, 426
gdcm::Element< TVR, VM::VM3_3n >, 428
gdcm::Element< TVR, VM::VM3_4 >, 430
gdcm::Element< TVR, VM::VM3_n >, 431
gdcm::RLECodec, 946
gdcm::SequenceOfFragments, 997
gdcm::SequenceOfItems, 1006
gdcm::Value, 1320
SetLengthOnly
gdcm::ByteValue, 232
gdcm::Value, 1320
SetLengthToUndefined
gdcm::SequenceOfItems, 1006
SetLoadMode
gdcm::SerieHelper, 1013
SetLookupTable
vtkImageMapToColors16, 1433
SetLossless
gdcm::JPEGCodec, 655
gdcm::JPEGLSCodec, 663
SetLossyError
gdcm::JPEGLSCodec, 663
SetLossyFlag
gdcm::Bitmap, 205
gdcm::ImageCodec, 572
gdcm::PVRGCodec, 902
SetLUT
gdcm::Bitmap, 205
gdcm::ImageCodec, 573
gdcm::LookupTable, 681
gdcm::SegmentedPaletteColorLookupTable, 982
SetManifold
gdcm::Surface, 1118
SetMaximumLength
gdcm::network::MaximumLengthSub, 691
SetMaximumPointDistance
gdcm::Surface, 1118
SetMaxPDULength
gdcm::network::ULConnectionInfo, 1288
SetMaxPDUSize
gdcm::network::ULConnection, 1282
SetMCT
gdcm::JPEG2000Codec, 643
SetMeanPointDistance
gdcm::Surface, 1118
SetMedicalImageProperties
vtkGDCMImageReader, 1353
vtkGDCMImageReader2, 1368
vtkGDCMImageWriter, 1381
vtkGDCMPolyDataWriter, 1397
SetMergeModeToAbstractSyntax
gdcm::PresentationContextGenerator, 873
SetMergeModeToTransferSyntax
gdcm::PresentationContextGenerator, 874
SetMeshPrimitive
gdcm::Surface, 1118
SetMessageHeader
gdcm::network::PresentationDataValue, 881

SetMinMaxForPixelType
gdcm::Rescaler, 939
setmode
gdcm::terminal, 86
SetName
gdcm::CSAElement, 294
gdcm::CSAHeaderDictEntry, 308
gdcm::DictEntry, 383
gdcm::IODEntry, 616
gdcm::Macro, 686
gdcm::Module, 724
gdcm::ModuleEntry, 728
gdcm::network::AbstractSyntax, 106
gdcm::network::ApplicationContext, 122
gdcm::network::TransferSyntaxSub, 1188
gdcm::PDBElement, 808
SetNameFromUID
gdcm::network::AbstractSyntax, 106
gdcm::network::TransferSyntaxSub, 1188
SetNeedByteSwap
gdcm::Bitmap, 206
gdcm::ImageCodec, 573
SetNeedOverlayCleanup
gdcm::ImageCodec, 573
SetNestedDataSet
gdcm::Item, 629
SetNoOfItems
gdcm::CSAElement, 294
SetNoSwap
gdcm::Element< TVR, TVM >, 414
gdcm::Element< TVR, VM::VM1_n >, 422
SetNumberOfCurves
gdcm::Pixmap, 842
SetNumberOfDimensions
gdcm::Bitmap, 206
gdcm::ImageCodec, 573
SetNumberOfFilenames
gdcm::FilenameGenerator, 505
SetNumberOfFrames
gdcm::Overlay, 793
SetNumberOfInputPorts
vtkGDCMPolyDataWriter, 1397
SetNumberOfItems
gdcm::SequenceOfItems, 1007
SetNumberOfOverlays
gdcm::Pixmap, 842
SetNumberOfPoints
gdcm::Curve, 319
SetNumberOfResolutions
gdcm::JPEG2000Codec, 643
SetNumberOfSegments
gdcm::SegmentWriter, 989
SetNumberOfSurfacePoints
gdcm::Surface, 1118
SetNumberOfSurfaces
gdcm::SurfaceWriter, 1131
SetNumberOfTableValues
vtkLookupTable16, 1451
SetNumberOfThreadsForDecompression
gdcm::JPEG2000Codec, 643
SetNumberOfValues
gdcm::Attribute< Group, Element, TVR, VM::VM1_n >, 160
SetNumberOfVectors
gdcm::Surface, 1118
SetObliquityThresholdCosineValue
gdcm::Orientation, 784
SetOffScreenRendering
vtkImageColorViewer, 1423
SetOrigin
gdcm::Image, 545, 546
gdcm::Overlay, 793
SetOriginValue
gdcm::ImageHelper, 588
SetOutputDimensions
gdcm::IconImageGenerator, 538
SetOutputDirectory
gdcm::EmptyMaskGenerator, 440
SetOutputFileName
gdcm::DictConverter, 379
gdcm::FileAnonymizer, 469
gdcm::FileChangeTransferSyntax, 473
gdcm::FileStreamer, 512
SetOutputFormatToLuminance
vtkImageMapToColors16, 1434
SetOutputFormatToLuminanceAlpha
vtkImageMapToColors16, 1434
SetOutputFormatToRGB
vtkImageMapToColors16, 1434
SetOutputFormatToRGBA
vtkImageMapToColors16, 1434
SetOutputType
gdcm::DictConverter, 379
SetOutsideValuePixel
gdcm::IconImageGenerator, 538
SetOverlay
gdcm::Overlay, 793
SetOverlayVisibility
vtkImageColorViewer, 1423
SetOwner
gdcm::PrivateTag, 894
SetParentId
vtkImageColorViewer, 1423
SetPassword
gdcm::CAPICryptographicMessageSyntax, 238
gdcm::CryptographicMessageSyntax, 288
gdcm::OpenSSLCryptographicMessageSyntax, 775

gdcm::OpenSSLP7CryptographicMessageSyntax, 780
SetPattern
gdcm::FilenameGenerator, 505
SetPDU
gdcm::network::ULEvent, 1298
SetPermissions
gdcm::System, 1142
SetPhotometricInterpretation
gdcm::Bitmap, 206
gdcm::ImageChangePhotometricInterpretation, 553
gdcm::ImageCodec, 573
SetPixelFormat
gdcm::Bitmap, 206
gdcm::ImageCodec, 574
gdcm::JPEGCodec, 656
gdcm::Rescaler, 939
SetPixelMinMax
gdcm::IconImageGenerator, 538
SetPixelRepresentation
gdcm::PixelFormat, 836
SetPixmap
gdcm::FileDecompressLookupTable, 477
gdcm::IconImageGenerator, 539
gdcm::PixmapWriter, 855
SetPlanarConfiguration
gdcm::Bitmap, 207
gdcm::ImageChangePlanarConfiguration, 557
gdcm::ImageCodec, 574
SetPMSRescaleInterceptSlope
gdcm::ImageHelper, 588
SetPointCoordinatesData
gdcm::Surface, 1119
SetPointPositionAccuracy
gdcm::Surface, 1119
SetPointsBoundingBoxCoordinates
gdcm::Surface, 1119
SetPort
gdcm::ServiceClassUser, 1023
SetPortSCP
gdcm::ServiceClassUser, 1023
SetPosition
vtkImageColorViewer, 1423, 1424
SetPreamble
gdcm::FileMetaInformation, 493
SetPrefix
gdcm::FilenameGenerator, 505
SetPresentationContextID
gdcm::network::PresentationContextAC, 869
gdcm::network::PresentationContextRQ, 877
gdcm::network::PresentationDataValue, 882
gdcm::PresentationContext, 867
SetPresentationContexts
gdcm::network::ULConnection, 1282
gdcm::ServiceClassUser, 1024
SetPrettyPrint
gdcm::JSON, 666
SetPrimitiveData
gdcm::MeshPrimitive, 714
SetPrimitivesData
gdcm::MeshPrimitive, 714
SetPrimitiveType
gdcm::MeshPrimitive, 714
SetPrivateCreator
gdcm::Tag, 1163
SetProcessingAlgorithm
gdcm::Surface, 1119
SetProgress
gdcm::ProgressEvent, 898
SetPropertyCategory
gdcm::Segment, 977
SetPropertyType
gdcm::Segment, 977
SetPropertyTypeModifiers
gdcm::Segment, 977
SetPurposeOfReferenceCodeSequenceCodeValue
gdcm::FileDerivation, 481
SetQuality
gdcm::JPEG2000Codec, 644
gdcm::JPEGCodec, 656
SetRate
gdcm::JPEG2000Codec, 644
SetReason
gdcm::network::AAbortPDU, 89
gdcm::network::PresentationContextAC, 870
SetRecommendedDisplayCIELabValue
gdcm::Surface, 1119, 1120
SetRecommendedDisplayGrayscaleValue
gdcm::Surface, 1120
SetRecommendedPresentationOpacity
gdcm::Surface, 1120
SetRecommendedPresentationType
gdcm::Surface, 1120
SetRecomputeItemLength
gdcm::FileExplicitFilter, 484
SetRecomputeSequenceLength
gdcm::FileExplicitFilter, 484
SetRedLUT
gdcm::LookupTable, 681
SetRef
gdcm::IODEntry, 617
SetRegion
gdcm::ImageRegionReader, 597
SetRenderer
vtkImageColorViewer, 1424
SetRenderWindow
vtkImageColorViewer, 1424
SetRescaleInterceptSlopeValue

gdcm::ImageHelper, 588
SetRetired
gdcm::DictEntry, 384
SetReversible
gdcm::JPEG2000Codec, 644
SetRGB8
gdcm::ImageApplyLookupTable, 549
SetRoot
gdcm::UIDGenerator, 1197
SetRootDirectory
gdcm::DICOMDIRGenerator, 370
SetRows
gdcm::Bitmap, 207
gdcm::Overlay, 794
SetRTStructSetProperties
vtkGDCMPolyDataWriter, 1397
SetSamplesPerPixel
gdcm::PixelFormat, 836
SetScalarType
gdcm::PixelFormat, 837
SetSearchParameter
gdcm::BaseQuery, 182, 183
SetSegmentAlgorithmName
gdcm::Segment, 977
SetSegmentAlgorithmType
gdcm::Segment, 977
SetSegmentDescription
gdcm::Segment, 978
SetSegmentLabel
gdcm::Segment, 978
SetSegmentNumber
gdcm::Segment, 978
SetSegments
gdcm::SegmentWriter, 990
SetSize
vtkImageColorViewer, 1424
SetSlice
vtkImageColorViewer, 1425
SetSliceOrientation
vtkImageColorViewer, 1425
SetSliceOrientationToXY
vtkImageColorViewer, 1425
SetSliceOrientationToXZ
vtkImageColorViewer, 1425
SetSliceOrientationToYZ
vtkImageColorViewer, 1425
SetSlope
gdcm::Image, 546
gdcm::Rescaler, 939
SetSOPClassUIDMode
gdcm::EmptyMaskGenerator, 440
SetSOPInstanceUID
gdcm::BaseQuery, 183
SetSortFunction
gdcm::Sorter, 1049
SetSource
gdcm::network::AAbortPDU, 89
SetSourceApplicationEntityTitle
gdcm::FileMetaInformation, 493
SetSpacing
gdcm::Image, 546
SetSpacingValue
gdcm::ImageHelper, 588
SetState
gdcm::network::ULConnection, 1282
SetStream
gdcm::Reader, 930
gdcm::StreamImageReader, 1064
gdcm::StreamImageWriter, 1068
gdcm::Trace, 1179
gdcm::Writer, 1472
SetStreamToFile
gdcm::Trace, 1179
SetStyle
gdcm::Printer, 886
gdcm::XMLPrinter, 1480
SetSurfaceComments
gdcm::Surface, 1120
SetSurfaceCount
gdcm::Segment, 978
SetSurfaceNumber
gdcm::Surface, 1120
SetSurfaceProcessing
gdcm::Surface, 1121
SetSurfaceProcessingDescription
gdcm::Surface, 1121
SetSurfaceProcessingRatio
gdcm::Surface, 1121
SetSyngoDT
gdcm::CSAElement, 294
SetTag
gdcm::AnonymizeEvent, 110
gdcm::DataElement, 330
SetTagsToRead
gdcm::Sorter, 1050
SetTargetPixelType
gdcm::Rescaler, 940
SetTemplateFileName
gdcm::FileStreamer, 512
SetTileSize
gdcm::JPEG2000Codec, 644
SetTimeout
gdcm::network::ARTIMTimer, 132
gdcm::ServiceClassUser, 1024
SetToUndefined
gdcm::VL, 1328
SetTransferSyntax
gdcm::Bitmap, 207

