GDCM

2.6.6

Generated by Doxygen 1.8.12

Contents

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 . . . 57

9.1.2 Typedef Documentation . . . 57

9.1.2.1 AEComp . . . 57

9.1.2.2 ASComp . . . 58

9.1.2.3 BOOL_FUNCTION_PFILE_PFILE_POINTER . . . 58

9.1.2.4 CSComp . . . 58

9.1.2.5 DAComp . . . 58

9.1.2.6 DTComp . . . 58

9.1.2.7 FileList . . . 58

9.1.2.8 IconImage . . . 58

9.1.2.9 LOComp . . . 58

9.1.2.10 LTComp . . . 58

9.1.2.11 MacroEntry . . . 58

9.1.2.12 NestedMacroEntries . . . 59

9.1.2.13 PNComp . . . 59

9.1.2.14 SHComp . . . 59

9.1.2.15 STComp . . . 59

9.1.2.16 TMComp . . . 59

9.1.2.17 UIComp . . . 59

9.1.2.18 UTComp . . . 59

9.1.3 Enumeration Type Documentation . . . 59

9.1.3.1 CompOperators . . . 59

9.1.3.2 ECharSet . . . 60

9.1.3.3 ENQueryType . . . 60

9.1.3.4 EQueryLevel . . . 60

9.1.3.5 EQueryType . . . 61

9.1.3.6 ERootType . . . 61
9.1.3.7 LodModeType . . . 61
9.1.4 Function Documentation . . . 61
9.1.4.1 backslash() . . . 61
9.1.4.2 GetVRFromTag() . . . 61
9.1.4.3 operator"!=() [1/2] . . . 62
9.1.4.4 operator"!=() [2/2] . . . 62
9.1.4.5 operator<<() [1/55] . . . 62
9.1.4.6 operator<<() [2/55] . . . 62
9.1.4.7 operator<<() [3/55] . . . 62
9.1.4.8 operator<<() [4/55] . . . 62
9.1.4.9 operator<<() [5/55] . . . 63
9.1.4.10 operator<<() [6/55] . . . 63
9.1.4.11 operator<<() [7/55] . . . 63
9.1.4.12 operator<<() [8/55] . . . 63
9.1.4.13 operator<<() [9/55] . . . 63
9.1.4.14 operator<<() [10/55] . . . 63
9.1.4.15 operator<<() [11/55] . . . 64
9.1.4.16 operator<<() [12/55] . . . 64
9.1.4.17 operator<<() [13/55] . . . 64
9.1.4.18 operator<<() [14/55] . . . 64
9.1.4.19 operator<<() [15/55] . . . 64
9.1.4.20 operator<<() [16/55] . . . 64
9.1.4.21 operator<<() [17/55] . . . 65
9.1.4.22 operator<<() [18/55] . . . 65
9.1.4.23 operator<<() [19/55] . . . 65
9.1.4.24 operator<<() [20/55] . . . 65
9.1.4.25 operator<<() [21/55] . . . 65

9.1.4.26 operator<<() [22/55] . . . 65
9.1.4.27 operator<<() [23/55] . . . 66
9.1.4.28 operator<<() [24/55] . . . 66
9.1.4.29 operator<<() [25/55] . . . 66
9.1.4.30 operator<<() [26/55] . . . 66
9.1.4.31 operator<<() [27/55] . . . 66
9.1.4.32 operator<<() [28/55] . . . 66
9.1.4.33 operator<<() [29/55] . . . 67
9.1.4.34 operator<<() [30/55] . . . 67
9.1.4.35 operator<<() [31/55] . . . 67
9.1.4.36 operator<<() [32/55] . . . 67
9.1.4.37 operator<<() [33/55] . . . 67
9.1.4.38 operator<<() [34/55] . . . 68
9.1.4.39 operator<<() [35/55] . . . 68
9.1.4.40 operator<<() [36/55] . . . 68
9.1.4.41 operator<<() [37/55] . . . 68
9.1.4.42 operator<<() [38/55] . . . 68
9.1.4.43 operator<<() [39/55] . . . 68
9.1.4.44 operator<<() [40/55] . . . 69
9.1.4.45 operator<<() [41/55] . . . 69
9.1.4.46 operator<<() [42/55] . . . 69
9.1.4.47 operator<<() [43/55] . . . 69
9.1.4.48 operator<<() [44/55] . . . 69
9.1.4.49 operator<<() [45/55] . . . 69
9.1.4.50 operator<<() [46/55] . . . 70
9.1.4.51 operator<<() [47/55] . . . 70
9.1.4.52 operator<<() [48/55] . . . 70
9.1.4.53 operator<<() [49/55] . . . 70

9.1.4.54 operator<<() [50/55] . . . 70
9.1.4.55 operator<<() [51/55] . . . 71
9.1.4.56 operator<<() [52/55] . . . 71
9.1.4.57 operator<<() [53/55] . . . 71
9.1.4.58 operator<<() [54/55] . . . 71
9.1.4.59 operator<<() [55/55] . . . 71
9.1.4.60 operator==() . . . 72
9.1.4.61 operator>>() [1/3] . . . 72
9.1.4.62 operator>>() [2/3] . . . 72
9.1.4.63 operator>>() [3/3] . . . 72
9.1.4.64 to_string() . . . 72
9.1.4.65 TYPETOENCODING() . . . 73
9.1.5 Variable Documentation . . . 73
9.1.5.1 GlobalInstance . . . 73
9.1.5.2 VRBINARY . . . 73
9.2 gdcm::network Namespace Reference . . . 73
9.2.1 Enumeration Type Documentation . . . 77
9.2.1.1 EEventID . . . 77
9.2.1.2 EStateID . . . 78
9.2.2 Function Documentation . . . 78
9.2.2.1 GetStateIndex() . . . 79
9.2.3 Variable Documentation . . . 79
9.2.3.1 cMaxEventID . . . 79
9.2.3.2 cMaxStateID . . . 79
9.3 gdcm::SegmentHelper Namespace Reference . . . 79
9.4 gdcm::terminal Namespace Reference . . . 79
9.4.1 Detailed Description . . . 80
9.4.2 Enumeration Type Documentation . . . 80
9.4.2.1 Attribute . . . 80
9.4.2.2 Color . . . 81
9.4.2.3 Mode . . . 81
9.4.3 Function Documentation . . . 81
9.4.3.1 setattribute() . . . 81
9.4.3.2 setbgcolor() . . . 82
9.4.3.3 setfgcolor() . . . 82
9.4.3.4 setmode() . . . 82

10 Class Documentation **83**

10.1 gdcm::network::AAbortPDU Class Reference . . . 83

10.1.1 Detailed Description . . . 84

10.1.2 Constructor & Destructor Documentation . . . 84

10.1.2.1 AAbortPDU() . . . 84

10.1.3 Member Function Documentation . . . 84

10.1.3.1 IsLastFragment() . . . 84

10.1.3.2 Print() . . . 84

10.1.3.3 Read() . . . 84

10.1.3.4 SetReason() . . . 85

10.1.3.5 SetSource() . . . 85

10.1.3.6 Size() . . . 85

10.1.3.7 Write() . . . 85

10.2 gdcm::network::AAssociateACPDU Class Reference . . . 85

10.2.1 Detailed Description . . . 87

10.2.2 Member Typedef Documentation . . . 87

10.2.2.1 SizeType . . . 87

10.2.3 Constructor & Destructor Documentation . . . 87

10.2.3.1 AAssociateACPDU() . . . 87

10.2.4 Member Function Documentation . . . 87

10.2.4.1 AddPresentationContextAC() . . . 87

10.2.4.2 GetNumberOfPresentationContextAC() . . . 87

10.2.4.3 GetPresentationContextAC() . . . 87

10.2.4.4 GetUserInformation() . . . 87

10.2.4.5 InitFromRQ() . . . 88

10.2.4.6 IsLastFragment() . . . 88

10.2.4.7 Print() . . . 88

10.2.4.8 Read() . . . 88

10.2.4.9 SetCalledAETitle() . . . 88
10.2.4.10 SetCallingAETitle() . . . 88
10.2.4.11 Size() . . . 89
10.2.4.12 Write() . . . 89
10.2.5 Friends And Related Function Documentation . . . 89
10.2.5.1 AAssociateRQPDU . . . 89
10.3 gdcm::network::AAssociateRJPDU Class Reference . . . 89
10.3.1 Detailed Description . . . 90
10.3.2 Constructor & Destructor Documentation . . . 90
10.3.2.1 AAssociateRJPDU() . . . 90
10.3.3 Member Function Documentation . . . 90
10.3.3.1 IsLastFragment() . . . 90
10.3.3.2 Print() . . . 91
10.3.3.3 Read() . . . 91
10.3.3.4 Size() . . . 91
10.3.3.5 Write() . . . 91
10.4 gdcm::network::AAssociateRQPDU Class Reference . . . 91
10.4.1 Detailed Description . . . 93
10.4.2 Member Typedef Documentation . . . 93
10.4.2.1 PresentationContextArrayType . . . 93
10.4.2.2 SizeType . . . 94
10.4.3 Constructor & Destructor Documentation . . . 94
10.4.3.1 AAssociateRQPDU() [1/2] . . . 94
10.4.3.2 AAssociateRQPDU() [2/2] . . . 94
10.4.4 Member Function Documentation . . . 94
10.4.4.1 AddPresentationContext() . . . 94
10.4.4.2 GetCalledAETitle() . . . 94
10.4.4.3 GetCallingAETitle() . . . 94

10.4.4.4 GetNumberOfPresentationContext() . . . 94
10.4.4.5 GetPresentationContext() . . . 95
10.4.4.6 GetPresentationContextByAbstractSyntax() . . . 95
10.4.4.7 GetPresentationContextByID() . . . 95
10.4.4.8 GetPresentationContexts() . . . 95
10.4.4.9 GetReserved43_74() . . . 95
10.4.4.10 GetUserInformation() . . . 95
10.4.4.11 IsAETitleValid() . . . 95
10.4.4.12 IsLastFragment() . . . 96
10.4.4.13 Print() . . . 96
10.4.4.14 Read() . . . 96
10.4.4.15 SetCalledAETitle() . . . 96
10.4.4.16 SetCallingAETitle() . . . 96
10.4.4.17 SetUserInformation() . . . 97
10.4.4.18 Size() . . . 97
10.4.4.19 Write() . . . 97
10.4.5 Friends And Related Function Documentation . . . 97
10.4.5.1 AAssociateACPDU . . . 97
10.5 gdcm::AbortEvent Class Reference . . . 97
10.6 gdcm::network::AbstractSyntax Class Reference . . . 99
10.6.1 Detailed Description . . . 99
10.6.2 Constructor & Destructor Documentation . . . 99
10.6.2.1 AbstractSyntax() . . . 99
10.6.3 Member Function Documentation . . . 99
10.6.3.1 GetAsDataElement() . . . 99
10.6.3.2 GetName() . . . 99
10.6.3.3 operator==() . . . 100
10.6.3.4 Print() . . . 100

10.6.3.5 Read() . . . 100
10.6.3.6 SetName() . . . 100
10.6.3.7 SetNameFromUID() . . . 100
10.6.3.8 Size() . . . 100
10.6.3.9 Write() . . . 100
10.7 gdcm::AnonymizeEvent Class Reference . . . 101
10.7.1 Detailed Description . . . 102
10.7.2 Member Typedef Documentation . . . 102
10.7.2.1 Self . . . 102
10.7.2.2 Superclass . . . 102
10.7.3 Constructor & Destructor Documentation . . . 102
10.7.3.1 AnonymizeEvent() [1/2] . . . 102
10.7.3.2 ∼AnonymizeEvent() . . . 102
10.7.3.3 AnonymizeEvent() [2/2] . . . 103
10.7.4 Member Function Documentation . . . 103
10.7.4.1 CheckEvent() . . . 103
10.7.4.2 GetEventName() . . . 103
10.7.4.3 GetTag() . . . 103
10.7.4.4 MakeObject() . . . 103
10.7.4.5 SetTag() . . . 103
10.8 gdcm::Anonymizer Class Reference . . . 104
10.8.1 Detailed Description . . . 105
10.8.2 Constructor & Destructor Documentation . . . 107
10.8.2.1 Anonymizer() . . . 107
10.8.2.2 ∼Anonymizer() . . . 107
10.8.3 Member Function Documentation . . . 107
10.8.3.1 BALCPProtect() . . . 107
10.8.3.2 BasicApplicationLevelConfidentialityProfile() . . . 107

10.8.3.3 CanEmptyTag() . . . 107
10.8.3.4 ClearInternalUIDs() . . . 107
10.8.3.5 Empty() . . . 108
10.8.3.6 GetBasicApplicationLevelConfidentialityProfileAttributes() . . . 108
10.8.3.7 GetCryptographicMessageSyntax() . . . 108
10.8.3.8 GetFile() . . . 108
10.8.3.9 New() . . . 108
10.8.3.10 RecurseDataSet() . . . 108
10.8.3.11 Remove() . . . 109
10.8.3.12 RemoveGroupLength() . . . 109
10.8.3.13 RemovePrivateTags() . . . 109
10.8.3.14 RemoveRetired() . . . 109
10.8.3.15 Replace() [1/2] . . . 109
10.8.3.16 Replace() [2/2] . . . 110
10.8.3.17 SetCryptographicMessageSyntax() . . . 110
10.8.3.18 SetFile() . . . 110
10.9 gdcm::AnyEvent Class Reference . . . 110
10.10 gdcm::network::ApplicationContext Class Reference . . . 112
10.10.1 Detailed Description . . . 112
10.10.2 Constructor & Destructor Documentation . . . 113
10.10.2.1 ApplicationContext() . . . 113
10.10.3 Member Function Documentation . . . 113
10.10.3.1 GetName() . . . 113
10.10.3.2 Print() . . . 113
10.10.3.3 Read() . . . 113
10.10.3.4 SetName() . . . 113
10.10.3.5 Size() . . . 113
10.10.3.6 Write() . . . 113

10.11gdcm::ApplicationEntity Class Reference . . . 114
10.11.1 Detailed Description . . . 115
10.11.2 Member Function Documentation . . . 115
10.11.2.1 IsValid() . . . 115
10.11.2.2 Print() . . . 115
10.11.2.3 SetBlob() . . . 115
10.11.2.4 Squeeze() . . . 115
10.11.3 Member Data Documentation . . . 115
10.11.3.1 Internal . . . 115
10.11.3.2 MaxLength . . . 115
10.11.3.3 MaxNumberOfComponents . . . 116
10.11.3.4 Padding . . . 116
10.11.3.5 Separator . . . 116
10.12gdcm::network::AReleaseRPPDU Class Reference . . . 116
10.12.1 Detailed Description . . . 117
10.12.2 Constructor & Destructor Documentation . . . 117
10.12.2.1 AReleaseRPPDU() . . . 117
10.12.3 Member Function Documentation . . . 117
10.12.3.1 IsLastFragment() . . . 117
10.12.3.2 Print() . . . 118
10.12.3.3 Read() . . . 118
10.12.3.4 Size() . . . 118
10.12.3.5 Write() . . . 118
10.13gdcm::network::AReleaseRQPDU Class Reference . . . 118
10.13.1 Detailed Description . . . 119
10.13.2 Constructor & Destructor Documentation . . . 120
10.13.2.1 AReleaseRQPDU() . . . 120
10.13.3 Member Function Documentation . . . 120

10.13.3.1 IsLastFragment() . . . 120
10.13.3.2 Print() . . . 120
10.13.3.3 Read() . . . 120
10.13.3.4 Size() . . . 120
10.13.3.5 Write() . . . 120
10.14gdcm::network::ARTIMTimer Class Reference . . . 121
10.14.1 Detailed Description . . . 121
10.14.2 Constructor & Destructor Documentation . . . 121
10.14.2.1 ARTIMTimer() . . . 121
10.14.3 Member Function Documentation . . . 121
10.14.3.1 GetElapsedTime() . . . 121
10.14.3.2 GetHasExpired() . . . 122
10.14.3.3 GetTimeout() . . . 122
10.14.3.4 SetTimeout() . . . 122
10.14.3.5 Start() . . . 122
10.14.3.6 Stop() . . . 122
10.15gdcm::ASN1 Class Reference . . . 122
10.15.1 Detailed Description . . . 123
10.15.2 Constructor & Destructor Documentation . . . 123
10.15.2.1 ASN1() . . . 123
10.15.2.2 ∼ASN1() . . . 123
10.15.3 Member Function Documentation . . . 123
10.15.3.1 ParseDump() . . . 123
10.15.3.2 ParseDumpFile() . . . 123
10.15.3.3 TestPBKDF2() . . . 123
10.16gdcm::network::AsynchronousOperationsWindowSub Class Reference . . . 124
10.16.1 Detailed Description . . . 124
10.16.2 Constructor & Destructor Documentation . . . 124

10.16.2.1 AsynchronousOperationsWindowSub() . . . 124
10.16.3 Member Function Documentation . . . 124
10.16.3.1 Print() . . . 124
10.16.3.2 Read() . . . 124
10.16.3.3 Size() . . . 124
10.16.3.4 Write() . . . 125
10.17 gdcm::Attribute< Group, Element, TVR, TVM > Class Template Reference . . . 125
10.17.1 Detailed Description . . . 127
10.17.2 Member Typedef Documentation . . . 127
10.17.2.1 ArrayType . . . 127
10.17.3 Member Enumeration Documentation . . . 127
10.17.3.1 anonymous enum . . . 127
10.17.4 Member Function Documentation . . . 128
10.17.4.1 GDCM_STATIC_ASSERT() [1/3] . . . 128
10.17.4.2 GDCM_STATIC_ASSERT() [2/3] . . . 128
10.17.4.3 GDCM_STATIC_ASSERT() [3/3] . . . 128
10.17.4.4 GetAsDataElement() . . . 128
10.17.4.5 GetDictVM() . . . 128
10.17.4.6 GetDictVR() . . . 128
10.17.4.7 GetNumberOfValues() . . . 129
10.17.4.8 GetTag() . . . 129
10.17.4.9 GetValue() [1/2] . . . 129
10.17.4.10 GetValue() [2/2] . . . 129
10.17.4.11 GetValues() . . . 129
10.17.4.12 GetVM() . . . 129
10.17.4.13 GetVR() . . . 130
10.17.4.14 operator"!=() . . . 130
10.17.4.15 operator<() . . . 130

10.17.4.16operator==() . . . 130
10.17.4.17operator[]() [1/2] . . . 130
10.17.4.18operator[]() [2/2] . . . 130
10.17.4.19Print() . . . 131
10.17.4.20Set() . . . 131
10.17.4.21SetByteValue() . . . 131
10.17.4.22SetByteValueNoSwap() . . . 131
10.17.4.23SetFromDataElement() . . . 131
10.17.4.24SetFromDataSet() . . . 132
10.17.4.25SetValue() . . . 132
10.17.4.26SetValues() . . . 132
10.17.5 Member Data Documentation . . . 132
10.17.5.1 Internal . . . 132
10.18gdcm::Attribute< Group, Element, TVR, VM::VM1 > Class Template Reference . . . 133
10.18.1 Member Typedef Documentation . . . 134
10.18.1.1 ArrayType . . . 134
10.18.2 Member Enumeration Documentation . . . 134
10.18.2.1 anonymous enum . . . 134
10.18.3 Member Function Documentation . . . 134
10.18.3.1 GDCM_STATIC_ASSERT() [1/4] . . . 134
10.18.3.2 GDCM_STATIC_ASSERT() [2/4] . . . 135
10.18.3.3 GDCM_STATIC_ASSERT() [3/4] . . . 135
10.18.3.4 GDCM_STATIC_ASSERT() [4/4] . . . 135
10.18.3.5 GetAsDataElement() . . . 135
10.18.3.6 GetDictVM() . . . 135
10.18.3.7 GetDictVR() . . . 135
10.18.3.8 GetNumberOfValues() . . . 135
10.18.3.9 GetTag() . . . 136

10.18.3.10GetValue() [1/2] . . . 136
10.18.3.11GetValue() [2/2] . . . 136
10.18.3.12GetValues() . . . 136
10.18.3.13GetVM() . . . 136
10.18.3.14GetVR() . . . 136
10.18.3.15operator"!=() . . . 136
10.18.3.16operator<() . . . 137
10.18.3.17operator==() . . . 137
10.18.3.18Print() . . . 137
10.18.3.19Set() . . . 137
10.18.3.20SetByteValue() . . . 137
10.18.3.21SetByteValueNoSwap() . . . 137
10.18.3.22SetFromDataElement() . . . 138
10.18.3.23SetFromDataSet() . . . 138
10.18.3.24SetValue() . . . 138
10.18.4 Member Data Documentation . . . 138
10.18.4.1 Internal . . . 138
10.19gdcm::Attribute< Group, Element, TVR, VM::VM1_3 > Class Template Reference . . . 139
10.19.1 Member Function Documentation . . . 140
10.19.1.1 GetVM() . . . 140
10.20gdcm::Attribute< Group, Element, TVR, VM::VM1_8 > Class Template Reference . . . 140
10.20.1 Member Function Documentation . . . 141
10.20.1.1 GetVM() . . . 141
10.21gdcm::Attribute< Group, Element, TVR, VM::VM1_n > Class Template Reference . . . 141
10.21.1 Member Typedef Documentation . . . 142
10.21.1.1 ArrayType . . . 142
10.21.2 Constructor & Destructor Documentation . . . 143
10.21.2.1 Attribute() . . . 143

10.21.2.2 ~Attribute() . . . 143
10.21.3 Member Function Documentation . . . 143
10.21.3.1 GDCM_STATIC_ASSERT() [1/3] . . . 143
10.21.3.2 GDCM_STATIC_ASSERT() [2/3] . . . 143
10.21.3.3 GDCM_STATIC_ASSERT() [3/3] . . . 143
10.21.3.4 GetAsDataElement() . . . 143
10.21.3.5 GetDictVM() . . . 144
10.21.3.6 GetDictVR() . . . 144
10.21.3.7 GetNumberOfValues() . . . 144
10.21.3.8 GetTag() . . . 144
10.21.3.9 GetValue() [1/2] . . . 144
10.21.3.10 GetValue() [2/2] . . . 144
10.21.3.11 GetValues() . . . 144
10.21.3.12 GetVM() . . . 144
10.21.3.13 GetVR() . . . 145
10.21.3.14 operator[]() [1/2] . . . 145
10.21.3.15 operator[]() [2/2] . . . 145
10.21.3.16 Print() . . . 145
10.21.3.17 Set() . . . 145
10.21.3.18 SetByteValue() . . . 145
10.21.3.19 SetFromDataElement() . . . 145
10.21.3.20 SetFromDataSet() . . . 146
10.21.3.21 SetNumberOfValues() . . . 146
10.21.3.22 SetValue() [1/2] . . . 146
10.21.3.23 SetValue() [2/2] . . . 146
10.21.3.24 SetValues() . . . 146
10.22 gdcm::Attribute< Group, Element, TVR, VM::VM2_2n > Class Template Reference . . . 147
10.22.1 Member Function Documentation . . . 148

10.22.1.1 GetVM() . . . 148
10.23gdcm::Attribute< Group, Element, TVR, VM::VM2_n > Class Template Reference . . . 148
10.23.1 Member Function Documentation . . . 149
10.23.1.1 GetVM() . . . 149
10.24gdcm::Attribute< Group, Element, TVR, VM::VM3_3n > Class Template Reference . . . 149
10.24.1 Member Function Documentation . . . 151
10.24.1.1 GetVM() . . . 151
10.25gdcm::Attribute< Group, Element, TVR, VM::VM3_n > Class Template Reference . . . 151
10.25.1 Member Function Documentation . . . 152
10.25.1.1 GetVM() . . . 152
10.26gdcm::AudioCodec Class Reference . . . 152
10.26.1 Detailed Description . . . 154
10.26.2 Constructor & Destructor Documentation . . . 154
10.26.2.1 AudioCodec() . . . 154
10.26.2.2 ∼AudioCodec() . . . 154
10.26.3 Member Function Documentation . . . 154
10.26.3.1 CanCode() . . . 154
10.26.3.2 CanDecode() . . . 154
10.26.3.3 Decode() . . . 154
10.27gdcm::Base64 Class Reference . . . 155
10.27.1 Detailed Description . . . 155
10.27.2 Member Function Documentation . . . 155
10.27.2.1 Decode() . . . 155
10.27.2.2 Encode() . . . 156
10.27.2.3 GetDecodeLength() . . . 156
10.27.2.4 GetEncodeLength() . . . 156
10.28gdcm::network::BaseCompositeMessage Class Reference . . . 157
10.28.1 Detailed Description . . . 157

10.28.2 Constructor & Destructor Documentation . . . 158
10.28.2.1 ∼BaseCompositeMessage() . . . 158
10.28.3 Member Function Documentation . . . 158
10.28.3.1 ConstructPDV() . . . 158
10.29gdcm::network::BaseNormalizedMessage Class Reference . . . 159
10.29.1 Detailed Description . . . 160
10.29.2 Constructor & Destructor Documentation . . . 160
10.29.2.1 ∼BaseNormalizedMessage() . . . 160
10.29.3 Member Function Documentation . . . 160
10.29.3.1 ConstructPDV() . . . 160
10.30gdcm::network::BasePDU Class Reference . . . 161
10.30.1 Detailed Description . . . 161
10.30.2 Constructor & Destructor Documentation . . . 162
10.30.2.1 ∼BasePDU() . . . 162
10.30.3 Member Function Documentation . . . 162
10.30.3.1 IsLastFragment() . . . 162
10.30.3.2 Print() . . . 162
10.30.3.3 Read() . . . 163
10.30.3.4 Size() . . . 163
10.30.3.5 Write() . . . 163
10.31gdcm::BaseQuery Class Reference . . . 164
10.31.1 Detailed Description . . . 165
10.31.2 Constructor & Destructor Documentation . . . 165
10.31.2.1 BaseQuery() . . . 165
10.31.2.2 ∼BaseQuery() . . . 165
10.31.3 Member Function Documentation . . . 166
10.31.3.1 AddQueryDataSet() . . . 166
10.31.3.2 GetAbstractSyntaxUID() . . . 166

10.31.3.3 GetQueryDataSet() [1/2] . . . 166
10.31.3.4 GetQueryDataSet() [2/2] . . . 166
10.31.3.5 GetSOPInstanceUID() . . . 166
10.31.3.6 Print() . . . 166
10.31.3.7 SetSearchParameter() [1/3] . . . 166
10.31.3.8 SetSearchParameter() [2/3] . . . 167
10.31.3.9 SetSearchParameter() [3/3] . . . 167
10.31.3.10 SetSOPInstanceUID() . . . 167
10.31.3.11 ValidateQuery() . . . 167
10.31.3.12 ValidDataSet() . . . 167
10.31.3.13 WriteHelpFile() . . . 167
10.31.3.14 WriteQuery() . . . 167
10.31.4 Friends And Related Function Documentation . . . 168
10.31.4.1 QueryFactory . . . 168
10.31.5 Member Data Documentation . . . 168
10.31.5.1 mDataSet . . . 168
10.31.5.2 mHelpDescription . . . 168
10.31.5.3 mSopInstanceUID . . . 168
10.32 gdcm::BaseRootQuery Class Reference . . . 168
10.32.1 Detailed Description . . . 170
10.32.2 Constructor & Destructor Documentation . . . 170
10.32.2.1 BaseRootQuery() . . . 170
10.32.2.2 ~BaseRootQuery() . . . 170
10.32.3 Member Function Documentation . . . 170
10.32.3.1 Construct() . . . 170
10.32.3.2 GetQueryLevelFromQueryRoot() . . . 170
10.32.3.3 GetQueryLevelFromString() . . . 170
10.32.3.4 GetQueryLevelString() . . . 171

10.32.3.5 GetTagListByLevel() . . . 171
10.32.3.6 InitializeDataSet() . . . 171
10.32.3.7 ValidateQuery() . . . 171
10.32.4 Friends And Related Function Documentation . . . 172
10.32.4.1 QueryFactory . . . 172
10.32.5 Member Data Documentation . . . 172
10.32.5.1 mHelpDescription . . . 172
10.32.5.2 mImage . . . 172
10.32.5.3 mPatient . . . 172
10.32.5.4 mRootType . . . 172
10.32.5.5 mSeries . . . 172
10.32.5.6 mStudy . . . 172
10.33gdcm::SegmentHelper::BasicCodedEntry Struct Reference . . . 173
10.33.1 Detailed Description . . . 174
10.33.2 Constructor & Destructor Documentation . . . 174
10.33.2.1 BasicCodedEntry() [1/3] . . . 174
10.33.2.2 BasicCodedEntry() [2/3] . . . 174
10.33.2.3 BasicCodedEntry() [3/3] . . . 174
10.33.3 Member Function Documentation . . . 174
10.33.3.1 IsEmpty() . . . 174
10.33.4 Member Data Documentation . . . 175
10.33.4.1 CM . . . 175
10.33.4.2 CSD . . . 175
10.33.4.3 CSV . . . 175
10.33.4.4 CV . . . 175
10.34gdcm::BasicOffsetTable Class Reference . . . 176
10.34.1 Detailed Description . . . 177
10.34.2 Constructor & Destructor Documentation . . . 177

10.34.2.1 BasicOffsetTable() . . . 177
10.34.3 Member Function Documentation . . . 177
10.34.3.1 Read() . . . 177
10.34.4 Friends And Related Function Documentation . . . 177
10.34.4.1 operator<< . . . 177
10.35gdcm::Bitmap Class Reference . . . 178
10.35.1 Detailed Description . . . 180
10.35.2 Member Typedef Documentation . . . 181
10.35.2.1 LUTPtr . . . 181
10.35.3 Constructor & Destructor Documentation . . . 181
10.35.3.1 Bitmap() . . . 181
10.35.3.2 ~Bitmap() . . . 181
10.35.4 Member Function Documentation . . . 181
10.35.4.1 AreOverlaysInPixelData() . . . 181
10.35.4.2 Clear() . . . 181
10.35.4.3 ComputeLossyFlag() . . . 181
10.35.4.4 GetBuffer() . . . 181
10.35.4.5 GetBuffer2() . . . 182
10.35.4.6 GetBufferLength() . . . 182
10.35.4.7 GetColumns() . . . 182
10.35.4.8 GetDataElement() [1/2] . . . 182
10.35.4.9 GetDataElement() [2/2] . . . 182
10.35.4.10 GetDimension() . . . 182
10.35.4.11 GetDimensions() . . . 183
10.35.4.12 GetLUT() [1/2] . . . 183
10.35.4.13 GetLUT() [2/2] . . . 183
10.35.4.14 GetNeedByteSwap() . . . 183
10.35.4.15 GetNumberOfDimensions() . . . 183

10.35.4.16 GetPhotometricInterpretation() . . . 183
10.35.4.17 GetPixelFormat() [1/2] . . . 184
10.35.4.18 GetPixelFormat() [2/2] . . . 184
10.35.4.19 GetPlanarConfiguration() . . . 184
10.35.4.20 GetRows() . . . 184
10.35.4.21 GetTransferSyntax() . . . 184
10.35.4.22 IsEmpty() . . . 184
10.35.4.23 IsLossy() . . . 184
10.35.4.24 IsTransferSyntaxCompatible() . . . 185
10.35.4.25 Print() . . . 185
10.35.4.26 SetColumns() . . . 185
10.35.4.27 SetDataElement() . . . 185
10.35.4.28 SetDimension() . . . 185
10.35.4.29 SetDimensions() . . . 186
10.35.4.30 SetLossyFlag() . . . 186
10.35.4.31 SetLUT() . . . 186
10.35.4.32 SetNeedByteSwap() . . . 186
10.35.4.33 SetNumberOfDimensions() . . . 186
10.35.4.34 SetPhotometricInterpretation() . . . 186
10.35.4.35 SetPixelFormat() . . . 187
10.35.4.36 SetPlanarConfiguration() . . . 187
10.35.4.37 SetRows() . . . 187
10.35.4.38 SetTransferSyntax() . . . 187
10.35.4.39 TryJPEG2000Codec() . . . 187
10.35.4.40 TryJPEG2000Codec2() . . . 188
10.35.4.41 TryJPEGCodec() . . . 188
10.35.4.42 TryJPEGCodec2() . . . 188
10.35.4.43 TryJPEGLSCodec() . . . 188