gdcm::FileChangeTransferSyntax, 473
gdcm::ImageChangeTransferSyntax, 561
gdcm::network::PresentationContextAC, 870
SetTuple
gdcm::network::RoleSelectionSub, 948
gdcm::network::ServiceClassApplicationInformation, 1016
gdcm::network::SOPClassExtendedNegociationSub, 1043
SetType
gdcm::ModuleEntry, 728
gdcm::Overlay, 794
SetTypeOfData
gdcm::Curve, 319
SetupInteractor
vtkImageColorViewer, 1425
SetUsage
gdcm::IODEntry, 617
SetUserCodec
gdcm::ImageChangeTransferSyntax, 562
SetUserData
gdcm::Parser, 802
SetUserInformation
gdcm::network::AAssociateRQPDU, 102
SetUseSeriesDetails
gdcm::SerieHelper, 1013
SetUseTargetPixelType
gdcm::Rescaler, 940
SetUseVRUN
gdcm::FileExplicitFilter, 484
SetValue
gdcm::Attribute< Group, Element, TVR, TVM >, 144
gdcm::Attribute< Group, Element, TVR, VM::VM1 >, 152
gdcm::Attribute< Group, Element, TVR, VM::VM1_n >, 161
gdcm::CSAElement, 295
gdcm::DataElement, 330
gdcm::Element< TVR, TVM >, 415
gdcm::Element< TVR, VM::VM1_n >, 422
gdcm::PDBElement, 808
SetValueFieldLength
gdcm::DataElement, 331
SetValues
gdcm::Attribute< Group, Element, TVR, TVM >, 145
gdcm::Attribute< Group, Element, TVR, VM::VM1_n >, 161
SetVectorAccuracy
gdcm::Surface, 1121
SetVectorCoordinateData
gdcm::Surface, 1121
SetVectorDimensionality
gdcm::Surface, 1121
SetVL
gdcm::DataElement, 331
SetVLToUndefined
gdcm::DataElement, 331
SetVM
gdcm::CSAElement, 295
gdcm::CSAHeaderDictEntry, 309
gdcm::DictEntry, 384
SetVR
gdcm::CSAElement, 295
gdcm::CSAHeaderDictEntry, 309
gdcm::DataElement, 331
gdcm::DictEntry, 384
SetWarning
gdcm::Trace, 1179
SetWarningStream
gdcm::Trace, 1179
SetWindowId
vtkImageColorViewer, 1426
SetWriteDataSetOnly
gdcm::Writer, 1472
SetZSpacingTolerance
gdcm::IPPSorter, 624
SH
gdcm::VR, 1336
SHA1
gdcm::SHA1, 1026
SHComp
gdcm, 60
Shift
vtkGDCMImageReader, 1362
vtkGDCMImageReader2, 1376
ShiftEnd
gdcm::ByteBuffer, 220
ShowAbort
gdcm::SimpleSubjectWatcher, 1033
ShowAnonymization
gdcm::SimpleSubjectWatcher, 1034
ShowData
gdcm::SimpleSubjectWatcher, 1034
ShowDataSet
gdcm::SimpleSubjectWatcher, 1034
ShowFileName
gdcm::SimpleSubjectWatcher, 1034
ShowIteration
gdcm::SimpleSubjectWatcher, 1034
ShowProgress
gdcm::SimpleSubjectWatcher, 1035
SIEMENS
gdcm::Dicts, 389
gdcm::EquipmentManufacturer, 447
SimpleMemberCommand
gdcm::SimpleMemberCommand< T >, 1029
SimpleSubjectWatcher
gdcm::SimpleSubjectWatcher, 1033

SimplifiedAdultEchoSRStorage
gdcm::UIDs, 1222
SINGLEBIT
gdcm::PixelFormat, 831
SingleSerieUIDFileSetHT
gdcm::SerieHelper, 1014
SingleSerieUIDFileSetmap
gdcm::SerieHelper, 1010
Size
gdcm::CodeString, 264
gdcm::DataSet, 350
gdcm::GroupDict, 531
gdcm::network::AAbortPDU, 89
gdcm::network::AAssociateACPDU, 93
gdcm::network::AAssociateRJPDU, 96
gdcm::network::AAssociateRQPDU, 102
gdcm::network::AbstractSyntax, 106
gdcm::network::ApplicationContext, 122
gdcm::network::AReleaseRPPDU, 128
gdcm::network::AReleaseRQPDU, 130
gdcm::network::AsynchronousOperationsWindowSub, 135
gdcm::network::BasePDU, 179
gdcm::network::ImplementationClassUIDSub, 605
gdcm::network::ImplementationVersionNameSub, 607
gdcm::network::MaximumLengthSub, 691
gdcm::network::PDataTFPDU, 805
gdcm::network::PresentationContextAC, 870
gdcm::network::PresentationContextRQ, 878
gdcm::network::PresentationDataValue, 882
gdcm::network::RoleSelectionSub, 948
gdcm::network::ServiceClassApplicationInformation, 1016
gdcm::network::SOPClassExtendedNegociationSub, 1043
gdcm::network::TransferSyntaxSub, 1188
gdcm::network::UserInformation, 1314
size_type
gdcm::CodeString, 262
gdcm::LO, 672
gdcm::String< TDelimiter, TMaxLength, TPadChar >, 1095
SizeType
gdcm::DataSet, 341
gdcm::FilenameGenerator, 503
gdcm::IOD, 613
gdcm::NestedModuleEntries, 751
gdcm::network::AAssociateACPDU, 91
gdcm::network::AAssociateRQPDU, 99
gdcm::network::PDataTFPDU, 804
gdcm::network::PresentationContextRQ, 875
gdcm::PresentationContext, 865
gdcm::PresentationContextGenerator, 872
gdcm::SequenceOfFragments, 993
gdcm::SequenceOfItems, 1002
SL
gdcm::VR, 1337
Slice
vtkImageColorViewer, 1430
SLICE_ORIENTATION_XY
vtkImageColorViewer, 1419
SLICE_ORIENTATION_XZ
vtkImageColorViewer, 1419
SLICE_ORIENTATION_YZ
vtkImageColorViewer, 1419
SliceOrientation
vtkImageColorViewer, 1430
Slices
gdcm::MrProtocol::SliceArray, 1037
SmartPointer
gdcm::Object, 771
gdcm::SmartPointer< ObjectType >, 1039, 1040
SOPClassExtendedNegociationSub
gdcm::network::SOPClassExtendedNegociationSub, 1043
SOPClassUIDMode
gdcm::EmptyMaskGenerator, 439
SOPInstanceUID
vtkRTStructSetProperties, 1462
Sort
gdcm::IPPSorter, 624
gdcm::Sorter, 1050
Sorter
gdcm::Sorter, 1048
SortFunc
gdcm::Sorter, 1051
SortFunction
gdcm::Sorter, 1048
SpacialFiducialsStorage
gdcm::MediaStorage, 697
SpacialRegistrationStorage
gdcm::MediaStorage, 697
Spacing
gdcm::Spacing, 1053
SpacingType
gdcm::Spacing, 1053
SpatialFiducialsStorage
gdcm::UIDs, 1217
SpatialRegistrationStorage
gdcm::UIDs, 1217
SpectaclePrescriptionReportStorage
gdcm::UIDs, 1221
Spectroscopy
gdcm::Spectroscopy, 1054
Split
gdcm::ImageFragmentSplitter, 582
gdcm::SplitMosaicFilter, 1058

SplitExtent
vtkGDCMThreadedImageReader2, 1410
SplitMosaicFilter
gdcm::SplitMosaicFilter, 1056
SPM2AVG152PDFrameofReference
gdcm::UIDs, 1215
SPM2AVG152T1FrameofReference
gdcm::UIDs, 1215
SPM2AVG152T2FrameofReference
gdcm::UIDs, 1215
SPM2AVG305T1FrameofReference
gdcm::UIDs, 1215
SPM2BRAINMASKFrameofReference
gdcm::UIDs, 1215
SPM2CSFFrameofReference
gdcm::UIDs, 1215
SPM2EPIFrameofReference
gdcm::UIDs, 1215
SPM2FILT1FrameofReference
gdcm::UIDs, 1215
SPM2GRAYFrameofReference
gdcm::UIDs, 1215
SPM2PDFrameofReference
gdcm::UIDs, 1215
SPM2PETFrameofReference
gdcm::UIDs, 1215
SPM2SINGLESUBJT1FrameofReference
gdcm::UIDs, 1215
SPM2SPECTFrameofReference
gdcm::UIDs, 1215
SPM2T1FrameofReference
gdcm::UIDs, 1215
SPM2T2FrameofReference
gdcm::UIDs, 1215
SPM2TRANSMFrameofReference
gdcm::UIDs, 1215
SPM2WHITEFrameofReference
gdcm::UIDs, 1215
SpringColorPaletteSOPInstance
gdcm::UIDs, 1220
SQ
gdcm::VR, 1337
Squeeze
gdcm::ApplicationEntity, 125
SS
gdcm::VR, 1337
ST
gdcm::VR, 1337
StableSort
gdcm::Sorter, 1050
StandaloneCurveStorage
gdcm::MediaStorage, 697
StandaloneCurveStorageRetired
gdcm::UIDs, 1217
StandaloneModalityLUTStorage
gdcm::MediaStorage, 697
StandaloneModalityLUTStorageRetired
gdcm::UIDs, 1217
StandaloneOverlayStorage
gdcm::MediaStorage, 697
StandaloneOverlayStorageRetired
gdcm::UIDs, 1217
StandalonePETCurveStorageRetired
gdcm::UIDs, 1218
StandaloneVOILUTStorage
gdcm::MediaStorage, 697
StandaloneVOILUTStorageRetired
gdcm::UIDs, 1217
Start
gdcm::network::ARTIMTimer, 132
StartAssociation
gdcm::ServiceClassUser, 1024
StartDataElement
gdcm::FileStreamer, 512
StartElement
gdcm::TableReader, 1151
gdcm::XMLDictReader, 1476
gdcm::XMLPrivateDictReader, 1483
StartElementHandler
gdcm::Parser, 798
StartEncode
gdcm::ImageCodec, 574
gdcm::JPEG2000Codec, 644
gdcm::JPEGCodec, 656
gdcm::JPEGLSCodec, 664
gdcm::RLECodec, 946
StartFilter
gdcm::SimpleSubjectWatcher, 1035
StartGroupDataElement
gdcm::FileStreamer, 513
STATES
gdcm::Surface, 1109
STATES_END
gdcm::Surface, 1110
STComp
gdcm, 60
StereometricRelationshipStorage
gdcm::UIDs, 1218
Stop
gdcm::network::ARTIMTimer, 132
StopAssociation
gdcm::ServiceClassUser, 1024
StopDataElement
gdcm::FileStreamer, 513
StopEncode
gdcm::ImageCodec, 574
gdcm::JPEG2000Codec, 644
gdcm::JPEGCodec, 656

gdcm::JPEGLSCodec, 664
gdcm::RLECodec, 946
StopGroupDataElement
gdcm::FileStreamer, 513
StopProtocol
gdcm::network::ULConnection, 1282
StorageCommitmentPullModelSOPClassRetired
gdcm::UIDs, 1215
StorageCommitmentPullModelSOPInstanceRetired
gdcm::UIDs, 1215
StorageCommitmentPushModelSOPClass
gdcm::UIDs, 1215
StorageCommitmentPushModelSOPInstance
gdcm::UIDs, 1215
StorageServiceClass
gdcm::UIDs, 1216
StoredPrintStorageSOPClassRetired
gdcm::UIDs, 1216
StrCaseCmp
gdcm::System, 1142
Stream
gdcm::Writer, 1473
StreamImageReader
gdcm::Reader, 930
gdcm::StreamImageReader, 1061
StreamImageWriter
gdcm::StreamImageWriter, 1066
gdcm::Writer, 1473
StrictScanner
gdcm::StrictScanner, 1075
StrictScanner2
gdcm::StrictScanner2, 1085
String
gdcm::String< TDelimiter, TMaxLength, TPadChar >, 1096
StringFilter
gdcm::StringFilter, 1099
StrNCaseCmp
gdcm::System, 1143
StrSep
gdcm::System, 1143
StrTokR
gdcm::System, 1143
StructureSetDate
vtkRTStructSetProperties, 1463
StructureSetLabel
vtkRTStructSetProperties, 1463
StructureSetName
vtkRTStructSetProperties, 1463
StructureSetTime
vtkRTStructSetProperties, 1463
Study
gdcm::Study, 1102
StudyComponentManagementSOPClass
gdcm::MediaStorage, 697
StudyComponentManagementSOPClassRetired
gdcm::UIDs, 1215
StudyInstanceUID
vtkRTStructSetProperties, 1463
StudyRootQueryRetrieveInformationModelFIND
gdcm::UIDs, 1218
StudyRootQueryRetrieveInformationModelGET
gdcm::UIDs, 1218
StudyRootQueryRetrieveInformationModelMOVE
gdcm::UIDs, 1218
Subject
gdcm::Subject, 1104
SubjectiveRefractionMeasurementsStorage
gdcm::UIDs, 1221
SubstanceAdministrationLoggingSOPClass
gdcm::UIDs, 1215
SubstanceAdministrationLoggingSOPInstance
gdcm::UIDs, 1215
SubstanceApprovalQuerySOPClass
gdcm::UIDs, 1219
SummerColorPaletteSOPInstance
gdcm::UIDs, 1220
Superclass
gdcm::AnonymizeEvent, 108
gdcm::DataEvent, 336
gdcm::DataSetEvent, 353
gdcm::FileNameEvent, 500
gdcm::LO, 672
gdcm::ProgressEvent, 896
SURFACE
gdcm::Surface, 1110
Surface
gdcm::Surface, 1110
SurfaceCount
gdcm::Segment, 980
SurfaceReader
gdcm::SurfaceReader, 1127
Surfaces
gdcm::Segment, 980
SurfaceScanMeshStorage
gdcm::UIDs, 1221
SurfaceScanPointCloudStorage
gdcm::UIDs, 1221
SurfaceSegmentationStorage
gdcm::MediaStorage, 698
gdcm::UIDs, 1220
SurfaceVector
gdcm::Segment, 972
SurfaceWriter
gdcm::SurfaceWriter, 1130
SV
gdcm::VR, 1337
SV10

gdcm::CSAHeader, 299
Swap
gdcm::ByteSwap< T >, 222
gdcm::SwapperDoOp, 1134
gdcm::SwapperNoOp, 1135
SwapArray
gdcm::SwapperDoOp, 1135
gdcm::SwapperNoOp, 1136
SwapCode
gdcm::SwapCode, 1133
SwapCodeType
gdcm::SwapCode, 1133
SwapFromSwapCodeIntoSystem
gdcm::ByteSwap< T >, 222
SwapRange
gdcm::ByteSwap< T >, 222
SwapRangeFromSwapCodeIntoSystem
gdcm::ByteSwap< T >, 222
SyngoDTField
gdcm::CSAElement, 296
SyntaxError
gdcm::Parser, 800
SystemIsBigEndian
gdcm::ByteSwap< T >, 222
SystemIsLittleEndian
gdcm::ByteSwap< T >, 223

T1
gdcm::Type, 1192
T1C
gdcm::Type, 1192
T2
gdcm::Type, 1192
T2C
gdcm::Type, 1192
T3
gdcm::Type, 1192
Table
gdcm::Table, 1145
Table16
vtkLookupTable16, 1451
TableEntry
gdcm::TableEntry, 1147
TableInternal
gdcm::Table, 1146
TableReader
gdcm::TableReader, 1149
TableRow
gdcm::network::TableRow, 1153
Tag
gdcm::Tag, 1155, 1156
tag
gdcm::Tag, 1164
TagField
gdcm::DataElement, 332
TagMismatchError
gdcm::Parser, 800
TagPath
gdcm::TagPath, 1165
tags
gdcm::Tag, 1164
TagsToRead
gdcm::Sorter, 1051
TagToValue
gdcm::Scanner, 952
gdcm::StrictScanner, 1075
TagToValueValueType
gdcm::Scanner, 952
gdcm::StrictScanner, 1075
TalairachBrainAtlasFrameofReference
gdcm::UIDs, 1215
TConstMemberFunctionPointer
gdcm::MemberCommand< T >, 706
TestAbortOff
gdcm::SimpleSubjectWatcher, 1035
TestAbortOn
gdcm::SimpleSubjectWatcher, 1035
Testing
gdcm::Testing, 1169
TestPBKDF2
gdcm::ASN1, 134
TestsList.txt, 1485
TextSRStorageTrialRetired
gdcm::UIDs, 1218
ThreadedExecute
vtkImageRGBToYBR, 1445
vtkImageYBRToRGB, 1447
ThreadedRequestData
vtkGDCMThreadedImageReader2, 1410
vtkImageMapToColors16, 1434
vtkImageMapToWindowLevelColors2, 1439
TM
gdcm::VR, 1337
TMComp
gdcm, 60
TMemberFunctionPointer
gdcm::MemberCommand< T >, 706
gdcm::SimpleMemberCommand< T >, 1029
ToPyObject
gdcm::PythonFilter, 903
TOSHIBA
gdcm::EquipmentManufacturer, 447
ToshibaPrivateDataStorage
gdcm::MediaStorage, 698
ToString
gdcm::StringFilter, 1100, 1101
ToStringPair
gdcm::StringFilter, 1101, 1102

ToUnixSlashes
gdcm::Filename, 497
ToWindowsSlashes
gdcm::Filename, 497
Trace
gdcm::Trace, 1176
TractographyResultsStorage
gdcm::UIDs, 1221
TransferSyntax
gdcm::TransferSyntax, 1183
TransferSyntaxArrayType
gdcm::PresentationContext, 865
TransferSyntaxes
gdcm::PresentationContext, 867
TransferSyntaxStringsType
gdcm::UIDs, 1213
TransferSyntaxSub
gdcm::network::TransferSyntaxSub, 1187
Transition
gdcm::network::Transition, 1190
transitions
gdcm::network::TableRow, 1153
TRIANGLE
gdcm::MeshPrimitive, 712
TRIANGLE_FAN
gdcm::MeshPrimitive, 712
TRIANGLE_STRIP
gdcm::MeshPrimitive, 712
Trim
gdcm::String< TDelimiter, TMaxLength, TPadChar >, 1097
TrimInternal
gdcm::CodeString, 264
Truncate
gdcm::String< TDelimiter, TMaxLength, TPadChar >, 1097
TryJPEG2000Codec
gdcm::Bitmap, 207
gdcm::ImageChangeTransferSyntax, 562
TryJPEG2000Codec2
gdcm::Bitmap, 208
TryJPEGCodec
gdcm::Bitmap, 208
gdcm::ImageChangeTransferSyntax, 562
TryJPEGCodec2
gdcm::Bitmap, 208
TryJPEGLSCodec
gdcm::Bitmap, 208
gdcm::ImageChangeTransferSyntax, 562
TryKAKADUCodec
gdcm::Bitmap, 208
TryPVRGCodec
gdcm::Bitmap, 208
TryRAWCodec
gdcm::Bitmap, 209
gdcm::ImageChangeTransferSyntax, 563
TryRLECodec
gdcm::Bitmap, 209
gdcm::ImageChangeTransferSyntax, 563
TS
gdcm::Bitmap, 211
TS_END
gdcm::TransferSyntax, 1183
TSName
gdcm::UIDs, 1214
TSType
gdcm::TransferSyntax, 1183
gdcm::UIDs, 1223
Type
gdcm::Element< TVR, TVM >, 412
gdcm::Element< TVR, VM::VM1_n >, 418
gdcm::EquipmentManufacturer, 447
gdcm::Type, 1193
gdcm::VL, 1326
TYPETOENCODING
gdcm, 76
gdcmVR.h, 1734
TYPETOLENGTH
gdcmVM.h, 1730
TypeToString
gdcm::EquipmentManufacturer, 448
TypeType
gdcm::Type, 1192

UberonOntology
gdcm::UIDs, 1220
UC
gdcm::VR, 1337
UCComp
gdcm, 60
UI
gdcm::VR, 1337
UIComp
gdcm, 60
uid_1_2_840_10008_15_0_3_1
gdcm::UIDs, 1228
uid_1_2_840_10008_15_0_3_10
gdcm::UIDs, 1229
uid_1_2_840_10008_15_0_3_11
gdcm::UIDs, 1229
uid_1_2_840_10008_15_0_3_12
gdcm::UIDs, 1229
uid_1_2_840_10008_15_0_3_13
gdcm::UIDs, 1229
uid_1_2_840_10008_15_0_3_14
gdcm::UIDs, 1229
uid_1_2_840_10008_15_0_3_15
gdcm::UIDs, 1229

uid_1_2_840_10008_15_0_3_16
gdcm::UIDs, 1229
uid_1_2_840_10008_15_0_3_17
gdcm::UIDs, 1229
uid_1_2_840_10008_15_0_3_18
gdcm::UIDs, 1229
uid_1_2_840_10008_15_0_3_19
gdcm::UIDs, 1229
uid_1_2_840_10008_15_0_3_2
gdcm::UIDs, 1228
uid_1_2_840_10008_15_0_3_20
gdcm::UIDs, 1229
uid_1_2_840_10008_15_0_3_21
gdcm::UIDs, 1229
uid_1_2_840_10008_15_0_3_22
gdcm::UIDs, 1229
uid_1_2_840_10008_15_0_3_23
gdcm::UIDs, 1229
uid_1_2_840_10008_15_0_3_24
gdcm::UIDs, 1229
uid_1_2_840_10008_15_0_3_25
gdcm::UIDs, 1229
uid_1_2_840_10008_15_0_3_26
gdcm::UIDs, 1229
uid_1_2_840_10008_15_0_3_27
gdcm::UIDs, 1229
uid_1_2_840_10008_15_0_3_28
gdcm::UIDs, 1229
uid_1_2_840_10008_15_0_3_29
gdcm::UIDs, 1229
uid_1_2_840_10008_15_0_3_3
gdcm::UIDs, 1229
uid_1_2_840_10008_15_0_3_30
gdcm::UIDs, 1229
uid_1_2_840_10008_15_0_3_31
gdcm::UIDs, 1229
uid_1_2_840_10008_15_0_3_4
gdcm::UIDs, 1229
uid_1_2_840_10008_15_0_3_5
gdcm::UIDs, 1229
uid_1_2_840_10008_15_0_3_6
gdcm::UIDs, 1229
uid_1_2_840_10008_15_0_3_7
gdcm::UIDs, 1229
uid_1_2_840_10008_15_0_3_8
gdcm::UIDs, 1229
uid_1_2_840_10008_15_0_3_9
gdcm::UIDs, 1229
uid_1_2_840_10008_15_0_4_1
gdcm::UIDs, 1229
uid_1_2_840_10008_15_0_4_2
gdcm::UIDs, 1229
uid_1_2_840_10008_15_0_4_3
gdcm::UIDs, 1229
uid_1_2_840_10008_15_0_4_4
gdcm::UIDs, 1229
uid_1_2_840_10008_15_0_4_5
gdcm::UIDs, 1229
uid_1_2_840_10008_15_0_4_6
gdcm::UIDs, 1229
uid_1_2_840_10008_15_0_4_7
gdcm::UIDs, 1229
uid_1_2_840_10008_15_0_4_8
gdcm::UIDs, 1229
uid_1_2_840_10008_15_1_1
gdcm::UIDs, 1232
uid_1_2_840_10008_1_1
gdcm::UIDs, 1223
uid_1_2_840_10008_1_2
gdcm::UIDs, 1223
uid_1_2_840_10008_1_20
gdcm::UIDs, 1230
uid_1_2_840_10008_1_20_1
gdcm::UIDs, 1225
uid_1_2_840_10008_1_20_1_1
gdcm::UIDs, 1225
uid_1_2_840_10008_1_20_2
gdcm::UIDs, 1225
uid_1_2_840_10008_1_20_2_1
gdcm::UIDs, 1225
uid_1_2_840_10008_1_2_1
gdcm::UIDs, 1223
uid_1_2_840_10008_1_2_1_99
gdcm::UIDs, 1223
uid_1_2_840_10008_1_2_2
gdcm::UIDs, 1223
uid_1_2_840_10008_1_2_4_100
gdcm::UIDs, 1224
uid_1_2_840_10008_1_2_4_101
gdcm::UIDs, 1229
uid_1_2_840_10008_1_2_4_102
gdcm::UIDs, 1229
uid_1_2_840_10008_1_2_4_103
gdcm::UIDs, 1230
uid_1_2_840_10008_1_2_4_104
gdcm::UIDs, 1230
uid_1_2_840_10008_1_2_4_105
gdcm::UIDs, 1230
uid_1_2_840_10008_1_2_4_106
gdcm::UIDs, 1230
uid_1_2_840_10008_1_2_4_107
gdcm::UIDs, 1230
uid_1_2_840_10008_1_2_4_108
gdcm::UIDs, 1230
uid_1_2_840_10008_1_2_4_50
gdcm::UIDs, 1223
uid_1_2_840_10008_1_2_4_51
gdcm::UIDs, 1223