10.35.4.44TryKAKADUCodec() . . . 188
10.35.4.45TryPVRGCodec() . . . 188
10.35.4.46TryRAWCodec() . . . 188
10.35.4.47TryRLECodec() . . . 189
10.35.5 Friends And Related Function Documentation . . . 189
10.35.5.1 ImageChangeTransferSyntax . . . 189
10.35.5.2 PixmapReader . . . 189
10.35.6 Member Data Documentation . . . 189
10.35.6.1 Dimensions . . . 189
10.35.6.2 LossyFlag . . . 189
10.35.6.3 LUT . . . 189
10.35.6.4 NeedByteSwap . . . 189
10.35.6.5 NumberOfDimensions . . . 189
10.35.6.6 PF . . . 190
10.35.6.7 PI . . . 190
10.35.6.8 PixelData . . . 190
10.35.6.9 PlanarConfiguration . . . 190
10.35.6.10TS . . . 190
10.36gdcm::BitmapToBitmapFilter Class Reference . . . 190
10.36.1 Detailed Description . . . 191
10.36.2 Constructor & Destructor Documentation . . . 192
10.36.2.1 BitmapToBitmapFilter() . . . 192
10.36.2.2 ∼BitmapToBitmapFilter() . . . 192
10.36.3 Member Function Documentation . . . 192
10.36.3.1 GetOutput() . . . 192
10.36.3.2 GetOutputAsBitmap() . . . 192
10.36.3.3 SetInput() . . . 192
10.36.4 Member Data Documentation . . . 192

10.36.4.1 Input . . . 192
10.36.4.2 Output . . . 193
10.37 gdcm::BoxRegion Class Reference . . . 193
10.37.1 Detailed Description . . . 194
10.37.2 Constructor & Destructor Documentation . . . 194
10.37.2.1 BoxRegion() [1/2] . . . 194
10.37.2.2 ~BoxRegion() . . . 195
10.37.2.3 BoxRegion() [2/2] . . . 195
10.37.3 Member Function Documentation . . . 195
10.37.3.1 Area() . . . 195
10.37.3.2 BoundingBox() . . . 195
10.37.3.3 Clone() . . . 195
10.37.3.4 ComputeBoundingBox() . . . 195
10.37.3.5 Empty() . . . 196
10.37.3.6 GetXMax() . . . 196
10.37.3.7 GetXMin() . . . 196
10.37.3.8 GetYMax() . . . 196
10.37.3.9 GetYMin() . . . 196
10.37.3.10 GetZMax() . . . 196
10.37.3.11 GetZMin() . . . 196
10.37.3.12 IsValid() . . . 196
10.37.3.13 operator=() . . . 197
10.37.3.14 Print() . . . 197
10.37.3.15 SetDomain() . . . 197
10.38 gdcm::ByteBuffer Class Reference . . . 197
10.38.1 Detailed Description . . . 198
10.38.2 Constructor & Destructor Documentation . . . 198
10.38.2.1 ByteBuffer() . . . 198

10.38.3 Member Function Documentation . . . 198
10.38.3.1 Get() . . . 198
10.38.3.2 GetStart() . . . 198
10.38.3.3 ShiftEnd() . . . 198
10.38.3.4 UpdatePosition() . . . 198
10.39 gdcm::ByteSwap< T > Class Template Reference . . . 199
10.39.1 Detailed Description . . . 199
10.39.2 Member Function Documentation . . . 199
10.39.2.1 Swap() . . . 199
10.39.2.2 SwapFromSwapCodeIntoSystem() . . . 199
10.39.2.3 SwapRange() . . . 200
10.39.2.4 SwapRangeFromSwapCodeIntoSystem() . . . 200
10.39.2.5 SystemIsBigEndian() . . . 200
10.39.2.6 SystemIsLittleEndian() . . . 200
10.40 gdcm::ByteSwapFilter Class Reference . . . 200
10.40.1 Detailed Description . . . 201
10.40.2 Constructor & Destructor Documentation . . . 201
10.40.2.1 ByteSwapFilter() . . . 201
10.40.2.2 ~ByteSwapFilter() . . . 201
10.40.3 Member Function Documentation . . . 201
10.40.3.1 ByteSwap() . . . 201
10.40.3.2 SetByteSwapTag() . . . 201
10.41 gdcm::ByteValue Class Reference . . . 202
10.41.1 Detailed Description . . . 203
10.41.2 Constructor & Destructor Documentation . . . 204
10.41.2.1 ByteValue() [1/2] . . . 204
10.41.2.2 ByteValue() [2/2] . . . 204
10.41.2.3 ~ByteValue() . . . 204

10.41.3 Member Function Documentation 204
10.41.3.1 Append() 204
10.41.3.2 Clear() 204
10.41.3.3 ComputeLength() 204
10.41.3.4 Fill() 205
10.41.3.5 GetBuffer() 205
10.41.3.6 GetLength() 205
10.41.3.7 GetPointer() 206
10.41.3.8 IsEmpty() 206
10.41.3.9 IsPrintable() 206
10.41.3.10 operator const std::vector< char > &() 206
10.41.3.11 operator=() 206
10.41.3.12 operator==() [1/2] 206
10.41.3.13 operator==() [2/2] 207
10.41.3.14 Print() 207
10.41.3.15 PrintASCII() 207
10.41.3.16 PrintASCIIXML() 207
10.41.3.17 PrintGroupLength() 207
10.41.3.18 PrintHex() 207
10.41.3.19 PrintHexXML() 207
10.41.3.20 PrintPNXML() 208
10.41.3.21 Read() [1/2] 208
10.41.3.22 Read() [2/2] 208
10.41.3.23 SetLength() 208
10.41.3.24 SetLengthOnly() 208
10.41.3.25 Write() [1/2] 208
10.41.3.26 Write() [2/2] 209
10.41.3.27 WriteBuffer() 209

10.42gdcm::CAPICryptoFactory Class Reference . . . 209
10.42.1 Constructor & Destructor Documentation . . . 210
10.42.1.1 CAPICryptoFactory() . . . 210
10.42.2 Member Function Documentation . . . 210
10.42.2.1 CreateCMSProvider() . . . 210
10.43gdcm::CAPICryptographicMessageSyntax Class Reference . . . 210
10.43.1 Constructor & Destructor Documentation . . . 211
10.43.1.1 CAPICryptographicMessageSyntax() . . . 211
10.43.1.2 ∼CAPICryptographicMessageSyntax() . . . 211
10.43.2 Member Function Documentation . . . 212
10.43.2.1 Decrypt() . . . 212
10.43.2.2 Encrypt() . . . 212
10.43.2.3 GetCipherType() . . . 212
10.43.2.4 GetInitialized() . . . 212
10.43.2.5 ParseCertificateFile() . . . 212
10.43.2.6 ParseKeyFile() . . . 213
10.43.2.7 SetCipherType() . . . 213
10.43.2.8 SetPassword() . . . 213
10.44gdcm::network::CEchoRQ Class Reference . . . 213
10.44.1 Detailed Description . . . 214
10.44.2 Member Function Documentation . . . 214
10.44.2.1 ConstructPDV() . . . 214
10.44.3 Member Data Documentation . . . 215
10.44.3.1 AffectedSOPClassUID . . . 215
10.44.3.2 MessageID . . . 215
10.45gdcm::network::CEchoRSP Class Reference . . . 215
10.45.1 Detailed Description . . . 216
10.45.2 Member Function Documentation . . . 216

10.45.2.1 ConstructPDVByDataSet() . . . 216
10.46gdcm::network::CFind Class Reference . . . 216
10.46.1 Detailed Description . . . 217
10.47gdcm::network::CFindCancelRQ Class Reference . . . 217
10.47.1 Detailed Description . . . 218
10.47.2 Member Function Documentation . . . 218
10.47.2.1 ConstructPDVByDataSet() . . . 218
10.48gdcm::network::CFindRQ Class Reference . . . 218
10.48.1 Detailed Description . . . 219
10.48.2 Member Function Documentation . . . 219
10.48.2.1 ConstructPDV() . . . 219
10.49gdcm::network::CFindRSP Class Reference . . . 220
10.49.1 Detailed Description . . . 220
10.49.2 Member Function Documentation . . . 221
10.49.2.1 ConstructPDVByDataSet() . . . 221
10.50gdcm::network::CMoveCancelRq Class Reference . . . 221
10.50.1 Member Function Documentation . . . 222
10.50.1.1 ConstructPDVByDataSet() . . . 222
10.51gdcm::network::CMoveRQ Class Reference . . . 222
10.51.1 Detailed Description . . . 223
10.51.2 Member Function Documentation . . . 223
10.51.2.1 ConstructPDV() . . . 223
10.52gdcm::network::CMoveRSP Class Reference . . . 224
10.52.1 Detailed Description . . . 224
10.52.2 Member Function Documentation . . . 225
10.52.2.1 ConstructPDVByDataSet() . . . 225
10.53gdcm::Codec Class Reference . . . 225
10.53.1 Detailed Description . . . 226

10.54gdcm::Coder Class Reference . . . 226
10.54.1 Detailed Description . . . 227
10.54.2 Constructor & Destructor Documentation . . . 227
10.54.2.1 ~Coder() . . . 227
10.54.3 Member Function Documentation . . . 227
10.54.3.1 CanCode() . . . 227
10.54.3.2 Code() . . . 227
10.54.3.3 InternalCode() . . . 227
10.55gdcm::CodeString Class Reference . . . 228
10.55.1 Detailed Description . . . 229
10.55.2 Member Typedef Documentation . . . 229
10.55.2.1 const_iterator . . . 229
10.55.2.2 const_reference . . . 229
10.55.2.3 const_reverse_iterator . . . 229
10.55.2.4 difference_type . . . 229
10.55.2.5 iterator . . . 229
10.55.2.6 pointer . . . 230
10.55.2.7 reference . . . 230
10.55.2.8 reverse_iterator . . . 230
10.55.2.9 size_type . . . 230
10.55.2.10value_type . . . 230
10.55.3 Constructor & Destructor Documentation . . . 230
10.55.3.1 CodeString() [1/4] . . . 230
10.55.3.2 CodeString() [2/4] . . . 230
10.55.3.3 CodeString() [3/4] . . . 230
10.55.3.4 CodeString() [4/4] . . . 231
10.55.4 Member Function Documentation . . . 231
10.55.4.1 GetAsString() . . . 231

10.55.4.2 IsValid() . . . 231
10.55.4.3 Size() . . . 231
10.55.4.4 TrimInternal() . . . 231
10.55.5 Friends And Related Function Documentation . . . 231
10.55.5.1 operator"!= . . . 231
10.55.5.2 operator<< . . . 232
10.55.5.3 operator== . . . 232
10.56gdcm::Command Class Reference . . . 232
10.56.1 Detailed Description . . . 233
10.56.2 Constructor & Destructor Documentation . . . 233
10.56.2.1 Command() . . . 233
10.56.2.2 ~Command() . . . 234
10.56.3 Member Function Documentation . . . 234
10.56.3.1 Execute() [1/2] . . . 234
10.56.3.2 Execute() [2/2] . . . 234
10.57gdcm::CommandDataSet Class Reference . . . 234
10.57.1 Detailed Description . . . 235
10.57.2 Constructor & Destructor Documentation . . . 235
10.57.2.1 CommandDataSet() . . . 235
10.57.2.2 ~CommandDataSet() . . . 236
10.57.3 Member Function Documentation . . . 236
10.57.3.1 Insert() . . . 236
10.57.3.2 Read() . . . 236
10.57.3.3 Replace() . . . 236
10.57.3.4 Write() . . . 236
10.57.4 Friends And Related Function Documentation . . . 236
10.57.4.1 operator<< . . . 236
10.58gdcm::network::CompositeMessageFactory Class Reference . . . 237

10.58.1 Detailed Description . . . 237
10.58.2 Member Function Documentation . . . 237
10.58.2.1 ConstructCEchoRQ() . . . 237
10.58.2.2 ConstructCFindRQ() . . . 237
10.58.2.3 ConstructCMoveRQ() . . . 238
10.58.2.4 ConstructCStoreRQ() . . . 238
10.58.2.5 ConstructCStoreRSP() . . . 238
10.59gdcm::CompositeNetworkFunctions Class Reference . . . 238
10.59.1 Detailed Description . . . 239
10.59.2 Member Typedef Documentation . . . 239
10.59.2.1 KeyValuePairArrayType . . . 239
10.59.2.2 KeyValuePairType . . . 239
10.59.3 Member Function Documentation . . . 239
10.59.3.1 CEcho() . . . 239
10.59.3.2 CFind() . . . 240
10.59.3.3 CMove() . . . 240
10.59.3.4 ConstructQuery() [1/2] . . . 241
10.59.3.5 ConstructQuery() [2/2] . . . 241
10.59.3.6 CStore() . . . 242
10.60gdcm::ConstCharWrapper Class Reference . . . 242
10.60.1 Detailed Description . . . 243
10.60.2 Constructor & Destructor Documentation . . . 243
10.60.2.1 ConstCharWrapper() . . . 243
10.60.3 Member Function Documentation . . . 243
10.60.3.1 operator const char *() . . . 243
10.61gdcm::CP246ExplicitDataElement Class Reference . . . 243
10.61.1 Detailed Description . . . 244
10.61.2 Member Function Documentation . . . 244

10.61.2.1 GetLength() . . . 244
10.61.2.2 Read() . . . 245
10.61.2.3 ReadPreValue() . . . 245
10.61.2.4 ReadValue() . . . 245
10.61.2.5 ReadWithLength() . . . 245
10.62gdcm::CryptoFactory Class Reference . . . 245
10.62.1 Detailed Description . . . 246
10.62.2 Member Enumeration Documentation . . . 246
10.62.2.1 CryptoLib . . . 246
10.62.3 Constructor & Destructor Documentation . . . 247
10.62.3.1 CryptoFactory() [1/2] . . . 247
10.62.3.2 CryptoFactory() [2/2] . . . 247
10.62.3.3 ∼CryptoFactory() . . . 247
10.62.4 Member Function Documentation . . . 247
10.62.4.1 CreateCMSProvider() . . . 247
10.62.4.2 GetFactoryInstance() . . . 247
10.63gdcm::CryptographicMessageSyntax Class Reference . . . 248
10.63.1 Member Enumeration Documentation . . . 248
10.63.1.1 CipherTypes . . . 248
10.63.2 Constructor & Destructor Documentation . . . 249
10.63.2.1 CryptographicMessageSyntax() . . . 249
10.63.2.2 ∼CryptographicMessageSyntax() . . . 249
10.63.3 Member Function Documentation . . . 249
10.63.3.1 Decrypt() . . . 249
10.63.3.2 Encrypt() . . . 249
10.63.3.3 GetCipherType() . . . 250
10.63.3.4 ParseCertificateFile() . . . 250
10.63.3.5 ParseKeyFile() . . . 250

10.63.3.6 SetCipherType() . . . 250
10.63.3.7 SetPassword() . . . 250
10.64 gdcm::CSAElement Class Reference . . . 251
10.64.1 Detailed Description . . . 252
10.64.2 Member Typedef Documentation . . . 252
10.64.2.1 DataPtr . . . 252
10.64.3 Constructor & Destructor Documentation . . . 253
10.64.3.1 CSAElement() [1/2] . . . 253
10.64.3.2 CSAElement() [2/2] . . . 253
10.64.4 Member Function Documentation . . . 253
10.64.4.1 GetByteValue() . . . 253
10.64.4.2 GetKey() . . . 253
10.64.4.3 GetName() . . . 253
10.64.4.4 GetNoOfItems() . . . 254
10.64.4.5 GetSyngoDT() . . . 254
10.64.4.6 GetValue() [1/2] . . . 254
10.64.4.7 GetValue() [2/2] . . . 254
10.64.4.8 GetVM() . . . 254
10.64.4.9 GetVR() . . . 254
10.64.4.10 IsEmpty() . . . 255
10.64.4.11 operator< () . . . 255
10.64.4.12 operator=() . . . 255
10.64.4.13 operator==() . . . 255
10.64.4.14 SetByteValue() . . . 255
10.64.4.15 SetKey() . . . 255
10.64.4.16 SetName() . . . 256
10.64.4.17 SetNoOfItems() . . . 256
10.64.4.18 SetSyngoDT() . . . 256

10.64.4.19 SetValue() . . . 256
10.64.4.20 SetVM() . . . 256
10.64.4.21 SetVR() . . . 256
10.64.5 Friends And Related Function Documentation . . . 256
10.64.5.1 operator<< . . . 256
10.64.6 Member Data Documentation . . . 257
10.64.6.1 DataField . . . 257
10.64.6.2 KeyField . . . 257
10.64.6.3 NameField . . . 257
10.64.6.4 NoOfItemsField . . . 257
10.64.6.5 SyngoDTField . . . 257
10.64.6.6 ValueMultiplicityField . . . 257
10.64.6.7 VRField . . . 258
10.65 gdcm::CSAHeader Class Reference . . . 258
10.65.1 Detailed Description . . . 259
10.65.2 Member Enumeration Documentation . . . 259
10.65.2.1 CSAHeaderType . . . 259
10.65.3 Constructor & Destructor Documentation . . . 260
10.65.3.1 CSAHeader() . . . 260
10.65.3.2 ~CSAHeader() . . . 260
10.65.4 Member Function Documentation . . . 260
10.65.4.1 FindCSAElementByName() . . . 260
10.65.4.2 GetCSADataInfo() . . . 260
10.65.4.3 GetCSAEEnd() . . . 261
10.65.4.4 GetCSAElementByName() . . . 261
10.65.4.5 GetCSAImageHeaderInfoTag() . . . 261
10.65.4.6 GetCSASeriesHeaderInfoTag() . . . 261
10.65.4.7 GetDataSet() . . . 261

10.65.4.8 GetFormat() . . . 262
10.65.4.9 GetInterfile() . . . 262
10.65.4.10 LoadFromDataElement() . . . 262
10.65.4.11 Print() . . . 262
10.65.4.12 Read() . . . 262
10.65.4.13 Write() . . . 263
10.65.5 Friends And Related Function Documentation . . . 263
10.65.5.1 operator<< . . . 263
10.66 gdcm::CSAHeaderDict Class Reference . . . 263
10.66.1 Detailed Description . . . 264
10.66.2 Member Typedef Documentation . . . 264
10.66.2.1 ConstIterator . . . 264
10.66.2.2 Iterator . . . 264
10.66.2.3 MapCSAHeaderDictEntry . . . 264
10.66.3 Constructor & Destructor Documentation . . . 264
10.66.3.1 CSAHeaderDict() . . . 264
10.66.4 Member Function Documentation . . . 264
10.66.4.1 AddCSAHeaderDictEntry() . . . 264
10.66.4.2 Begin() . . . 265
10.66.4.3 End() . . . 265
10.66.4.4 GetCSAHeaderDictEntry() . . . 265
10.66.4.5 IsEmpty() . . . 265
10.66.4.6 LoadDefault() . . . 265
10.66.5 Friends And Related Function Documentation . . . 265
10.66.5.1 Dicts . . . 265
10.66.5.2 operator<< . . . 265
10.67 gdcm::CSAHeaderDictEntry Class Reference . . . 266
10.67.1 Detailed Description . . . 266

10.67.2 Constructor & Destructor Documentation . . . 267
10.67.2.1 CSAHeaderDictEntry() . . . 267
10.67.3 Member Function Documentation . . . 267
10.67.3.1 GetDescription() . . . 267
10.67.3.2 GetName() . . . 267
10.67.3.3 GetVM() . . . 267
10.67.3.4 GetVR() . . . 267
10.67.3.5 operator<() . . . 267
10.67.3.6 SetDescription() . . . 268
10.67.3.7 SetName() . . . 268
10.67.3.8 SetVM() . . . 268
10.67.3.9 SetVR() . . . 268
10.67.4 Friends And Related Function Documentation . . . 268
10.67.4.1 operator<< . . . 268
10.68gdcm::CSAHeaderDictException Class Reference . . . 269
10.69gdcm::network::CStoreRQ Class Reference . . . 269
10.69.1 Detailed Description . . . 270
10.69.2 Member Function Documentation . . . 271
10.69.2.1 ConstructPDV() . . . 271
10.70gdcm::network::CStoreRSP Class Reference . . . 271
10.70.1 Detailed Description . . . 272
10.70.2 Member Function Documentation . . . 272
10.70.2.1 ConstructPDV() . . . 272
10.71gdcm::Curve Class Reference . . . 272
10.71.1 Detailed Description . . . 274
10.71.2 Constructor & Destructor Documentation . . . 274
10.71.2.1 Curve() [1/2] . . . 274
10.71.2.2 ~Curve() . . . 274

10.71.2.3 Curve() [2/2] . . . 274
10.71.3 Member Function Documentation . . . 274
10.71.3.1 Decode() . . . 274
10.71.3.2 GetAsPoints() . . . 275
10.71.3.3 GetCurveDataDescriptor() . . . 275
10.71.3.4 GetDataValueRepresentation() . . . 275
10.71.3.5 GetDimensions() . . . 275
10.71.3.6 GetGroup() . . . 275
10.71.3.7 GetNumberOfCurves() . . . 275
10.71.3.8 GetNumberOfPoints() . . . 275
10.71.3.9 GetTypeOfData() . . . 275
10.71.3.10 GetTypeOfDataDescription() . . . 275
10.71.3.11 IsEmpty() . . . 276
10.71.3.12 Print() . . . 276
10.71.3.13 SetCoordinateStartValue() . . . 276
10.71.3.14 SetCoordinateStepValue() . . . 276
10.71.3.15 SetCurve() . . . 276
10.71.3.16 SetCurveDataDescriptor() . . . 276
10.71.3.17 SetCurveDescription() . . . 276
10.71.3.18 SetDataValueRepresentation() . . . 276
10.71.3.19 SetDimensions() . . . 277
10.71.3.20 SetGroup() . . . 277
10.71.3.21 SetNumberOfPoints() . . . 277
10.71.3.22 SetTypeOfData() . . . 277
10.71.3.23 Update() . . . 277
10.72 gdcm::DataElement Class Reference . . . 277
10.72.1 Detailed Description . . . 280
10.72.2 Member Typedef Documentation . . . 281

10.72.2.1 ValuePtr . . . 281
10.72.3 Constructor & Destructor Documentation . . . 281
10.72.3.1 DataElement() [1/2] . . . 281
10.72.3.2 DataElement() [2/2] . . . 281
10.72.4 Member Function Documentation . . . 281
10.72.4.1 Clear() . . . 281
10.72.4.2 Empty() . . . 281
10.72.4.3 GetByteValue() . . . 282
10.72.4.4 GetLength() . . . 282
10.72.4.5 GetSequenceOfFragments() [1/2] . . . 282
10.72.4.6 GetSequenceOfFragments() [2/2] . . . 282
10.72.4.7 GetTag() [1/2] . . . 283
10.72.4.8 GetTag() [2/2] . . . 283
10.72.4.9 GetValue() [1/2] . . . 283
10.72.4.10 GetValue() [2/2] . . . 283
10.72.4.11 GetValueAsSQ() . . . 284
10.72.4.12 GetVL() [1/2] . . . 284
10.72.4.13 GetVL() [2/2] . . . 284
10.72.4.14 GetVR() . . . 284
10.72.4.15 IsEmpty() . . . 285
10.72.4.16 IsUndefinedLength() . . . 285
10.72.4.17 operator< () . . . 285
10.72.4.18 operator=() . . . 285
10.72.4.19 operator==() . . . 285
10.72.4.20 Read() . . . 286
10.72.4.21 ReadOrSkip() . . . 286
10.72.4.22 ReadPreValue() . . . 286
10.72.4.23 ReadValue() . . . 286

10.72.4.24 ReadValueWithLength() . . . 286
10.72.4.25 ReadWithLength() . . . 286
10.72.4.26 SetByteValue() . . . 287
10.72.4.27 SetTag() . . . 287
10.72.4.28 SetValue() . . . 287
10.72.4.29 SetValueFieldLength() . . . 288
10.72.4.30 SetVL() . . . 288
10.72.4.31 SetVLToUndefined() . . . 288
10.72.4.32 SetVR() . . . 288
10.72.4.33 Write() . . . 289
10.72.5 Friends And Related Function Documentation . . . 289
10.72.5.1 operator<< . . . 289
10.72.6 Member Data Documentation . . . 289
10.72.6.1 TagField . . . 289
10.72.6.2 ValueField . . . 289
10.72.6.3 ValueLengthField . . . 289
10.72.6.4 VRField . . . 289
10.73 gdcm::DataElementException Class Reference . . . 290
10.74 gdcm::DataEvent Class Reference . . . 290
10.74.1 Detailed Description . . . 292
10.74.2 Member Typedef Documentation . . . 292
10.74.2.1 Self . . . 292
10.74.2.2 Superclass . . . 292
10.74.3 Constructor & Destructor Documentation . . . 292
10.74.3.1 DataEvent() [1/2] . . . 292
10.74.3.2 ~DataEvent() . . . 292
10.74.3.3 DataEvent() [2/2] . . . 292
10.74.4 Member Function Documentation . . . 293

10.74.4.1 CheckEvent() . . . 293
10.74.4.2 GetData() . . . 293
10.74.4.3 GetDataLength() . . . 293
10.74.4.4 GetEventName() . . . 293
10.74.4.5 MakeObject() . . . 293
10.74.4.6 SetData() . . . 293
10.75gdcm::DataSet Class Reference . . . 294
10.75.1 Detailed Description . . . 296
10.75.2 Member Typedef Documentation . . . 296
10.75.2.1 ConstIterator . . . 296
10.75.2.2 DataElementSet . . . 296
10.75.2.3 Iterator . . . 297
10.75.2.4 SizeType . . . 297
10.75.3 Member Function Documentation . . . 297
10.75.3.1 Begin() [1/2] . . . 297
10.75.3.2 Begin() [2/2] . . . 297
10.75.3.3 Clear() . . . 297
10.75.3.4 ComputeDataElement() . . . 297
10.75.3.5 ComputeGroupLength() . . . 297
10.75.3.6 End() [1/2] . . . 298
10.75.3.7 End() [2/2] . . . 298
10.75.3.8 FindDataElement() [1/2] . . . 298
10.75.3.9 FindDataElement() [2/2] . . . 298
10.75.3.10 FindNextDataElement() . . . 298
10.75.3.11 GetDataElement() [1/2] . . . 299
10.75.3.12 GetDataElement() [2/2] . . . 299
10.75.3.13 GetDEEnd() . . . 299
10.75.3.14 GetDES() [1/2] . . . 299

10.75.3.15 GetDES() [2/2] . . . 300
10.75.3.16 GetLength() . . . 300
10.75.3.17 GetMediaStorage() . . . 300
10.75.3.18 GetPrivateCreator() . . . 300
10.75.3.19 Insert() . . . 300
10.75.3.20 InsertDataElement() . . . 301
10.75.3.21 IsEmpty() . . . 301
10.75.3.22 operator()() . . . 301
10.75.3.23 operator=() . . . 301
10.75.3.24 operator[]() . . . 301
10.75.3.25 Print() . . . 301
10.75.3.26 Read() . . . 301
10.75.3.27 ReadNested() . . . 302
10.75.3.28 ReadSelectedPrivateTags() . . . 302
10.75.3.29 ReadSelectedPrivateTagsWithLength() . . . 302
10.75.3.30 ReadSelectedTags() . . . 302
10.75.3.31 ReadSelectedTagsWithLength() . . . 302
10.75.3.32 ReadUpToTag() . . . 302
10.75.3.33 ReadUpToTagWithLength() . . . 303
10.75.3.34 ReadWithLength() . . . 303
10.75.3.35 Remove() . . . 303
10.75.3.36 Replace() . . . 303
10.75.3.37 ReplaceEmpty() . . . 304
10.75.3.38 Size() . . . 304
10.75.3.39 Write() . . . 304
10.75.4 Friends And Related Function Documentation . . . 304
10.75.4.1 CSAHeader . . . 304
10.75.4.2 operator<< . . . 304

10.76gdcm::DataSetEvent Class Reference . . . 305
10.76.1 Detailed Description . . . 306
10.76.2 Member Typedef Documentation . . . 306
10.76.2.1 Self . . . 306
10.76.2.2 Superclass . . . 306
10.76.3 Constructor & Destructor Documentation . . . 306
10.76.3.1 DataSetEvent() [1/2] . . . 306
10.76.3.2 ∼DataSetEvent() . . . 306
10.76.3.3 DataSetEvent() [2/2] . . . 307
10.76.4 Member Function Documentation . . . 307
10.76.4.1 CheckEvent() . . . 307
10.76.4.2 GetDataSet() . . . 307
10.76.4.3 GetEventName() . . . 307
10.76.4.4 MakeObject() . . . 307
10.77gdcm::DataSetHelper Class Reference . . . 307
10.77.1 Detailed Description . . . 308
10.77.2 Member Function Documentation . . . 308
10.77.2.1 ComputeVR() . . . 308
10.78gdcm::Decoder Class Reference . . . 308
10.78.1 Detailed Description . . . 309
10.78.2 Constructor & Destructor Documentation . . . 309
10.78.2.1 ∼Decoder() . . . 309
10.78.3 Member Function Documentation . . . 309
10.78.3.1 CanDecode() . . . 309
10.78.3.2 Decode() . . . 309
10.78.3.3 DecodeByStreams() . . . 310
10.79gdcm::DefinedTerms Class Reference . . . 310
10.79.1 Detailed Description . . . 310

10.79.2 Constructor & Destructor Documentation . . . 310
10.79.2.1 DefinedTerms() . . . 310
10.80 gdcm::Defs Class Reference . . . 311
10.80.1 Detailed Description . . . 311
10.80.2 Constructor & Destructor Documentation . . . 312
10.80.2.1 Defs() . . . 312
10.80.2.2 ~Defs() . . . 312
10.80.3 Member Function Documentation . . . 312
10.80.3.1 GetIODFromFile() . . . 312
10.80.3.2 GetIODNameFromMediaStorage() . . . 312
10.80.3.3 GetIODs() [1/2] . . . 312
10.80.3.4 GetIODs() [2/2] . . . 312
10.80.3.5 GetMacros() [1/2] . . . 313
10.80.3.6 GetMacros() [2/2] . . . 313
10.80.3.7 GetModules() [1/2] . . . 313
10.80.3.8 GetModules() [2/2] . . . 313
10.80.3.9 GetTypeFromTag() . . . 313
10.80.3.10 IsEmpty() . . . 313
10.80.3.11 LoadDefaults() . . . 313
10.80.3.12 LoadFromFile() . . . 314
10.80.3.13 Verify() [1/2] . . . 314
10.80.3.14 Verify() [2/2] . . . 314
10.80.4 Friends And Related Function Documentation . . . 314
10.80.4.1 Global . . . 314
10.81 gdcm::DeltaEncodingCodec Class Reference . . . 314
10.81.1 Detailed Description . . . 316
10.81.2 Constructor & Destructor Documentation . . . 316
10.81.2.1 DeltaEncodingCodec() . . . 316

10.81.2.2 ~DeltaEncodingCodec() . . . 316
10.81.3 Member Function Documentation . . . 316
10.81.3.1 CanDecode() . . . 316
10.81.3.2 Decode() [1/2] . . . 316
10.81.3.3 Decode() [2/2] . . . 316
10.82gdcm::DICOMDIR Class Reference . . . 317
10.82.1 Detailed Description . . . 317
10.82.2 Constructor & Destructor Documentation . . . 317
10.82.2.1 DICOMDIR() [1/2] . . . 317
10.82.2.2 DICOMDIR() [2/2] . . . 317
10.83gdcm::DICOMDIRGenerator Class Reference . . . 317
10.83.1 Detailed Description . . . 318
10.83.2 Member Typedef Documentation . . . 319
10.83.2.1 FilenamesType . . . 319
10.83.2.2 FilenameType . . . 319
10.83.3 Constructor & Destructor Documentation . . . 319
10.83.3.1 DICOMDIRGenerator() . . . 319
10.83.3.2 ~DICOMDIRGenerator() . . . 319
10.83.4 Member Function Documentation . . . 319
10.83.4.1 AddImageDirectoryRecord() . . . 319
10.83.4.2 AddPatientDirectoryRecord() . . . 319
10.83.4.3 AddSeriesDirectoryRecord() . . . 319
10.83.4.4 AddStudyDirectoryRecord() . . . 319
10.83.4.5 Generate() . . . 320
10.83.4.6 GetFile() . . . 320
10.83.4.7 GetScanner() . . . 320
10.83.4.8 SetDescriptor() . . . 320
10.83.4.9 SetFile() . . . 320