uid_1_2_840_10008_1_2_4_52
gdcm::UIDs, 1224
uid_1_2_840_10008_1_2_4_53
gdcm::UIDs, 1224
uid_1_2_840_10008_1_2_4_54
gdcm::UIDs, 1224
uid_1_2_840_10008_1_2_4_55
gdcm::UIDs, 1224
uid_1_2_840_10008_1_2_4_56
gdcm::UIDs, 1224
uid_1_2_840_10008_1_2_4_57
gdcm::UIDs, 1224
uid_1_2_840_10008_1_2_4_58
gdcm::UIDs, 1224
uid_1_2_840_10008_1_2_4_59
gdcm::UIDs, 1224
uid_1_2_840_10008_1_2_4_60
gdcm::UIDs, 1224
uid_1_2_840_10008_1_2_4_61
gdcm::UIDs, 1224
uid_1_2_840_10008_1_2_4_62
gdcm::UIDs, 1224
uid_1_2_840_10008_1_2_4_63
gdcm::UIDs, 1224
uid_1_2_840_10008_1_2_4_64
gdcm::UIDs, 1224
uid_1_2_840_10008_1_2_4_65
gdcm::UIDs, 1224
uid_1_2_840_10008_1_2_4_66
gdcm::UIDs, 1224
uid_1_2_840_10008_1_2_4_70
gdcm::UIDs, 1224
uid_1_2_840_10008_1_2_4_80
gdcm::UIDs, 1224
uid_1_2_840_10008_1_2_4_81
gdcm::UIDs, 1224
uid_1_2_840_10008_1_2_4_90
gdcm::UIDs, 1224
uid_1_2_840_10008_1_2_4_91
gdcm::UIDs, 1224
uid_1_2_840_10008_1_2_4_92
gdcm::UIDs, 1224
uid_1_2_840_10008_1_2_4_93
gdcm::UIDs, 1224
uid_1_2_840_10008_1_2_4_94
gdcm::UIDs, 1224
uid_1_2_840_10008_1_2_4_95
gdcm::UIDs, 1224
uid_1_2_840_10008_1_2_5
gdcm::UIDs, 1224
uid_1_2_840_10008_1_2_6_1
gdcm::UIDs, 1224
uid_1_2_840_10008_1_2_6_2
gdcm::UIDs, 1224
uid_1_2_840_10008_1_3_10
gdcm::UIDs, 1224
uid_1_2_840_10008_1_40
gdcm::UIDs, 1225
uid_1_2_840_10008_1_40_1
gdcm::UIDs, 1225
uid_1_2_840_10008_1_42
gdcm::UIDs, 1225
uid_1_2_840_10008_1_42_1
gdcm::UIDs, 1225
uid_1_2_840_10008_1_4_1_1
gdcm::UIDs, 1224
uid_1_2_840_10008_1_4_1_10
gdcm::UIDs, 1224
uid_1_2_840_10008_1_4_1_11
gdcm::UIDs, 1224
uid_1_2_840_10008_1_4_1_12
gdcm::UIDs, 1224
uid_1_2_840_10008_1_4_1_13
gdcm::UIDs, 1224
uid_1_2_840_10008_1_4_1_14
gdcm::UIDs, 1224
uid_1_2_840_10008_1_4_1_15
gdcm::UIDs, 1224
uid_1_2_840_10008_1_4_1_16
gdcm::UIDs, 1224
uid_1_2_840_10008_1_4_1_17
gdcm::UIDs, 1224
uid_1_2_840_10008_1_4_1_18
gdcm::UIDs, 1225
uid_1_2_840_10008_1_4_1_2
gdcm::UIDs, 1224
uid_1_2_840_10008_1_4_1_3
gdcm::UIDs, 1224
uid_1_2_840_10008_1_4_1_4
gdcm::UIDs, 1224
uid_1_2_840_10008_1_4_1_5
gdcm::UIDs, 1224
uid_1_2_840_10008_1_4_1_6
gdcm::UIDs, 1224
uid_1_2_840_10008_1_4_1_7
gdcm::UIDs, 1224
uid_1_2_840_10008_1_4_1_8
gdcm::UIDs, 1224
uid_1_2_840_10008_1_4_1_9
gdcm::UIDs, 1224
uid_1_2_840_10008_1_4_2_1
gdcm::UIDs, 1225
uid_1_2_840_10008_1_4_2_2
gdcm::UIDs, 1225
uid_1_2_840_10008_1_5_1
gdcm::UIDs, 1230
uid_1_2_840_10008_1_5_2
gdcm::UIDs, 1230

uid_1_2_840_10008_1_5_3
gdcm::UIDs, 1230
uid_1_2_840_10008_1_5_4
gdcm::UIDs, 1230
uid_1_2_840_10008_1_5_5
gdcm::UIDs, 1230
uid_1_2_840_10008_1_5_6
gdcm::UIDs, 1230
uid_1_2_840_10008_1_5_7
gdcm::UIDs, 1230
uid_1_2_840_10008_1_5_8
gdcm::UIDs, 1230
uid_1_2_840_10008_1_9
gdcm::UIDs, 1225
uid_1_2_840_10008_2_16_10
gdcm::UIDs, 1230
uid_1_2_840_10008_2_16_11
gdcm::UIDs, 1230
uid_1_2_840_10008_2_16_12
gdcm::UIDs, 1230
uid_1_2_840_10008_2_16_13
gdcm::UIDs, 1230
uid_1_2_840_10008_2_16_14
gdcm::UIDs, 1230
uid_1_2_840_10008_2_16_4
gdcm::UIDs, 1225
uid_1_2_840_10008_2_16_5
gdcm::UIDs, 1230
uid_1_2_840_10008_2_16_6
gdcm::UIDs, 1230
uid_1_2_840_10008_2_16_7
gdcm::UIDs, 1230
uid_1_2_840_10008_2_16_8
gdcm::UIDs, 1230
uid_1_2_840_10008_2_16_9
gdcm::UIDs, 1230
uid_1_2_840_10008_2_6_1
gdcm::UIDs, 1225
uid_1_2_840_10008_3_1_1_1
gdcm::UIDs, 1225
uid_1_2_840_10008_3_1_2_1_1
gdcm::UIDs, 1225
uid_1_2_840_10008_3_1_2_1_4
gdcm::UIDs, 1225
uid_1_2_840_10008_3_1_2_2_1
gdcm::UIDs, 1225
uid_1_2_840_10008_3_1_2_3_1
gdcm::UIDs, 1225
uid_1_2_840_10008_3_1_2_3_2
gdcm::UIDs, 1225
uid_1_2_840_10008_3_1_2_3_3
gdcm::UIDs, 1225
uid_1_2_840_10008_3_1_2_3_4
gdcm::UIDs, 1225
uid_1_2_840_10008_3_1_2_3_5
gdcm::UIDs, 1225
uid_1_2_840_10008_3_1_2_5_1
gdcm::UIDs, 1225
uid_1_2_840_10008_3_1_2_5_4
gdcm::UIDs, 1225
uid_1_2_840_10008_3_1_2_5_5
gdcm::UIDs, 1225
uid_1_2_840_10008_3_1_2_6_1
gdcm::UIDs, 1225
uid_1_2_840_10008_4_2
gdcm::UIDs, 1225
uid_1_2_840_10008_5_1_1_1
gdcm::UIDs, 1225
uid_1_2_840_10008_5_1_1_14
gdcm::UIDs, 1225
uid_1_2_840_10008_5_1_1_15
gdcm::UIDs, 1225
uid_1_2_840_10008_5_1_1_16
gdcm::UIDs, 1225
uid_1_2_840_10008_5_1_1_16_376
gdcm::UIDs, 1225
uid_1_2_840_10008_5_1_1_17
gdcm::UIDs, 1225
uid_1_2_840_10008_5_1_1_17_376
gdcm::UIDs, 1225
uid_1_2_840_10008_5_1_1_18
gdcm::UIDs, 1225
uid_1_2_840_10008_5_1_1_18_1
gdcm::UIDs, 1225
uid_1_2_840_10008_5_1_1_2
gdcm::UIDs, 1225
uid_1_2_840_10008_5_1_1_22
gdcm::UIDs, 1225
uid_1_2_840_10008_5_1_1_23
gdcm::UIDs, 1225
uid_1_2_840_10008_5_1_1_24
gdcm::UIDs, 1225
uid_1_2_840_10008_5_1_1_24_1
gdcm::UIDs, 1226
uid_1_2_840_10008_5_1_1_25
gdcm::UIDs, 1226
uid_1_2_840_10008_5_1_1_26
gdcm::UIDs, 1226
uid_1_2_840_10008_5_1_1_27
gdcm::UIDs, 1226
uid_1_2_840_10008_5_1_1_29
gdcm::UIDs, 1226
uid_1_2_840_10008_5_1_1_30
gdcm::UIDs, 1226
uid_1_2_840_10008_5_1_1_31
gdcm::UIDs, 1226
uid_1_2_840_10008_5_1_1_32
gdcm::UIDs, 1226

uid_1_2_840_10008_5_1_1_33
gdcm::UIDs, 1226
uid_1_2_840_10008_5_1_1_4
gdcm::UIDs, 1225
uid_1_2_840_10008_5_1_1_40
gdcm::UIDs, 1230
uid_1_2_840_10008_5_1_1_40_1
gdcm::UIDs, 1230
uid_1_2_840_10008_5_1_1_4_1
gdcm::UIDs, 1225
uid_1_2_840_10008_5_1_1_4_2
gdcm::UIDs, 1225
uid_1_2_840_10008_5_1_1_9
gdcm::UIDs, 1225
uid_1_2_840_10008_5_1_1_9_1
gdcm::UIDs, 1225
uid_1_2_840_10008_5_1_4_1_1_1
gdcm::UIDs, 1226
uid_1_2_840_10008_5_1_4_1_1_10
gdcm::UIDs, 1226
uid_1_2_840_10008_5_1_4_1_1_104_1
gdcm::UIDs, 1227
uid_1_2_840_10008_5_1_4_1_1_104_2
gdcm::UIDs, 1228
uid_1_2_840_10008_5_1_4_1_1_104_3
gdcm::UIDs, 1231
uid_1_2_840_10008_5_1_4_1_1_11
gdcm::UIDs, 1226
uid_1_2_840_10008_5_1_4_1_1_11_1
gdcm::UIDs, 1226
uid_1_2_840_10008_5_1_4_1_1_11_10
gdcm::UIDs, 1230
uid_1_2_840_10008_5_1_4_1_1_11_11
gdcm::UIDs, 1230
uid_1_2_840_10008_5_1_4_1_1_11_2
gdcm::UIDs, 1226
uid_1_2_840_10008_5_1_4_1_1_11_3
gdcm::UIDs, 1226
uid_1_2_840_10008_5_1_4_1_1_11_4
gdcm::UIDs, 1226
uid_1_2_840_10008_5_1_4_1_1_11_5
gdcm::UIDs, 1230
uid_1_2_840_10008_5_1_4_1_1_11_6
gdcm::UIDs, 1230
uid_1_2_840_10008_5_1_4_1_1_11_7
gdcm::UIDs, 1230
uid_1_2_840_10008_5_1_4_1_1_11_8
gdcm::UIDs, 1230
uid_1_2_840_10008_5_1_4_1_1_11_9
gdcm::UIDs, 1230
uid_1_2_840_10008_5_1_4_1_1_128
gdcm::UIDs, 1228
uid_1_2_840_10008_5_1_4_1_1_128_1
gdcm::UIDs, 1229
uid_1_2_840_10008_5_1_4_1_1_129
gdcm::UIDs, 1228
uid_1_2_840_10008_5_1_4_1_1_12_1
gdcm::UIDs, 1227
uid_1_2_840_10008_5_1_4_1_1_12_1_1
gdcm::UIDs, 1227
uid_1_2_840_10008_5_1_4_1_1_12_2
gdcm::UIDs, 1227
uid_1_2_840_10008_5_1_4_1_1_12_2_1
gdcm::UIDs, 1227
uid_1_2_840_10008_5_1_4_1_1_12_3
gdcm::UIDs, 1227
uid_1_2_840_10008_5_1_4_1_1_12_77
gdcm::UIDs, 1230
uid_1_2_840_10008_5_1_4_1_1_130
gdcm::UIDs, 1231
uid_1_2_840_10008_5_1_4_1_1_131
gdcm::UIDs, 1231
uid_1_2_840_10008_5_1_4_1_1_13_1_1
gdcm::UIDs, 1227
uid_1_2_840_10008_5_1_4_1_1_13_1_2
gdcm::UIDs, 1227
uid_1_2_840_10008_5_1_4_1_1_13_1_3
gdcm::UIDs, 1229
uid_1_2_840_10008_5_1_4_1_1_13_1_4
gdcm::UIDs, 1230
uid_1_2_840_10008_5_1_4_1_1_13_1_5
gdcm::UIDs, 1230
uid_1_2_840_10008_5_1_4_1_1_14_1
gdcm::UIDs, 1230
uid_1_2_840_10008_5_1_4_1_1_14_2
gdcm::UIDs, 1230
uid_1_2_840_10008_5_1_4_1_1_1_1
gdcm::UIDs, 1226
uid_1_2_840_10008_5_1_4_1_1_1_1_1
gdcm::UIDs, 1226
uid_1_2_840_10008_5_1_4_1_1_1_2
gdcm::UIDs, 1226
uid_1_2_840_10008_5_1_4_1_1_1_2_1
gdcm::UIDs, 1226
uid_1_2_840_10008_5_1_4_1_1_1_3
gdcm::UIDs, 1226
uid_1_2_840_10008_5_1_4_1_1_1_3_1
gdcm::UIDs, 1226
uid_1_2_840_10008_5_1_4_1_1_2
gdcm::UIDs, 1226
uid_1_2_840_10008_5_1_4_1_1_20
gdcm::UIDs, 1227
uid_1_2_840_10008_5_1_4_1_1_200_1
gdcm::UIDs, 1231
uid_1_2_840_10008_5_1_4_1_1_200_2
gdcm::UIDs, 1231
uid_1_2_840_10008_5_1_4_1_1_200_3
gdcm::UIDs, 1231