10.83.4.10SetFilenames() . . . 320
10.83.4.11SetRootDirectory() . . . 321
10.84gdcm::Dict Class Reference . . . 321
10.84.1 Detailed Description . . . 322
10.84.2 Member Typedef Documentation . . . 322
10.84.2.1 ConstIterator . . . 322
10.84.2.2 Iterator . . . 322
10.84.2.3 MapDictEntry . . . 322
10.84.3 Constructor & Destructor Documentation . . . 322
10.84.3.1 Dict() . . . 322
10.84.4 Member Function Documentation . . . 322
10.84.4.1 AddDictEntry() . . . 322
10.84.4.2 Begin() . . . 323
10.84.4.3 End() . . . 323
10.84.4.4 GetDictEntry() . . . 323
10.84.4.5 GetDictEntryByKeyword() . . . 323
10.84.4.6 GetDictEntryByName() . . . 323
10.84.4.7 GetKeywordFromTag() . . . 324
10.84.4.8 IsEmpty() . . . 324
10.84.4.9 LoadDefault() . . . 324
10.84.5 Friends And Related Function Documentation . . . 324
10.84.5.1 Dicts . . . 324
10.84.5.2 operator<< . . . 324
10.85gdcm::DictConverter Class Reference . . . 324
10.85.1 Detailed Description . . . 325
10.85.2 Member Enumeration Documentation . . . 325
10.85.2.1 OutputTypes . . . 325
10.85.3 Constructor & Destructor Documentation . . . 326

10.85.3.1 DictConverter() . . . 326
10.85.3.2 ~DictConverter() . . . 326
10.85.4 Member Function Documentation . . . 326
10.85.4.1 AddGroupLength() . . . 326
10.85.4.2 Convert() . . . 326
10.85.4.3 ConvertToCXX() . . . 326
10.85.4.4 ConvertToXML() . . . 326
10.85.4.5 GetDictName() . . . 327
10.85.4.6 GetInputFilename() . . . 327
10.85.4.7 GetOutputFilename() . . . 327
10.85.4.8 GetOutputType() . . . 327
10.85.4.9 Readuint16() . . . 327
10.85.4.10 ReadVM() . . . 327
10.85.4.11 ReadVR() . . . 327
10.85.4.12 SetDictName() . . . 327
10.85.4.13 SetInputFileName() . . . 328
10.85.4.14 SetOutputFileName() . . . 328
10.85.4.15 SetOutputType() . . . 328
10.85.4.16 WriteFooter() . . . 328
10.85.4.17 WriteHeader() . . . 328
10.86 gdcm::DictEntry Class Reference . . . 328
10.86.1 Detailed Description . . . 329
10.86.2 Constructor & Destructor Documentation . . . 330
10.86.2.1 DictEntry() . . . 330
10.86.3 Member Function Documentation . . . 330
10.86.3.1 GetKeyword() . . . 330
10.86.3.2 GetName() . . . 330
10.86.3.3 GetRetired() . . . 330

10.86.3.4 GetVM() . . . 330
10.86.3.5 GetVR() . . . 331
10.86.3.6 IsUnique() . . . 331
10.86.3.7 SetElementXX() . . . 331
10.86.3.8 SetGroupXX() . . . 331
10.86.3.9 SetKeyword() . . . 331
10.86.3.10 SetName() . . . 331
10.86.3.11 SetRetired() . . . 332
10.86.3.12 SetVM() . . . 332
10.86.3.13 SetVR() . . . 332
10.86.4 Friends And Related Function Documentation . . . 332
10.86.4.1 Dict . . . 332
10.86.4.2 operator<< . . . 332
10.87 gdcm::DictPrinter Class Reference . . . 333
10.87.1 Detailed Description . . . 334
10.87.2 Constructor & Destructor Documentation . . . 334
10.87.2.1 DictPrinter() . . . 334
10.87.2.2 ~DictPrinter() . . . 334
10.87.3 Member Function Documentation . . . 334
10.87.3.1 Print() . . . 334
10.87.3.2 PrintDataElement2() . . . 334
10.87.3.3 PrintDataSet2() . . . 335
10.88 gdcm::Dicts Class Reference . . . 335
10.88.1 Detailed Description . . . 336
10.88.2 Member Enumeration Documentation . . . 336
10.88.2.1 ConstructorType . . . 336
10.88.3 Constructor & Destructor Documentation . . . 336
10.88.3.1 Dicts() . . . 336

10.88.3.2 ~Dicts() . . . 336
10.88.4 Member Function Documentation . . . 336
10.88.4.1 GetConstructorString() . . . 336
10.88.4.2 GetCSAHeaderDict() . . . 337
10.88.4.3 GetDictEntry() [1/2] . . . 337
10.88.4.4 GetDictEntry() [2/2] . . . 337
10.88.4.5 GetPrivateDict() [1/2] . . . 337
10.88.4.6 GetPrivateDict() [2/2] . . . 337
10.88.4.7 GetPublicDict() . . . 338
10.88.4.8 IsEmpty() . . . 338
10.88.4.9 LoadDefaults() . . . 338
10.88.5 Friends And Related Function Documentation . . . 338
10.88.5.1 Global . . . 338
10.88.5.2 operator<< . . . 338
10.89gdcm::network::DIMSE Class Reference . . . 338
10.89.1 Detailed Description . . . 339
10.89.2 Member Enumeration Documentation . . . 339
10.89.2.1 CommandTypes . . . 339
10.90gdcm::DirectionCosines Class Reference . . . 340
10.90.1 Detailed Description . . . 341
10.90.2 Constructor & Destructor Documentation . . . 341
10.90.2.1 DirectionCosines() [1/2] . . . 341
10.90.2.2 DirectionCosines() [2/2] . . . 341
10.90.2.3 ~DirectionCosines() . . . 341
10.90.3 Member Function Documentation . . . 341
10.90.3.1 ComputeDistAlongNormal() . . . 341
10.90.3.2 Cross() . . . 342
10.90.3.3 CrossDot() . . . 342

10.90.3.4 Dot() . . . 342
10.90.3.5 IsValid() . . . 342
10.90.3.6 Normalize() . . . 342
10.90.3.7 operator const double *() . . . 342
10.90.3.8 Print() . . . 343
10.90.3.9 SetFromString() . . . 343
10.91 gdcm::Directory Class Reference . . . 343
10.91.1 Detailed Description . . . 344
10.91.2 Member Typedef Documentation . . . 344
10.91.2.1 FilenamesType . . . 344
10.91.2.2 FilenameType . . . 344
10.91.3 Constructor & Destructor Documentation . . . 344
10.91.3.1 Directory() . . . 344
10.91.3.2 ~Directory() . . . 345
10.91.4 Member Function Documentation . . . 345
10.91.4.1 Explore() . . . 345
10.91.4.2 GetDirectories() . . . 345
10.91.4.3 GetFilenames() . . . 345
10.91.4.4 GetToplevel() . . . 345
10.91.4.5 Load() . . . 346
10.91.4.6 Print() . . . 346
10.91.5 Friends And Related Function Documentation . . . 346
10.91.5.1 operator<< . . . 346
10.92 gdcm::DirectoryHelper Class Reference . . . 347
10.92.1 Detailed Description . . . 347
10.92.2 Member Function Documentation . . . 347
10.92.2.1 GetCTImageSeriesUIDs() . . . 347
10.92.2.2 GetFilenamesFromSeriesUIDs() . . . 347

10.92.2.3 GetFrameOfReference() . . . 348
10.92.2.4 GetMRImageSeriesUIDs() . . . 348
10.92.2.5 GetRTStructSeriesUIDs() . . . 348
10.92.2.6 GetSeriesUIDsBySOPClassUID() . . . 348
10.92.2.7 GetSOPClassUID() . . . 348
10.92.2.8 GetStringValueFromTag() . . . 348
10.92.2.9 LoadImageFromFiles() . . . 348
10.92.2.10 RetrieveSOPInstanceUIDFromIndex() . . . 349
10.92.2.11 RetrieveSOPInstanceUIDFromZPosition() . . . 349
10.93 gdcm::DummyValueGenerator Class Reference . . . 349
10.93.1 Detailed Description . . . 349
10.93.2 Member Function Documentation . . . 349
10.93.2.1 Generate() . . . 349
10.94 gdcm::Dumper Class Reference . . . 350
10.94.1 Detailed Description . . . 351
10.94.2 Constructor & Destructor Documentation . . . 351
10.94.2.1 Dumper() . . . 351
10.94.2.2 ∼Dumper() . . . 351
10.95 gdcm::Element< TVR, TVM > Class Template Reference . . . 351
10.95.1 Detailed Description . . . 353
10.95.2 Member Typedef Documentation . . . 353
10.95.2.1 Type . . . 353
10.95.3 Member Function Documentation . . . 353
10.95.3.1 GetAsDataElement() . . . 353
10.95.3.2 GetLength() . . . 353
10.95.3.3 GetValue() [1/2] . . . 354
10.95.3.4 GetValue() [2/2] . . . 354
10.95.3.5 GetValues() . . . 354

10.95.3.6 GetVM() . . . 354
10.95.3.7 GetVR() . . . 354
10.95.3.8 operator[]() . . . 354
10.95.3.9 Print() . . . 354
10.95.3.10 Read() . . . 355
10.95.3.11 Set() . . . 355
10.95.3.12 SetFromDataElement() . . . 355
10.95.3.13 SetNoSwap() . . . 355
10.95.3.14 SetValue() . . . 355
10.95.3.15 Write() . . . 355
10.95.4 Member Data Documentation . . . 355
10.95.4.1 Internal . . . 355
10.96 gdcm::Element< TVR, VM::VM1_2 > Class Template Reference . . . 356
10.96.1 Member Typedef Documentation . . . 357
10.96.1.1 Parent . . . 357
10.96.2 Member Function Documentation . . . 357
10.96.2.1 SetLength() . . . 357
10.97 gdcm::Element< TVR, VM::VM1_n > Class Template Reference . . . 357
10.97.1 Member Typedef Documentation . . . 358
10.97.1.1 Type . . . 358
10.97.2 Constructor & Destructor Documentation . . . 358
10.97.2.1 Element() [1/2] . . . 358
10.97.2.2 ~Element() . . . 359
10.97.2.3 Element() [2/2] . . . 359
10.97.3 Member Function Documentation . . . 359
10.97.3.1 GetAsDataElement() . . . 359
10.97.3.2 GetLength() . . . 359
10.97.3.3 GetValue() [1/2] . . . 359

10.97.3.4 GetValue() [2/2] . . . 359
10.97.3.5 GetVM() . . . 359
10.97.3.6 GetVR() . . . 360
10.97.3.7 operator=() . . . 360
10.97.3.8 operator[]() . . . 360
10.97.3.9 Print() . . . 360
10.97.3.10 Read() . . . 360
10.97.3.11 Set() . . . 360
10.97.3.12 SetArray() . . . 360
10.97.3.13 SetFromDataElement() . . . 361
10.97.3.14 SetLength() . . . 361
10.97.3.15 SetNoSwap() . . . 361
10.97.3.16 SetValue() . . . 361
10.97.3.17 Write() . . . 361
10.97.3.18 WriteASCII() . . . 361
10.98 gdcm::Element< TVR, VM::VM2_2n > Class Template Reference . . . 362
10.98.1 Member Typedef Documentation . . . 363
10.98.1.1 Parent . . . 363
10.98.2 Member Function Documentation . . . 363
10.98.2.1 SetLength() . . . 363
10.99 gdcm::Element< TVR, VM::VM2_n > Class Template Reference . . . 363
10.99.1 Member Typedef Documentation . . . 365
10.99.1.1 Parent . . . 365
10.99.2 Member Function Documentation . . . 365
10.99.2.1 SetLength() . . . 365
10.100 gdcm::Element< TVR, VM::VM3_3n > Class Template Reference . . . 365
10.100.1 Member Typedef Documentation . . . 366
10.100.1.1 Parent . . . 366

10.100.2 Member Function Documentation . . . 366
10.100.2.1 SetLength() . . . 366
10.101 gdcm::Element< TVR, VM::VM3_n > Class Template Reference . . . 367
10.101.1 Member Typedef Documentation . . . 368
10.101.1.1 Parent . . . 368
10.101.2 Member Function Documentation . . . 368
10.101.2.1 SetLength() . . . 368
10.102 gdcm::Element< VR::AS, VM::VM5 > Class Template Reference . . . 368
10.102.1 Member Function Documentation . . . 369
10.102.1.1 GetLength() . . . 369
10.102.1.2 Print() . . . 369
10.102.2 Member Data Documentation . . . 369
10.102.2.1 Internal . . . 369
10.103 gdcm::Element< VR::OB, VM::VM1 > Class Template Reference . . . 369
10.104 gdcm::Element< VR::OW, VM::VM1 > Class Template Reference . . . 370
10.105 gdcm::ElementDisableCombinations< TVR, TVM > Class Template Reference . . . 372
10.105.1 Detailed Description . . . 373
10.106 gdcm::ElementDisableCombinations< VR::OB, VM::VM1_n > Class Template Reference . . . 373
10.107 gdcm::ElementDisableCombinations< VR::OW, VM::VM1_n > Class Template Reference . . . 373
10.108 gdcm::EncapsulatedDocument Class Reference . . . 373
10.108.1 Detailed Description . . . 374
10.108.2 Constructor & Destructor Documentation . . . 374
10.108.2.1 EncapsulatedDocument() . . . 374
10.109 gdcm::EncodingImplementation< T > Class Template Reference . . . 374
10.109.1 Detailed Description . . . 374
10.110 gdcm::EncodingImplementation< VR::VRASCII > Class Template Reference . . . 374
10.110.1 Member Function Documentation . . . 375
10.110.1.1 Read() . . . 375

10.110.1.2 ReadComputeLength() . . . 375
10.110.1.3 ReadNoSwap() . . . 375
10.110.1.4 Write() [1/3] . . . 376
10.110.1.5 Write() [2/3] . . . 376
10.110.1.6 Write() [3/3] . . . 376
10.111 gdcm::EncodingImplementation< VR::VRBINARY > Class Template Reference . . . 376
10.111.1 Member Function Documentation . . . 377
10.111.1.1 Read() . . . 377
10.111.1.2 ReadComputeLength() . . . 377
10.111.1.3 ReadNoSwap() . . . 377
10.111.1.4 Write() . . . 377
10.112 gdcm::EndEvent Class Reference . . . 378
10.113 gdcm::EnumeratedValues Class Reference . . . 379
10.113.1 Detailed Description . . . 379
10.113.2 Constructor & Destructor Documentation . . . 379
10.113.2.1 EnumeratedValues() . . . 379
10.114 gdcm::Event Class Reference . . . 380
10.114.1 Detailed Description . . . 381
10.114.2 Constructor & Destructor Documentation . . . 381
10.114.2.1 Event() [1/2] . . . 381
10.114.2.2 Event() [2/2] . . . 381
10.114.2.3 ~Event() . . . 381
10.114.3 Member Function Documentation . . . 381
10.114.3.1 CheckEvent() . . . 381
10.114.3.2 GetEventName() . . . 381
10.114.3.3 MakeObject() . . . 382
10.114.3.4 Print() . . . 382
10.115 gdcm::Exception Class Reference . . . 382

10.115.1 Detailed Description . . . 383
10.115.2 Constructor & Destructor Documentation . . . 383
10.115.2.1 Exception() . . . 383
10.115.2.2 ~Exception() . . . 384
10.115.3 Member Function Documentation . . . 384
10.115.3.1 GetDescription() . . . 384
10.115.3.2 what() . . . 384
10.116 gdcm::ExitEvent Class Reference . . . 384
10.117 gdcm::ExplicitDataElement Class Reference . . . 385
10.117.1 Detailed Description . . . 386
10.117.2 Member Function Documentation . . . 387
10.117.2.1 GetLength() . . . 387
10.117.2.2 Read() . . . 387
10.117.2.3 ReadPreValue() . . . 387
10.117.2.4 ReadValue() . . . 387
10.117.2.5 ReadWithLength() . . . 387
10.117.2.6 Write() . . . 387
10.118 gdcm::ExplicitImplicitDataElement Class Reference . . . 388
10.118.1 Detailed Description . . . 389
10.118.2 Member Function Documentation . . . 389
10.118.2.1 GetLength() . . . 389
10.118.2.2 Read() . . . 389
10.118.2.3 ReadPreValue() . . . 389
10.118.2.4 ReadValue() . . . 389
10.118.2.5 ReadWithLength() . . . 390
10.119 gdcm::Fiducials Class Reference . . . 390
10.119.1 Detailed Description . . . 390
10.119.2 Constructor & Destructor Documentation . . . 390

10.119.2.1 Fiducials() . . . 390
10.120 gdcm::File Class Reference . . . 391
10.120.1 Detailed Description . . . 392
10.120.2 Constructor & Destructor Documentation . . . 392
10.120.2.1 File() . . . 392
10.120.2.2 ~File() . . . 393
10.120.3 Member Function Documentation . . . 393
10.120.3.1 GetDataSet() [1/2] . . . 393
10.120.3.2 GetDataSet() [2/2] . . . 393
10.120.3.3 GetHeader() [1/2] . . . 393
10.120.3.4 GetHeader() [2/2] . . . 394
10.120.3.5 Read() . . . 394
10.120.3.6 SetDataSet() . . . 394
10.120.3.7 SetHeader() . . . 394
10.120.3.8 Write() . . . 394
10.120.4 Friends And Related Function Documentation . . . 394
10.120.4.1 operator<< . . . 394
10.121 gdcm::FileAnonymizer Class Reference . . . 395
10.121.1 Detailed Description . . . 396
10.121.2 Constructor & Destructor Documentation . . . 397
10.121.2.1 FileAnonymizer() . . . 397
10.121.2.2 ~FileAnonymizer() . . . 397
10.121.3 Member Function Documentation . . . 397
10.121.3.1 Empty() . . . 397
10.121.3.2 Remove() . . . 397
10.121.3.3 Replace() [1/2] . . . 397
10.121.3.4 Replace() [2/2] . . . 398
10.121.3.5 SetInputFileName() . . . 398

10.121.3.6 SetOutputFileName() . . . 398
10.121.3.7 Write() . . . 398
10.122 gdcm::FileChangeTransferSyntax Class Reference . . . 399
10.122.1 Detailed Description . . . 400
10.122.2 Constructor & Destructor Documentation . . . 400
10.122.2.1 FileChangeTransferSyntax() . . . 400
10.122.2.2 ~FileChangeTransferSyntax() . . . 400
10.122.3 Member Function Documentation . . . 401
10.122.3.1 Change() . . . 401
10.122.3.2 GetCodec() . . . 401
10.122.3.3 New() . . . 401
10.122.3.4 SetInputFileName() . . . 401
10.122.3.5 SetOutputFileName() . . . 401
10.122.3.6 SetTransferSyntax() . . . 401
10.123 gdcm::FileDecompressLookupTable Class Reference . . . 402
10.123.1 Detailed Description . . . 403
10.123.2 Constructor & Destructor Documentation . . . 403
10.123.2.1 FileDecompressLookupTable() . . . 403
10.123.2.2 ~FileDecompressLookupTable() . . . 403
10.123.3 Member Function Documentation . . . 403
10.123.3.1 Change() . . . 403
10.123.3.2 GetFile() . . . 403
10.123.3.3 GetPixmap() [1/2] . . . 404
10.123.3.4 GetPixmap() [2/2] . . . 404
10.123.3.5 SetFile() . . . 404
10.123.3.6 SetPixmap() . . . 404
10.124 gdcm::FileDerivation Class Reference . . . 404
10.124.1 Detailed Description . . . 405

10.124.2 Constructor & Destructor Documentation . . . 405
10.124.2.1 FileDerivation() . . . 405
10.124.2.2 ~FileDerivation() . . . 405
10.124.3 Member Function Documentation . . . 405
10.124.3.1 AddDerivationDescription() . . . 405
10.124.3.2 AddPurposeOfReferenceCodeSequence() . . . 405
10.124.3.3 AddReference() . . . 406
10.124.3.4 AddSourceImageSequence() . . . 406
10.124.3.5 Derive() . . . 406
10.124.3.6 GetFile() [1/2] . . . 406
10.124.3.7 GetFile() [2/2] . . . 406
10.124.3.8 SetDerivationCodeSequenceCodeValue() . . . 407
10.124.3.9 SetDerivationDescription() . . . 407
10.124.3.10 SetFile() . . . 407
10.124.3.11 SetPurposeOfReferenceCodeSequenceCodeValue() . . . 407
10.125 gdcm::FileExplicitFilter Class Reference . . . 408
10.125.1 Detailed Description . . . 408
10.125.2 Constructor & Destructor Documentation . . . 409
10.125.2.1 FileExplicitFilter() . . . 409
10.125.2.2 ~FileExplicitFilter() . . . 409
10.125.3 Member Function Documentation . . . 409
10.125.3.1 Change() . . . 409
10.125.3.2 ChangeFMI() . . . 409
10.125.3.3 GetFile() . . . 409
10.125.3.4 ProcessDataSet() . . . 409
10.125.3.5 SetChangePrivateTags() . . . 409
10.125.3.6 SetFile() . . . 410
10.125.3.7 SetRecomputeItemLength() . . . 410

10.125.3.8 SetRecomputeSequenceLength() . . . 410
10.125.3.9 SetUseVRUN() . . . 410
10.126 gdcm::FileMetaInformation Class Reference . . . 411
10.126.1 Detailed Description . . . 413
10.126.2 Constructor & Destructor Documentation . . . 413
10.126.2.1 FileMetaInformation() [1/2] . . . 413
10.126.2.2 ~FileMetaInformation() . . . 413
10.126.2.3 FileMetaInformation() [2/2] . . . 413
10.126.3 Member Function Documentation . . . 414
10.126.3.1 AppendImplementationClassUID() . . . 414
10.126.3.2 ComputeDataSetMediaStorageSOPClass() . . . 414
10.126.3.3 ComputeDataSetTransferSyntax() . . . 414
10.126.3.4 Default() . . . 414
10.126.3.5 FillFromDataSet() . . . 414
10.126.3.6 GetDataSetTransferSyntax() . . . 414
10.126.3.7 GetFileMetaInformationVersion() . . . 414
10.126.3.8 GetFullLength() . . . 415
10.126.3.9 GetGDCMImplementationClassUID() . . . 415
10.126.3.10 GetGDCMImplementationVersionName() . . . 415
10.126.3.11 GetGDCMSourceApplicationEntityTitle() . . . 415
10.126.3.12 GetImplementationClassUID() . . . 415
10.126.3.13 GetImplementationVersionName() . . . 415
10.126.3.14 GetMediaStorage() . . . 415
10.126.3.15 GetMediaStorageAsString() . . . 415
10.126.3.16 GetMetaInformationTS() . . . 415
10.126.3.17 GetPreamble() [1/2] . . . 416
10.126.3.18 GetPreamble() [2/2] . . . 416
10.126.3.19 GetSourceApplicationEntityTitle() . . . 416

10.126.3.20 Insert() . . . 416
10.126.3.21 IsValid() . . . 416
10.126.3.22 Read() . . . 416
10.126.3.23 ReadCompat() . . . 416
10.126.3.24 ReadCompatInternal() . . . 417
10.126.3.25 Replace() . . . 417
10.126.3.26 SetDataSetTransferSyntax() . . . 417
10.126.3.27 SetImplementationClassUID() . . . 417
10.126.3.28 SetImplementationVersionName() . . . 417
10.126.3.29 SetPreamble() . . . 417
10.126.3.30 SetSourceApplicationEntityTitle() . . . 418
10.126.3.31 Write() . . . 418
10.126.4 Friends And Related Function Documentation . . . 418
10.126.4.1 operator<< . . . 418
10.126.5 Member Data Documentation . . . 418
10.126.5.1 DataSetMS . . . 418
10.126.5.2 DataSetTS . . . 418
10.126.5.3 MetaInformationTS . . . 419
10.127 gdcm::Filename Class Reference . . . 419
10.127.1 Detailed Description . . . 420
10.127.2 Constructor & Destructor Documentation . . . 420
10.127.2.1 Filename() . . . 420
10.127.3 Member Function Documentation . . . 420
10.127.3.1 EndWith() . . . 420
10.127.3.2 GetExtension() . . . 420
10.127.3.3 GetFileName() . . . 420
10.127.3.4 GetName() . . . 420
10.127.3.5 GetPath() . . . 421

10.127.3.6 IsEmpty() . . . 421
10.127.3.7 IsIdentical() . . . 421
10.127.3.8 Join() . . . 421
10.127.3.9 operator const char *() . . . 421
10.127.3.10 ToUnixSlashes() . . . 421
10.127.3.11 ToWindowsSlashes() . . . 422
10.128 gdcm::FileNameEvent Class Reference . . . 422
10.128.1 Detailed Description . . . 423
10.128.2 Member Typedef Documentation . . . 424
10.128.2.1 Self . . . 424
10.128.2.2 Superclass . . . 424
10.128.3 Constructor & Destructor Documentation . . . 424
10.128.3.1 FileNameEvent() [1/2] . . . 424
10.128.3.2 ~FileNameEvent() . . . 424
10.128.3.3 FileNameEvent() [2/2] . . . 424
10.128.4 Member Function Documentation . . . 424
10.128.4.1 CheckEvent() . . . 424
10.128.4.2 GetEventName() . . . 424
10.128.4.3 GetFileName() . . . 425
10.128.4.4 MakeObject() . . . 425
10.128.4.5 SetFileName() . . . 425
10.129 gdcm::FilenameGenerator Class Reference . . . 425
10.129.1 Detailed Description . . . 426
10.129.2 Member Typedef Documentation . . . 426
10.129.2.1 FilenamesType . . . 426
10.129.2.2 FilenameType . . . 426
10.129.2.3 SizeType . . . 427
10.129.3 Constructor & Destructor Documentation . . . 427

10.129.3.1 FilenameGenerator() . . . 427
10.129.3.2 ~FilenameGenerator() . . . 427
10.129.4 Member Function Documentation . . . 427
10.129.4.1 Generate() . . . 427
10.129.4.2 GetFilename() . . . 427
10.129.4.3 GetFilenames() . . . 427
10.129.4.4 GetNumberOfFilenames() . . . 428
10.129.4.5 GetPattern() . . . 428
10.129.4.6 GetPrefix() . . . 428
10.129.4.7 SetNumberOfFilenames() . . . 428
10.129.4.8 SetPattern() . . . 428
10.129.4.9 SetPrefix() . . . 429
10.130 gdcm::FileSet Class Reference . . . 429
10.130.1 Detailed Description . . . 429
10.130.2 Member Typedef Documentation . . . 429
10.130.2.1 FilesType . . . 429
10.130.2.2 FileType . . . 430
10.130.3 Constructor & Destructor Documentation . . . 430
10.130.3.1 FileSet() . . . 430
10.130.4 Member Function Documentation . . . 430
10.130.4.1 AddFile() [1/2] . . . 430
10.130.4.2 AddFile() [2/2] . . . 430
10.130.4.3 GetFiles() . . . 430
10.130.4.4 SetFiles() . . . 430
10.130.5 Friends And Related Function Documentation . . . 431
10.130.5.1 operator<< . . . 431
10.131 gdcm::FileStreamer Class Reference . . . 431
10.131.1 Detailed Description . . . 433

10.131.2 Constructor & Destructor Documentation . . . 433
10.131.2.1 FileStreamer() . . . 433
10.131.2.2 ~FileStreamer() . . . 433
10.131.3 Member Function Documentation . . . 433
10.131.3.1 AppendToDataElement() . . . 433
10.131.3.2 AppendToGroupDataElement() . . . 434
10.131.3.3 CheckDataElement() . . . 434
10.131.3.4 CheckTemplateFileName() . . . 434
10.131.3.5 New() . . . 434
10.131.3.6 ReserveDataElement() . . . 434
10.131.3.7 ReserveGroupDataElement() . . . 435
10.131.3.8 SetOutputFileName() . . . 435
10.131.3.9 SetTemplateFileName() . . . 435
10.131.3.10 StartDataElement() . . . 435
10.131.3.11 StartGroupDataElement() . . . 435
10.131.3.12 StopDataElement() . . . 436
10.131.3.13 StopGroupDataElement() . . . 436
10.132 gdcm::FileWithName Class Reference . . . 436
10.132.1 Detailed Description . . . 437
10.132.2 Constructor & Destructor Documentation . . . 437
10.132.2.1 FileWithName() . . . 437
10.132.3 Member Data Documentation . . . 438
10.132.3.1 filename . . . 438
10.133 gdcm::FindPatientRootQuery Class Reference . . . 438
10.133.1 Detailed Description . . . 439
10.133.2 Constructor & Destructor Documentation . . . 439
10.133.2.1 FindPatientRootQuery() . . . 439
10.133.3 Member Function Documentation . . . 440

10.133.3.1 GetAbstractSyntaxUID() . . . 440
10.133.3.2 GetTagListByLevel() . . . 440
10.133.3.3 InitializeDataSet() . . . 440
10.133.3.4 ValidateQuery() . . . 440
10.133.4 Friends And Related Function Documentation . . . 441
10.133.4.1 QueryFactory . . . 441
10.134 gdcm::FindStudyRootQuery Class Reference . . . 441
10.134.1 Detailed Description . . . 442
10.134.2 Constructor & Destructor Documentation . . . 442
10.134.2.1 FindStudyRootQuery() . . . 442
10.134.3 Member Function Documentation . . . 443
10.134.3.1 GetAbstractSyntaxUID() . . . 443
10.134.3.2 GetTagListByLevel() . . . 443
10.134.3.3 InitializeDataSet() . . . 443
10.134.3.4 ValidateQuery() . . . 443
10.134.4 Friends And Related Function Documentation . . . 443
10.134.4.1 QueryFactory . . . 443
10.135 gdcm::Fragment Class Reference . . . 444
10.135.1 Detailed Description . . . 445
10.135.2 Constructor & Destructor Documentation . . . 445
10.135.2.1 Fragment() . . . 445
10.135.3 Member Function Documentation . . . 445
10.135.3.1 ComputeLength() . . . 445
10.135.3.2 GetLength() . . . 446
10.135.3.3 Read() . . . 446
10.135.3.4 ReadBacktrack() . . . 446
10.135.3.5 ReadPreValue() . . . 446
10.135.3.6 ReadValue() . . . 446

10.135.3.7 Write() . . . 446
10.135.4 Friends And Related Function Documentation . . . 447
10.135.4.1 operator<< . . . 447
10.136 gdcm::Global Class Reference . . . 447
10.136.1 Detailed Description . . . 448
10.136.2 Constructor & Destructor Documentation . . . 448
10.136.2.1 Global() . . . 448
10.136.2.2 ~Global() . . . 448
10.136.3 Member Function Documentation . . . 448
10.136.3.1 Append() . . . 448
10.136.3.2 GetDefs() . . . 448
10.136.3.3 GetDicts() [1/2] . . . 449
10.136.3.4 GetDicts() [2/2] . . . 449
10.136.3.5 GetInstance() . . . 449
10.136.3.6 LoadResourcesFiles() . . . 449
10.136.3.7 Locate() . . . 449
10.136.3.8 Prepend() . . . 450
10.136.4 Friends And Related Function Documentation . . . 450
10.136.4.1 operator<< . . . 450
10.137 gdcm::GroupDict Class Reference . . . 450
10.137.1 Detailed Description . . . 451
10.137.2 Member Typedef Documentation . . . 451
10.137.2.1 GroupStringVector . . . 451
10.137.3 Constructor & Destructor Documentation . . . 451
10.137.3.1 GroupDict() . . . 451
10.137.3.2 ~GroupDict() . . . 451
10.137.4 Member Function Documentation . . . 451
10.137.4.1 Add() . . . 451

10.137.4.2 GetAbbreviation() . . . 451
10.137.4.3 GetName() . . . 452
10.137.4.4 Insert() . . . 452
10.137.4.5 Size() . . . 452
10.137.5 Friends And Related Function Documentation . . . 452
10.137.5.1 operator<< . . . 452
10.138 gdcm::IconImageFilter Class Reference . . . 452
10.138.1 Detailed Description . . . 453
10.138.2 Constructor & Destructor Documentation . . . 454
10.138.2.1 IconImageFilter() . . . 454
10.138.2.2 ~IconImageFilter() . . . 454
10.138.3 Member Function Documentation . . . 454
10.138.3.1 Extract() . . . 454
10.138.3.2 ExtractIconImages() . . . 454
10.138.3.3 ExtractVeproIconImages() . . . 454
10.138.3.4 GetFile() [1/2] . . . 454
10.138.3.5 GetFile() [2/2] . . . 454
10.138.3.6 GetIconImage() . . . 455
10.138.3.7 GetNumberOfIconImages() . . . 455
10.138.3.8 SetFile() . . . 455
10.139 gdcm::IconImageGenerator Class Reference . . . 455
10.139.1 Detailed Description . . . 456
10.139.2 Constructor & Destructor Documentation . . . 456
10.139.2.1 IconImageGenerator() . . . 456
10.139.2.2 ~IconImageGenerator() . . . 457
10.139.3 Member Function Documentation . . . 457
10.139.3.1 AutoPixelMinMax() . . . 457
10.139.3.2 ConvertRGBToPaletteColor() . . . 457

10.139.3.3 Generate() . . . 457
10.139.3.4 GetIconImage() . . . 457
10.139.3.5 GetPixmap() [1/2] . . . 458
10.139.3.6 GetPixmap() [2/2] . . . 458
10.139.3.7 SetOutputDimensions() . . . 458
10.139.3.8 SetOutsideValuePixel() . . . 458
10.139.3.9 SetPixelMinMax() . . . 458
10.139.3.10 SetPixmap() . . . 458
10.140 gdcm::ignore_char Struct Reference . . . 459
10.140.1 Constructor & Destructor Documentation . . . 459
10.140.1.1 ignore_char() . . . 459
10.140.2 Member Data Documentation . . . 459
10.140.2.1 m_char . . . 459
10.141 gdcm::Image Class Reference . . . 460
10.141.1 Detailed Description . . . 461
10.141.2 Constructor & Destructor Documentation . . . 462
10.141.2.1 Image() . . . 462
10.141.2.2 ~Image() . . . 462
10.141.3 Member Function Documentation . . . 462
10.141.3.1 GetDirectionCosines() [1/2] . . . 462
10.141.3.2 GetDirectionCosines() [2/2] . . . 462
10.141.3.3 GetIntercept() . . . 462
10.141.3.4 GetOrigin() [1/2] . . . 463
10.141.3.5 GetOrigin() [2/2] . . . 463
10.141.3.6 GetSlope() . . . 463
10.141.3.7 GetSpacing() [1/2] . . . 463
10.141.3.8 GetSpacing() [2/2] . . . 463
10.141.3.9 Print() . . . 463

10.141.3.15SetDirectionCosines() [1/3] . . . 464
10.141.3.16SetDirectionCosines() [2/3] . . . 464
10.141.3.17SetDirectionCosines() [3/3] . . . 464
10.141.3.18SetIntercept() . . . 464
10.141.3.19SetOrigin() [1/3] . . . 464
10.141.3.20SetOrigin() [2/3] . . . 464
10.141.3.21SetOrigin() [3/3] . . . 464
10.141.3.22SetSlope() . . . 465
10.141.3.23SetSpacing() [1/2] . . . 465
10.141.3.24SetSpacing() [2/2] . . . 465
10.142gdcm::ImageApplyLookupTable Class Reference . . . 465
10.142.1Detailed Description . . . 468
10.142.2Constructor & Destructor Documentation . . . 468
10.142.2.1ImageApplyLookupTable() . . . 468
10.142.2.2~ImageApplyLookupTable() . . . 468
10.142.3Member Function Documentation . . . 468
10.142.3.1Apply() . . . 468
10.143gdcm::ImageChangePhotometricInterpretation Class Reference . . . 468
10.143.1Detailed Description . . . 471
10.143.2Constructor & Destructor Documentation . . . 471
10.143.2.1ImageChangePhotometricInterpretation() . . . 471
10.143.2.2~ImageChangePhotometricInterpretation() . . . 471
10.143.3Member Function Documentation . . . 471
10.143.3.1Change() . . . 471
10.143.3.2ChangeMonochrome() . . . 471
10.143.3.3GetPhotometricInterpretation() . . . 472
10.143.3.4RGB2YBR() . . . 472
10.143.3.5SetPhotometricInterpretation() . . . 472

10.143.3.6 YBR2RGB() . . . 472
10.144 gdcm::ImageChangePlanarConfiguration Class Reference . . . 472
10.144.1 Detailed Description . . . 475
10.144.2 Constructor & Destructor Documentation . . . 475
10.144.2.1 ImageChangePlanarConfiguration() . . . 475
10.144.2.2 ~ImageChangePlanarConfiguration() . . . 475
10.144.3 Member Function Documentation . . . 475
10.144.3.1 Change() . . . 475
10.144.3.2 GetPlanarConfiguration() . . . 475
10.144.3.3 RGBPixelsToRGBPlanes() . . . 476
10.144.3.4 RGBPlanesToRGBPixels() . . . 476
10.144.3.5 SetPlanarConfiguration() . . . 476
10.145 gdcm::ImageChangeTransferSyntax Class Reference . . . 477
10.145.1 Detailed Description . . . 479
10.145.2 Constructor & Destructor Documentation . . . 479
10.145.2.1 ImageChangeTransferSyntax() . . . 479
10.145.2.2 ~ImageChangeTransferSyntax() . . . 479
10.145.3 Member Function Documentation . . . 480
10.145.3.1 Change() . . . 480
10.145.3.2 GetTransferSyntax() . . . 480
10.145.3.3 SetCompressIconImage() . . . 480
10.145.3.4 SetForce() . . . 480
10.145.3.5 SetTransferSyntax() . . . 480
10.145.3.6 SetUserCodec() . . . 481
10.145.3.7 TryJPEG2000Codec() . . . 481
10.145.3.8 TryJPEGCodec() . . . 481
10.145.3.9 TryJPEGLSCodec() . . . 481
10.145.3.10 TryRAWCodec() . . . 481

10.145.3.1 TryRLECodec() . . . 482
10.146 gdcm::ImageCodec Class Reference . . . 482
10.146.1 Detailed Description . . . 484
10.146.2 Member Typedef Documentation . . . 484
10.146.2.1 LUTPtr . . . 484
10.146.3 Constructor & Destructor Documentation . . . 484
10.146.3.1 ImageCodec() . . . 484
10.146.3.2 ~ImageCodec() . . . 484
10.146.4 Member Function Documentation . . . 485
10.146.4.1 AppendFrameEncode() . . . 485
10.146.4.2 AppendRowEncode() . . . 485
10.146.4.3 CanCode() . . . 485
10.146.4.4 CanDecode() . . . 485
10.146.4.5 Clone() . . . 486
10.146.4.6 Decode() . . . 486
10.146.4.7 DecodeByStreams() . . . 486
10.146.4.8 DoByteSwap() . . . 486
10.146.4.9 DoInvertMonochrome() . . . 486
10.146.4.10 DoOverlayCleanup() . . . 487
10.146.4.11 DoPaddedCompositePixelCode() . . . 487
10.146.4.12 DoPlanarConfiguration() . . . 487
10.146.4.13 DoSimpleCopy() . . . 487
10.146.4.14 DoYBR() . . . 487
10.146.4.15 GetDimensions() . . . 487
10.146.4.16 GetHeaderInfo() . . . 487
10.146.4.17 GetLossyFlag() . . . 488
10.146.4.18 GetLUT() . . . 488
10.146.4.19 GetNeedByteSwap() . . . 488

10.146.4.20 GetNumberOfDimensions() . . . 488
10.146.4.21 GetPhotometricInterpretation() . . . 488
10.146.4.22 GetPixelFormat() [1/2] . . . 488
10.146.4.23 GetPixelFormat() [2/2] . . . 488
10.146.4.24 GetPlanarConfiguration() . . . 488
10.146.4.25 IsFrameEncoder() . . . 489
10.146.4.26 IsLossy() . . . 489
10.146.4.27 IsRowEncoder() . . . 489
10.146.4.28 IsValid() . . . 489
10.146.4.29 SetDimensions() [1/2] . . . 489
10.146.4.30 SetDimensions() [2/2] . . . 489
10.146.4.31 SetLossyFlag() . . . 489
10.146.4.32 SetLUT() . . . 490
10.146.4.33 SetNeedByteSwap() . . . 490
10.146.4.34 SetNeedOverlayCleanup() . . . 490
10.146.4.35 SetNumberOfDimensions() . . . 490
10.146.4.36 SetPhotometricInterpretation() . . . 490
10.146.4.37 SetPixelFormat() . . . 490
10.146.4.38 SetPlanarConfiguration() . . . 491
10.146.4.39 StartEncode() . . . 491
10.146.4.40 StopEncode() . . . 491
10.146.5 Friends And Related Function Documentation . . . 491
10.146.5.1 FileChangeTransferSyntax . . . 491
10.146.5.2 ImageChangePhotometricInterpretation . . . 491
10.146.6 Member Data Documentation . . . 491
10.146.6.1 Dimensions . . . 491
10.146.6.2 LossyFlag . . . 492
10.146.6.3 LUT . . . 492

10.146.6.4 NeedByteSwap . . . 492
10.146.6.5 NeedOverlayCleanup . . . 492
10.146.6.6 NumberOfDimensions . . . 492
10.146.6.7 PF . . . 492
10.146.6.8 PI . . . 492
10.146.6.9 PlanarConfiguration . . . 492
10.146.6.10 RequestPaddedCompositePixelCode . . . 492
10.146.6.11 RequestPlanarConfiguration . . . 493
10.147 gdcm::ImageConverter Class Reference . . . 493
10.147.1 Detailed Description . . . 493
10.147.2 Constructor & Destructor Documentation . . . 493
10.147.2.1 ImageConverter() . . . 493
10.147.2.2 ~ImageConverter() . . . 493
10.147.3 Member Function Documentation . . . 494
10.147.3.1 Convert() . . . 494
10.147.3.2 GetOuput() . . . 494
10.147.3.3 SetInput() . . . 494
10.148 gdcm::ImageFragmentSplitter Class Reference . . . 494
10.148.1 Detailed Description . . . 496
10.148.2 Constructor & Destructor Documentation . . . 496
10.148.2.1 ImageFragmentSplitter() . . . 496
10.148.2.2 ~ImageFragmentSplitter() . . . 496
10.148.3 Member Function Documentation . . . 496
10.148.3.1 GetFragmentSizeMax() . . . 496
10.148.3.2 SetForce() . . . 496
10.148.3.3 SetFragmentSizeMax() . . . 496
10.148.3.4 Split() . . . 497
10.149 gdcm::ImageHelper Class Reference . . . 497

10.149.1 Detailed Description . . . 498
10.149.2 Member Function Documentation . . . 498
10.149.2.1 ComputeMediaStorageFromModality() . . . 498
10.149.2.2 ComputeSpacingFromImagePositionPatient() . . . 498
10.149.2.3 GetDimensionsValue() . . . 499
10.149.2.4 GetDirectionCosinesFromDataSet() . . . 499
10.149.2.5 GetDirectionCosinesValue() . . . 499
10.149.2.6 GetForcePixelSpacing() . . . 499
10.149.2.7 GetForceRescaleInterceptSlope() . . . 499
10.149.2.8 GetLUT() . . . 499
10.149.2.9 GetOriginValue() . . . 500
10.149.2.10 GetPhotometricInterpretationValue() . . . 500
10.149.2.11 GetPixelFormatValue() . . . 500
10.149.2.12 GetPlanarConfigurationValue() . . . 500
10.149.2.13 GetPMSRescaleInterceptSlope() . . . 500
10.149.2.14 GetPointerFromElement() . . . 500
10.149.2.15 GetRealWorldValueMappingContent() . . . 500
10.149.2.16 GetRescaleInterceptSlopeValue() . . . 501
10.149.2.17 GetSpacingTagFromMediaStorage() . . . 501
10.149.2.18 GetSpacingValue() . . . 501
10.149.2.19 GetZSpacingTagFromMediaStorage() . . . 501
10.149.2.20 SetDimensionsValue() . . . 501
10.149.2.21 SetDirectionCosinesValue() . . . 501
10.149.2.22 SetForcePixelSpacing() . . . 502
10.149.2.23 SetForceRescaleInterceptSlope() . . . 502
10.149.2.24 SetOriginValue() . . . 502
10.149.2.25 SetPMSRescaleInterceptSlope() . . . 502
10.149.2.26 SetRescaleInterceptSlopeValue() . . . 502

10.149.2.25 SetSpacingValue() . . . 502
10.150 gdcm::ImageReader Class Reference . . . 503
10.150.1 Detailed Description . . . 505
10.150.2 Constructor & Destructor Documentation . . . 505
10.150.2.1 ImageReader() . . . 505
10.150.2.2 ~ImageReader() . . . 505
10.150.3 Member Function Documentation . . . 505
10.150.3.1 GetImage() [1/2] . . . 505
10.150.3.2 GetImage() [2/2] . . . 506
10.150.3.3 Read() . . . 506
10.150.3.4 ReadACRNEMAImage() . . . 506
10.150.3.5 ReadImage() . . . 506
10.151 gdcm::ImageRegionReader Class Reference . . . 507
10.151.1 Detailed Description . . . 509
10.151.2 Constructor & Destructor Documentation . . . 509
10.151.2.1 ImageRegionReader() . . . 509
10.151.2.2 ~ImageRegionReader() . . . 509
10.151.3 Member Function Documentation . . . 509
10.151.3.1 ComputeBufferLength() . . . 509
10.151.3.2 GetRegion() . . . 510
10.151.3.3 Read() . . . 510
10.151.3.4 ReadInformation() . . . 510
10.151.3.5 ReadIntoBuffer() . . . 510
10.151.3.6 SetRegion() . . . 510
10.152 gdcm::ImageToImageFilter Class Reference . . . 511
10.152.1 Detailed Description . . . 512
10.152.2 Constructor & Destructor Documentation . . . 512
10.152.2.1 ImageToImageFilter() . . . 512

10.152.2.2 ~ImageToImageFilter() . . . 512
10.152.3 Member Function Documentation . . . 512
10.152.3.1 GetInput() . . . 512
10.152.3.2 GetOutput() . . . 512
10.153 gdcm::ImageWriter Class Reference . . . 513
10.153.1 Detailed Description . . . 515
10.153.2 Constructor & Destructor Documentation . . . 515
10.153.2.1 ImageWriter() . . . 515
10.153.2.2 ~ImageWriter() . . . 515
10.153.3 Member Function Documentation . . . 515
10.153.3.1 ComputeTargetMediaStorage() . . . 515
10.153.3.2 GetImage() [1/2] . . . 515
10.153.3.3 GetImage() [2/2] . . . 516
10.153.3.4 Write() . . . 516
10.154 gdcm::network::ImplementationClassUIDSub Class Reference . . . 516
10.154.1 Detailed Description . . . 516
10.154.2 Constructor & Destructor Documentation . . . 517
10.154.2.1 ImplementationClassUIDSub() . . . 517
10.154.3 Member Function Documentation . . . 517
10.154.3.1 Print() . . . 517
10.154.3.2 Read() . . . 517
10.154.3.3 Size() . . . 517
10.154.3.4 Write() . . . 517
10.155 gdcm::network::ImplementationUIDSub Class Reference . . . 517
10.155.1 Detailed Description . . . 518
10.155.2 Constructor & Destructor Documentation . . . 518
10.155.2.1 ImplementationUIDSub() . . . 518
10.155.3 Member Function Documentation . . . 518

10.155.3.1 Write() . . . 518
10.156 gdcm::network::ImplementationVersionNameSub Class Reference . . . 518
10.156.1 Detailed Description . . . 518
10.156.2 Constructor & Destructor Documentation . . . 519
10.156.2.1 ImplementationVersionNameSub() . . . 519
10.156.3 Member Function Documentation . . . 519
10.156.3.1 Print() . . . 519
10.156.3.2 Read() . . . 519
10.156.3.3 Size() . . . 519
10.156.3.4 Write() . . . 519
10.157 gdcm::ImplicitDataElement Class Reference . . . 520
10.157.1 Detailed Description . . . 521
10.157.2 Member Function Documentation . . . 521
10.157.2.1 GetLength() . . . 521
10.157.2.2 Read() . . . 521
10.157.2.3 ReadPreValue() . . . 521
10.157.2.4 ReadValue() . . . 522
10.157.2.5 ReadValueWithLength() . . . 522
10.157.2.6 ReadWithLength() . . . 522
10.157.2.7 Write() . . . 522
10.158 gdcm::InitializeEvent Class Reference . . . 522
10.159 gdcm::IOD Class Reference . . . 524
10.159.1 Detailed Description . . . 524
10.159.2 Member Typedef Documentation . . . 524
10.159.2.1 MapIODEntry . . . 524
10.159.2.2 SizeType . . . 525
10.159.3 Constructor & Destructor Documentation . . . 525
10.159.3.1 IOD() . . . 525

10.159.4 Member Function Documentation . . . 525
10.159.4.1 AddIODEntry() . . . 525
10.159.4.2 Clear() . . . 525
10.159.4.3 GetIODEntry() . . . 525
10.159.4.4 GetNumberOfIODs() . . . 525
10.159.4.5 GetTypeFromTag() . . . 526
10.159.5 Friends And Related Function Documentation . . . 526
10.159.5.1 operator<< . . . 526
10.160 gdcm::IODEntry Class Reference . . . 526
10.160.1 Detailed Description . . . 527
10.160.2 Constructor & Destructor Documentation . . . 527
10.160.2.1 IODEntry() . . . 527
10.160.3 Member Function Documentation . . . 527
10.160.3.1 GetIE() . . . 527
10.160.3.2 GetName() . . . 528
10.160.3.3 GetRef() . . . 528
10.160.3.4 GetUsage() . . . 528
10.160.3.5 GetUsageType() . . . 528
10.160.3.6 SetIE() . . . 528
10.160.3.7 SetName() . . . 528
10.160.3.8 SetRef() . . . 528
10.160.3.9 SetUsage() . . . 528
10.160.4 Friends And Related Function Documentation . . . 529
10.160.4.1 operator<< . . . 529
10.161 gdcm::IODs Class Reference . . . 529
10.161.1 Detailed Description . . . 530
10.161.2 Member Typedef Documentation . . . 530
10.161.2.1 IODMapType . . . 530

10.161.2.2 IODMapTypeConstIterator . . . 530
10.161.2.3 IODName . . . 530
10.161.3 Constructor & Destructor Documentation . . . 530
10.161.3.1 IODs() . . . 530
10.161.4 Member Function Documentation . . . 530
10.161.4.1 AddIOD() . . . 530
10.161.4.2 Begin() . . . 531
10.161.4.3 Clear() . . . 531
10.161.4.4 End() . . . 531
10.161.4.5 GetIOD() . . . 531
10.161.5 Friends And Related Function Documentation . . . 531
10.161.5.1 operator<< . . . 531
10.162 gdcm::IPPSorter Class Reference . . . 532
10.162.1 Detailed Description . . . 533
10.162.2 Constructor & Destructor Documentation . . . 534
10.162.2.1 IPPSorter() . . . 534
10.162.3 Member Function Documentation . . . 534
10.162.3.1 GetDirectionCosinesTolerance() . . . 534
10.162.3.2 GetZSpacing() . . . 534
10.162.3.3 GetZSpacingTolerance() . . . 534
10.162.3.4 SetComputeZSpacing() . . . 534
10.162.3.5 SetDirectionCosinesTolerance() . . . 535
10.162.3.6 SetDropDuplicatePositions() . . . 535
10.162.3.7 SetZSpacingTolerance() . . . 535
10.162.3.8 Sort() . . . 535
10.162.4 Member Data Documentation . . . 536
10.162.4.1 ComputeZSpacing . . . 536
10.162.4.2 DirCosTolerance . . . 536

10.162.4.3 DropDuplicatePositions . . . 536
10.162.4.4 ZSpacing . . . 536
10.162.4.5 ZTolerance . . . 536
10.163 gdcm::Item Class Reference . . . 536
10.163.1 Detailed Description . . . 538
10.163.2 Constructor & Destructor Documentation . . . 538
10.163.2.1 Item() [1/2] . . . 538
10.163.2.2 Item() [2/2] . . . 538
10.163.3 Member Function Documentation . . . 538
10.163.3.1 Clear() . . . 538
10.163.3.2 FindDataElement() . . . 539
10.163.3.3 GetDataElement() . . . 539
10.163.3.4 GetLength() . . . 539
10.163.3.5 GetNestedDataSet() [1/2] . . . 539
10.163.3.6 GetNestedDataSet() [2/2] . . . 539
10.163.3.7 InsertDataElement() . . . 539
10.163.3.8 Read() . . . 540
10.163.3.9 SetNestedDataSet() . . . 540
10.163.3.10 Write() . . . 540
10.163.4 Friends And Related Function Documentation . . . 540
10.163.4.1 operator<< . . . 540
10.164 gdcm::IterationEvent Class Reference . . . 541
10.165 gdcm::JPEG12Codec Class Reference . . . 542
10.165.1 Detailed Description . . . 543
10.165.2 Constructor & Destructor Documentation . . . 543
10.165.2.1 JPEG12Codec() . . . 543
10.165.2.2 ~JPEG12Codec() . . . 543
10.165.3 Member Function Documentation . . . 543

10.165.3.1 DecodeByStreams() . . . 543
10.165.3.2 EncodeBuffer() . . . 544
10.165.3.3 GetHeaderInfo() . . . 544
10.165.3.4 InternalCode() . . . 544
10.165.3.5 IsStateSuspension() . . . 544
10.166 gdcm::JPEG16Codec Class Reference . . . 545
10.166.1 Detailed Description . . . 546
10.166.2 Constructor & Destructor Documentation . . . 546
10.166.2.1 JPEG16Codec() . . . 546
10.166.2.2 ~JPEG16Codec() . . . 546
10.166.3 Member Function Documentation . . . 546
10.166.3.1 DecodeByStreams() . . . 546
10.166.3.2 EncodeBuffer() . . . 547
10.166.3.3 GetHeaderInfo() . . . 547
10.166.3.4 InternalCode() . . . 547
10.166.3.5 IsStateSuspension() . . . 547
10.167 gdcm::JPEG2000Codec Class Reference . . . 548
10.167.1 Detailed Description . . . 549
10.167.2 Constructor & Destructor Documentation . . . 550
10.167.2.1 JPEG2000Codec() . . . 550
10.167.2.2 ~JPEG2000Codec() . . . 550
10.167.3 Member Function Documentation . . . 550
10.167.3.1 AppendFrameEncode() . . . 550
10.167.3.2 AppendRowEncode() . . . 550
10.167.3.3 CanCode() . . . 550
10.167.3.4 CanDecode() . . . 550
10.167.3.5 Clone() . . . 551
10.167.3.6 Code() . . . 551

10.167.3.7 Decode() . . . 551
10.167.3.8 DecodeByStreams() . . . 551
10.167.3.9 DecodeExtent() . . . 551
10.167.3.10 GetHeaderInfo() . . . 552
10.167.3.11 GetQuality() . . . 552
10.167.3.12 GetRate() . . . 552
10.167.3.13 IsFrameEncoder() . . . 552
10.167.3.14 IsRowEncoder() . . . 552
10.167.3.15 SetNumberOfResolutions() . . . 552
10.167.3.16 SetQuality() . . . 552
10.167.3.17 SetRate() . . . 553
10.167.3.18 SetReversible() . . . 553
10.167.3.19 SetTileSize() . . . 553
10.167.3.20 StartEncode() . . . 553
10.167.3.21 StopEncode() . . . 553
10.167.4 Friends And Related Function Documentation . . . 553
10.167.4.1 Bitmap . . . 553
10.167.4.2 ImageRegionReader . . . 553
10.168 gdcm::JPEG8Codec Class Reference . . . 554
10.168.1 Detailed Description . . . 555
10.168.2 Constructor & Destructor Documentation . . . 555
10.168.2.1 JPEG8Codec() . . . 555
10.168.2.2 ~JPEG8Codec() . . . 555
10.168.3 Member Function Documentation . . . 555
10.168.3.1 DecodeByStreams() . . . 555
10.168.3.2 EncodeBuffer() . . . 556
10.168.3.3 GetHeaderInfo() . . . 556
10.168.3.4 InternalCode() . . . 556

10.168.3.5 IsStateSuspension() . . . 556
10.169 gdcm::JPEGCodec Class Reference . . . 557
10.169.1 Detailed Description . . . 559
10.169.2 Constructor & Destructor Documentation . . . 559
10.169.2.1 JPEGCodec() . . . 559
10.169.2.2 ~JPEGCodec() . . . 559
10.169.3 Member Function Documentation . . . 559
10.169.3.1 AppendFrameEncode() . . . 559
10.169.3.2 AppendRowEncode() . . . 560
10.169.3.3 CanCode() . . . 560
10.169.3.4 CanDecode() . . . 560
10.169.3.5 Clone() . . . 560
10.169.3.6 Code() . . . 560
10.169.3.7 ComputeOffsetTable() . . . 561
10.169.3.8 Decode() . . . 561
10.169.3.9 DecodeByStreams() . . . 561
10.169.3.10 DecodeExtent() . . . 561
10.169.3.11 EncodeBuffer() . . . 561
10.169.3.12 GetHeaderInfo() . . . 562
10.169.3.13 GetLossless() . . . 562
10.169.3.14 GetQuality() . . . 562
10.169.3.15 IsFrameEncoder() . . . 562
10.169.3.16 IsRowEncoder() . . . 562
10.169.3.17 IsStateSuspension() . . . 562
10.169.3.18 IsValid() . . . 563
10.169.3.19 SetBitSample() . . . 563
10.169.3.20 SetLossless() . . . 563
10.169.3.21 SetPixelFormat() . . . 563

10.169.3.22 SetQuality() . . . 563
10.169.3.23 StartEncode() . . . 563
10.169.3.24 StopEncode() . . . 564
10.169.4 Friends And Related Function Documentation . . . 564
10.169.4.1 ImageRegionReader . . . 564
10.169.5 Member Data Documentation . . . 564
10.169.5.1 BitSample . . . 564
10.169.5.2 Quality . . . 564
10.170 gdcm::JPEGLSCodec Class Reference . . . 564
10.170.1 Detailed Description . . . 566
10.170.2 Constructor & Destructor Documentation . . . 566
10.170.2.1 JPEGLSCodec() . . . 566
10.170.2.2 ~JPEGLSCodec() . . . 567
10.170.3 Member Function Documentation . . . 567
10.170.3.1 AppendFrameEncode() . . . 567
10.170.3.2 AppendRowEncode() . . . 567
10.170.3.3 CanCode() . . . 567
10.170.3.4 CanDecode() . . . 567
10.170.3.5 Clone() . . . 568
10.170.3.6 Code() . . . 568
10.170.3.7 Decode() [1/2] . . . 568
10.170.3.8 Decode() [2/2] . . . 568
10.170.3.9 DecodeExtent() . . . 568
10.170.3.10 GetBufferLength() . . . 569
10.170.3.11 GetHeaderInfo() . . . 569
10.170.3.12 GetLossless() . . . 569
10.170.3.13 IsFrameEncoder() . . . 569
10.170.3.14 IsRowEncoder() . . . 569

10.170.3.15 SetBufferLength() . . . 569
10.170.3.16 SetLossless() . . . 569
10.170.3.17 SetLossyError() . . . 569
10.170.3.18 StartEncode() . . . 570
10.170.3.19 StopEncode() . . . 570
10.170.4 Friends And Related Function Documentation . . . 570
10.170.4.1 ImageRegionReader . . . 570
10.171 gdcm::JSON Class Reference . . . 570
10.171.1 Detailed Description . . . 570
10.171.2 Constructor & Destructor Documentation . . . 571
10.171.2.1 JSON() . . . 571
10.171.2.2 ~JSON() . . . 571
10.171.3 Member Function Documentation . . . 571
10.171.3.1 Code() . . . 571
10.171.3.2 Decode() . . . 571
10.171.3.3 GetPrettyPrint() . . . 571
10.171.3.4 PrettyPrintOff() . . . 571
10.171.3.5 PrettyPrintOn() . . . 572
10.171.3.6 SetPrettyPrint() . . . 572
10.172 gdcm::KAKADUCodec Class Reference . . . 572
10.172.1 Detailed Description . . . 573
10.172.2 Constructor & Destructor Documentation . . . 573
10.172.2.1 KAKADUCodec() . . . 573
10.172.2.2 ~KAKADUCodec() . . . 573
10.172.3 Member Function Documentation . . . 574
10.172.3.1 CanCode() . . . 574
10.172.3.2 CanDecode() . . . 574
10.172.3.3 Clone() . . . 574

10.172.3.4 Code() . . . 574
10.172.3.5 Decode() . . . 574
10.173 gdcm::LO Class Reference . . . 575
10.173.1 Detailed Description . . . 576
10.173.2 Member Typedef Documentation . . . 576
10.173.2.1 const_iterator . . . 576
10.173.2.2 const_reference . . . 576
10.173.2.3 const_reverse_iterator . . . 576
10.173.2.4 difference_type . . . 576
10.173.2.5 iterator . . . 576
10.173.2.6 pointer . . . 576
10.173.2.7 reference . . . 577
10.173.2.8 reverse_iterator . . . 577
10.173.2.9 size_type . . . 577
10.173.2.10 Superclass . . . 577
10.173.2.11 value_type . . . 577
10.173.3 Constructor & Destructor Documentation . . . 577
10.173.3.1 LO() [1/4] . . . 577
10.173.3.2 LO() [2/4] . . . 577
10.173.3.3 LO() [3/4] . . . 577
10.173.3.4 LO() [4/4] . . . 578
10.173.4 Member Function Documentation . . . 578
10.173.4.1 IsValid() . . . 578
10.174 gdcm::LookupTable Class Reference . . . 578
10.174.1 Detailed Description . . . 580
10.174.2 Member Enumeration Documentation . . . 580
10.174.2.1 LookupTableType . . . 580
10.174.3 Constructor & Destructor Documentation . . . 580

10.174.3.1 LookupTable() [1/2] . . . 581
10.174.3.2 ~LookupTable() . . . 581
10.174.3.3 LookupTable() [2/2] . . . 581
10.174.4 Member Function Documentation . . . 581
10.174.4.1 Allocate() . . . 581
10.174.4.2 Clear() . . . 581
10.174.4.3 Decode() [1/2] . . . 581
10.174.4.4 Decode() [2/2] . . . 581
10.174.4.5 GetBitSample() . . . 582
10.174.4.6 GetBufferAsRGBA() . . . 582
10.174.4.7 GetLUT() . . . 582
10.174.4.8 GetLUTDescriptor() . . . 582
10.174.4.9 GetLUTLength() . . . 582
10.174.4.10 GetPointer() . . . 582
10.174.4.11 InitializeBlueLUT() . . . 583
10.174.4.12 Initialized() . . . 583
10.174.4.13 InitializeGreenLUT() . . . 583
10.174.4.14 InitializeLUT() . . . 583
10.174.4.15 InitializeRedLUT() . . . 583
10.174.4.16 Print() . . . 583
10.174.4.17 SetBlueLUT() . . . 584
10.174.4.18 SetGreenLUT() . . . 584
10.174.4.19 SetLUT() . . . 584
10.174.4.20 SetRedLUT() . . . 584
10.174.4.21 WriteBufferAsRGBA() . . . 584
10.174.5 Member Data Documentation . . . 584
10.174.5.1 BitSample . . . 584
10.174.5.2 IncompleteLUT . . . 584