uid_1_2_840_10008_5_1_4_1_1_200_4
gdcm::UIDs, 1231
uid_1_2_840_10008_5_1_4_1_1_200_5
gdcm::UIDs, 1231
uid_1_2_840_10008_5_1_4_1_1_200_6
gdcm::UIDs, 1231
uid_1_2_840_10008_5_1_4_1_1_2_1
gdcm::UIDs, 1226
uid_1_2_840_10008_5_1_4_1_1_2_2
gdcm::UIDs, 1229
uid_1_2_840_10008_5_1_4_1_1_3
gdcm::UIDs, 1226
uid_1_2_840_10008_5_1_4_1_1_30
gdcm::UIDs, 1230
uid_1_2_840_10008_5_1_4_1_1_3_1
gdcm::UIDs, 1226
uid_1_2_840_10008_5_1_4_1_1_4
gdcm::UIDs, 1226
uid_1_2_840_10008_5_1_4_1_1_40
gdcm::UIDs, 1230
uid_1_2_840_10008_5_1_4_1_1_481_1
gdcm::UIDs, 1228
uid_1_2_840_10008_5_1_4_1_1_481_10
gdcm::UIDs, 1231
uid_1_2_840_10008_5_1_4_1_1_481_11
gdcm::UIDs, 1231
uid_1_2_840_10008_5_1_4_1_1_481_2
gdcm::UIDs, 1228
uid_1_2_840_10008_5_1_4_1_1_481_3
gdcm::UIDs, 1228
uid_1_2_840_10008_5_1_4_1_1_481_4
gdcm::UIDs, 1228
uid_1_2_840_10008_5_1_4_1_1_481_5
gdcm::UIDs, 1228
uid_1_2_840_10008_5_1_4_1_1_481_6
gdcm::UIDs, 1228
uid_1_2_840_10008_5_1_4_1_1_481_7
gdcm::UIDs, 1228
uid_1_2_840_10008_5_1_4_1_1_481_8
gdcm::UIDs, 1228
uid_1_2_840_10008_5_1_4_1_1_481_9
gdcm::UIDs, 1228
uid_1_2_840_10008_5_1_4_1_1_4_1
gdcm::UIDs, 1226
uid_1_2_840_10008_5_1_4_1_1_4_2
gdcm::UIDs, 1226
uid_1_2_840_10008_5_1_4_1_1_4_3
gdcm::UIDs, 1232
uid_1_2_840_10008_5_1_4_1_1_4_4
gdcm::UIDs, 1229
uid_1_2_840_10008_5_1_4_1_1_5
gdcm::UIDs, 1226
uid_1_2_840_10008_5_1_4_1_1_501_1
gdcm::UIDs, 1231
uid_1_2_840_10008_5_1_4_1_1_501_2_1
gdcm::UIDs, 1231
uid_1_2_840_10008_5_1_4_1_1_501_2_2
gdcm::UIDs, 1231
uid_1_2_840_10008_5_1_4_1_1_501_3
gdcm::UIDs, 1231
uid_1_2_840_10008_5_1_4_1_1_501_4
gdcm::UIDs, 1232
uid_1_2_840_10008_5_1_4_1_1_501_5
gdcm::UIDs, 1232
uid_1_2_840_10008_5_1_4_1_1_501_6
gdcm::UIDs, 1232
uid_1_2_840_10008_5_1_4_1_1_6
gdcm::UIDs, 1226
uid_1_2_840_10008_5_1_4_1_1_601_1
gdcm::UIDs, 1232
uid_1_2_840_10008_5_1_4_1_1_601_2
gdcm::UIDs, 1232
uid_1_2_840_10008_5_1_4_1_1_66
gdcm::UIDs, 1227
uid_1_2_840_10008_5_1_4_1_1_66_1
gdcm::UIDs, 1227
uid_1_2_840_10008_5_1_4_1_1_66_2
gdcm::UIDs, 1227
uid_1_2_840_10008_5_1_4_1_1_66_3
gdcm::UIDs, 1227
uid_1_2_840_10008_5_1_4_1_1_66_4
gdcm::UIDs, 1227
uid_1_2_840_10008_5_1_4_1_1_66_5
gdcm::UIDs, 1229
uid_1_2_840_10008_5_1_4_1_1_66_6
gdcm::UIDs, 1230
uid_1_2_840_10008_5_1_4_1_1_67
gdcm::UIDs, 1227
uid_1_2_840_10008_5_1_4_1_1_68_1
gdcm::UIDs, 1230
uid_1_2_840_10008_5_1_4_1_1_68_2
gdcm::UIDs, 1231
uid_1_2_840_10008_5_1_4_1_1_6_1
gdcm::UIDs, 1226
uid_1_2_840_10008_5_1_4_1_1_6_2
gdcm::UIDs, 1229
uid_1_2_840_10008_5_1_4_1_1_7
gdcm::UIDs, 1226
uid_1_2_840_10008_5_1_4_1_1_77_1
gdcm::UIDs, 1227
uid_1_2_840_10008_5_1_4_1_1_77_1_1
gdcm::UIDs, 1227
uid_1_2_840_10008_5_1_4_1_1_77_1_1_1
gdcm::UIDs, 1227
uid_1_2_840_10008_5_1_4_1_1_77_1_2
gdcm::UIDs, 1227
uid_1_2_840_10008_5_1_4_1_1_77_1_2_1
gdcm::UIDs, 1227

uid_1_2_840_10008_5_1_4_1_1_77_1_3
gdcm::UIDs, 1227
uid_1_2_840_10008_5_1_4_1_1_77_1_4
gdcm::UIDs, 1227
uid_1_2_840_10008_5_1_4_1_1_77_1_4_1
gdcm::UIDs, 1227
uid_1_2_840_10008_5_1_4_1_1_77_1_5_1
gdcm::UIDs, 1227
uid_1_2_840_10008_5_1_4_1_1_77_1_5_2
gdcm::UIDs, 1227
uid_1_2_840_10008_5_1_4_1_1_77_1_5_3
gdcm::UIDs, 1227
uid_1_2_840_10008_5_1_4_1_1_77_1_5_4
gdcm::UIDs, 1227
uid_1_2_840_10008_5_1_4_1_1_77_1_5_5
gdcm::UIDs, 1231
uid_1_2_840_10008_5_1_4_1_1_77_1_5_6
gdcm::UIDs, 1231
uid_1_2_840_10008_5_1_4_1_1_77_1_5_7
gdcm::UIDs, 1231
uid_1_2_840_10008_5_1_4_1_1_77_1_5_8
gdcm::UIDs, 1231
uid_1_2_840_10008_5_1_4_1_1_77_1_6
gdcm::UIDs, 1229
uid_1_2_840_10008_5_1_4_1_1_77_2
gdcm::UIDs, 1227
uid_1_2_840_10008_5_1_4_1_1_78_1
gdcm::UIDs, 1231
uid_1_2_840_10008_5_1_4_1_1_78_2
gdcm::UIDs, 1231
uid_1_2_840_10008_5_1_4_1_1_78_3
gdcm::UIDs, 1231
uid_1_2_840_10008_5_1_4_1_1_78_4
gdcm::UIDs, 1231
uid_1_2_840_10008_5_1_4_1_1_78_5
gdcm::UIDs, 1231
uid_1_2_840_10008_5_1_4_1_1_78_6
gdcm::UIDs, 1231
uid_1_2_840_10008_5_1_4_1_1_78_7
gdcm::UIDs, 1231
uid_1_2_840_10008_5_1_4_1_1_78_8
gdcm::UIDs, 1231
uid_1_2_840_10008_5_1_4_1_1_79_1
gdcm::UIDs, 1231
uid_1_2_840_10008_5_1_4_1_1_7_1
gdcm::UIDs, 1226
uid_1_2_840_10008_5_1_4_1_1_7_2
gdcm::UIDs, 1226
uid_1_2_840_10008_5_1_4_1_1_7_3
gdcm::UIDs, 1226
uid_1_2_840_10008_5_1_4_1_1_7_4
gdcm::UIDs, 1226
uid_1_2_840_10008_5_1_4_1_1_8
gdcm::UIDs, 1226
uid_1_2_840_10008_5_1_4_1_1_80_1
gdcm::UIDs, 1231
uid_1_2_840_10008_5_1_4_1_1_81_1
gdcm::UIDs, 1231
uid_1_2_840_10008_5_1_4_1_1_82_1
gdcm::UIDs, 1231
uid_1_2_840_10008_5_1_4_1_1_88_1
gdcm::UIDs, 1227
uid_1_2_840_10008_5_1_4_1_1_88_11
gdcm::UIDs, 1227
uid_1_2_840_10008_5_1_4_1_1_88_2
gdcm::UIDs, 1227
uid_1_2_840_10008_5_1_4_1_1_88_22
gdcm::UIDs, 1227
uid_1_2_840_10008_5_1_4_1_1_88_3
gdcm::UIDs, 1227
uid_1_2_840_10008_5_1_4_1_1_88_33
gdcm::UIDs, 1227
uid_1_2_840_10008_5_1_4_1_1_88_34
gdcm::UIDs, 1231
uid_1_2_840_10008_5_1_4_1_1_88_35
gdcm::UIDs, 1231
uid_1_2_840_10008_5_1_4_1_1_88_4
gdcm::UIDs, 1227
uid_1_2_840_10008_5_1_4_1_1_88_40
gdcm::UIDs, 1227
uid_1_2_840_10008_5_1_4_1_1_88_50
gdcm::UIDs, 1227
uid_1_2_840_10008_5_1_4_1_1_88_59
gdcm::UIDs, 1227
uid_1_2_840_10008_5_1_4_1_1_88_65
gdcm::UIDs, 1227
uid_1_2_840_10008_5_1_4_1_1_88_67
gdcm::UIDs, 1227
uid_1_2_840_10008_5_1_4_1_1_88_68
gdcm::UIDs, 1231
uid_1_2_840_10008_5_1_4_1_1_88_69
gdcm::UIDs, 1231
uid_1_2_840_10008_5_1_4_1_1_88_70
gdcm::UIDs, 1231
uid_1_2_840_10008_5_1_4_1_1_88_71
gdcm::UIDs, 1231
uid_1_2_840_10008_5_1_4_1_1_88_72
gdcm::UIDs, 1231
uid_1_2_840_10008_5_1_4_1_1_88_73
gdcm::UIDs, 1231
uid_1_2_840_10008_5_1_4_1_1_88_74
gdcm::UIDs, 1231
uid_1_2_840_10008_5_1_4_1_1_88_75
gdcm::UIDs, 1231
uid_1_2_840_10008_5_1_4_1_1_9
gdcm::UIDs, 1226
uid_1_2_840_10008_5_1_4_1_1_90_1
gdcm::UIDs, 1231

uid_1_2_840_10008_5_1_4_1_1_9_1
gdcm::UIDs, 1226
uid_1_2_840_10008_5_1_4_1_1_9_1_1
gdcm::UIDs, 1226
uid_1_2_840_10008_5_1_4_1_1_9_1_2
gdcm::UIDs, 1226
uid_1_2_840_10008_5_1_4_1_1_9_1_3
gdcm::UIDs, 1226
uid_1_2_840_10008_5_1_4_1_1_9_2_1
gdcm::UIDs, 1226
uid_1_2_840_10008_5_1_4_1_1_9_3_1
gdcm::UIDs, 1226
uid_1_2_840_10008_5_1_4_1_1_9_4_1
gdcm::UIDs, 1226
uid_1_2_840_10008_5_1_4_1_1_9_4_2
gdcm::UIDs, 1230
uid_1_2_840_10008_5_1_4_1_1_9_5_1
gdcm::UIDs, 1230
uid_1_2_840_10008_5_1_4_1_1_9_6_1
gdcm::UIDs, 1230
uid_1_2_840_10008_5_1_4_1_2_1_1
gdcm::UIDs, 1228
uid_1_2_840_10008_5_1_4_1_2_1_2
gdcm::UIDs, 1228
uid_1_2_840_10008_5_1_4_1_2_1_3
gdcm::UIDs, 1228
uid_1_2_840_10008_5_1_4_1_2_2_1
gdcm::UIDs, 1228
uid_1_2_840_10008_5_1_4_1_2_2_2
gdcm::UIDs, 1228
uid_1_2_840_10008_5_1_4_1_2_2_3
gdcm::UIDs, 1228
uid_1_2_840_10008_5_1_4_1_2_3_1
gdcm::UIDs, 1228
uid_1_2_840_10008_5_1_4_1_2_3_2
gdcm::UIDs, 1228
uid_1_2_840_10008_5_1_4_1_2_3_3
gdcm::UIDs, 1228
uid_1_2_840_10008_5_1_4_1_2_4_2
gdcm::UIDs, 1232
uid_1_2_840_10008_5_1_4_1_2_4_3
gdcm::UIDs, 1232
uid_1_2_840_10008_5_1_4_1_2_5_3
gdcm::UIDs, 1232
uid_1_2_840_10008_5_1_4_20_1
gdcm::UIDs, 1232
uid_1_2_840_10008_5_1_4_20_2
gdcm::UIDs, 1232
uid_1_2_840_10008_5_1_4_20_3
gdcm::UIDs, 1232
uid_1_2_840_10008_5_1_4_31
gdcm::UIDs, 1228
uid_1_2_840_10008_5_1_4_32
gdcm::UIDs, 1228
uid_1_2_840_10008_5_1_4_32_1
gdcm::UIDs, 1228
uid_1_2_840_10008_5_1_4_32_2
gdcm::UIDs, 1228
uid_1_2_840_10008_5_1_4_32_3
gdcm::UIDs, 1228
uid_1_2_840_10008_5_1_4_33
gdcm::UIDs, 1228
uid_1_2_840_10008_5_1_4_34_1
gdcm::UIDs, 1228
uid_1_2_840_10008_5_1_4_34_10
gdcm::UIDs, 1232
uid_1_2_840_10008_5_1_4_34_2
gdcm::UIDs, 1228
uid_1_2_840_10008_5_1_4_34_3
gdcm::UIDs, 1228
uid_1_2_840_10008_5_1_4_34_4
gdcm::UIDs, 1228
uid_1_2_840_10008_5_1_4_34_4_1
gdcm::UIDs, 1228
uid_1_2_840_10008_5_1_4_34_4_2
gdcm::UIDs, 1228
uid_1_2_840_10008_5_1_4_34_4_3
gdcm::UIDs, 1228
uid_1_2_840_10008_5_1_4_34_4_4
gdcm::UIDs, 1228
uid_1_2_840_10008_5_1_4_34_5
gdcm::UIDs, 1228
uid_1_2_840_10008_5_1_4_34_5_1
gdcm::UIDs, 1232
uid_1_2_840_10008_5_1_4_34_6
gdcm::UIDs, 1232
uid_1_2_840_10008_5_1_4_34_6_1
gdcm::UIDs, 1232
uid_1_2_840_10008_5_1_4_34_6_2
gdcm::UIDs, 1232
uid_1_2_840_10008_5_1_4_34_6_3
gdcm::UIDs, 1232
uid_1_2_840_10008_5_1_4_34_6_4
gdcm::UIDs, 1232
uid_1_2_840_10008_5_1_4_34_7
gdcm::UIDs, 1232
uid_1_2_840_10008_5_1_4_34_8
gdcm::UIDs, 1232
uid_1_2_840_10008_5_1_4_34_9
gdcm::UIDs, 1232
uid_1_2_840_10008_5_1_4_37_1
gdcm::UIDs, 1228
uid_1_2_840_10008_5_1_4_37_2
gdcm::UIDs, 1228
uid_1_2_840_10008_5_1_4_37_3
gdcm::UIDs, 1228
uid_1_2_840_10008_5_1_4_38_1
gdcm::UIDs, 1228