10.174.5.3 Internal . . . 585
10.175 gdcm::Scanner::ltstr Struct Reference . . . 585
10.175.1 Member Function Documentation . . . 585
10.175.1.1 operator()() . . . 585
10.176 gdcm::StrictScanner::ltstr Struct Reference . . . 585
10.176.1 Member Function Documentation . . . 586
10.176.1.1 operator()() . . . 586
10.177 gdcm::Macro Class Reference . . . 586
10.177.1 Detailed Description . . . 587
10.177.2 Member Typedef Documentation . . . 587
10.177.2.1 ArrayIncludeMacrosType . . . 587
10.177.2.2 MapModuleEntry . . . 587
10.177.3 Constructor & Destructor Documentation . . . 587
10.177.3.1 Macro() . . . 587
10.177.4 Member Function Documentation . . . 587
10.177.4.1 AddMacroEntry() . . . 587
10.177.4.2 Clear() . . . 588
10.177.4.3 FindMacroEntry() . . . 588
10.177.4.4 GetMacroEntry() . . . 588
10.177.4.5 GetName() . . . 588
10.177.4.6 SetName() . . . 588
10.177.4.7 Verify() . . . 588
10.177.5 Friends And Related Function Documentation . . . 588
10.177.5.1 operator<< . . . 588
10.178 gdcm::Macros Class Reference . . . 589
10.178.1 Detailed Description . . . 589
10.178.2 Member Typedef Documentation . . . 589
10.178.2.1 ModuleMapType . . . 589

10.178.2 Constructor & Destructor Documentation . . . 590
10.178.2.1 Macros() . . . 590
10.178.3 Member Function Documentation . . . 590
10.178.3.1 AddMacro() . . . 590
10.178.3.2 Clear() . . . 590
10.178.3.3 GetMacro() . . . 590
10.178.3.4 IsEmpty() . . . 590
10.178.4 Friends And Related Function Documentation . . . 590
10.178.4.1 operator<< . . . 590
10.179 gdcm::network::MaximumLengthSub Class Reference . . . 591
10.179.1 Detailed Description . . . 591
10.179.2 Constructor & Destructor Documentation . . . 591
10.179.2.1 MaximumLengthSub() . . . 591
10.179.3 Member Function Documentation . . . 591
10.179.3.1 GetMaximumLength() . . . 591
10.179.3.2 Print() . . . 592
10.179.3.3 Read() . . . 592
10.179.3.4 SetMaximumLength() . . . 592
10.179.3.5 Size() . . . 592
10.179.3.6 Write() . . . 592
10.180 gdcm::MD5 Class Reference . . . 592
10.180.1 Detailed Description . . . 593
10.180.2 Constructor & Destructor Documentation . . . 593
10.180.2.1 MD5() . . . 593
10.180.2.2 ~MD5() . . . 593
10.180.3 Member Function Documentation . . . 593
10.180.3.1 Compute() . . . 593
10.180.3.2 ComputeFile() . . . 594

10.181 gdcm::MediaStorage Class Reference . . . 594
10.181.1 Detailed Description . . . 597
10.181.2 Member Enumeration Documentation . . . 597
10.181.2.1 MSType . . . 597
10.181.2.2 ObjectType . . . 599
10.181.3 Constructor & Destructor Documentation . . . 600
10.181.3.1 MediaStorage() . . . 600
10.181.4 Member Function Documentation . . . 600
10.181.4.1 GetModality() . . . 600
10.181.4.2 GetModalityDimension() . . . 600
10.181.4.3 GetMSString() . . . 600
10.181.4.4 GetMSType() . . . 600
10.181.4.5 GetNumberOfModality() . . . 601
10.181.4.6 GetNumberOfMSString() . . . 601
10.181.4.7 GetNumberOfMSType() . . . 601
10.181.4.8 GetString() . . . 601
10.181.4.9 GuessFromModality() . . . 601
10.181.4.10 IsImage() . . . 601
10.181.4.11 IsUndefined() . . . 602
10.181.4.12 operator MSType() . . . 602
10.181.4.13 SetFromDataSet() . . . 602
10.181.4.14 SetFromFile() . . . 602
10.181.4.15 SetFromHeader() . . . 602
10.181.4.16 SetFromModality() . . . 602
10.181.4.17 SetFromSourceImageSequence() . . . 603
10.181.5 Friends And Related Function Documentation . . . 603
10.181.5.1 operator<< . . . 603
10.182 gdcm::MemberCommand< T > Class Template Reference . . . 603

10.182.1 Detailed Description . . . 605
10.182.2 Member Typedef Documentation . . . 605
10.182.2.1 Self . . . 605
10.182.2.2 TConstMemberFunctionPointer . . . 605
10.182.2.3 TMemberFunctionPointer . . . 605
10.182.3 Constructor & Destructor Documentation . . . 605
10.182.3.1 MemberCommand() . . . 605
10.182.3.2 ~MemberCommand() . . . 606
10.182.4 Member Function Documentation . . . 606
10.182.4.1 Execute() [1/2] . . . 606
10.182.4.2 Execute() [2/2] . . . 606
10.182.4.3 New() . . . 606
10.182.4.4 SetCallbackFunction() [1/2] . . . 606
10.182.4.5 SetCallbackFunction() [2/2] . . . 607
10.182.5 Member Data Documentation . . . 607
10.182.5.1 m_ConstMemberFunction . . . 607
10.182.5.2 m_MemberFunction . . . 607
10.182.5.3 m_This . . . 607
10.183 gdcm::MeshPrimitive Class Reference . . . 607
10.183.1 Detailed Description . . . 609
10.183.2 Member Typedef Documentation . . . 609
10.183.2.1 PrimitivesData . . . 609
10.183.3 Member Enumeration Documentation . . . 609
10.183.3.1 MPType . . . 609
10.183.4 Constructor & Destructor Documentation . . . 610
10.183.4.1 MeshPrimitive() . . . 610
10.183.4.2 ~MeshPrimitive() . . . 610
10.183.5 Member Function Documentation . . . 610

10.183.5.1 AddPrimitiveData() . . . 610
10.183.5.2 GetMPType() . . . 610
10.183.5.3 GetMPTypeString() . . . 610
10.183.5.4 GetNumberOfPrimitivesData() . . . 610
10.183.5.5 GetPrimitiveData() [1/4] . . . 611
10.183.5.6 GetPrimitiveData() [2/4] . . . 611
10.183.5.7 GetPrimitiveData() [3/4] . . . 611
10.183.5.8 GetPrimitiveData() [4/4] . . . 611
10.183.5.9 GetPrimitivesData() [1/2] . . . 611
10.183.5.10 GetPrimitivesData() [2/2] . . . 611
10.183.5.11 GetPrimitiveType() . . . 611
10.183.5.12 SetPrimitiveData() [1/2] . . . 611
10.183.5.13 SetPrimitiveData() [2/2] . . . 611
10.183.5.14 SetPrimitivesData() . . . 612
10.183.5.15 SetPrimitiveType() . . . 612
10.183.6 Member Data Documentation . . . 612
10.183.6.1 PrimitiveData . . . 612
10.183.6.2 PrimitiveType . . . 612
10.184 gdcm::ModalityPerformedProcedureStepCreateQuery Class Reference . . . 612
10.184.1 Detailed Description . . . 613
10.184.2 Constructor & Destructor Documentation . . . 614
10.184.2.1 ModalityPerformedProcedureStepCreateQuery() . . . 614
10.184.3 Member Function Documentation . . . 614
10.184.3.1 GetAbstractSyntaxUID() . . . 614
10.184.3.2 GetRequiredDataSet() . . . 614
10.184.3.3 ValidateQuery() . . . 614
10.184.4 Friends And Related Function Documentation . . . 614
10.184.4.1 QueryFactory . . . 614

10.185 gdcm::ModalityPerformedProcedureStepSetQuery Class Reference . . . 615
10.185.1 Detailed Description . . . 616
10.185.2 Constructor & Destructor Documentation . . . 616
10.185.2.1 ModalityPerformedProcedureStepSetQuery() . . . 616
10.185.3 Member Function Documentation . . . 616
10.185.3.1 GetAbstractSyntaxUID() . . . 616
10.185.3.2 GetRequiredDataSet() . . . 616
10.185.3.3 ValidateQuery() . . . 616
10.185.4 Friends And Related Function Documentation . . . 617
10.185.4.1 QueryFactory . . . 617
10.186 gdcm::ModifiedEvent Class Reference . . . 617
10.187 gdcm::Module Class Reference . . . 618
10.187.1 Detailed Description . . . 619
10.187.2 Member Typedef Documentation . . . 619
10.187.2.1 ArrayIncludeMacrosType . . . 619
10.187.2.2 MapModuleEntry . . . 619
10.187.3 Constructor & Destructor Documentation . . . 619
10.187.3.1 Module() . . . 619
10.187.4 Member Function Documentation . . . 619
10.187.4.1 AddMacro() . . . 619
10.187.4.2 AddModuleEntry() . . . 620
10.187.4.3 Clear() . . . 620
10.187.4.4 FindModuleEntryInMacros() . . . 620
10.187.4.5 GetModuleEntryInMacros() . . . 620
10.187.4.6 GetName() . . . 620
10.187.4.7 SetName() . . . 620
10.187.4.8 Verify() . . . 621
10.187.5 Friends And Related Function Documentation . . . 621

10.187.5.1 operator<< . . . 621
10.188 gdcm::ModuleEntry Class Reference . . . 621
10.188.1 Detailed Description . . . 623
10.188.2 Member Typedef Documentation . . . 623
10.188.2.1 Description . . . 623
10.188.3 Constructor & Destructor Documentation . . . 623
10.188.3.1 ModuleEntry() . . . 623
10.188.3.2 ~ModuleEntry() . . . 623
10.188.4 Member Function Documentation . . . 623
10.188.4.1 GetDescription() . . . 623
10.188.4.2 GetName() . . . 624
10.188.4.3 GetType() . . . 624
10.188.4.4 SetDescription() . . . 624
10.188.4.5 SetName() . . . 624
10.188.4.6 SetType() . . . 624
10.188.5 Friends And Related Function Documentation . . . 624
10.188.5.1 operator<< . . . 624
10.188.6 Member Data Documentation . . . 624
10.188.6.1 DataElementType . . . 624
10.188.6.2 DescriptionField . . . 625
10.188.6.3 Name . . . 625
10.189 gdcm::Modules Class Reference . . . 625
10.189.1 Detailed Description . . . 626
10.189.2 Member Typedef Documentation . . . 626
10.189.2.1 ModuleMapType . . . 626
10.189.3 Constructor & Destructor Documentation . . . 626
10.189.3.1 Modules() . . . 626
10.189.4 Member Function Documentation . . . 626

10.189.4.1 AddModule() . . . 626
10.189.4.2 Clear() . . . 626
10.189.4.3 GetModule() . . . 627
10.189.4.4 IsEmpty() . . . 627
10.189.5 Friends And Related Function Documentation . . . 627
10.189.5.1 operator<< . . . 627
10.190 gdcm::MovePatientRootQuery Class Reference . . . 627
10.190.1 Detailed Description . . . 629
10.190.2 Constructor & Destructor Documentation . . . 629
10.190.2.1 MovePatientRootQuery() . . . 629
10.190.3 Member Function Documentation . . . 629
10.190.3.1 GetAbstractSyntaxUID() . . . 629
10.190.3.2 GetTagListByLevel() . . . 629
10.190.3.3 InitializeDataSet() . . . 629
10.190.3.4 ValidateQuery() . . . 630
10.190.4 Friends And Related Function Documentation . . . 630
10.190.4.1 QueryFactory . . . 630
10.191 gdcm::MoveStudyRootQuery Class Reference . . . 630
10.191.1 Detailed Description . . . 632
10.191.2 Constructor & Destructor Documentation . . . 632
10.191.2.1 MoveStudyRootQuery() . . . 632
10.191.3 Member Function Documentation . . . 632
10.191.3.1 GetAbstractSyntaxUID() . . . 632
10.191.3.2 GetTagListByLevel() . . . 632
10.191.3.3 InitializeDataSet() . . . 632
10.191.3.4 ValidateQuery() . . . 633
10.191.4 Friends And Related Function Documentation . . . 633
10.191.4.1 QueryFactory . . . 633

10.192 gdcm::network::NActionRQ Class Reference . . . 633
10.192.1 Detailed Description . . . 634
10.192.2 Member Function Documentation . . . 634
10.192.2.1 ConstructPDV() . . . 634
10.193 gdcm::network::NActionRSP Class Reference . . . 635
10.193.1 Detailed Description . . . 635
10.193.2 Member Function Documentation . . . 636
10.193.2.1 ConstructPDVByDataSet() . . . 636
10.194 gdcm::network::NCreateRQ Class Reference . . . 636
10.194.1 Detailed Description . . . 637
10.194.2 Member Function Documentation . . . 637
10.194.2.1 ConstructPDV() . . . 637
10.195 gdcm::network::NCreateRSP Class Reference . . . 637
10.195.1 Detailed Description . . . 638
10.195.2 Member Function Documentation . . . 638
10.195.2.1 ConstructPDVByDataSet() . . . 638
10.196 gdcm::network::NDeleteRQ Class Reference . . . 639
10.196.1 Detailed Description . . . 639
10.196.2 Member Function Documentation . . . 640
10.196.2.1 ConstructPDV() . . . 640
10.197 gdcm::network::NDeleteRSP Class Reference . . . 640
10.197.1 Detailed Description . . . 641
10.197.2 Member Function Documentation . . . 641
10.197.2.1 ConstructPDVByDataSet() . . . 641
10.198 gdcm::NestedModuleEntries Class Reference . . . 641
10.198.1 Detailed Description . . . 643
10.198.2 Member Typedef Documentation . . . 643
10.198.2.1 SizeType . . . 643

10.198.3 Constructor & Destructor Documentation . . . 643
10.198.3.1 NestedModuleEntries() . . . 643
10.198.4 Member Function Documentation . . . 643
10.198.4.1 AddModuleEntry() . . . 643
10.198.4.2 GetModuleEntry() [1/2] . . . 643
10.198.4.3 GetModuleEntry() [2/2] . . . 644
10.198.4.4 GetNumberOfModuleEntries() . . . 644
10.198.5 Friends And Related Function Documentation . . . 644
10.198.5.1 operator<< . . . 644
10.199 gdcm::network::NEventReportRQ Class Reference . . . 644
10.199.1 Detailed Description . . . 645
10.199.2 Member Function Documentation . . . 645
10.199.2.1 ConstructPDV() . . . 645
10.200 gdcm::network::NEventReportRSP Class Reference . . . 646
10.200.1 Detailed Description . . . 646
10.200.2 Member Function Documentation . . . 647
10.200.2.1 ConstructPDVByDataSet() . . . 647
10.201 gdcm::network::NGetRQ Class Reference . . . 647
10.201.1 Detailed Description . . . 648
10.201.2 Member Function Documentation . . . 648
10.201.2.1 ConstructPDV() . . . 648
10.202 gdcm::network::NGetRSP Class Reference . . . 648
10.202.1 Detailed Description . . . 649
10.202.2 Member Function Documentation . . . 649
10.202.2.1 ConstructPDVByDataSet() . . . 649
10.203 gdcm::NoEvent Class Reference . . . 650
10.203.1 Detailed Description . . . 650
10.204 gdcm::network::NormalizedMessageFactory Class Reference . . . 651

10.204.1 Member Function Documentation . . . 651
10.204.1.1 ConstructNAction() . . . 651
10.204.1.2 ConstructNCreate() . . . 651
10.204.1.3 ConstructNDelete() . . . 651
10.204.1.4 ConstructNEventReport() . . . 652
10.204.1.5 ConstructNGet() . . . 652
10.204.1.6 ConstructNSet() . . . 652
10.205 gdcm::NormalizedNetworkFunctions Class Reference . . . 652
10.205.1 Detailed Description . . . 653
10.205.2 Member Function Documentation . . . 653
10.205.2.1 ConstructQuery() . . . 653
10.205.2.2 NAction() . . . 653
10.205.2.3 NCreate() . . . 653
10.205.2.4 NDelete() . . . 654
10.205.2.5 NEventReport() . . . 654
10.205.2.6 NGet() . . . 654
10.205.2.7 NSet() . . . 654
10.206 gdcm::network::NSetRQ Class Reference . . . 655
10.206.1 Detailed Description . . . 655
10.206.2 Member Function Documentation . . . 656
10.206.2.1 ConstructPDV() . . . 656
10.207 gdcm::network::NSetRSP Class Reference . . . 656
10.207.1 Detailed Description . . . 657
10.207.2 Member Function Documentation . . . 657
10.207.2.1 ConstructPDVByDataSet() . . . 657
10.208 gdcm::Object Class Reference . . . 657
10.208.1 Detailed Description . . . 659
10.208.2 Constructor & Destructor Documentation . . . 659

10.208.2.1Object() [1/2] . . . 659
10.208.2.2~Object() . . . 659
10.208.2.3Object() [2/2] . . . 659
10.208.3Member Function Documentation . . . 659
10.208.3.1operator=() . . . 659
10.208.3.2Print() . . . 660
10.208.3.3Register() . . . 660
10.208.3.4UnRegister() . . . 660
10.208.4Friends And Related Function Documentation . . . 660
10.208.4.1operator<< . . . 660
10.208.4.2SmartPointer . . . 660
10.209gdcm::OpenSSLCryptoFactory Class Reference . . . 661
10.209.1Constructor & Destructor Documentation . . . 662
10.209.1.1OpenSSLCryptoFactory() . . . 662
10.209.2Member Function Documentation . . . 662
10.209.2.1CreateCMSProvider() . . . 662
10.209.2.2InitOpenSSL() . . . 662
10.210gdcm::OpenSSLCryptographicMessageSyntax Class Reference . . . 662
10.210.1Constructor & Destructor Documentation . . . 663
10.210.1.1OpenSSLCryptographicMessageSyntax() . . . 663
10.210.1.2~OpenSSLCryptographicMessageSyntax() . . . 663
10.210.2Member Function Documentation . . . 664
10.210.2.1Decrypt() . . . 664
10.210.2.2Encrypt() . . . 664
10.210.2.3GetCipherType() . . . 664
10.210.2.4ParseCertificateFile() . . . 664
10.210.2.5ParseKeyFile() . . . 664
10.210.2.6SetCipherType() . . . 665

10.210.2.7 SetPassword() . . . 665
10.211 gdcm::OpenSSLP7CryptoFactory Class Reference . . . 665
10.211.1 Constructor & Destructor Documentation . . . 666
10.211.1.1 OpenSSLP7CryptoFactory() . . . 666
10.211.2 Member Function Documentation . . . 666
10.211.2.1 CreateCMSProvider() . . . 666
10.212 gdcm::OpenSSLP7CryptographicMessageSyntax Class Reference . . . 667
10.212.1 Detailed Description . . . 668
10.212.2 Constructor & Destructor Documentation . . . 668
10.212.2.1 OpenSSLP7CryptographicMessageSyntax() . . . 668
10.212.2.2 ~OpenSSLP7CryptographicMessageSyntax() . . . 668
10.212.3 Member Function Documentation . . . 668
10.212.3.1 Decrypt() . . . 668
10.212.3.2 Encrypt() . . . 669
10.212.3.3 GetCipherType() . . . 669
10.212.3.4 ParseCertificateFile() . . . 669
10.212.3.5 ParseKeyFile() . . . 669
10.212.3.6 SetCipherType() . . . 669
10.212.3.7 SetPassword() . . . 670
10.213 gdcm::Orientation Class Reference . . . 670
10.213.1 Detailed Description . . . 671
10.213.2 Member Enumeration Documentation . . . 671
10.213.2.1 OrientationType . . . 671
10.213.3 Constructor & Destructor Documentation . . . 671
10.213.3.1 Orientation() . . . 671
10.213.3.2 ~Orientation() . . . 671
10.213.4 Member Function Documentation . . . 672
10.213.4.1 GetLabel() . . . 672

10.213.4.2 GetMajorAxisFromPatientRelativeDirectionCosine() . . . 672
10.213.4.3 GetObliquityThresholdCosineValue() . . . 672
10.213.4.4 GetType() . . . 672
10.213.4.5 Print() . . . 672
10.213.4.6 SetObliquityThresholdCosineValue() . . . 673
10.213.5 Friends And Related Function Documentation . . . 673
10.213.5.1 operator<< . . . 673
10.214 gdcm::Overlay Class Reference . . . 673
10.214.1 Detailed Description . . . 676
10.214.2 Member Enumeration Documentation . . . 676
10.214.2.1 OverlayType . . . 676
10.214.3 Constructor & Destructor Documentation . . . 676
10.214.3.1 Overlay() [1/2] . . . 676
10.214.3.2 ~Overlay() . . . 676
10.214.3.3 Overlay() [2/2] . . . 676
10.214.4 Member Function Documentation . . . 677
10.214.4.1 Decompress() . . . 677
10.214.4.2 GetBitPosition() . . . 677
10.214.4.3 GetBitsAllocated() . . . 677
10.214.4.4 GetColumns() . . . 677
10.214.4.5 GetDescription() . . . 677
10.214.4.6 GetGroup() . . . 677
10.214.4.7 GetOrigin() . . . 677
10.214.4.8 GetOverlayData() . . . 678
10.214.4.9 GetOverlayTypeAsString() . . . 678
10.214.4.10 GetOverlayTypeFromString() . . . 678
10.214.4.11 GetRows() . . . 678
10.214.4.12 GetType() . . . 678

10.214.4.13GetTypeAsEnum() . . . 678
10.214.4.14GetUnpackBuffer() . . . 678
10.214.4.15GetUnpackBufferLength() . . . 679
10.214.4.16GrabOverlayFromPixelData() . . . 679
10.214.4.17IsEmpty() . . . 679
10.214.4.18IsInPixelData() [1/2] . . . 679
10.214.4.19IsInPixelData() [2/2] . . . 679
10.214.4.20IsZero() . . . 679
10.214.4.21operator=() . . . 679
10.214.4.22Print() . . . 680
10.214.4.23SetBitPosition() . . . 680
10.214.4.24SetBitsAllocated() . . . 680
10.214.4.25SetColumns() . . . 680
10.214.4.26SetDescription() . . . 680
10.214.4.27SetFrameOrigin() . . . 680
10.214.4.28SetGroup() . . . 681
10.214.4.29SetNumberOfFrames() . . . 681
10.214.4.30SetOrigin() . . . 681
10.214.4.31SetOverlay() . . . 681
10.214.4.32SetRows() . . . 681
10.214.4.33SetType() . . . 681
10.214.4.34Update() . . . 682
10.215gdcm::ParseException Class Reference . . . 682
10.215.1Detailed Description . . . 683
10.215.2Constructor & Destructor Documentation . . . 683
10.215.2.1ParseException() . . . 683
10.215.2.2~ParseException() . . . 683
10.215.3Member Function Documentation . . . 684

10.215.3.1 GetLastElement() . . . 684
10.215.3.2 operator=() . . . 684
10.215.3.3 SetLastElement() . . . 684
10.216 gdcm::Parser Class Reference . . . 684
10.216.1 Detailed Description . . . 685
10.216.2 Member Typedef Documentation . . . 685
10.216.2.1 EndElementHandler . . . 685
10.216.2.2 StartElementHandler . . . 685
10.216.3 Member Enumeration Documentation . . . 685
10.216.3.1 ErrorType . . . 685
10.216.4 Constructor & Destructor Documentation . . . 686
10.216.4.1 Parser() . . . 686
10.216.4.2 ~Parser() . . . 686
10.216.5 Member Function Documentation . . . 686
10.216.5.1 GetBuffer() . . . 686
10.216.5.2 GetCurrentByteIndex() . . . 686
10.216.5.3 GetErrorCode() . . . 686
10.216.5.4 GetErrorString() . . . 686
10.216.5.5 GetUserData() . . . 687
10.216.5.6 Parse() . . . 687
10.216.5.7 ParseBuffer() . . . 687
10.216.5.8 Process() . . . 687
10.216.5.9 SetElementHandler() . . . 687
10.216.5.10 SetUserData() . . . 687
10.217 gdcm::Patient Class Reference . . . 687
10.217.1 Detailed Description . . . 688
10.217.2 Constructor & Destructor Documentation . . . 688
10.217.2.1 Patient() . . . 688

10.218 dcm::network::PDataTFPDU Class Reference . . . 688
10.218.1 Detailed Description . . . 689
10.218.2 Member Typedef Documentation . . . 689
10.218.2.1 SizeType . . . 689
10.218.3 Constructor & Destructor Documentation . . . 690
10.218.3.1 PDataTFPDU() . . . 690
10.218.4 Member Function Documentation . . . 690
10.218.4.1 AddPresentationDataValue() . . . 690
10.218.4.2 GetNumberOfPresentationDataValues() . . . 690
10.218.4.3 GetPresentationDataValue() . . . 690
10.218.4.4 IsLastFragment() . . . 690
10.218.4.5 Print() . . . 690
10.218.4.6 Read() . . . 690
10.218.4.7 ReadInto() . . . 691
10.218.4.8 Size() . . . 691
10.218.4.9 Write() . . . 691
10.219 dcm::PDBElement Class Reference . . . 691
10.219.1 Detailed Description . . . 692
10.219.2 Constructor & Destructor Documentation . . . 692
10.219.2.1 PDBElement() . . . 692
10.219.3 Member Function Documentation . . . 692
10.219.3.1 GetName() . . . 692
10.219.3.2 GetValue() . . . 693
10.219.3.3 operator==() . . . 693
10.219.3.4 SetName() . . . 693
10.219.3.5 SetValue() . . . 693
10.219.4 Friends And Related Function Documentation . . . 693
10.219.4.1 operator<< . . . 693

10.219.5 Member Data Documentation . . . 693
10.219.5.1 NameField . . . 693
10.219.5.2 ValueField . . . 694
10.220 gdcm::PDBHeader Class Reference . . . 694
10.220.1 Detailed Description . . . 695
10.220.2 Constructor & Destructor Documentation . . . 695
10.220.2.1 PDBHeader() . . . 695
10.220.2.2 ~PDBHeader() . . . 695
10.220.3 Member Function Documentation . . . 695
10.220.3.1 FindPDBElementByName() . . . 695
10.220.3.2 GetPDBEEnd() . . . 695
10.220.3.3 GetPDBElementByName() . . . 696
10.220.3.4 GetPDBInfoTag() . . . 696
10.220.3.5 LoadFromDataElement() . . . 696
10.220.3.6 Print() . . . 696
10.220.4 Friends And Related Function Documentation . . . 696
10.220.4.1 operator<< . . . 696
10.221 gdcm::PDFCodec Class Reference . . . 697
10.221.1 Detailed Description . . . 698
10.221.2 Constructor & Destructor Documentation . . . 698
10.221.2.1 PDFCodec() . . . 698
10.221.2.2 ~PDFCodec() . . . 698
10.221.3 Member Function Documentation . . . 698
10.221.3.1 CanCode() . . . 698
10.221.3.2 CanDecode() . . . 698
10.221.3.3 Decode() . . . 699
10.222 gdcm::network::PDUFactory Class Reference . . . 699
10.222.1 Detailed Description . . . 700

10.222.2 Member Function Documentation . . . 700
10.222.2.1 ConstructAbortPDU() . . . 700
10.222.2.2 ConstructPDU() . . . 700
10.222.2.3 ConstructReleasePDU() . . . 700
10.222.2.4 CreateCEchoPDU() . . . 700
10.222.2.5 CreateCFindPDU() . . . 700
10.222.2.6 CreateCMovePDU() . . . 700
10.222.2.7 CreateCStoreRQPDU() . . . 701
10.222.2.8 CreateCStoreRSPPDU() . . . 701
10.222.2.9 CreateNActionPDU() . . . 701
10.222.2.10 CreateNCreatePDU() . . . 701
10.222.2.11 CreateNDeletePDU() . . . 701
10.222.2.12 CreateNEventReportPDU() . . . 701
10.222.2.13 CreateNGetPDU() . . . 701
10.222.2.14 CreateNSetPDU() . . . 702
10.222.2.15 DetermineEventByPDU() . . . 702
10.222.2.16 GetPDVs() . . . 702
10.223 gdcm::PersonName Class Reference . . . 702
10.223.1 Detailed Description . . . 703
10.223.2 Member Function Documentation . . . 703
10.223.2.1 GetMaxLength() . . . 703
10.223.2.2 GetNumberOfComponents() . . . 703
10.223.2.3 Print() . . . 703
10.223.2.4 SetBlob() . . . 703
10.223.2.5 SetComponents() [1/2] . . . 703
10.223.2.6 SetComponents() [2/2] . . . 704
10.223.3 Member Data Documentation . . . 704
10.223.3.1 Component . . . 704

10.223.3.2 MaxLength . . . 704
10.223.3.3 MaxNumberOfComponents . . . 704
10.223.3.4 Padding . . . 704
10.223.3.5 Separator . . . 704
10.224 gdcm::PGXCodec Class Reference . . . 705
10.224.1 Detailed Description . . . 706
10.224.2 Constructor & Destructor Documentation . . . 706
10.224.2.1 PGXCodec() . . . 706
10.224.2.2 ~PGXCodec() . . . 706
10.224.3 Member Function Documentation . . . 706
10.224.3.1 CanCode() . . . 706
10.224.3.2 CanDecode() . . . 706
10.224.3.3 Clone() . . . 707
10.224.3.4 GetHeaderInfo() . . . 707
10.224.3.5 Read() . . . 707
10.224.3.6 Write() . . . 707
10.225 gdcm::PhotometricInterpretation Class Reference . . . 707
10.225.1 Detailed Description . . . 708
10.225.2 Member Enumeration Documentation . . . 709
10.225.2.1 PIType . . . 709
10.225.3 Constructor & Destructor Documentation . . . 710
10.225.3.1 PhotometricInterpretation() . . . 710
10.225.4 Member Function Documentation . . . 710
10.225.4.1 GetPIString() . . . 710
10.225.4.2 GetPIType() . . . 710
10.225.4.3 GetSamplesPerPixel() . . . 711
10.225.4.4 GetString() . . . 711
10.225.4.5 GetType() . . . 711

10.225.4.6 IsLossless() . . . 711
10.225.4.7 IsLossy() . . . 711
10.225.4.8 IsRetired() . . . 711
10.225.4.9 IsSameColorSpace() . . . 711
10.225.4.10 operator PIType() . . . 711
10.225.5 Friends And Related Function Documentation . . . 712
10.225.5.1 operator<< . . . 712
10.226 gdcm::PixelFormat Class Reference . . . 712
10.226.1 Detailed Description . . . 714
10.226.2 Member Enumeration Documentation . . . 714
10.226.2.1 ScalarType . . . 714
10.226.3 Constructor & Destructor Documentation . . . 715
10.226.3.1 PixelFormat() [1/2] . . . 715
10.226.3.2 PixelFormat() [2/2] . . . 715
10.226.4 Member Function Documentation . . . 715
10.226.4.1 GetBitsAllocated() . . . 715
10.226.4.2 GetBitsStored() . . . 715
10.226.4.3 GetHighBit() . . . 715
10.226.4.4 GetMax() . . . 716
10.226.4.5 GetMin() . . . 716
10.226.4.6 GetPixelRepresentation() . . . 716
10.226.4.7 GetPixelSize() . . . 716
10.226.4.8 GetSamplesPerPixel() . . . 716
10.226.4.9 GetScalarType() . . . 717
10.226.4.10 GetScalarTypeAsString() . . . 717
10.226.4.11 IsCompatible() . . . 717
10.226.4.12 IsValid() . . . 717
10.226.4.13 operator ScalarType() . . . 717

10.226.4.14operator"!=() [1/2] . . . 717
10.226.4.15operator"!=() [2/2] . . . 717
10.226.4.16operator==() [1/2] . . . 717
10.226.4.17operator==() [2/2] . . . 718
10.226.4.18Print() . . . 718
10.226.4.19SetBitsAllocated() . . . 718
10.226.4.20SetBitsStored() . . . 718
10.226.4.21SetHighBit() . . . 718
10.226.4.22SetPixelRepresentation() . . . 718
10.226.4.23SetSamplesPerPixel() . . . 718
10.226.4.24SetScalarType() . . . 719
10.226.4.25Validate() . . . 719
10.226.5Friends And Related Function Documentation . . . 719
10.226.5.1Bitmap . . . 719
10.226.5.2operator<< . . . 719
10.227gdcm::Pixmap Class Reference . . . 720
10.227.1Detailed Description . . . 721
10.227.2Constructor & Destructor Documentation . . . 721
10.227.2.1Pixmap() . . . 721
10.227.2.2~Pixmap() . . . 722
10.227.3Member Function Documentation . . . 722
10.227.3.1AreOverlaysInPixelData() . . . 722
10.227.3.2GetCurve() [1/2] . . . 722
10.227.3.3GetCurve() [2/2] . . . 722
10.227.3.4GetIconImage() [1/2] . . . 722
10.227.3.5GetIconImage() [2/2] . . . 722
10.227.3.6GetNumberOfCurves() . . . 722
10.227.3.7GetNumberOfOverlays() . . . 723

10.227.3.8 GetOverlay() [1/2] . . . 723
10.227.3.9 GetOverlay() [2/2] . . . 723
10.227.3.10 Print() . . . 723
10.227.3.11 RemoveOverlay() . . . 723
10.227.3.12 SetIconImage() . . . 723
10.227.3.13 SetNumberOfCurves() . . . 723
10.227.3.14 SetNumberOfOverlays() . . . 723
10.227.4 Member Data Documentation . . . 724
10.227.4.1 Curves . . . 724
10.227.4.2 Icon . . . 724
10.227.4.3 Overlays . . . 724
10.228 gdcm::PixmapReader Class Reference . . . 724
10.228.1 Detailed Description . . . 726
10.228.2 Constructor & Destructor Documentation . . . 726
10.228.2.1 PixmapReader() . . . 726
10.228.2.2 ~PixmapReader() . . . 726
10.228.3 Member Function Documentation . . . 726
10.228.3.1 GetPixmap() [1/2] . . . 726
10.228.3.2 GetPixmap() [2/2] . . . 726
10.228.3.3 Read() . . . 727
10.228.3.4 ReadACRNEMAImage() . . . 727
10.228.3.5 ReadImage() . . . 727
10.228.3.6 ReadImageInternal() . . . 727
10.228.4 Member Data Documentation . . . 727
10.228.4.1 PixelData . . . 727
10.229 gdcm::PixmapToPixmapFilter Class Reference . . . 728
10.229.1 Detailed Description . . . 729
10.229.2 Constructor & Destructor Documentation . . . 729

10.229.2.1 PixmapToPixmapFilter() . . . 729
10.229.2.2 ~PixmapToPixmapFilter() . . . 729
10.229.3 Member Function Documentation . . . 729
10.229.3.1 GetInput() . . . 729
10.229.3.2 GetOutput() . . . 729
10.229.3.3 GetOutputAsPixmap() . . . 730
10.230 gdcm::PixmapWriter Class Reference . . . 730
10.230.1 Detailed Description . . . 732
10.230.2 Constructor & Destructor Documentation . . . 732
10.230.2.1 PixmapWriter() . . . 732
10.230.2.2 ~PixmapWriter() . . . 732
10.230.3 Member Function Documentation . . . 732
10.230.3.1 DoIconImage() . . . 732
10.230.3.2 GetImage() [1/2] . . . 733
10.230.3.3 GetImage() [2/2] . . . 733
10.230.3.4 GetPixmap() [1/2] . . . 733
10.230.3.5 GetPixmap() [2/2] . . . 733
10.230.3.6 PrepareWrite() [1/2] . . . 733
10.230.3.7 PrepareWrite() [2/2] . . . 733
10.230.3.8 SetImage() . . . 733
10.230.3.9 SetPixmap() . . . 734
10.230.3.10 Write() . . . 734
10.230.4 Member Data Documentation . . . 734
10.230.4.1 PixelData . . . 734
10.231 gdcm::PNMCodec Class Reference . . . 734
10.231.1 Detailed Description . . . 735
10.231.2 Constructor & Destructor Documentation . . . 736
10.231.2.1 PNMCodec() . . . 736

10.231.2.2 ~PNMCodec() . . . 736
10.231.3 Member Function Documentation . . . 736
10.231.3.1 CanCode() . . . 736
10.231.3.2 CanDecode() . . . 736
10.231.3.3 Clone() . . . 736
10.231.3.4 GetBufferLength() . . . 736
10.231.3.5 GetHeaderInfo() . . . 737
10.231.3.6 Read() . . . 737
10.231.3.7 SetBufferLength() . . . 737
10.231.3.8 Write() . . . 737
10.232 gdcm::Preamble Class Reference . . . 737
10.232.1 Detailed Description . . . 738
10.232.2 Constructor & Destructor Documentation . . . 738
10.232.2.1 Preamble() [1/2] . . . 738
10.232.2.2 ~Preamble() . . . 739
10.232.2.3 Preamble() [2/2] . . . 739
10.232.3 Member Function Documentation . . . 739
10.232.3.1 Clear() . . . 739
10.232.3.2 Create() . . . 739
10.232.3.3 GetInternal() . . . 739
10.232.3.4 GetLength() . . . 739
10.232.3.5 IsEmpty() . . . 739
10.232.3.6 IsValid() . . . 740
10.232.3.7 operator=() . . . 740
10.232.3.8 Print() . . . 740
10.232.3.9 Read() . . . 740
10.232.3.10 Remove() . . . 740
10.232.3.11 Valid() . . . 740

10.232.3.12 Write() . . . 740
10.232.4 Friends And Related Function Documentation . . . 741
10.232.4.1 operator<< . . . 741
10.233 gdcm::PresentationContext Class Reference . . . 741
10.233.1 Detailed Description . . . 742
10.233.2 Member Typedef Documentation . . . 742
10.233.2.1 SizeType . . . 742
10.233.2.2 TransferSyntaxArrayType . . . 742
10.233.3 Constructor & Destructor Documentation . . . 743
10.233.3.1 PresentationContext() [1/2] . . . 743
10.233.3.2 PresentationContext() [2/2] . . . 743
10.233.4 Member Function Documentation . . . 743
10.233.4.1 AddTransferSyntax() . . . 743
10.233.4.2 GetAbstractSyntax() . . . 743
10.233.4.3 GetNumberOfTransferSyntaxes() . . . 743
10.233.4.4 GetPresentationContextID() . . . 743
10.233.4.5 GetTransferSyntax() . . . 743
10.233.4.6 operator==() . . . 744
10.233.4.7 Print() . . . 744
10.233.4.8 SetAbstractSyntax() . . . 744
10.233.4.9 SetPresentationContextID() . . . 744
10.233.5 Member Data Documentation . . . 744
10.233.5.1 AbstractSyntax . . . 744
10.233.5.2 ID . . . 744
10.233.5.3 TransferSyntaxes . . . 744
10.234 gdcm::network::PresentationContextAC Class Reference . . . 745
10.234.1 Detailed Description . . . 745
10.234.2 Constructor & Destructor Documentation . . . 745

10.234.2.1 PresentationContextAC() . . . 745
10.234.3 Member Function Documentation . . . 745
10.234.3.1 GetPresentationContextID() . . . 745
10.234.3.2 GetReason() . . . 746
10.234.3.3 GetTransferSyntax() . . . 746
10.234.3.4 Print() . . . 746
10.234.3.5 Read() . . . 746
10.234.3.6 SetPresentationContextID() . . . 746
10.234.3.7 SetReason() . . . 746
10.234.3.8 SetTransferSyntax() . . . 746
10.234.3.9 Size() . . . 746
10.234.3.10 Write() . . . 747
10.235 dcm::PresentationContextGenerator Class Reference . . . 747
10.235.1 Detailed Description . . . 748
10.235.2 Member Typedef Documentation . . . 748
10.235.2.1 PresentationContextArrayType . . . 748
10.235.2.2 SizeType . . . 748
10.235.3 Constructor & Destructor Documentation . . . 748
10.235.3.1 PresentationContextGenerator() . . . 748
10.235.4 Member Function Documentation . . . 749
10.235.4.1 AddFromFile() . . . 749
10.235.4.2 AddPresentationContext() . . . 749
10.235.4.3 GenerateFromFilenames() . . . 749
10.235.4.4 GenerateFromUID() . . . 749
10.235.4.5 GetDefaultTransferSyntax() . . . 749
10.235.4.6 GetPresentationContexts() . . . 749
10.235.4.7 SetDefaultTransferSyntax() . . . 750
10.235.4.8 SetMergeModeToAbstractSyntax() . . . 750

10.235.4.9 SetMergeModeToTransferSyntax() . . . 750
10.236 gdcm::network::PresentationContextRQ Class Reference . . . 750
10.236.1 Detailed Description . . . 751
10.236.2 Member Typedef Documentation . . . 751
10.236.2.1 SizeType . . . 751
10.236.3 Constructor & Destructor Documentation . . . 751
10.236.3.1 PresentationContextRQ() [1/3] . . . 751
10.236.3.2 PresentationContextRQ() [2/3] . . . 751
10.236.3.3 PresentationContextRQ() [3/3] . . . 751
10.236.4 Member Function Documentation . . . 751
10.236.4.1 AddTransferSyntax() . . . 751
10.236.4.2 GetAbstractSyntax() [1/2] . . . 752
10.236.4.3 GetAbstractSyntax() [2/2] . . . 752
10.236.4.4 GetNumberOfTransferSyntaxes() . . . 752
10.236.4.5 GetPresentationContextID() . . . 752
10.236.4.6 GetTransferSyntax() [1/2] . . . 752
10.236.4.7 GetTransferSyntax() [2/2] . . . 752
10.236.4.8 GetTransferSyntaxes() . . . 752
10.236.4.9 operator==() . . . 752
10.236.4.10 Print() . . . 752
10.236.4.11 Read() . . . 753
10.236.4.12 SetAbstractSyntax() . . . 753
10.236.4.13 SetPresentationContextID() . . . 753
10.236.4.14 Size() . . . 753
10.236.4.15 Write() . . . 753
10.237 gdcm::network::PresentationDataValue Class Reference . . . 753
10.237.1 Detailed Description . . . 754
10.237.2 Constructor & Destructor Documentation . . . 754

10.237.2.1 PresentationDataValue() . . . 754
10.237.3 Member Function Documentation . . . 754
10.237.3.1 ConcatenatePDVBlobs() . . . 754
10.237.3.2 ConcatenatePDVBlobsAsExplicit() . . . 755
10.237.3.3 GetBlob() . . . 755
10.237.3.4 GetIsCommand() . . . 755
10.237.3.5 GetIsLastFragment() . . . 755
10.237.3.6 GetMessageHeader() . . . 755
10.237.3.7 GetPresentationContextID() . . . 755
10.237.3.8 Print() . . . 755
10.237.3.9 Read() . . . 755
10.237.3.10 ReadInto() . . . 755
10.237.3.11 SetBlob() . . . 756
10.237.3.12 SetCommand() . . . 756
10.237.3.13 SetDataSet() . . . 756
10.237.3.14 SetLastFragment() . . . 756
10.237.3.15 SetMessageHeader() . . . 756
10.237.3.16 SetPresentationContextID() . . . 756
10.237.3.17 Size() . . . 756
10.237.3.18 Write() . . . 757
10.238 gdcm::Printer Class Reference . . . 757
10.238.1 Detailed Description . . . 758
10.238.2 Member Enumeration Documentation . . . 758
10.238.2.1 PrintStyles . . . 758
10.238.3 Constructor & Destructor Documentation . . . 759
10.238.3.1 Printer() . . . 759
10.238.3.2 ~Printer() . . . 759
10.238.4 Member Function Documentation . . . 759

10.238.4.1 GetPrintStyle() . . . 759
10.238.4.2 Print() . . . 759
10.238.4.3 PrintDataElement() . . . 759
10.238.4.4 PrintDataSet() . . . 760
10.238.4.5 PrintSQ() . . . 760
10.238.4.6 SetColor() . . . 760
10.238.4.7 SetFile() . . . 760
10.238.4.8 SetStyle() . . . 760
10.238.5 Member Data Documentation . . . 761
10.238.5.1 F . . . 761
10.238.5.2 MaxPrintLength . . . 761
10.238.5.3 PrintStyle . . . 761
10.239 gdcm::PrivateDict Class Reference . . . 761
10.239.1 Detailed Description . . . 762
10.239.2 Constructor & Destructor Documentation . . . 762
10.239.2.1 PrivateDict() . . . 762
10.239.2.2 ~PrivateDict() . . . 762
10.239.3 Member Function Documentation . . . 762
10.239.3.1 AddDictEntry() . . . 762
10.239.3.2 FindDictEntry() . . . 762
10.239.3.3 GetDictEntry() . . . 762
10.239.3.4 IsEmpty() . . . 762
10.239.3.5 LoadDefault() . . . 762
10.239.3.6 PrintXML() . . . 763
10.239.3.7 RemoveDictEntry() . . . 763
10.239.4 Friends And Related Function Documentation . . . 763
10.239.4.1 Dicts . . . 763
10.239.4.2 operator<< . . . 763

10.240gdcm::PrivateTag Class Reference . . . 763
10.240.1Detailed Description . . . 765
10.240.2Constructor & Destructor Documentation . . . 765
10.240.2.1PrivateTag() [1/2] . . . 765
10.240.2.2PrivateTag() [2/2] . . . 765
10.240.3Member Function Documentation . . . 765
10.240.3.1GetAsDataElement() . . . 765
10.240.3.2GetOwner() . . . 765
10.240.3.3operator<() . . . 766
10.240.3.4ReadFromCommaSeparatedString() . . . 766
10.240.3.5SetOwner() . . . 766
10.240.4Friends And Related Function Documentation . . . 766
10.240.4.1operator<< . . . 766
10.241gdcm::ProgressEvent Class Reference . . . 766
10.241.1Detailed Description . . . 768
10.241.2Member Typedef Documentation . . . 768
10.241.2.1Self . . . 768
10.241.2.2Superclass . . . 768
10.241.3Constructor & Destructor Documentation . . . 768
10.241.3.1ProgressEvent() [1/2] . . . 768
10.241.3.2~ProgressEvent() . . . 768
10.241.3.3ProgressEvent() [2/2] . . . 769
10.241.4Member Function Documentation . . . 769
10.241.4.1CheckEvent() . . . 769
10.241.4.2GetEventName() . . . 769
10.241.4.3GetProgress() . . . 769
10.241.4.4MakeObject() . . . 769
10.241.4.5SetProgress() . . . 769

10.242gdcm::PVRGCodec Class Reference . . . 770
10.242.1Detailed Description . . . 771
10.242.2Constructor & Destructor Documentation . . . 771
10.242.2.1PVRGCodec() . . . 771
10.242.2.2~PVRGCodec() . . . 771
10.242.3Member Function Documentation . . . 771
10.242.3.1CanCode() . . . 771
10.242.3.2CanDecode() . . . 772
10.242.3.3Clone() . . . 772
10.242.3.4Code() . . . 772
10.242.3.5Decode() . . . 772
10.242.3.6SetLossyFlag() . . . 772
10.243gdcm::PythonFilter Class Reference . . . 773
10.243.1Detailed Description . . . 773
10.243.2Constructor & Destructor Documentation . . . 773
10.243.2.1PythonFilter() . . . 773
10.243.2.2~PythonFilter() . . . 773
10.243.3Member Function Documentation . . . 773
10.243.3.1GetFile() [1/2] . . . 773
10.243.3.2GetFile() [2/2] . . . 773
10.243.3.3SetDicts() . . . 774
10.243.3.4SetFile() . . . 774
10.243.3.5ToPyObject() . . . 774
10.243.3.6UseDictAlways() . . . 774
10.244gdcm::QueryBase Class Reference . . . 774
10.244.1Detailed Description . . . 775
10.244.2Constructor & Destructor Documentation . . . 775
10.244.2.1~QueryBase() . . . 775

10.244.3 Member Function Documentation . . . 776
10.244.3.1 GetAllRequiredTags() . . . 776
10.244.3.2 GetAllTags() . . . 776
10.244.3.3 GetHierachicalSearchTags() . . . 776
10.244.3.4 GetName() . . . 776
10.244.3.5 GetOptionalTags() . . . 776
10.244.3.6 GetQueryLevel() . . . 776
10.244.3.7 GetRequiredTags() . . . 777
10.244.3.8 GetUniqueTags() . . . 777
10.245 gdcm::QueryFactory Class Reference . . . 777
10.245.1 Detailed Description . . . 777
10.245.2 Member Function Documentation . . . 778
10.245.2.1 GetCharacterFromCurrentLocale() . . . 778
10.245.2.2 ListCharSets() . . . 778
10.245.2.3 ProduceCharacterSetDataElement() . . . 778
10.245.2.4 ProduceQuery() [1/2] . . . 778
10.245.2.5 ProduceQuery() [2/2] . . . 778
10.246 gdcm::QueryImage Class Reference . . . 779
10.246.1 Detailed Description . . . 780
10.246.2 Member Function Documentation . . . 780
10.246.2.1 GetHierachicalSearchTags() . . . 780
10.246.2.2 GetName() . . . 780
10.246.2.3 GetOptionalTags() . . . 780
10.246.2.4 GetQueryLevel() . . . 780
10.246.2.5 GetRequiredTags() . . . 780
10.246.2.6 GetUniqueTags() . . . 781
10.247 gdcm::QueryPatient Class Reference . . . 781
10.247.1 Detailed Description . . . 782

10.247.2 Member Function Documentation . . . 782
10.247.2.1 GetHierachicalSearchTags() . . . 782
10.247.2.2 GetName() . . . 782
10.247.2.3 GetOptionalTags() . . . 782
10.247.2.4 GetQueryLevel() . . . 782
10.247.2.5 GetRequiredTags() . . . 783
10.247.2.6 GetUniqueTags() . . . 783
10.248 gdcm::QuerySeries Class Reference . . . 783
10.248.1 Detailed Description . . . 784
10.248.2 Member Function Documentation . . . 784
10.248.2.1 GetHierachicalSearchTags() . . . 784
10.248.2.2 GetName() . . . 784
10.248.2.3 GetOptionalTags() . . . 785
10.248.2.4 GetQueryLevel() . . . 785
10.248.2.5 GetRequiredTags() . . . 785
10.248.2.6 GetUniqueTags() . . . 785
10.249 gdcm::QueryStudy Class Reference . . . 785
10.249.1 Detailed Description . . . 786
10.249.2 Member Function Documentation . . . 787
10.249.2.1 GetHierachicalSearchTags() . . . 787
10.249.2.2 GetName() . . . 787
10.249.2.3 GetOptionalTags() . . . 787
10.249.2.4 GetQueryLevel() . . . 787
10.249.2.5 GetRequiredTags() . . . 787
10.249.2.6 GetUniqueTags() . . . 787
10.250 gdcm::RAWCodec Class Reference . . . 788
10.250.1 Detailed Description . . . 789
10.250.2 Constructor & Destructor Documentation . . . 789

10.250.2.1 RAWCodec() . . . 789
10.250.2.2 ~RAWCodec() . . . 789
10.250.3 Member Function Documentation . . . 789
10.250.3.1 CanCode() . . . 789
10.250.3.2 CanDecode() . . . 790
10.250.3.3 Clone() . . . 790
10.250.3.4 Code() . . . 790
10.250.3.5 Decode() . . . 790
10.250.3.6 DecodeByStreams() . . . 790
10.250.3.7 DecodeBytes() . . . 791
10.250.3.8 GetHeaderInfo() . . . 791
10.251 gdcm::Reader Class Reference . . . 791
10.251.1 Detailed Description . . . 793
10.251.2 Constructor & Destructor Documentation . . . 794
10.251.2.1 Reader() . . . 794
10.251.2.2 ~Reader() . . . 794
10.251.3 Member Function Documentation . . . 794
10.251.3.1 CanRead() . . . 794
10.251.3.2 GetFile() [1/2] . . . 794
10.251.3.3 GetFile() [2/2] . . . 795
10.251.3.4 GetStreamCurrentPosition() . . . 795
10.251.3.5 GetStreamPtr() . . . 795
10.251.3.6 Read() . . . 795
10.251.3.7 ReadDataSet() . . . 795
10.251.3.8 ReadMetaInformation() . . . 795
10.251.3.9 ReadPreamble() . . . 796
10.251.3.10 ReadSelectedPrivateTags() . . . 796
10.251.3.11 ReadSelectedTags() . . . 796

10.251.3.12 ReadUpToTag() . . . 796
10.251.3.13 SetFile() . . . 796
10.251.3.14 SetFileName() . . . 796
10.251.3.15 SetStream() . . . 797
10.251.4 Friends And Related Function Documentation . . . 797
10.251.4.1 StreamImageReader . . . 797
10.251.5 Member Data Documentation . . . 797
10.251.5.1 F . . . 797
10.252 gdcm::RealWorldValueMappingContent Struct Reference . . . 798
10.252.1 Member Data Documentation . . . 798
10.252.1.1 CodeMeaning . . . 798
10.252.1.2 CodeValue . . . 798
10.252.1.3 RealWorldValueIntercept . . . 799
10.252.1.4 RealWorldValueSlope . . . 799
10.253 gdcm::Region Class Reference . . . 799
10.253.1 Detailed Description . . . 800
10.253.2 Constructor & Destructor Documentation . . . 800
10.253.2.1 Region() . . . 800
10.253.2.2 ~Region() . . . 800
10.253.3 Member Function Documentation . . . 800
10.253.3.1 Area() . . . 800
10.253.3.2 Clone() . . . 800
10.253.3.3 ComputeBoundingBox() . . . 800
10.253.3.4 Empty() . . . 801
10.253.3.5 IsValid() . . . 801
10.253.3.6 Print() . . . 801
10.254 gdcm::Rescaler Class Reference . . . 801
10.254.1 Detailed Description . . . 802

10.254.2 Constructor & Destructor Documentation . . . 803
10.254.2.1 Rescaler() . . . 803
10.254.2.2 ~Rescaler() . . . 803
10.254.3 Member Function Documentation . . . 803
10.254.3.1 ComputeInterceptSlopePixelType() . . . 803
10.254.3.2 ComputePixelTypeFromMinMax() . . . 803
10.254.3.3 GetIntercept() . . . 804
10.254.3.4 GetSlope() . . . 804
10.254.3.5 InverseRescale() . . . 804
10.254.3.6 InverseRescaleFunctionIntoBestFit() . . . 804
10.254.3.7 Rescale() . . . 804
10.254.3.8 RescaleFunctionIntoBestFit() . . . 804
10.254.3.9 SetIntercept() . . . 805
10.254.3.10 SetMinMaxForPixelType() . . . 805
10.254.3.11 SetPixelFormat() . . . 805
10.254.3.12 SetSlope() . . . 805
10.254.3.13 SetTargetPixelType() . . . 805
10.254.3.14 SetUseTargetPixelType() . . . 805
10.255 gdcm::RLECodec Class Reference . . . 806
10.255.1 Detailed Description . . . 807
10.255.2 Constructor & Destructor Documentation . . . 807
10.255.2.1 RLECodec() . . . 807
10.255.2.2 ~RLECodec() . . . 808
10.255.3 Member Function Documentation . . . 808
10.255.3.1 AppendFrameEncode() . . . 808
10.255.3.2 AppendRowEncode() . . . 808
10.255.3.3 CanCode() . . . 808
10.255.3.4 CanDecode() . . . 808

10.255.3.5 Clone() . . . 809
10.255.3.6 Code() . . . 809
10.255.3.7 Decode() . . . 809
10.255.3.8 DecodeByStreams() . . . 809
10.255.3.9 DecodeExtent() . . . 809
10.255.3.10 GetBufferLength() . . . 810
10.255.3.11 GetHeaderInfo() . . . 810
10.255.3.12 SetFrameEncoder() . . . 810
10.255.3.13 SetRowEncoder() . . . 810
10.255.3.14 SetBufferLength() . . . 810
10.255.3.15 SetLength() . . . 810
10.255.3.16 StartEncode() . . . 810
10.255.3.17 StopEncode() . . . 811
10.255.4 Friends And Related Function Documentation . . . 811
10.255.4.1 ImageRegionReader . . . 811
10.256 gdcm::network::RoleSelectionSub Class Reference . . . 811
10.256.1 Detailed Description . . . 811
10.256.2 Constructor & Destructor Documentation . . . 811
10.256.2.1 RoleSelectionSub() . . . 811
10.256.3 Member Function Documentation . . . 812
10.256.3.1 Print() . . . 812
10.256.3.2 Read() . . . 812
10.256.3.3 SetTuple() . . . 812
10.256.3.4 Size() . . . 812
10.256.3.5 Write() . . . 812
10.257 gdcm::SerieHelper::Rule Struct Reference . . . 813
10.257.1 Member Data Documentation . . . 813
10.257.1.1 elem . . . 813

10.257.1.2 group . . . 813
10.257.1.3 op . . . 813
10.257.1.4 value . . . 814
10.258 gdcm::Scanner Class Reference . . . 814
10.258.1 Detailed Description . . . 816
10.258.2 Member Typedef Documentation . . . 817
10.258.2.1 ConstIterator . . . 817
10.258.2.2 MappingType . . . 817
10.258.2.3 TagToValue . . . 817
10.258.2.4 TagToValueValueType . . . 817
10.258.2.5 ValuesType . . . 818
10.258.3 Constructor & Destructor Documentation . . . 818
10.258.3.1 Scanner() . . . 818
10.258.3.2 ~Scanner() . . . 818
10.258.4 Member Function Documentation . . . 818
10.258.4.1 AddPrivateTag() . . . 818
10.258.4.2 AddSkipTag() . . . 818
10.258.4.3 AddTag() . . . 818
10.258.4.4 Begin() . . . 818
10.258.4.5 ClearSkipTags() . . . 819
10.258.4.6 ClearTags() . . . 819
10.258.4.7 End() . . . 819
10.258.4.8 GetAllFilenamesFromTagToValue() . . . 819
10.258.4.9 GetFilenameFromTagToValue() . . . 819
10.258.4.10 GetFilenames() . . . 819
10.258.4.11 GetKeys() . . . 819
10.258.4.12 GetMapping() . . . 820
10.258.4.13 GetMappingFromTagToValue() . . . 820

10.258.4.14 GetMappings() . . . 820
10.258.4.15 GetOrderedValues() . . . 820
10.258.4.16 GetValue() . . . 820
10.258.4.17 GetValues() [1/2] . . . 821
10.258.4.18 GetValues() [2/2] . . . 821
10.258.4.19 IsKey() . . . 821
10.258.4.20 New() . . . 821
10.258.4.21 Print() . . . 821
10.258.4.22 ProcessPublicTag() . . . 822
10.258.4.23 Scan() . . . 822
10.258.5 Friends And Related Function Documentation . . . 822
10.258.5.1 operator<< . . . 822
10.259 gdcm::Segment Class Reference . . . 822
10.259.1 Detailed Description . . . 824
10.259.2 Member Typedef Documentation . . . 824
10.259.2.1 SurfaceVector . . . 824
10.259.3 Member Enumeration Documentation . . . 824
10.259.3.1 ALGOType . . . 824
10.259.4 Constructor & Destructor Documentation . . . 825
10.259.4.1 Segment() . . . 825
10.259.4.2 ~Segment() . . . 825
10.259.5 Member Function Documentation . . . 825
10.259.5.1 AddSurface() . . . 825
10.259.5.2 GetALGOType() . . . 825
10.259.5.3 GetALGOTypeString() . . . 825
10.259.5.4 GetAnatomicRegion() [1/2] . . . 825
10.259.5.5 GetAnatomicRegion() [2/2] . . . 825
10.259.5.6 GetPropertyCategory() [1/2] . . . 826

10.259.5.7 GetPropertyCategory() [2/2] . . . 826
10.259.5.8 GetPropertyType() [1/2] . . . 826
10.259.5.9 GetPropertyType() [2/2] . . . 826
10.259.5.10 GetSegmentAlgorithmName() . . . 826
10.259.5.11 GetSegmentAlgorithmType() . . . 826
10.259.5.12 GetSegmentDescription() . . . 826
10.259.5.13 GetSegmentLabel() . . . 826
10.259.5.14 GetSegmentNumber() . . . 826
10.259.5.15 GetSurface() . . . 827
10.259.5.16 GetSurfaceCount() . . . 827
10.259.5.17 GetSurfaces() [1/2] . . . 827
10.259.5.18 GetSurfaces() [2/2] . . . 827
10.259.5.19 SetAnatomicRegion() . . . 827
10.259.5.20 SetPropertyCategory() . . . 827
10.259.5.21 SetPropertyType() . . . 827
10.259.5.22 SetSegmentAlgorithmName() . . . 827
10.259.5.23 SetSegmentAlgorithmType() [1/2] . . . 827
10.259.5.24 SetSegmentAlgorithmType() [2/2] . . . 828
10.259.5.25 SetSegmentDescription() . . . 828
10.259.5.26 SetSegmentLabel() . . . 828
10.259.5.27 SetSegmentNumber() . . . 828
10.259.5.28 SetSurfaceCount() . . . 828
10.259.6 Member Data Documentation . . . 828
10.259.6.1 AnatomicRegion . . . 828
10.259.6.2 PropertyCategory . . . 828
10.259.6.3 PropertyType . . . 828
10.259.6.4 SegmentAlgorithmName . . . 829
10.259.6.5 SegmentAlgorithmType . . . 829

10.259.6.6 SegmentDescription . . . 829
10.259.6.7 SegmentLabel . . . 829
10.259.6.8 SegmentNumber . . . 829
10.259.6.9 SurfaceCount . . . 829
10.259.6.10 Surfaces . . . 829
10.260 gdcm::SegmentedPaletteColorLookupTable Class Reference . . . 830
10.260.1 Detailed Description . . . 831
10.260.2 Constructor & Destructor Documentation . . . 831
10.260.2.1 SegmentedPaletteColorLookupTable() . . . 831
10.260.2.2 ~SegmentedPaletteColorLookupTable() . . . 831
10.260.3 Member Function Documentation . . . 831
10.260.3.1 Print() . . . 831
10.260.3.2 SetLUT() . . . 831
10.261 gdcm::SegmentReader Class Reference . . . 832
10.261.1 Detailed Description . . . 833
10.261.2 Member Typedef Documentation . . . 833
10.261.2.1 SegmentMap . . . 833
10.261.2.2 SegmentVector . . . 834
10.261.3 Constructor & Destructor Documentation . . . 834
10.261.3.1 SegmentReader() . . . 834
10.261.3.2 ~SegmentReader() . . . 834
10.261.4 Member Function Documentation . . . 834
10.261.4.1 GetSegments() [1/2] . . . 834
10.261.4.2 GetSegments() [2/2] . . . 834
10.261.4.3 Read() . . . 834
10.261.4.4 ReadSegment() . . . 834
10.261.4.5 ReadSegments() . . . 835
10.261.5 Member Data Documentation . . . 835

10.261.5.1 Segments . . . 835
10.262 gdcm::SegmentWriter Class Reference . . . 835
10.262.1 Detailed Description . . . 836
10.262.2 Member Typedef Documentation . . . 836
10.262.2.1 SegmentVector . . . 836
10.262.3 Constructor & Destructor Documentation . . . 837
10.262.3.1 SegmentWriter() . . . 837
10.262.3.2 ~SegmentWriter() . . . 837
10.262.4 Member Function Documentation . . . 837
10.262.4.1 AddSegment() . . . 837
10.262.4.2 GetNumberOfSegments() . . . 837
10.262.4.3 GetSegment() . . . 837
10.262.4.4 GetSegments() [1/2] . . . 837
10.262.4.5 GetSegments() [2/2] . . . 837
10.262.4.6 PrepareWrite() . . . 837
10.262.4.7 SetNumberOfSegments() . . . 838
10.262.4.8 SetSegments() . . . 838
10.262.4.9 Write() . . . 838
10.262.5 Member Data Documentation . . . 838
10.262.5.1 Segments . . . 838
10.263 gdcm::SequenceOfFragments Class Reference . . . 838
10.263.1 Detailed Description . . . 840
10.263.2 Member Typedef Documentation . . . 841
10.263.2.1 ConstIterator . . . 841
10.263.2.2 FragmentVector . . . 841
10.263.2.3 Iterator . . . 841
10.263.2.4 SizeType . . . 841
10.263.3 Constructor & Destructor Documentation . . . 841