uid_1_2_840_10008_5_1_4_38_2
gdcm::UIDs, 1228
uid_1_2_840_10008_5_1_4_38_3
gdcm::UIDs, 1228
uid_1_2_840_10008_5_1_4_38_4
gdcm::UIDs, 1232
uid_1_2_840_10008_5_1_4_39_1
gdcm::UIDs, 1232
uid_1_2_840_10008_5_1_4_39_2
gdcm::UIDs, 1232
uid_1_2_840_10008_5_1_4_39_3
gdcm::UIDs, 1232
uid_1_2_840_10008_5_1_4_39_4
gdcm::UIDs, 1232
uid_1_2_840_10008_5_1_4_41
gdcm::UIDs, 1228
uid_1_2_840_10008_5_1_4_42
gdcm::UIDs, 1228
uid_1_2_840_10008_5_1_4_43_1
gdcm::UIDs, 1232
uid_1_2_840_10008_5_1_4_43_2
gdcm::UIDs, 1232
uid_1_2_840_10008_5_1_4_43_3
gdcm::UIDs, 1232
uid_1_2_840_10008_5_1_4_43_4
gdcm::UIDs, 1232
uid_1_2_840_10008_5_1_4_44_1
gdcm::UIDs, 1232
uid_1_2_840_10008_5_1_4_44_2
gdcm::UIDs, 1232
uid_1_2_840_10008_5_1_4_44_3
gdcm::UIDs, 1232
uid_1_2_840_10008_5_1_4_44_4
gdcm::UIDs, 1232
uid_1_2_840_10008_5_1_4_45_1
gdcm::UIDs, 1232
uid_1_2_840_10008_5_1_4_45_2
gdcm::UIDs, 1232
uid_1_2_840_10008_5_1_4_45_3
gdcm::UIDs, 1232
uid_1_2_840_10008_5_1_4_45_4
gdcm::UIDs, 1232
uid_1_2_840_10008_7_1_1
gdcm::UIDs, 1232
uid_1_2_840_10008_7_1_2
gdcm::UIDs, 1232
uid_1_2_840_10008_8_1_1
gdcm::UIDs, 1232
UIDGenerator
gdcm::UIDGenerator, 1195
UINT12
gdcm::PixelFormat, 831
UINT16
gdcm::PixelFormat, 831
UINT32
gdcm::PixelFormat, 831
UINT64
gdcm::PixelFormat, 831
UINT8
gdcm::PixelFormat, 831
UL
gdcm::VR, 1337
ULAction
gdcm::network::ULAction, 1236, 1237
ULActionAE6
gdcm::network::ULConnection, 1283
ULBasicCallback
gdcm::network::ULBasicCallback, 1276
ULConnection
gdcm::network::ULConnection, 1279
ULConnectionCallback
gdcm::network::ULConnectionCallback, 1284
ULConnectionInfo
gdcm::network::ULConnectionInfo, 1286
ULConnectionManager
gdcm::network::ULConnection, 1283
gdcm::network::ULConnectionManager, 1290
ULEvent
gdcm::network::ULEvent, 1296, 1297
ULTransitionTable
gdcm::network::ULTransitionTable, 1299
UltrasoundImageStorage
gdcm::MediaStorage, 696
gdcm::UIDs, 1217
UltrasoundImageStorageRetired
gdcm::MediaStorage, 696
gdcm::UIDs, 1217
UltrasoundMultiFrameImageStorage
gdcm::MediaStorage, 696
UltrasoundMultiframeImageStorage
gdcm::UIDs, 1216
UltrasoundMultiFrameImageStorageRetired
gdcm::MediaStorage, 696
UltrasoundMultiframeImageStorageRetired
gdcm::UIDs, 1216
ULWritingCallback
gdcm::network::ULWritingCallback, 1301
UN
gdcm::VR, 1337
UndefinedEntityError
gdcm::Parser, 800
underline
gdcm::terminal, 85
UnexpectedStateError
gdcm::Parser, 800
UnifiedProcedureStepEventSOPClass
gdcm::UIDs, 1219
UnifiedProcedureStepEventSOPClass1

gdcm::UIDs, 1223
UnifiedProcedureStepPullSOPClass
gdcm::UIDs, 1219
UnifiedProcedureStepPullSOPClass1
gdcm::UIDs, 1222
UnifiedProcedureStepPushSOPClass
gdcm::UIDs, 1219
UnifiedProcedureStepPushSOPClass1
gdcm::UIDs, 1222
UnifiedProcedureStepWatchSOPClass
gdcm::UIDs, 1219
UnifiedProcedureStepWatchSOPClass1
gdcm::UIDs, 1222
UnifiedWorklistandProcedureStepServiceClass
gdcm::UIDs, 1219
UnifiedWorklistandProcedureStepServiceClass1
gdcm::UIDs, 1222
UnifiedWorklistandProcedureStepSOPInstance
gdcm::UIDs, 1219
UnInstallPipeline
vtkImageColorViewer, 1426
UniversalCoordinatedTime
gdcm::UIDs, 1223
UNKNOWN
gdcm::CSAHeader, 299
gdcm::EquipmentManufacturer, 447
gdcm::LookupTable, 676
gdcm::Orientation, 782
gdcm::PhotometricInterpretation, 826
gdcm::PixelFormat, 831
gdcm::Spacing, 1053
gdcm::Surface, 1110
gdcm::Type, 1192
Unknown
gdcm::SwapCode, 1133
gdcm::TransferSyntax, 1182
Unpack
gdcm::Unpacker12Bits, 1308
UnRegister
gdcm::Object, 770
UnusedBitsPresentInPixelData
gdcm::Bitmap, 209
gdcm::Pixmap, 843
Update
gdcm::Curve, 319
gdcm::Overlay, 794
UpdateDisplayExtent
vtkImageColorViewer, 1426
UpdateOrientation
vtkImageColorViewer, 1426
UpdatePosition
gdcm::ByteBuffer, 221
UPSFilteredGlobalSubscriptionSOPInstance
gdcm::UIDs, 1222
UR
gdcm::VR, 1337
URComp
gdcm, 61
URI
gdcm::MediaStorage, 699
US
gdcm::VR, 1337
US_OW
gdcm::VR, 1337
US_SS
gdcm::VR, 1337
US_SS_OW
gdcm::VR, 1337
Usage
gdcm::Usage, 1310
UsageType
gdcm::Usage, 1309
UseDictAlways
gdcm::PythonFilter, 904
gdcm::StringFilter, 1102
UseGrayscaleSecondaryImageStorage
gdcm::EmptyMaskGenerator, 439
UseOriginalSOPClassUID
gdcm::EmptyMaskGenerator, 439
UserInformation
gdcm::network::UserInformation, 1312
UserOption
gdcm::Usage, 1309
UserOrdering
gdcm::SerieHelper, 1013
UT
gdcm::VR, 1337
UTComp
gdcm, 61
UV
gdcm::VR, 1337

V
gdcm::Validate, 1317
Valid
gdcm::Preamble, 863
Validate
gdcm::PixelFormat, 837
gdcm::Validate, 1316
ValidateQuery
gdcm::BaseQuery, 183
gdcm::BaseRootQuery, 188
gdcm::FindPatientRootQuery, 518
gdcm::FindStudyRootQuery, 521
gdcm::ModalityPerformedProcedureStepCreateQuery, 717
gdcm::ModalityPerformedProcedureStepSetQuery, 720

gdcm::MovePatientRootQuery, 734
gdcm::MoveStudyRootQuery, 737
gdcm::WLMFindQuery, 1466
Validation
gdcm::Validate, 1317
ValidDataSet
gdcm::BaseQuery, 183
Value
gdcm::Value, 1319
value
gdcm::SerieHelper, 1014
gdcm::STATIC_ASSERTION_FAILURE< true >, 1060
value_type
gdcm::CodeString, 262
gdcm::LO, 672
gdcm::String< TDelimiter, TMaxLength, TPadChar >, 1095
ValueField
gdcm::DataElement, 333
gdcm::PDBElement, 809
ValueLengthField
gdcm::DataElement, 333
ValueMultiplicityField
gdcm::CSAElement, 296
ValuePtr
gdcm::DataElement, 323
ValuesType
gdcm::Scanner, 952
gdcm::Scanner2, 962
gdcm::StrictScanner, 1075
gdcm::StrictScanner2, 1085
VERBOSE_STYLE
gdcm::Printer, 884
VerificationSOPClass
gdcm::UIDs, 1214
Verify
gdcm::Defs, 362
gdcm::Macro, 686
gdcm::Module, 724
Version
gdcm::Version, 1323
VERTEX
gdcm::MeshPrimitive, 712
Video
gdcm::MediaStorage, 699
VideoEndoscopicImageStorage
gdcm::MediaStorage, 698
gdcm::UIDs, 1217
VideoMicroscopicImageStorage
gdcm::MediaStorage, 698
gdcm::UIDs, 1217
VideoPhotographicImageStorage
gdcm::MediaStorage, 698
gdcm::UIDs, 1218
VIEWType
gdcm::Surface, 1110
VIEWType_END
gdcm::Surface, 1110
VisualAcuityMeasurementsStorage
gdcm::UIDs, 1221
VL
gdcm::VL, 1326
VL16
gdcm::VR, 1337
VL32
gdcm::VR, 1337
VLEndoscopicImageStorage
gdcm::MediaStorage, 698
gdcm::UIDs, 1217
VLImageStorageTrialRetired
gdcm::UIDs, 1217
VLMicroscopicImageStorage
gdcm::MediaStorage, 698
gdcm::UIDs, 1217
VLMultiframeImageStorageTrialRetired
gdcm::UIDs, 1217
VLPhotographicImageStorage
gdcm::MediaStorage, 698
gdcm::UIDs, 1218
VLSlideCoordinatesMicroscopicImageStorage
gdcm::UIDs, 1218
VLWholeSlideMicroscopyImageStorage
gdcm::MediaStorage, 698
gdcm::UIDs, 1220
VM
gdcm::VM, 1332
VM0
gdcm::VM, 1331
VM1
gdcm::VM, 1331
VM10
gdcm::VM, 1331
VM12
gdcm::VM, 1331
VM16
gdcm::VM, 1331
VM18
gdcm::VM, 1331
VM1_2
gdcm::VM, 1331
VM1_3
gdcm::VM, 1331
VM1_32
gdcm::VM, 1331
VM1_4
gdcm::VM, 1331
VM1_5

gdcm::VM, 1331
VM1_8
gdcm::VM, 1331
VM1_99
gdcm::VM, 1331
VM1_n
gdcm::VM, 1331
VM2
gdcm::VM, 1331
VM24
gdcm::VM, 1331
VM256
gdcm::VM, 1331
VM28
gdcm::VM, 1331
VM2_2n
gdcm::VM, 1331
VM2_n
gdcm::VM, 1331
VM3
gdcm::VM, 1331
VM30_30n
gdcm::VM, 1331
VM32
gdcm::VM, 1331
VM35
gdcm::VM, 1331
VM3_3n
gdcm::VM, 1331
VM3_4
gdcm::VM, 1331
VM3_n
gdcm::VM, 1331
VM4
gdcm::VM, 1331
VM47_47n
gdcm::VM, 1331
VM4_4n
gdcm::VM, 1331
VM5
gdcm::VM, 1331
VM6
gdcm::VM, 1331
VM6_6n
gdcm::VM, 1331
VM6_n
gdcm::VM, 1331
VM7_7n
gdcm::VM, 1331
VM8
gdcm::VM, 1331
VM9
gdcm::VM, 1331
VM99
gdcm::VM, 1331
VM_END
gdcm::VM, 1331
VMType
gdcm::Attribute< Group, Element, TVR, TVM >, 139
gdcm::Attribute< Group, Element, TVR, VM::VM1 >, 148
gdcm::VM, 1331
VOILUTBoxSOPClass
gdcm::UIDs, 1216
VolumeRenderingVolumetricPresentationStateStorage
gdcm::UIDs, 1221
VR
gdcm::VR, 1337
VR_END
gdcm::VR, 1337
VR_VM1
gdcm::VR, 1337
VRALL
gdcm::VR, 1337
VRASCII
gdcm::VR, 1337
VRBINARY
gdcm, 77
gdcm::VR, 1337
VRField
gdcm::CSAElement, 296
gdcm::DataElement, 333
VRType
gdcm::VR, 1336
VRTypeTemplateCase
gdcmVR.h, 1734
VT100
gdcm::terminal, 85
VTK_CMYK
vtkGDCMImageReader.h, 2113
vtkGDCMImageReader2.h, 2118
VTK_INVERSE_LUMINANCE
vtkGDCMImageReader.h, 2113
vtkGDCMImageReader2.h, 2118
VTK_LEGACY
vtkImageColorViewer, 1426, 1427
VTK_LOOKUP_TABLE
vtkGDCMImageReader.h, 2113
vtkGDCMImageReader2.h, 2118
VTK_YBR
vtkGDCMImageReader.h, 2113
vtkGDCMImageReader2.h, 2119
vtkBooleanMacro
vtkGDCMImageReader, 1353, 1354
vtkGDCMImageReader2, 1368, 1369
vtkGDCMImageWriter, 1381
vtkGDCMThreadedImageReader, 1406
vtkGDCMThreadedImageReader2, 1411

vtkImageColorViewer, 1427
vtkImageMapToColors16, 1434
vtkGDCMImageReader, 1347
~vtkGDCMImageReader, 1350
ApplyInverseVideo, 1358
ApplyLookupTable, 1359
ApplyPlanarConfiguration, 1359
ApplyShiftScale, 1359
ApplyYBRToRGB, 1359
CanReadFile, 1350
Curve, 1359
DirectionCosines, 1359
ExecuteData, 1350
ExecuteInformation, 1351
FileNames, 1359
FillMedicalImageInformation, 1351
ForceRescale, 1360
GetDescriptiveName, 1351
GetFileExtensions, 1351
GetIconImage, 1351
GetOverlay, 1351
IconDataScalarType, 1360
IconImageDataExtent, 1360
IconNumberOfScalarComponents, 1360
ImageFormat, 1360
ImageOrientationPatient, 1360
ImagePositionPatient, 1360
LoadIconImage, 1361
LoadOverlays, 1361
LoadSingleFile, 1351
LossyFlag, 1361
MedicalImageProperties, 1361
New, 1352
NumberOfIconImages, 1361
NumberOfOverlays, 1361
PlanarConfiguration, 1361
PrintSelf, 1352
RequestDataCompat, 1352
RequestInformationCompat, 1352
Scale, 1362
SetCurve, 1352
SetFileNames, 1353
SetFilePattern, 1353
SetFilePrefix, 1353
SetMedicalImageProperties, 1353
Shift, 1362
vtkBooleanMacro, 1353, 1354
vtkGDCMImageReader, 1350
vtkGDCMMedicalImageProperties, 1388
vtkGetMacro, 1354–1356
vtkGetObjectMacro, 1356
vtkGetStringMacro, 1357
vtkGetVector3Macro, 1357
vtkGetVector6Macro, 1357
vtkSetMacro, 1357, 1358
vtkSetVector6Macro, 1358
vtkTypeMacro, 1358
vtkGDCMImageReader.h, 2111, 2113
VTK_CMYK, 2113
VTK_INVERSE_LUMINANCE, 2113
VTK_LOOKUP_TABLE, 2113
VTK_YBR, 2113
vtkGDCMImageReader2, 1362
~vtkGDCMImageReader2, 1365
ApplyInverseVideo, 1373
ApplyLookupTable, 1373
ApplyPlanarConfiguration, 1373
ApplyShiftScale, 1374
ApplyYBRToRGB, 1374
CanReadFile, 1365
Curve, 1374
DirectionCosines, 1374
FillMedicalImageInformation, 1365
ForceRescale, 1374
GetDescriptiveName, 1365
GetFileExtensions, 1365
GetIconImage, 1366
GetIconImagePort, 1366
GetOverlay, 1366
GetOverlayPort, 1366
IconDataScalarType, 1374
IconImageDataExtent, 1374
IconNumberOfScalarComponents, 1375
ImageFormat, 1375
ImageOrientationPatient, 1375
ImagePositionPatient, 1375
LoadIconImage, 1375
LoadOverlays, 1375
LoadSingleFile, 1366
LossyFlag, 1375
New, 1366
NumberOfIconImages, 1376
NumberOfOverlays, 1376
PlanarConfiguration, 1376
PrintSelf, 1367
ProcessRequest, 1367
RequestData, 1367
RequestDataCompat, 1367
RequestInformation, 1367
RequestInformationCompat, 1368
Scale, 1376
SetCurve, 1368
SetFilePattern, 1368
SetFilePrefix, 1368
SetMedicalImageProperties, 1368
Shift, 1376
vtkBooleanMacro, 1368, 1369
vtkGDCMImageReader2, 1365

vtkGDCMMedicalImageProperties, 1389
vtkGetMacro, 1369–1371
vtkGetObjectMacro, 1371
vtkGetStringMacro, 1371, 1372
vtkGetVector3Macro, 1372
vtkGetVector6Macro, 1372
vtkSetMacro, 1372, 1373
vtkSetVector6Macro, 1373
vtkTypeMacro, 1373
vtkGDCMImageReader2.h, 2117, 2119
VTK_CMYK, 2118
VTK_INVERSE_LUMINANCE, 2118
VTK_LOOKUP_TABLE, 2118
VTK_YBR, 2119
vtkGDCMImageWriter, 1377
∼vtkGDCMImageWriter, 1379
CompressionTypes, 1379
GetDescriptiveName, 1379
GetFileExtensions, 1380
GetFileName, 1380
JPEG2000_COMPRESSION, 1379
JPEG_COMPRESSION, 1379
JPEGLS_COMPRESSION, 1379
New, 1380
NO_COMPRESSION, 1379
PrintSelf, 1380
RLE_COMPRESSION, 1379
SetDirectionCosines, 1380
SetDirectionCosinesFromImageOrientationPatient, 1380
SetFileNames, 1381
SetMedicalImageProperties, 1381
vtkBooleanMacro, 1381
vtkGDCMImageWriter, 1379
vtkGDCMMedicalImageProperties, 1389
vtkGetMacro, 1381, 1382
vtkGetObjectMacro, 1383
vtkGetStringMacro, 1383
vtkSetMacro, 1383, 1384
vtkSetStringMacro, 1385
vtkTypeMacro, 1385
Write, 1385
WriteGDCMData, 1385
WriteSlice, 1386
vtkGDCMImageWriter.h, 2122, 2123
vtkGDCMMedicalImageProperties, 1386
∼vtkGDCMMedicalImageProperties, 1387
Clear, 1387
GetFile, 1388
New, 1388
PrintSelf, 1388
PushBackFile, 1388
vtkGDCMImageReader, 1388
vtkGDCMImageReader2, 1389
vtkGDCMImageWriter, 1389
vtkGDCMMedicalImageProperties, 1387
vtkTypeMacro, 1388
vtkGDCMMedicalImageProperties.h, 2126
vtkGDCMPolyDataReader, 1389
∼vtkGDCMPolyDataReader, 1391
FileName, 1394
FillMedicalImageInformation, 1391
MedicalImageProperties, 1394
New, 1391
PrintSelf, 1391
RequestData, 1392
RequestData_HemodynamicWaveformStorage, 1392
RequestData_RTStructureSetStorage, 1392
RequestInformation, 1392
RequestInformation_HemodynamicWaveformStorage, 1392
RequestInformation_RTStructureSetStorage, 1392
RTStructSetProperties, 1394
vtkGDCMPolyDataReader, 1391
vtkGetObjectMacro, 1393
vtkGetStringMacro, 1393
vtkSetStringMacro, 1393
vtkTypeMacro, 1393
vtkGDCMPolyDataReader.h, 2131, 2132
vtkGDCMPolyDataWriter, 1394
∼vtkGDCMPolyDataWriter, 1396
InitializeRTStructSet, 1396
MedicalImageProperties, 1398
New, 1396
PrintSelf, 1397
RTStructSetProperties, 1399
SetMedicalImageProperties, 1397
SetNumberOfInputPorts, 1397
SetRTStructSetProperties, 1397
vtkGDCMPolyDataWriter, 1396
vtkTypeMacro, 1398
WriteData, 1398
WriteRTSTRUCTData, 1398
WriteRTSTRUCTInfo, 1398
vtkGDCMPolyDataWriter.h, 2133
vtkGDCMTesting, 1399
∼vtkGDCMTesting, 1401
GetGDCMDataRoot, 1401
GetMD5MetaImage, 1401
GetMHDMD5FromFile, 1401
GetNumberOfMD5MetaImages, 1401
GetRAWMD5FromFile, 1402
GetVTKDataRoot, 1402
MD5MetaImagesType, 1400
New, 1402
PrintSelf, 1402
vtkGDCMTesting, 1400

vtkTypeMacro, 1402
vtkGDCMTesting.h, 2135
vtkGDCMThreadedImageReader, 1403
~vtkGDCMThreadedImageReader, 1405
ExecuteData, 1405
ExecuteInformation, 1405
New, 1405
PrintSelf, 1405
ReadFiles, 1405
RequestDataCompat, 1406
vtkBooleanMacro, 1406
vtkGDCMThreadedImageReader, 1404
vtkGetMacro, 1406
vtkSetMacro, 1406
vtkTypeMacro, 1407
vtkGDCMThreadedImageReader.h, 2136
vtkGDCMThreadedImageReader2, 1407
~vtkGDCMThreadedImageReader2, 1409
GetFileName, 1409
New, 1409
PrintSelf, 1409
RequestInformation, 1410
SetFileName, 1410
SetFileNames, 1410
SplitExtent, 1410
ThreadedRequestData, 1410
vtkBooleanMacro, 1411
vtkGDCMThreadedImageReader2, 1409
vtkGetMacro, 1411, 1412
vtkGetObjectMacro, 1413
vtkGetVector3Macro, 1413
vtkGetVector6Macro, 1413
vtkSetMacro, 1413, 1414
vtkSetVector3Macro, 1414, 1415
vtkSetVector6Macro, 1415
vtkTypeMacro, 1415
vtkGDCMThreadedImageReader2.h, 2138
vtkGetMacro
vtkGDCMImageReader, 1354–1356
vtkGDCMImageReader2, 1369–1371
vtkGDCMImageWriter, 1381, 1382
vtkGDCMThreadedImageReader, 1406
vtkGDCMThreadedImageReader2, 1411, 1412
vtkImageColorViewer, 1427
vtkImageMapToColors16, 1435
vtkImageMapToWindowLevelColors2, 1439, 1440
vtkGetObjectMacro
vtkGDCMImageReader, 1356
vtkGDCMImageReader2, 1371
vtkGDCMImageWriter, 1383
vtkGDCMPolyDataReader, 1393
vtkGDCMThreadedImageReader2, 1413
vtkImageColorViewer, 1427, 1428
vtkImageMapToColors16, 1435
vtkGetStringMacro
vtkGDCMImageReader, 1357
vtkGDCMImageReader2, 1371, 1372
vtkGDCMImageWriter, 1383
vtkGDCMPolyDataReader, 1393
vtkRTStructSetProperties, 1459, 1460
vtkGetVector3Macro
vtkGDCMImageReader, 1357
vtkGDCMImageReader2, 1372
vtkGDCMThreadedImageReader2, 1413
vtkGetVector6Macro
vtkGDCMImageReader, 1357
vtkGDCMImageReader2, 1372
vtkGDCMThreadedImageReader2, 1413
vtkImageColorViewer, 1416
~vtkImageColorViewer, 1419
AddInput, 1419
AddInputConnection, 1419
FirstRender, 1429
GetColorLevel, 1419
GetColorWindow, 1420
GetInput, 1420
GetOffScreenRendering, 1420
GetOverlayVisibility, 1420
GetPosition, 1420
GetSize, 1420
GetSliceMax, 1420
GetSliceMin, 1421
GetSliceRange, 1421
GetWindowName, 1421
ImageActor, 1429
InstallPipeline, 1421
Interactor, 1429
InteractorStyle, 1429
New, 1421
OverlayImageActor, 1429
PrintSelf, 1422
Render, 1422
Renderer, 1429
RenderWindow, 1430
SetColorLevel, 1422
SetColorWindow, 1422
SetDisplayId, 1422
SetInput, 1423
SetInputConnection, 1423
SetOffScreenRendering, 1423
SetOverlayVisibility, 1423
SetParentId, 1423
SetPosition, 1423, 1424
SetRenderer, 1424
SetRenderWindow, 1424
SetSize, 1424
SetSlice, 1425
SetSliceOrientation, 1425