10.263.3.1 SequenceOfFragments() . . . 841
10.263.4 Member Function Documentation . . . 841
10.263.4.1 AddFragment() . . . 841
10.263.4.2 Begin() [1/2] . . . 842
10.263.4.3 Begin() [2/2] . . . 842
10.263.4.4 Clear() . . . 842
10.263.4.5 ComputeByteLength() . . . 842
10.263.4.6 ComputeLength() . . . 842
10.263.4.7 End() [1/2] . . . 842
10.263.4.8 End() [2/2] . . . 842
10.263.4.9 GetBuffer() . . . 842
10.263.4.10 GetFragBuffer() . . . 843
10.263.4.11 GetFragment() . . . 843
10.263.4.12 GetLength() . . . 843
10.263.4.13 GetNumberOfFragments() . . . 843
10.263.4.14 GetTable() [1/2] . . . 843
10.263.4.15 GetTable() [2/2] . . . 843
10.263.4.16 New() . . . 844
10.263.4.17 operator==() . . . 844
10.263.4.18 Print() . . . 844
10.263.4.19 Read() . . . 844
10.263.4.20 ReadPreValue() . . . 844
10.263.4.21 ReadValue() . . . 844
10.263.4.22 SetLength() . . . 845
10.263.4.23 Write() . . . 845
10.263.4.24 WriteBuffer() . . . 845
10.264 gdcm::SequenceOfItems Class Reference . . . 845
10.264.1 Detailed Description . . . 848

10.264.2 Member Typedef Documentation . . . 848
10.264.2.1 ConstIterator . . . 848
10.264.2.2 ItemVector . . . 848
10.264.2.3 Iterator . . . 848
10.264.2.4 SizeType . . . 849
10.264.3 Constructor & Destructor Documentation . . . 849
10.264.3.1 SequenceOfItems() . . . 849
10.264.4 Member Function Documentation . . . 849
10.264.4.1 AddItem() . . . 849
10.264.4.2 AddNewUndefinedLengthItem() . . . 849
10.264.4.3 Begin() [1/2] . . . 849
10.264.4.4 Begin() [2/2] . . . 849
10.264.4.5 Clear() . . . 850
10.264.4.6 ComputeLength() . . . 850
10.264.4.7 End() [1/2] . . . 850
10.264.4.8 End() [2/2] . . . 850
10.264.4.9 FindDataElement() . . . 850
10.264.4.10 GetItem() [1/2] . . . 850
10.264.4.11 GetItem() [2/2] . . . 850
10.264.4.12 GetLength() . . . 851
10.264.4.13 GetNumberOfItems() . . . 851
10.264.4.14 IsUndefinedLength() . . . 851
10.264.4.15 New() . . . 851
10.264.4.16 operator=() . . . 851
10.264.4.17 operator==() . . . 851
10.264.4.18 Print() . . . 852
10.264.4.19 Read() . . . 852
10.264.4.20 RemoveItemByIndex() . . . 852

10.264.4.23 SetLength() . . . 852
10.264.4.24 SetLengthToUndefined() . . . 853
10.264.4.25 SetNumberOfItems() . . . 853
10.264.4.26 Write() . . . 853
10.264.5 Member Data Documentation . . . 853
10.264.5.1 Items . . . 853
10.264.5.2 SequenceLengthField . . . 853
10.265 gdcm::SerieHelper Class Reference . . . 854
10.265.1 Detailed Description . . . 855
10.265.2 Member Typedef Documentation . . . 855
10.265.2.1 SerieRestrictions . . . 855
10.265.2.2 SingleSerieUIDFileSetmap . . . 855
10.265.3 Constructor & Destructor Documentation . . . 855
10.265.3.1 SerieHelper() . . . 855
10.265.3.2 ~SerieHelper() . . . 856
10.265.4 Member Function Documentation . . . 856
10.265.4.1 AddFile() . . . 856
10.265.4.2 AddFileName() . . . 856
10.265.4.3 AddRestriction() [1/3] . . . 856
10.265.4.4 AddRestriction() [2/3] . . . 856
10.265.4.5 AddRestriction() [3/3] . . . 856
10.265.4.6 Clear() . . . 856
10.265.4.7 CreateDefaultUniqueSeriesIdentifier() . . . 856
10.265.4.8 CreateUniqueSeriesIdentifier() . . . 857
10.265.4.9 FileNameOrdering() . . . 857
10.265.4.10 GetFirstSingleSerieUIDFileSet() . . . 857
10.265.4.11 GetNextSingleSerieUIDFileSet() . . . 857
10.265.4.12 ImagePositionPatientOrdering() . . . 857

10.265.4.13 OrderFileList() . . . 857
10.265.4.14 SetDirectory() . . . 857
10.265.4.15 SetLoadMode() . . . 857
10.265.4.16 SetUseSeriesDetails() . . . 858
10.265.4.17 UserOrdering() . . . 858
10.265.5 Member Data Documentation . . . 858
10.265.5.1 ltFileSetHt . . . 858
10.265.5.2 SingleSerieUIDFileSetHT . . . 858
10.266 gdcm::Series Class Reference . . . 858
10.266.1 Detailed Description . . . 858
10.266.2 Constructor & Destructor Documentation . . . 859
10.266.2.1 Series() . . . 859
10.267 gdcm::network::ServiceClassApplicationInformation Class Reference . . . 859
10.267.1 Detailed Description . . . 859
10.267.2 Constructor & Destructor Documentation . . . 859
10.267.2.1 ServiceClassApplicationInformation() . . . 859
10.267.3 Member Function Documentation . . . 859
10.267.3.1 Print() . . . 859
10.267.3.2 Read() . . . 860
10.267.3.3 SetTuple() . . . 860
10.267.3.4 Size() . . . 860
10.267.3.5 Write() . . . 860
10.268 gdcm::ServiceClassUser Class Reference . . . 860
10.268.1 Detailed Description . . . 862
10.268.2 Constructor & Destructor Documentation . . . 862
10.268.2.1 ServiceClassUser() . . . 862
10.268.2.2 ~ServiceClassUser() . . . 863
10.268.3 Member Function Documentation . . . 863

10.268.3.1 GetAETitle() . . . 863
10.268.3.2 GetCalledAETitle() . . . 863
10.268.3.3 GetTimeout() . . . 863
10.268.3.4 InitializeConnection() . . . 863
10.268.3.5 IsPresentationContextAccepted() . . . 863
10.268.3.6 New() . . . 863
10.268.3.7 SendEcho() . . . 864
10.268.3.8 SendFind() . . . 864
10.268.3.9 SendMove() [1/3] . . . 864
10.268.3.10 SendMove() [2/3] . . . 864
10.268.3.11 SendMove() [3/3] . . . 864
10.268.3.12 SendStore() [1/3] . . . 864
10.268.3.13 SendStore() [2/3] . . . 865
10.268.3.14 SendStore() [3/3] . . . 865
10.268.3.15 SetAETitle() . . . 865
10.268.3.16 SetCalledAETitle() . . . 865
10.268.3.17 SetHostname() . . . 865
10.268.3.18 SetPort() . . . 866
10.268.3.19 SetPortSCP() . . . 866
10.268.3.20 SetPresentationContexts() . . . 866
10.268.3.21 SetTimeout() . . . 866
10.268.3.22 StartAssociation() . . . 867
10.268.3.23 StopAssociation() . . . 867
10.269 gdcm::SHA1 Class Reference . . . 867
10.269.1 Detailed Description . . . 868
10.269.2 Constructor & Destructor Documentation . . . 868
10.269.2.1 SHA1() . . . 868
10.269.2.2 ~SHA1() . . . 868

10.269.3 Member Function Documentation . . . 868
10.269.3.1 Compute() . . . 868
10.269.3.2 ComputeFile() . . . 868
10.270 gdcm::SimpleMemberCommand< T > Class Template Reference . . . 869
10.270.1 Detailed Description . . . 871
10.270.2 Member Typedef Documentation . . . 871
10.270.2.1 Self . . . 871
10.270.2.2 TMemberFunctionPointer . . . 871
10.270.3 Constructor & Destructor Documentation . . . 871
10.270.3.1 SimpleMemberCommand() . . . 871
10.270.3.2 ~SimpleMemberCommand() . . . 871
10.270.4 Member Function Documentation . . . 872
10.270.4.1 Execute() [1/2] . . . 872
10.270.4.2 Execute() [2/2] . . . 872
10.270.4.3 New() . . . 872
10.270.4.4 SetCallbackFunction() . . . 872
10.270.5 Member Data Documentation . . . 872
10.270.5.1 m_MemberFunction . . . 872
10.270.5.2 m_This . . . 873
10.271 gdcm::SimpleSubjectWatcher Class Reference . . . 873
10.271.1 Detailed Description . . . 873
10.271.2 Constructor & Destructor Documentation . . . 874
10.271.2.1 SimpleSubjectWatcher() . . . 874
10.271.2.2 ~SimpleSubjectWatcher() . . . 874
10.271.3 Member Function Documentation . . . 874
10.271.3.1 EndFilter() . . . 874
10.271.3.2 ShowAbort() . . . 874
10.271.3.3 ShowAnonymization() . . . 874

10.271.3.4 ShowData() . . . 874
10.271.3.5 ShowDataSet() . . . 874
10.271.3.6 ShowFileName() . . . 875
10.271.3.7 ShowIteration() . . . 875
10.271.3.8 ShowProgress() . . . 875
10.271.3.9 StartFilter() . . . 875
10.271.3.10 TestAbortOff() . . . 875
10.271.3.11 TestAbortOn() . . . 875
10.272 gdcm::SmartPointer< ObjectType > Class Template Reference . . . 876
10.272.1 Detailed Description . . . 877
10.272.2 Constructor & Destructor Documentation . . . 877
10.272.2.1 SmartPointer() [1/4] . . . 877
10.272.2.2 SmartPointer() [2/4] . . . 878
10.272.2.3 SmartPointer() [3/4] . . . 878
10.272.2.4 SmartPointer() [4/4] . . . 878
10.272.2.5 ~SmartPointer() . . . 878
10.272.3 Member Function Documentation . . . 878
10.272.3.1 GetPointer() . . . 878
10.272.3.2 operator ObjectType *() . . . 878
10.272.3.3 operator*() . . . 878
10.272.3.4 operator->() . . . 879
10.272.3.5 operator=() [1/3] . . . 879
10.272.3.6 operator=() [2/3] . . . 879
10.272.3.7 operator=() [3/3] . . . 879
10.273 gdcm::network::SOPClassExtendedNegociationSub Class Reference . . . 879
10.273.1 Detailed Description . . . 880
10.273.2 Constructor & Destructor Documentation . . . 880
10.273.2.1 SOPClassExtendedNegociationSub() . . . 880

10.273.3 Member Function Documentation . . . 880
10.273.3.1 Print() . . . 880
10.273.3.2 Read() . . . 880
10.273.3.3 SetTuple() . . . 880
10.273.3.4 Size() . . . 881
10.273.3.5 Write() . . . 881
10.274 gdcm::SOPClassUIDToIOD Class Reference . . . 881
10.274.1 Detailed Description . . . 881
10.274.2 Member Typedef Documentation . . . 882
10.274.2.1 const . . . 882
10.274.3 Member Function Documentation . . . 882
10.274.3.1 GetIOD() . . . 882
10.274.3.2 GetIODFromSOPClassUID() . . . 882
10.274.3.3 GetNumberOfSOPClassToIOD() . . . 882
10.274.3.4 GetSOPClassUIDFromIOD() . . . 882
10.274.3.5 GetSOPClassUIDToIOD() . . . 882
10.274.3.6 GetSOPClassUIDToIODs() . . . 883
10.275 gdcm::Sorter Class Reference . . . 883
10.275.1 Detailed Description . . . 884
10.275.2 Member Typedef Documentation . . . 885
10.275.2.1 SelectionMap . . . 885
10.275.2.2 SortFunction . . . 885
10.275.3 Constructor & Destructor Documentation . . . 885
10.275.3.1 Sorter() . . . 885
10.275.3.2 ~Sorter() . . . 885
10.275.4 Member Function Documentation . . . 885
10.275.4.1 AddSelect() . . . 885
10.275.4.2 GetFilenames() . . . 885

10.275.4.3 Print() . . . 886
10.275.4.4 SetSortFunction() . . . 886
10.275.4.5 Sort() . . . 886
10.275.4.6 StableSort() . . . 886
10.275.5 Friends And Related Function Documentation . . . 887
10.275.5.1 operator<< . . . 887
10.275.6 Member Data Documentation . . . 887
10.275.6.1 Filenames . . . 887
10.275.6.2 Selection . . . 887
10.275.6.3 SortFunc . . . 887
10.276 gdcm::Spacing Class Reference . . . 887
10.276.1 Detailed Description . . . 888
10.276.2 Member Enumeration Documentation . . . 889
10.276.2.1 SpacingType . . . 889
10.276.3 Constructor & Destructor Documentation . . . 890
10.276.3.1 Spacing() . . . 890
10.276.3.2 ~Spacing() . . . 890
10.276.4 Member Function Documentation . . . 890
10.276.4.1 ComputePixelAspectRatioFromPixelSpacing() . . . 890
10.277 gdcm::Spectroscopy Class Reference . . . 890
10.277.1 Detailed Description . . . 891
10.277.2 Constructor & Destructor Documentation . . . 891
10.277.2.1 Spectroscopy() . . . 891
10.278 gdcm::SplitMosaicFilter Class Reference . . . 891
10.278.1 Detailed Description . . . 891
10.278.2 Constructor & Destructor Documentation . . . 892
10.278.2.1 SplitMosaicFilter() . . . 892
10.278.2.2 ~SplitMosaicFilter() . . . 892

10.278.3 Member Function Documentation . . . 892
10.278.3.1 ComputeMOSAICDimensions() . . . 892
10.278.3.2 GetFile() [1/2] . . . 892
10.278.3.3 GetFile() [2/2] . . . 892
10.278.3.4 GetImage() [1/2] . . . 892
10.278.3.5 GetImage() [2/2] . . . 892
10.278.3.6 SetFile() . . . 892
10.278.3.7 SetImage() . . . 893
10.278.3.8 Split() . . . 893
10.279 gdcm::StartEvent Class Reference . . . 893
10.280 gdcm::static_assert_test< x > Struct Template Reference . . . 894
10.281 gdcm::STATIC_ASSERTION_FAILURE< x > Struct Template Reference . . . 894
10.282 gdcm::STATIC_ASSERTION_FAILURE< true > Struct Template Reference . . . 895
10.282.1 Member Enumeration Documentation . . . 895
10.282.1.1 anonymous enum . . . 895
10.283 gdcm::StreamImageReader Class Reference . . . 895
10.283.1 Detailed Description . . . 896
10.283.2 Constructor & Destructor Documentation . . . 896
10.283.2.1 StreamImageReader() . . . 896
10.283.2.2 ~StreamImageReader() . . . 896
10.283.3 Member Function Documentation . . . 896
10.283.3.1 CanReadImage() . . . 896
10.283.3.2 DefinePixelExtent() . . . 897
10.283.3.3 DefineProperBufferLength() . . . 897
10.283.3.4 GetDimensionsValueForResolution() . . . 897
10.283.3.5 GetFile() . . . 897
10.283.3.6 Read() . . . 898
10.283.3.7 ReadImageInformation() . . . 898

10.283.3.8 SetFileName() . . . 898
10.283.3.9 SetStream() . . . 899
10.284 gdcm::StreamImageWriter Class Reference . . . 899
10.284.1 Detailed Description . . . 901
10.284.2 Constructor & Destructor Documentation . . . 901
10.284.2.1 StreamImageWriter() . . . 901
10.284.2.2 ~StreamImageWriter() . . . 902
10.284.3 Member Function Documentation . . . 902
10.284.3.1 CanWriteFile() . . . 902
10.284.3.2 DefinePixelExtent() . . . 902
10.284.3.3 DefineProperBufferLength() . . . 902
10.284.3.4 SetFile() . . . 903
10.284.3.5 SetFileName() . . . 903
10.284.3.6 SetStream() . . . 903
10.284.3.7 Write() . . . 903
10.284.3.8 WriteImageInformation() . . . 904
10.284.3.9 WriteImageSubregionRAW() . . . 904
10.284.3.10 WriteRawHeader() . . . 904
10.284.4 Member Data Documentation . . . 904
10.284.4.1 mElementOffsets . . . 904
10.284.4.2 mElementOffsets1 . . . 904
10.284.4.3 mspFile . . . 905
10.284.4.4 mWriter . . . 905
10.284.4.5 mXMax . . . 905
10.284.4.6 mXMin . . . 905
10.284.4.7 mYMax . . . 905
10.284.4.8 mYMin . . . 905
10.284.4.9 mZMax . . . 905

10.284.4.10 ZMin . . . 905
10.285 gdcm::StrictScanner Class Reference . . . 906
10.285.1 Detailed Description . . . 908
10.285.2 Member Typedef Documentation . . . 908
10.285.2.1 ConstIterator . . . 908
10.285.2.2 MappingType . . . 909
10.285.2.3 TagToValue . . . 909
10.285.2.4 TagToValueValueType . . . 909
10.285.2.5 ValuesType . . . 909
10.285.3 Constructor & Destructor Documentation . . . 909
10.285.3.1 StrictScanner() . . . 909
10.285.3.2 ~StrictScanner() . . . 909
10.285.4 Member Function Documentation . . . 909
10.285.4.1 AddPrivateTag() . . . 909
10.285.4.2 AddSkipTag() . . . 910
10.285.4.3 AddTag() . . . 910
10.285.4.4 Begin() . . . 910
10.285.4.5 ClearSkipTags() . . . 910
10.285.4.6 ClearTags() . . . 910
10.285.4.7 End() . . . 910
10.285.4.8 GetAllFilenamesFromTagToValue() . . . 910
10.285.4.9 GetFilenameFromTagToValue() . . . 911
10.285.4.10 GetFilenames() . . . 911
10.285.4.11 GetKeys() . . . 911
10.285.4.12 GetMapping() . . . 911
10.285.4.13 GetMappingFromTagToValue() . . . 911
10.285.4.14 GetMappings() . . . 911
10.285.4.15 GetOrderedValues() . . . 912

10.285.4.16 GetValue() . . . 912
10.285.4.17 GetValues() [1/2] . . . 912
10.285.4.18 GetValues() [2/2] . . . 912
10.285.4.19 IsKey() . . . 912
10.285.4.20 New() . . . 913
10.285.4.21 Print() . . . 913
10.285.4.22 ProcessPublicTag() . . . 913
10.285.4.23 Scan() . . . 913
10.285.5 Friends And Related Function Documentation . . . 913
10.285.5.1 operator<< . . . 913
10.286 gdcm::String< TDelimiter, TMaxLength, TPadChar > Class Template Reference . . . 914
10.286.1 Detailed Description . . . 915
10.286.2 Member Typedef Documentation . . . 916
10.286.2.1 const_iterator . . . 916
10.286.2.2 const_reference . . . 916
10.286.2.3 const_reverse_iterator . . . 916
10.286.2.4 difference_type . . . 916
10.286.2.5 iterator . . . 916
10.286.2.6 pointer . . . 916
10.286.2.7 reference . . . 916
10.286.2.8 reverse_iterator . . . 917
10.286.2.9 size_type . . . 917
10.286.2.10 value_type . . . 917
10.286.3 Constructor & Destructor Documentation . . . 917
10.286.3.1 String() [1/4] . . . 917
10.286.3.2 String() [2/4] . . . 917
10.286.3.3 String() [3/4] . . . 917
10.286.3.4 String() [4/4] . . . 917

10.286.4 Member Function Documentation . . . 918
10.286.4.1 IsValid() . . . 918
10.286.4.2 operator const char *() . . . 918
10.286.4.3 Trim() [1/2] . . . 918
10.286.4.4 Trim() [2/2] . . . 918
10.286.4.5 Truncate() . . . 918
10.287 gdcm::StringFilter Class Reference . . . 919
10.287.1 Detailed Description . . . 919
10.287.2 Constructor & Destructor Documentation . . . 920
10.287.2.1 StringFilter() . . . 920
10.287.2.2 ~StringFilter() . . . 920
10.287.3 Member Function Documentation . . . 920
10.287.3.1 ExecuteQuery() [1/2] . . . 920
10.287.3.2 ExecuteQuery() [2/2] . . . 920
10.287.3.3 FromString() [1/2] . . . 920
10.287.3.4 FromString() [2/2] . . . 920
10.287.3.5 GetFile() [1/2] . . . 921
10.287.3.6 GetFile() [2/2] . . . 921
10.287.3.7 SetDicts() . . . 921
10.287.3.8 SetFile() . . . 921
10.287.3.9 ToString() [1/2] . . . 921
10.287.3.10 ToString() [2/2] . . . 921
10.287.3.11 ToStringPair() [1/3] . . . 922
10.287.3.12 ToStringPair() [2/3] . . . 922
10.287.3.13 ToStringPair() [3/3] . . . 922
10.287.3.14 UseDictAlways() . . . 922
10.288 gdcm::Study Class Reference . . . 922
10.288.1 Detailed Description . . . 923

10.288.2 Constructor & Destructor Documentation . . . 923
10.288.2.1 Study() . . . 923
10.289 gdcm::Subject Class Reference . . . 923
10.289.1 Detailed Description . . . 924
10.289.2 Constructor & Destructor Documentation . . . 924
10.289.2.1 Subject() . . . 924
10.289.2.2 ~Subject() . . . 925
10.289.3 Member Function Documentation . . . 925
10.289.3.1 AddObserver() [1/2] . . . 925
10.289.3.2 AddObserver() [2/2] . . . 925
10.289.3.3 GetCommand() . . . 925
10.289.3.4 HasObserver() . . . 925
10.289.3.5 InvokeEvent() [1/2] . . . 925
10.289.3.6 InvokeEvent() [2/2] . . . 926
10.289.3.7 RemoveAllObservers() . . . 926
10.289.3.8 RemoveObserver() . . . 926
10.290 gdcm::Surface Class Reference . . . 926
10.290.1 Detailed Description . . . 929
10.290.2 Member Enumeration Documentation . . . 929
10.290.2.1 STATES . . . 929
10.290.2.2 VIEWType . . . 929
10.290.3 Constructor & Destructor Documentation . . . 930
10.290.3.1 Surface() . . . 930
10.290.3.2 ~Surface() . . . 930
10.290.4 Member Function Documentation . . . 930
10.290.4.1 GetAlgorithmFamily() [1/2] . . . 930
10.290.4.2 GetAlgorithmFamily() [2/2] . . . 930
10.290.4.3 GetAlgorithmName() . . . 930

10.290.4.4 GetAlgorithmVersion() . . . 930
10.290.4.5 GetAxisOfRotation() . . . 931
10.290.4.6 GetCenterOfRotation() . . . 931
10.290.4.7 GetFiniteVolume() . . . 931
10.290.4.8 GetManifold() . . . 931
10.290.4.9 GetMaximumPointDistance() . . . 931
10.290.4.10 GetMeanPointDistance() . . . 931
10.290.4.11 GetMeshPrimitive() [1/2] . . . 931
10.290.4.12 GetMeshPrimitive() [2/2] . . . 931
10.290.4.13 GetNumberOfSurfacePoints() . . . 932
10.290.4.14 GetNumberOfVectors() . . . 932
10.290.4.15 GetPointCoordinatesData() [1/2] . . . 932
10.290.4.16 GetPointCoordinatesData() [2/2] . . . 932
10.290.4.17 GetPointPositionAccuracy() . . . 932
10.290.4.18 GetPointsBoundingBoxCoordinates() . . . 932
10.290.4.19 GetProcessingAlgorithm() [1/2] . . . 932
10.290.4.20 GetProcessingAlgorithm() [2/2] . . . 932
10.290.4.21 GetRecommendedDisplayCIELabValue() [1/2] . . . 933
10.290.4.22 GetRecommendedDisplayCIELabValue() [2/2] . . . 933
10.290.4.23 GetRecommendedDisplayGrayscaleValue() . . . 933
10.290.4.24 GetRecommendedPresentationOpacity() . . . 933
10.290.4.25 GetRecommendedPresentationType() . . . 933
10.290.4.26 GetSTATES() . . . 933
10.290.4.27 GetSTATESString() . . . 933
10.290.4.28 GetSurfaceComments() . . . 933
10.290.4.29 GetSurfaceNumber() . . . 933
10.290.4.30 GetSurfaceProcessing() . . . 934
10.290.4.31 GetSurfaceProcessingDescription() . . . 934

10.290.4.32 GetSurfaceProcessingRatio() . . . 934
10.290.4.33 GetVectorAccuracy() . . . 934
10.290.4.34 GetVectorCoordinateData() [1/2] . . . 934
10.290.4.35 GetVectorCoordinateData() [2/2] . . . 934
10.290.4.36 GetVectorDimensionality() . . . 934
10.290.4.37 GetVIEWType() . . . 934
10.290.4.38 GetVIEWTypeString() . . . 934
10.290.4.39 SetAlgorithmFamily() . . . 935
10.290.4.40 SetAlgorithmName() . . . 935
10.290.4.41 SetAlgorithmVersion() . . . 935
10.290.4.42 SetAxisOfRotation() . . . 935
10.290.4.43 SetCenterOfRotation() . . . 935
10.290.4.44 SetFiniteVolume() . . . 935
10.290.4.45 SetManifold() . . . 935
10.290.4.46 SetMaximumPointDistance() . . . 935
10.290.4.47 SetMeanPointDistance() . . . 936
10.290.4.48 SetMeshPrimitive() . . . 936
10.290.4.49 SetNumberOfSurfacePoints() . . . 936
10.290.4.50 SetNumberOfVectors() . . . 936
10.290.4.51 SetPointCoordinatesData() . . . 936
10.290.4.52 SetPointPositionAccuracy() . . . 936
10.290.4.53 SetPointsBoundingBoxCoordinates() . . . 936
10.290.4.54 SetProcessingAlgorithm() . . . 936
10.290.4.55 SetRecommendedDisplayCIELabValue() [1/3] . . . 937
10.290.4.56 SetRecommendedDisplayCIELabValue() [2/3] . . . 937
10.290.4.57 SetRecommendedDisplayCIELabValue() [3/3] . . . 937
10.290.4.58 SetRecommendedDisplayGrayscaleValue() . . . 937
10.290.4.59 SetRecommendedPresentationOpacity() . . . 937

10.290.4.60 SetRecommendedPresentationType() . . . 937
10.290.4.61 SetSurfaceComments() . . . 937
10.290.4.62 SetSurfaceNumber() . . . 937
10.290.4.63 SetSurfaceProcessing() . . . 938
10.290.4.64 SetSurfaceProcessingDescription() . . . 938
10.290.4.65 SetSurfaceProcessingRatio() . . . 938
10.290.4.66 SetVectorAccuracy() . . . 938
10.290.4.67 SetVectorCoordinateData() . . . 938
10.290.4.68 SetVectorDimensionality() . . . 938
10.291 gdcm::SurfaceHelper Class Reference . . . 938
10.291.1 Detailed Description . . . 939
10.291.2 Member Typedef Documentation . . . 939
10.291.2.1 ColorArray . . . 939
10.291.3 Member Function Documentation . . . 939
10.291.3.1 RecommendedDisplayCIELabToRGB() [1/2] . . . 939
10.291.3.2 RecommendedDisplayCIELabToRGB() [2/2] . . . 940
10.291.3.3 RGBToRecommendedDisplayCIELab() . . . 940
10.291.3.4 RGBToRecommendedDisplayGrayscale() . . . 941
10.292 gdcm::SurfaceReader Class Reference . . . 941
10.292.1 Detailed Description . . . 943
10.292.2 Constructor & Destructor Documentation . . . 943
10.292.2.1 SurfaceReader() . . . 943
10.292.2.2 ~SurfaceReader() . . . 943
10.292.3 Member Function Documentation . . . 943
10.292.3.1 GetNumberOfSurfaces() . . . 943
10.292.3.2 Read() . . . 944
10.292.3.3 ReadPointMacro() . . . 944
10.292.3.4 ReadSurface() . . . 944

10.292.3.5 ReadSurfaces() . . . 944
10.293 gdcm::SurfaceWriter Class Reference . . . 944
10.293.1 Detailed Description . . . 946
10.293.2 Constructor & Destructor Documentation . . . 946
10.293.2.1 SurfaceWriter() . . . 946
10.293.2.2 ~SurfaceWriter() . . . 946
10.293.3 Member Function Documentation . . . 946
10.293.3.1 ComputeNumberOfSurfaces() . . . 946
10.293.3.2 GetNumberOfSurfaces() . . . 946
10.293.3.3 PrepareWrite() . . . 946
10.293.3.4 PrepareWritePointMacro() . . . 946
10.293.3.5 SetNumberOfSurfaces() . . . 947
10.293.3.6 Write() . . . 947
10.293.4 Member Data Documentation . . . 947
10.293.4.1 NumberOfSurfaces . . . 947
10.294 gdcm::SwapCode Class Reference . . . 947
10.294.1 Detailed Description . . . 948
10.294.2 Member Enumeration Documentation . . . 948
10.294.2.1 SwapCodeType . . . 948
10.294.3 Constructor & Destructor Documentation . . . 948
10.294.3.1 SwapCode() . . . 948
10.294.4 Member Function Documentation . . . 949
10.294.4.1 GetIndex() . . . 949
10.294.4.2 GetSwapCodeString() . . . 949
10.294.4.3 operator SwapCode::SwapCodeType() . . . 949
10.294.5 Friends And Related Function Documentation . . . 949
10.294.5.1 operator<< . . . 949
10.295 gdcm::SwapperDoOp Class Reference . . . 949

10.295.1 Member Function Documentation . . . 950
10.295.1.1 Swap() . . . 950
10.295.1.2 SwapArray() . . . 950
10.296 gdcm::SwapperNoOp Class Reference . . . 950
10.296.1 Detailed Description . . . 950
10.296.2 Member Function Documentation . . . 950
10.296.2.1 Swap() . . . 950
10.296.2.2 SwapArray() . . . 951
10.297 gdcm::System Class Reference . . . 951
10.297.1 Detailed Description . . . 952
10.297.2 Member Function Documentation . . . 952
10.297.2.1 DeleteDirectory() . . . 952
10.297.2.2 EncodeBytes() . . . 952
10.297.2.3 FileExists() . . . 953
10.297.2.4 FileIsDirectory() . . . 953
10.297.2.5 FileIsSymlink() . . . 953
10.297.2.6 FileSize() . . . 953
10.297.2.7 FileTime() . . . 954
10.297.2.8 FormatDateTime() . . . 954
10.297.2.9 GetCurrentDateTime() . . . 954
10.297.2.10 GetCurrentModuleFileName() . . . 954
10.297.2.11 GetCurrentProcessFileName() . . . 954
10.297.2.12 GetCurrentResourcesDirectory() . . . 954
10.297.2.13 GetCWD() . . . 955
10.297.2.14 GetHostName() . . . 955
10.297.2.15 GetLastSystemError() . . . 955
10.297.2.16 GetLocaleCharset() . . . 955
10.297.2.17 GetPermissions() . . . 955

10.297.2.18 GetTimezoneOffsetFromUTC() . . . 955
10.297.2.19 MakeDirectory() . . . 956
10.297.2.20 ParseDateTime() [1/2] . . . 956
10.297.2.21 ParseDateTime() [2/2] . . . 956
10.297.2.22 RemoveFile() . . . 956
10.297.2.23 SetPermissions() . . . 956
10.297.2.24 StrCaseCmp() . . . 957
10.297.2.25 StrNCaseCmp() . . . 957
10.297.2.26 StrSep() . . . 957
10.297.2.27 StrTokR() . . . 957
10.298 gdcm::Table Class Reference . . . 957
10.298.1 Detailed Description . . . 958
10.298.2 Member Typedef Documentation . . . 958
10.298.2.1 MapTableEntry . . . 958
10.298.3 Constructor & Destructor Documentation . . . 958
10.298.3.1 Table() . . . 958
10.298.3.2 ~Table() . . . 958
10.298.4 Member Function Documentation . . . 959
10.298.4.1 GetTableEntry() . . . 959
10.298.4.2 InsertEntry() . . . 959
10.298.5 Friends And Related Function Documentation . . . 959
10.298.5.1 operator<< . . . 959
10.299 gdcm::TableEntry Class Reference . . . 959
10.299.1 Detailed Description . . . 959
10.299.2 Constructor & Destructor Documentation . . . 960
10.299.2.1 TableEntry() . . . 960
10.299.2.2 ~TableEntry() . . . 960
10.300 gdcm::TableReader Class Reference . . . 960