SetSliceOrientationToXY, 1425
SetSliceOrientationToXZ, 1425
SetSliceOrientationToYZ, 1425
SetupInteractor, 1425
SetWindowId, 1426
Slice, 1430
SLICE_ORIENTATION_XY, 1419
SLICE_ORIENTATION_XZ, 1419
SLICE_ORIENTATION_YZ, 1419
SliceOrientation, 1430
UnInstallPipeline, 1426
UpdateDisplayExtent, 1426
UpdateOrientation, 1426
VTK_LEGACY, 1426, 1427
vtkBooleanMacro, 1427
vtkGetMacro, 1427
vtkGetObjectMacro, 1427, 1428
vtkImageColorViewer, 1419
vtkImageColorViewerCallback, 1429
vtkTypeMacro, 1428
WindowLevel, 1430
vtkImageColorViewer.h, 2140, 2141
vtkImageColorViewerCallback
vtkImageColorViewer, 1429
vtkImageMapToColors16, 1431
~vtkImageMapToColors16, 1432
ActiveComponent, 1436
DataWasPassed, 1436
GetMTime, 1433
LookupTable, 1436
New, 1433
OutputFormat, 1436
PassAlphaToOutput, 1437
PrintSelf, 1433
RequestData, 1433
RequestInformation, 1433
SetLookupTable, 1433
SetOutputFormatToLuminance, 1434
SetOutputFormatToLuminanceAlpha, 1434
SetOutputFormatToRGB, 1434
SetOutputFormatToRGBA, 1434
ThreadedRequestData, 1434
vtkBooleanMacro, 1434
vtkGetMacro, 1435
vtkGetObjectMacro, 1435
vtkImageMapToColors16, 1432
vtkSetMacro, 1435, 1436
vtkTypeMacro, 1436
vtkImageMapToColors16.h, 2144, 2145
vtkImageMapToWindowLevelColors2, 1437
~vtkImageMapToWindowLevelColors2, 1438
Level, 1440
New, 1439
PrintSelf, 1439
RequestData, 1439
RequestInformation, 1439
ThreadedRequestData, 1439
vtkGetMacro, 1439, 1440
vtkImageMapToWindowLevelColors2, 1438
vtkSetMacro, 1440
vtkTypeMacro, 1440
Window, 1441
vtkImageMapToWindowLevelColors2.h, 2147
vtkImagePlanarComponentsToComponents, 1441
~vtkImagePlanarComponentsToComponents, 1442
New, 1442
PrintSelf, 1442
RequestData, 1443
vtkImagePlanarComponentsToComponents, 1442
vtkTypeMacro, 1443
vtkImagePlanarComponentsToComponents.h, 2149
vtkImageRGBToYBR, 1443
~vtkImageRGBToYBR, 1444
New, 1445
PrintSelf, 1445
ThreadedExecute, 1445
vtkImageRGBToYBR, 1444
vtkTypeMacro, 1445
vtkImageRGBToYBR.h, 2150, 2151
vtkImageYBRToRGB, 1446
~vtkImageYBRToRGB, 1447
New, 1447
PrintSelf, 1447
ThreadedExecute, 1447
vtkImageYBRToRGB, 1447
vtkTypeMacro, 1448
vtkImageYBRToRGB.h, 2152
vtkLookupTable16, 1448
~vtkLookupTable16, 1450
Build, 1450
GetPointer, 1450
MapScalarsThroughTable2, 1450
New, 1450
PrintSelf, 1451
SetNumberOfTableValues, 1451
Table16, 1451
vtkLookupTable16, 1449
vtkTypeMacro, 1451
WritePointer, 1451
vtkLookupTable16.h, 2153, 2154
vtkRTStructSetProperties, 1452
~vtkRTStructSetProperties, 1454
AddContourReferencedFrameOfReference, 1454
AddReferencedFrameOfReference, 1455
AddStructureSetROI, 1455
AddStructureSetROIObservation, 1455
Clear, 1455
DeepCopy, 1455

GetContourReferencedFrameOfReferenceClassUID, 1456
GetContourReferencedFrameOfReferenceInstanceUID, 1456
GetNumberOfContourReferencedFrameOfReferences, 1456
GetNumberOfReferencedFrameOfReferences, 1456
GetNumberOfStructureSetROIs, 1456
GetReferencedFrameOfReferenceClassUID, 1457
GetReferencedFrameOfReferenceInstanceUID, 1457
GetStructureSetObservationNumber, 1457
GetStructureSetROIDescription, 1457
GetStructureSetROIGenerationAlgorithm, 1457
GetStructureSetROIName, 1457
GetStructureSetROINumber, 1458
GetStructureSetROIObservationLabel, 1458
GetStructureSetROIRefFrameRefUID, 1458
GetStructureSetRTROIInterpretedType, 1458
Internals, 1462
New, 1458
PrintSelf, 1458
ReferenceFrameOfReferenceUID, 1462
ReferenceSeriesInstanceUID, 1462
SeriesInstanceUID, 1462
SOPInstanceUID, 1462
StructureSetDate, 1463
StructureSetLabel, 1463
StructureSetName, 1463
StructureSetTime, 1463
StudyInstanceUID, 1463
vtkGetStringMacro, 1459, 1460
vtkRTStructSetProperties, 1454
vtkSetStringMacro, 1460, 1461
vtkTypeMacro, 1462
vtkRTStructSetProperties.h, 2155
vtkSetMacro
vtkGDCMImageReader, 1357, 1358
vtkGDCMImageReader2, 1372, 1373
vtkGDCMImageWriter, 1383, 1384
vtkGDCMThreadedImageReader, 1406
vtkGDCMThreadedImageReader2, 1413, 1414
vtkImageMapToColors16, 1435, 1436
vtkImageMapToWindowLevelColors2, 1440
vtkSetStringMacro
vtkGDCMImageWriter, 1385
vtkGDCMPolyDataReader, 1393
vtkRTStructSetProperties, 1460, 1461
vtkSetVector3Macro
vtkGDCMThreadedImageReader2, 1414, 1415
vtkSetVector6Macro
vtkGDCMImageReader, 1358
vtkGDCMImageReader2, 1373
vtkGDCMThreadedImageReader2, 1415
vtkTypeMacro
vtkGDCMImageReader, 1358
vtkGDCMImageReader2, 1373
vtkGDCMImageWriter, 1385
vtkGDCMMedicalImageProperties, 1388
vtkGDCMPolyDataReader, 1393
vtkGDCMPolyDataWriter, 1398
vtkGDCMTesting, 1402
vtkGDCMThreadedImageReader, 1407
vtkGDCMThreadedImageReader2, 1415
vtkImageColorViewer, 1428
vtkImageMapToColors16, 1436
vtkImageMapToWindowLevelColors2, 1440
vtkImagePlanarComponentsToComponents, 1443
vtkImageRGBToYBR, 1445
vtkImageYBRToRGB, 1448
vtkLookupTable16, 1451
vtkRTStructSetProperties, 1462

WarningOff
gdcm::Trace, 1180
WarningOn
gdcm::Trace, 1180
Waveform
gdcm::MediaStorage, 699
gdcm::Waveform, 1464
WaveformStorageTrialRetired
gdcm::UIDs, 1217
WeirdPapryus
gdcm::TransferSyntax, 1183
what
gdcm::Exception, 454
white
gdcm::terminal, 85
WideFieldOphthalmicPhotography3DCoordinatesImageStorage
gdcm::UIDs, 1221
WideFieldOphthalmicPhotographyStereographicProjectionImageStorage
gdcm::UIDs, 1221
Window
vtkImageMapToWindowLevelColors2, 1441
WindowLevel
vtkImageColorViewer, 1430
WinterColorPaletteSOPInstance
gdcm::UIDs, 1220
WIREFRAME
gdcm::Surface, 1110
WLMFindQuery
gdcm::WLMFindQuery, 1465
Write
gdcm::ByteValue, 233
gdcm::CommandDataSet, 270
gdcm::DataElement, 332
gdcm::DataSet, 350
gdcm::Element< TVR, TVM >, 415
gdcm::Element< TVR, VM::VM1_n >, 422

gdcm::EncodingImplementation< VR::VRASCII >, 443
gdcm::EncodingImplementation< VR::VRBINARY >, 444
gdcm::ExplicitDataElement, 457
gdcm::File, 465
gdcm::FileAnonymizer, 470
gdcm::FileMetaInformation, 493
gdcm::Fragment, 525
gdcm::ImageWriter, 603
gdcm::ImplicitDataElement, 610
gdcm::Item, 629
gdcm::network::AAbortPDU, 89
gdcm::network::AAssociateACPDU, 93
gdcm::network::AAssociateRJPDU, 96
gdcm::network::AAssociateRQPDU, 102
gdcm::network::AbstractSyntax, 106
gdcm::network::ApplicationContext, 123
gdcm::network::AReleaseRPPDU, 128
gdcm::network::AReleaseRQPDU, 130
gdcm::network::AsynchronousOperationsWindowSub, 136
gdcm::network::BasePDU, 179
gdcm::network::ImplementationClassUIDSub, 605
gdcm::network::ImplementationUIDSub, 606
gdcm::network::ImplementationVersionNameSub, 607
gdcm::network::MaximumLengthSub, 691
gdcm::network::PDataTFPDU, 806
gdcm::network::PresentationContextAC, 870
gdcm::network::PresentationContextRQ, 878
gdcm::network::PresentationDataValue, 882
gdcm::network::RoleSelectionSub, 948
gdcm::network::ServiceClassApplicationInformation, 1016
gdcm::network::SOPClassExtendedNegociationSub, 1044
gdcm::network::TransferSyntaxSub, 1189
gdcm::network::UserInformation, 1314
gdcm::PGXCodec, 824
gdcm::PixmapWriter, 855
gdcm::PNMCodec, 859
gdcm::Preamble, 863
gdcm::SegmentWriter, 990
gdcm::SequenceOfFragments, 998
gdcm::SequenceOfItems, 1007
gdcm::StreamImageWriter, 1068
gdcm::SurfaceWriter, 1131
gdcm::Tag, 1163
gdcm::ValueIO< TDE, TSwap, TType >, 1321
gdcm::VL, 1328
gdcm::VR, 1341
gdcm::VRVLSize< 0 >, 1346
gdcm::VRVLSize< 1 >, 1347
gdcm::Writer, 1473
vtkGDCMImageWriter, 1385
Write16
gdcm::VL, 1328
WriteASCII
gdcm::Element< TVR, VM::VM1_n >, 422
WriteBuffer
gdcm::ByteValue, 233
gdcm::SequenceOfFragments, 998
WriteBufferAsRGBA
gdcm::LookupTable, 681
WriteData
vtkGDCMPolyDataWriter, 1398
WriteFooter
gdcm::DictConverter, 379
WriteGDCMData
vtkGDCMImageWriter, 1385
WriteHeader
gdcm::DictConverter, 380
WriteHelpFile
gdcm::BaseQuery, 184
WriteImageInformation
gdcm::StreamImageWriter, 1069
WriteImageSubregionRAW
gdcm::StreamImageWriter, 1069
WritePointer
vtkLookupTable16, 1451
WriteQuery
gdcm::BaseQuery, 184
Writer
gdcm::Writer, 1470
WriteRawHeader
gdcm::StreamImageWriter, 1069
WriteRTSTRUCTData
vtkGDCMPolyDataWriter, 1398
WriteRTSTRUCTInfo
vtkGDCMPolyDataWriter, 1398
WriteSlice
vtkGDCMImageWriter, 1386

x16printf
gdcm, 76
XAXRFGrayscaleSoftcopyPresentationStateStorage
gdcm::UIDs, 1221
XML
gdcm::Printer, 884
XMLDictReader
gdcm::XMLDictReader, 1475
XMLEncoding
gdcm::UIDs, 1214
XMLPrinter
gdcm::XMLPrinter, 1478
XMLPrivateDictReader
gdcm::XMLPrivateDictReader, 1482

XRay3DAngiographicImageStorage
gdcm::MediaStorage, 698
gdcm::UIDs, 1217
XRay3DCraniofacialImageStorage
gdcm::MediaStorage, 698
gdcm::UIDs, 1217
XRayAngiographicBiPlaneImageStorageRetired
gdcm::MediaStorage, 697
gdcm::UIDs, 1217
XRayAngiographicImageStorage
gdcm::MediaStorage, 697
gdcm::UIDs, 1217
XRayRadiationDoseSR
gdcm::MediaStorage, 698
XRayRadiationDoseSRStorage
gdcm::UIDs, 1218
XRayRadiofluoroscopicImageStorage
gdcm::UIDs, 1217
XRayRadiofluoroscopingImageStorage
gdcm::MediaStorage, 697

YBR2RGB
gdcm::ImageChangePhotometricInterpretation, 554
YBR_FULL
gdcm::PhotometricInterpretation, 826
YBR_FULL_422
gdcm::PhotometricInterpretation, 826
YBR_ICT
gdcm::PhotometricInterpretation, 826
YBR_PARTIAL_420
gdcm::PhotometricInterpretation, 826
YBR_PARTIAL_422
gdcm::PhotometricInterpretation, 826
YBR_RCT
gdcm::PhotometricInterpretation, 826
yellow
gdcm::terminal, 85
YES
gdcm::Surface, 1110

ZEROED_OUT
gdcm::CSAHeader, 299
ZSpacing
gdcm::IPPSorter, 625
ZTolerance
gdcm::IPPSorter, 625