10.300.1 Detailed Description . . . 961
10.300.2 Constructor & Destructor Documentation . . . 961
10.300.2.1 TableReader() . . . 961
10.300.2.2 ~TableReader() . . . 961
10.300.3 Member Function Documentation . . . 961
10.300.3.1 CharacterDataHandler() . . . 961
10.300.3.2 EndElement() . . . 962
10.300.3.3 GetDefs() . . . 962
10.300.3.4 GetFilename() . . . 962
10.300.3.5 HandleIOD() . . . 962
10.300.3.6 HandleIODEntry() . . . 962
10.300.3.7 HandleMacro() . . . 962
10.300.3.8 HandleMacroEntry() . . . 962
10.300.3.9 HandleMacroEntryDescription() . . . 962
10.300.3.10 HandleModule() . . . 963
10.300.3.11 HandleModuleEntry() . . . 963
10.300.3.12 HandleModuleEntryDescription() . . . 963
10.300.3.13 HandleModuleInclude() . . . 963
10.300.3.14 Read() . . . 963
10.300.3.15 SetFilename() . . . 963
10.300.3.16 StartElement() . . . 963
10.301 gdcm::network::TableRow Class Reference . . . 964
10.301.1 Constructor & Destructor Documentation . . . 964
10.301.1.1 TableRow() . . . 964
10.301.1.2 ~TableRow() . . . 965
10.301.2 Member Data Documentation . . . 965
10.301.2.1 transitions . . . 965
10.302 gdcm::Tag Class Reference . . . 965

10.302.1 Detailed Description . . . 967
10.302.2 Constructor & Destructor Documentation . . . 967
10.302.2.1 Tag() [1/3] . . . 967
10.302.2.2 Tag() [2/3] . . . 968
10.302.2.3 Tag() [3/3] . . . 968
10.302.3 Member Function Documentation . . . 968
10.302.3.1 GetElement() . . . 968
10.302.3.2 GetElementTag() . . . 968
10.302.3.3 GetGroup() . . . 968
10.302.3.4 GetLength() . . . 969
10.302.3.5 GetPrivateCreator() . . . 969
10.302.3.6 IsGroupLength() . . . 969
10.302.3.7 IsGroupXX() . . . 969
10.302.3.8 IsIllegal() . . . 969
10.302.3.9 IsPrivate() . . . 969
10.302.3.10 IsPrivateCreator() . . . 970
10.302.3.11 IsPublic() . . . 970
10.302.3.12 operator"!=() . . . 970
10.302.3.13 operator<() . . . 970
10.302.3.14 operator<=() . . . 970
10.302.3.15 operator=() . . . 970
10.302.3.16 operator==() . . . 971
10.302.3.17 operator[]() [1/2] . . . 971
10.302.3.18 operator[]() [2/2] . . . 971
10.302.3.19 PrintAsContinuousString() . . . 971
10.302.3.20 PrintAsContinuousUpperCaseString() . . . 971
10.302.3.21 PrintAsPipeSeparatedString() . . . 971
10.302.3.22 Read() . . . 972

10.302.3.23 ReadFromCommaSeparatedString() . . . 972
10.302.3.24 ReadFromContinuousString() . . . 972
10.302.3.25 ReadFromPipeSeparatedString() . . . 972
10.302.3.26 SetElement() . . . 972
10.302.3.27 SetElementTag() [1/2] . . . 973
10.302.3.28 SetElementTag() [2/2] . . . 973
10.302.3.29 SetGroup() . . . 973
10.302.3.30 SetPrivateCreator() . . . 973
10.302.3.31 Write() . . . 973
10.302.4 Friends And Related Function Documentation . . . 974
10.302.4.1 operator<< . . . 974
10.302.4.2 operator>> . . . 974
10.302.5 Member Data Documentation . . . 974
10.302.5.1 bytes . . . 974
10.302.5.2 tag . . . 974
10.302.5.3 tags . . . 974
10.303 gdcm::TagPath Class Reference . . . 974
10.303.1 Detailed Description . . . 975
10.303.2 Constructor & Destructor Documentation . . . 975
10.303.2.1 TagPath() . . . 975
10.303.2.2 ~TagPath() . . . 975
10.303.3 Member Function Documentation . . . 975
10.303.3.1 ConstructFromString() . . . 975
10.303.3.2 ConstructFromTagList() . . . 976
10.303.3.3 IsValid() . . . 976
10.303.3.4 Print() . . . 976
10.303.3.5 Push() [1/2] . . . 976
10.303.3.6 Push() [2/2] . . . 976

10.304gdcm::Testing Class Reference 976
10.304.1Detailed Description 978
10.304.2Member Typedef Documentation 978
10.304.2.1MD5DataImagesType 978
10.304.2.2MediaStorageDataFilesType 978
10.304.3Constructor & Destructor Documentation 978
10.304.3.1Testing() 978
10.304.3.2~Testing() 978
10.304.4Member Function Documentation 978
10.304.4.1ComputeFileMD5() 978
10.304.4.2ComputeMD5() 979
10.304.4.3GetDataExtraRoot() 979
10.304.4.4GetDataRoot() 979
10.304.4.5GetFileName() 979
10.304.4.6GetFileNames() 980
10.304.4.7GetLossyFlagFromFile() 980
10.304.4.8GetMD5DataImage() 980
10.304.4.9GetMD5DataImages() 980
10.304.4.10GetMD5FromBrokenFile() 980
10.304.4.11GetMD5FromFile() 980
10.304.4.12GetMediaStorageDataFile() 980
10.304.4.13GetMediaStorageDataFiles() 981
10.304.4.14GetMediaStorageFromFile() 981
10.304.4.15GetNumberOfFileNames() 981
10.304.4.16GetNumberOfMD5DataImages() 981
10.304.4.17GetNumberOfMediaStorageDataFiles() 981
10.304.4.18GetPixelSpacingDataRoot() 981
10.304.4.19GetSelectedPrivateGroupOffsetFromFile() 981

10.304.4.20 GetSelectedTagsOffsetFromFile() . . . 982
10.304.4.21 GetSourceDirectory() . . . 982
10.304.4.22 GetStreamOffsetFromFile() . . . 982
10.304.4.23 GetTempDirectory() . . . 982
10.304.4.24 GetTempDirectoryW() . . . 982
10.304.4.25 GetTempFilename() . . . 983
10.304.4.26 GetTempFilenameW() . . . 983
10.304.4.27 Print() . . . 983
10.305 dcm::Trace Class Reference . . . 983
10.305.1 Detailed Description . . . 984
10.305.2 Constructor & Destructor Documentation . . . 985
10.305.2.1 Trace() . . . 985
10.305.2.2 ~Trace() . . . 985
10.305.3 Member Function Documentation . . . 985
10.305.3.1 DebugOff() . . . 985
10.305.3.2 DebugOn() . . . 985
10.305.3.3 ErrorOff() . . . 985
10.305.3.4 ErrorOn() . . . 985
10.305.3.5 GetDebugFlag() . . . 986
10.305.3.6 GetDebugStream() . . . 986
10.305.3.7 GetErrorFlag() . . . 986
10.305.3.8 GetErrorStream() . . . 986
10.305.3.9 GetStream() . . . 986
10.305.3.10 GetWarningFlag() . . . 986
10.305.3.11 GetWarningStream() . . . 986
10.305.3.12 SetDebug() . . . 986
10.305.3.13 SetDebugStream() . . . 987
10.305.3.14 SetError() . . . 987

10.305.3.15 SetErrorStream() . . . 987
10.305.3.16 SetStream() . . . 987
10.305.3.17 SetStreamToFile() . . . 987
10.305.3.18 SetWarning() . . . 988
10.305.3.19 SetWarningStream() . . . 988
10.305.3.20 WarningOff() . . . 988
10.305.3.21 WarningOn() . . . 988
10.306 gdcm::TransferSyntax Class Reference . . . 988
10.306.1 Detailed Description . . . 990
10.306.2 Member Enumeration Documentation . . . 990
10.306.2.1 NegociatedType . . . 990
10.306.2.2 TSType . . . 991
10.306.3 Constructor & Destructor Documentation . . . 991
10.306.3.1 TransferSyntax() . . . 991
10.306.4 Member Function Documentation . . . 991
10.306.4.1 CanStoreLossy() . . . 992
10.306.4.2 GetNegociatedType() . . . 992
10.306.4.3 GetString() . . . 992
10.306.4.4 GetSwapCode() . . . 992
10.306.4.5 GetTSString() . . . 992
10.306.4.6 GetTSType() . . . 992
10.306.4.7 IsEncapsulated() . . . 993
10.306.4.8 IsEncoded() . . . 993
10.306.4.9 IsExplicit() . . . 993
10.306.4.10 IsImplicit() . . . 993
10.306.4.11 IsLossless() . . . 993
10.306.4.12 IsLossy() . . . 993
10.306.4.13 IsValid() . . . 993

10.306.4.14 operator TSType() . . . 993
10.306.5 Friends And Related Function Documentation . . . 994
10.306.5.1 operator<< . . . 994
10.307 gdcm::network::TransferSyntaxSub Class Reference . . . 994
10.307.1 Detailed Description . . . 994
10.307.2 Constructor & Destructor Documentation . . . 994
10.307.2.1 TransferSyntaxSub() . . . 994
10.307.3 Member Function Documentation . . . 995
10.307.3.1 GetName() . . . 995
10.307.3.2 operator==() . . . 995
10.307.3.3 Print() . . . 995
10.307.3.4 Read() . . . 995
10.307.3.5 SetName() . . . 995
10.307.3.6 SetNameFromUID() . . . 995
10.307.3.7 Size() . . . 995
10.307.3.8 Write() . . . 996
10.308 gdcm::network::Transition Struct Reference . . . 996
10.308.1 Constructor & Destructor Documentation . . . 997
10.308.1.1 Transition() [1/2] . . . 997
10.308.1.2 ~Transition() . . . 997
10.308.1.3 Transition() [2/2] . . . 997
10.308.2 Member Function Documentation . . . 997
10.308.2.1 MakeNew() . . . 997
10.308.3 Member Data Documentation . . . 997
10.308.3.1 mAction . . . 997
10.308.3.2 mEnd . . . 998
10.309 gdcm::Type Class Reference . . . 998
10.309.1 Detailed Description . . . 999

10.309.2 Member Enumeration Documentation . . . 999
10.309.2.1 TypeType . . . 999
10.309.3 Constructor & Destructor Documentation . . . 999
10.309.3.1 Type() . . . 999
10.309.4 Member Function Documentation . . . 999
10.309.4.1 GetTypeString() . . . 1000
10.309.4.2 GetTypeType() . . . 1000
10.309.4.3 operator TypeType() . . . 1000
10.309.5 Friends And Related Function Documentation . . . 1000
10.309.5.1 operator<< . . . 1000
10.310 gdcm::UI Struct Reference . . . 1000
10.310.1 Friends And Related Function Documentation . . . 1001
10.310.1.1 operator<< . . . 1001
10.310.2 Member Data Documentation . . . 1001
10.310.2.1 Internal . . . 1001
10.311 gdcm::UIDGenerator Class Reference . . . 1001
10.311.1 Detailed Description . . . 1002
10.311.2 Constructor & Destructor Documentation . . . 1002
10.311.2.1 UIDGenerator() . . . 1002
10.311.3 Member Function Documentation . . . 1002
10.311.3.1 Generate() . . . 1002
10.311.3.2 GenerateUUID() . . . 1002
10.311.3.3 GetGDCMUID() . . . 1003
10.311.3.4 GetRoot() . . . 1003
10.311.3.5 IsValid() . . . 1003
10.311.3.6 SetRoot() . . . 1003
10.312 gdcm::UIDs Class Reference . . . 1004
10.312.1 Detailed Description . . . 1014

10.312.2 Member Typedef Documentation . . . 1014
10.312.2.1 TransferSyntaxStringsType . . . 1014
10.312.3 Member Enumeration Documentation . . . 1015
10.312.3.1 TSName . . . 1015
10.312.3.2 TSType . . . 1021
10.312.4 Member Function Documentation . . . 1027
10.312.4.1 GetName() . . . 1027
10.312.4.2 GetNumberOfTransferSyntaxStrings() . . . 1028
10.312.4.3 GetString() . . . 1028
10.312.4.4 GetTransferSyntaxString() . . . 1028
10.312.4.5 GetTransferSyntaxStrings() . . . 1028
10.312.4.6 GetUIDName() . . . 1028
10.312.4.7 GetUIDString() . . . 1028
10.312.4.8 operator TSType() . . . 1028
10.312.4.9 SetFromUID() . . . 1029
10.313 gdcm::network::ULAction Class Reference . . . 1029
10.313.1 Detailed Description . . . 1031
10.313.2 Constructor & Destructor Documentation . . . 1031
10.313.2.1 ULAction() . . . 1031
10.313.2.2 ~ULAction() . . . 1031
10.313.3 Member Function Documentation . . . 1032
10.313.3.1 PerformAction() . . . 1032
10.314 gdcm::network::ULActionAA1 Class Reference . . . 1032
10.314.1 Member Function Documentation . . . 1033
10.314.1.1 PerformAction() . . . 1033
10.315 gdcm::network::ULActionAA2 Class Reference . . . 1033
10.315.1 Member Function Documentation . . . 1034
10.315.1.1 PerformAction() . . . 1034

10.316gdcm::network::ULActionAA3 Class Reference 1035
10.316.1Member Function Documentation 1035
10.316.1.1PerformAction() 1035
10.317gdcm::network::ULActionAA4 Class Reference 1036
10.317.1Member Function Documentation 1037
10.317.1.1PerformAction() 1037
10.318gdcm::network::ULActionAA5 Class Reference 1037
10.318.1Member Function Documentation 1038
10.318.1.1PerformAction() 1038
10.319gdcm::network::ULActionAA6 Class Reference 1038
10.319.1Member Function Documentation 1039
10.319.1.1PerformAction() 1039
10.320gdcm::network::ULActionAA7 Class Reference 1040
10.320.1Member Function Documentation 1040
10.320.1.1PerformAction() 1040
10.321gdcm::network::ULActionAA8 Class Reference 1041
10.321.1Member Function Documentation 1042
10.321.1.1PerformAction() 1042
10.322gdcm::network::ULActionAE1 Class Reference 1042
10.322.1Member Function Documentation 1043
10.322.1.1PerformAction() 1043
10.323gdcm::network::ULActionAE2 Class Reference 1043
10.323.1Member Function Documentation 1044
10.323.1.1PerformAction() 1044
10.324gdcm::network::ULActionAE3 Class Reference 1045
10.324.1Member Function Documentation 1045
10.324.1.1PerformAction() 1045
10.325gdcm::network::ULActionAE4 Class Reference 1046

10.325.1 Member Function Documentation 1047
10.325.1.1 PerformAction() 1047
10.326 gdcm::network::ULActionAE5 Class Reference 1047
10.326.1 Member Function Documentation 1048
10.326.1.1 PerformAction() 1048
10.327 gdcm::network::ULActionAE6 Class Reference 1048
10.327.1 Member Function Documentation 1049
10.327.1.1 PerformAction() 1049
10.328 gdcm::network::ULActionAE7 Class Reference 1050
10.328.1 Member Function Documentation 1050
10.328.1.1 PerformAction() 1050
10.329 gdcm::network::ULActionAE8 Class Reference 1051
10.329.1 Member Function Documentation 1052
10.329.1.1 PerformAction() 1052
10.330 gdcm::network::ULActionAR1 Class Reference 1052
10.330.1 Member Function Documentation 1053
10.330.1.1 PerformAction() 1053
10.331 gdcm::network::ULActionAR10 Class Reference 1053
10.331.1 Member Function Documentation 1054
10.331.1.1 PerformAction() 1054
10.332 gdcm::network::ULActionAR2 Class Reference 1055
10.332.1 Member Function Documentation 1055
10.332.1.1 PerformAction() 1055
10.333 gdcm::network::ULActionAR3 Class Reference 1056
10.333.1 Member Function Documentation 1057
10.333.1.1 PerformAction() 1057
10.334 gdcm::network::ULActionAR4 Class Reference 1057
10.334.1 Member Function Documentation 1058

10.334.1.1PerformAction() . . . 1058
10.335gdcm::network::ULActionAR5 Class Reference . . . 1058
10.335.1Member Function Documentation . . . 1059
10.335.1.1PerformAction() . . . 1059
10.336gdcm::network::ULActionAR6 Class Reference . . . 1060
10.336.1Member Function Documentation . . . 1060
10.336.1.1PerformAction() . . . 1060
10.337gdcm::network::ULActionAR7 Class Reference . . . 1061
10.337.1Member Function Documentation . . . 1062
10.337.1.1PerformAction() . . . 1062
10.338gdcm::network::ULActionAR8 Class Reference . . . 1062
10.338.1Member Function Documentation . . . 1063
10.338.1.1PerformAction() . . . 1063
10.339gdcm::network::ULActionAR9 Class Reference . . . 1063
10.339.1Member Function Documentation . . . 1064
10.339.1.1PerformAction() . . . 1064
10.340gdcm::network::ULActionDT1 Class Reference . . . 1065
10.340.1Member Function Documentation . . . 1065
10.340.1.1PerformAction() . . . 1065
10.341gdcm::network::ULActionDT2 Class Reference . . . 1066
10.341.1Member Function Documentation . . . 1067
10.341.1.1PerformAction() . . . 1067
10.342gdcm::network::ULBasicCallback Class Reference . . . 1067
10.342.1Detailed Description . . . 1068
10.342.2Constructor & Destructor Documentation . . . 1068
10.342.2.1ULBasicCallback() . . . 1068
10.342.2.2~ULBasicCallback() . . . 1068
10.342.3Member Function Documentation . . . 1069

10.342.3.1 GetDataSets() . . . 1069
10.342.3.2 GetResponses() . . . 1069
10.342.3.3 HandleDataSet() . . . 1069
10.342.3.4 HandleResponse() . . . 1069
10.343 dcm::network::ULConnection Class Reference . . . 1069
10.343.1 Detailed Description . . . 1070
10.343.2 Constructor & Destructor Documentation . . . 1071
10.343.2.1 ULConnection() . . . 1071
10.343.2.2 ~ULConnection() . . . 1071
10.343.3 Member Function Documentation . . . 1071
10.343.3.1 AddAcceptedPresentationContext() . . . 1071
10.343.3.2 FindContext() . . . 1071
10.343.3.3 GetAcceptedPresentationContexts() [1/2] . . . 1071
10.343.3.4 GetAcceptedPresentationContexts() [2/2] . . . 1071
10.343.3.5 GetConnectionInfo() . . . 1071
10.343.3.6 GetMaxPDUSize() . . . 1071
10.343.3.7 GetPresentationContextACByID() . . . 1072
10.343.3.8 GetPresentationContextIDFromPresentationContext() . . . 1072
10.343.3.9 GetPresentationContextRQByID() . . . 1072
10.343.3.10 GetPresentationContexts() . . . 1072
10.343.3.11 GetProtocol() . . . 1072
10.343.3.12 GetState() . . . 1072
10.343.3.13 GetTimer() . . . 1072
10.343.3.14 InitializeConnection() . . . 1072
10.343.3.15 InitializeIncomingConnection() . . . 1073
10.343.3.16 SetMaxPDUSize() . . . 1073
10.343.3.17 SetPresentationContexts() [1/2] . . . 1073
10.343.3.18 SetPresentationContexts() [2/2] . . . 1073

10.343.3.1 SetState() . . . 1073
10.343.3.2 StopProtocol() . . . 1073
10.343.4 Friends And Related Function Documentation . . . 1073
10.343.4.1 ULActionAE6 . . . 1073
10.343.4.2 ULConnectionManager . . . 1073
10.344 dcm::network::ULConnectionCallback Class Reference . . . 1074
10.344.1 Detailed Description . . . 1074
10.344.2 Constructor & Destructor Documentation . . . 1075
10.344.2.1 ULConnectionCallback() . . . 1075
10.344.2.2 ~ULConnectionCallback() . . . 1075
10.344.3 Member Function Documentation . . . 1075
10.344.3.1 DataSetHandled() . . . 1075
10.344.3.2 DataSetHandles() . . . 1075
10.344.3.3 HandleDataSet() . . . 1075
10.344.3.4 HandleResponse() . . . 1075
10.344.3.5 ResetHandledDataSet() . . . 1075
10.344.3.6 SetImplicitFlag() . . . 1076
10.344.4 Member Data Documentation . . . 1076
10.344.4.1 mImplicit . . . 1076
10.345 dcm::network::ULConnectionInfo Class Reference . . . 1076
10.345.1 Detailed Description . . . 1076
10.345.2 Constructor & Destructor Documentation . . . 1077
10.345.2.1 ULConnectionInfo() . . . 1077
10.345.3 Member Function Documentation . . . 1077
10.345.3.1 GetCalledAETitle() . . . 1077
10.345.3.2 GetCalledComputerName() . . . 1077
10.345.3.3 GetCalledIPAddress() . . . 1077
10.345.3.4 GetCalledIPPort() . . . 1077

10.345.3.5 GetCallingAETitle() . . . 1077
10.345.3.6 GetMaxPDULength() . . . 1077
10.345.3.7 Initialize() . . . 1077
10.345.3.8 SetMaxPDULength() . . . 1078
10.346 gdcm::network::ULConnectionManager Class Reference . . . 1078
10.346.1 Detailed Description . . . 1080
10.346.2 Constructor & Destructor Documentation . . . 1080
10.346.2.1 ULConnectionManager() [1/2] . . . 1080
10.346.2.2 ULConnectionManager() [2/2] . . . 1080
10.346.2.3 ~ULConnectionManager() . . . 1080
10.346.3 Member Function Documentation . . . 1080
10.346.3.1 BreakConnection() . . . 1080
10.346.3.2 BreakConnectionNow() . . . 1080
10.346.3.3 EstablishConnection() . . . 1081
10.346.3.4 EstablishConnectionMove() . . . 1081
10.346.3.5 RunEventLoop() . . . 1081
10.346.3.6 RunMoveEventLoop() . . . 1081
10.346.3.7 SendEcho() . . . 1081
10.346.3.8 SendFind() [1/2] . . . 1082
10.346.3.9 SendFind() [2/2] . . . 1082
10.346.3.10 SendMove() [1/2] . . . 1082
10.346.3.11 SendMove() [2/2] . . . 1082
10.346.3.12 SendNAction() [1/2] . . . 1082
10.346.3.13 SendNAction() [2/2] . . . 1082
10.346.3.14 SendNCreate() [1/2] . . . 1082
10.346.3.15 SendNCreate() [2/2] . . . 1083
10.346.3.16 SendNDelete() [1/2] . . . 1083
10.346.3.17 SendNDelete() [2/2] . . . 1083

10.346.3.18 SendNEventReport() [1/2] . . . 1083
10.346.3.19 SendNEventReport() [2/2] . . . 1083
10.346.3.20 SendNGet() [1/2] . . . 1083
10.346.3.21 SendNGet() [2/2] . . . 1083
10.346.3.22 SendNSet() [1/2] . . . 1083
10.346.3.23 SendNSet() [2/2] . . . 1084
10.346.3.24 SendStore() [1/2] . . . 1084
10.346.3.25 SendStore() [2/2] . . . 1084
10.346.4 Member Data Documentation . . . 1084
10.346.4.1 mConnection . . . 1084
10.346.4.2 mSecondaryConnection . . . 1084
10.346.4.3 mTransitions . . . 1084
10.347 gdcm::network::ULEvent Class Reference . . . 1085
10.347.1 Detailed Description . . . 1085
10.347.2 Constructor & Destructor Documentation . . . 1085
10.347.2.1 ULEvent() [1/2] . . . 1085
10.347.2.2 ULEvent() [2/2] . . . 1085
10.347.2.3 ~ULEvent() . . . 1086
10.347.3 Member Function Documentation . . . 1086
10.347.3.1 GetDataSetPos() . . . 1086
10.347.3.2 GetEvent() . . . 1086
10.347.3.3 GetIStream() . . . 1086
10.347.3.4 GetPDUs() . . . 1086
10.347.3.5 SetEvent() . . . 1086
10.347.3.6 SetPDU() . . . 1086
10.348 gdcm::network::ULTransitionTable Class Reference . . . 1086
10.348.1 Detailed Description . . . 1087
10.348.2 Constructor & Destructor Documentation . . . 1087

10.348.2.1 ULTransitionTable() . . . 1087
10.348.3 Member Function Documentation . . . 1087
10.348.3.1 HandleEvent() . . . 1087
10.348.3.2 PrintTable() . . . 1087
10.349 gdcm::network::ULWritingCallback Class Reference . . . 1088
10.349.1 Constructor & Destructor Documentation . . . 1089
10.349.1.1 ULWritingCallback() . . . 1089
10.349.1.2 ~ULWritingCallback() . . . 1089
10.349.2 Member Function Documentation . . . 1089
10.349.2.1 HandleDataSet() . . . 1089
10.349.2.2 HandleResponse() . . . 1089
10.349.2.3 SetDirectory() . . . 1089
10.350 gdcm::UNExplicitDataElement Class Reference . . . 1090
10.350.1 Detailed Description . . . 1091
10.350.2 Member Function Documentation . . . 1091
10.350.2.1 GetLength() . . . 1091
10.350.2.2 Read() . . . 1091
10.350.2.3 ReadPreValue() . . . 1091
10.350.2.4 ReadValue() . . . 1091
10.350.2.5 ReadWithLength() . . . 1092
10.351 gdcm::UNExplicitImplicitDataElement Class Reference . . . 1092
10.351.1 Detailed Description . . . 1093
10.351.2 Member Function Documentation . . . 1094
10.351.2.1 GetLength() . . . 1094
10.351.2.2 Read() . . . 1094
10.351.2.3 ReadPreValue() . . . 1094
10.351.2.4 ReadValue() . . . 1094
10.352 gdcm::Unpacker12Bits Class Reference . . . 1094

10.352.1 Detailed Description . . . 1095
10.352.2 Member Function Documentation . . . 1095
10.352.2.1 Pack() . . . 1095
10.352.2.2 Unpack() . . . 1095
10.353 gdcm::Usage Class Reference . . . 1095
10.353.1 Detailed Description . . . 1096
10.353.2 Member Enumeration Documentation . . . 1096
10.353.2.1 UsageType . . . 1096
10.353.3 Constructor & Destructor Documentation . . . 1097
10.353.3.1 Usage() . . . 1097
10.353.4 Member Function Documentation . . . 1097
10.353.4.1 GetUsageString() . . . 1097
10.353.4.2 GetUsageType() . . . 1097
10.353.4.3 operator UsageType() . . . 1097
10.353.5 Friends And Related Function Documentation . . . 1097
10.353.5.1 operator<< . . . 1097
10.354 gdcm::UserEvent Class Reference . . . 1098
10.355 gdcm::network::UserInformation Class Reference . . . 1099
10.355.1 Detailed Description . . . 1099
10.355.2 Constructor & Destructor Documentation . . . 1099
10.355.2.1 UserInformation() . . . 1099
10.355.2.2 ~UserInformation() . . . 1099
10.355.3 Member Function Documentation . . . 1100
10.355.3.1 AddRoleSelectionSub() . . . 1100
10.355.3.2 AddSOPClassExtendedNegociationSub() . . . 1100
10.355.3.3 GetMaximumLengthSub() [1/2] . . . 1100
10.355.3.4 GetMaximumLengthSub() [2/2] . . . 1100
10.355.3.5 operator=() . . . 1100

10.355.3.6 Print() . . . 1100
10.355.3.7 Read() . . . 1100
10.355.3.8 Size() . . . 1101
10.355.3.9 Write() . . . 1101
10.356 gdcm::UUIDGenerator Class Reference . . . 1101
10.356.1 Detailed Description . . . 1101
10.356.2 Member Function Documentation . . . 1101
10.356.2.1 Generate() . . . 1101
10.356.2.2 IsValid() . . . 1102
10.357 gdcm::Validate Class Reference . . . 1102
10.357.1 Detailed Description . . . 1103
10.357.2 Constructor & Destructor Documentation . . . 1103
10.357.2.1 Validate() . . . 1103
10.357.2.2 ~Validate() . . . 1103
10.357.3 Member Function Documentation . . . 1103
10.357.3.1 GetValidatedFile() . . . 1103
10.357.3.2 SetFile() . . . 1103
10.357.3.3 Validation() . . . 1103
10.357.4 Member Data Documentation . . . 1103
10.357.4.1 F . . . 1103
10.357.4.2 V . . . 1104
10.358 gdcm::Value Class Reference . . . 1104
10.358.1 Detailed Description . . . 1105
10.358.2 Constructor & Destructor Documentation . . . 1105
10.358.2.1 Value() . . . 1105
10.358.2.2 ~Value() . . . 1105
10.358.3 Member Function Documentation . . . 1105
10.358.3.1 Clear() . . . 1105

10.358.3.2 GetLength() . . . 1106
10.358.3.3 operator==() . . . 1106
10.358.3.4 SetLength() . . . 1106
10.358.3.5 SetLengthOnly() . . . 1106
10.358.4 Friends And Related Function Documentation . . . 1106
10.358.4.1 DataElement . . . 1106
10.359 gdcm::ValueIO< TDE, TSwap, TType > Class Template Reference . . . 1106
10.359.1 Detailed Description . . . 1107
10.359.2 Member Function Documentation . . . 1107
10.359.2.1 Read() . . . 1107
10.359.2.2 Write() . . . 1107
10.360 gdcm::Version Class Reference . . . 1107
10.360.1 Detailed Description . . . 1108
10.360.2 Constructor & Destructor Documentation . . . 1108
10.360.2.1 Version() . . . 1108
10.360.2.2 ~Version() . . . 1108
10.360.3 Member Function Documentation . . . 1108
10.360.3.1 GetBuildVersion() . . . 1108
10.360.3.2 GetMajorVersion() . . . 1108
10.360.3.3 GetMinorVersion() . . . 1108
10.360.3.4 GetVersion() . . . 1109
10.360.3.5 Print() . . . 1109
10.360.4 Friends And Related Function Documentation . . . 1109
10.360.4.1 operator<< . . . 1109
10.361 gdcm::VL Class Reference . . . 1109
10.361.1 Detailed Description . . . 1110
10.361.2 Member Typedef Documentation . . . 1110
10.361.2.1 Type . . . 1110

10.361.3 Constructor & Destructor Documentation . . . 1111
10.361.3.1 VL() . . . 1111
10.361.4 Member Function Documentation . . . 1111
10.361.4.1 GetLength() . . . 1111
10.361.4.2 GetVL16Max() . . . 1111
10.361.4.3 GetVL32Max() . . . 1111
10.361.4.4 IsOdd() . . . 1111
10.361.4.5 IsUndefined() . . . 1111
10.361.4.6 operator uint32_t() . . . 1111
10.361.4.7 operator++() [1/2] . . . 1112
10.361.4.8 operator++() [2/2] . . . 1112
10.361.4.9 operator+=() . . . 1112
10.361.4.10 Read() . . . 1112
10.361.4.11 Read16() . . . 1112
10.361.4.12 SetToUndefined() . . . 1112
10.361.4.13 Write() . . . 1112
10.361.4.14 Write16() . . . 1113
10.361.5 Friends And Related Function Documentation . . . 1113
10.361.5.1 operator<< . . . 1113
10.362 gdcm::VM Class Reference . . . 1113
10.362.1 Detailed Description . . . 1115
10.362.2 Member Enumeration Documentation . . . 1115
10.362.2.1 VMType . . . 1115
10.362.3 Constructor & Destructor Documentation . . . 1116
10.362.3.1 VM() . . . 1116
10.362.4 Member Function Documentation . . . 1116
10.362.4.1 Compatible() . . . 1116
10.362.4.2 GetIndex() . . . 1116

10.362.4.3 GetLength() . . . 1116
10.362.4.4 GetNumberOfElementsFromArray() . . . 1116
10.362.4.5 GetVMString() . . . 1117
10.362.4.6 GetVMType() . . . 1117
10.362.4.7 GetVMTypeFromLength() . . . 1117
10.362.4.8 IsValid() . . . 1117
10.362.4.9 operator VMType() . . . 1117
10.362.5 Friends And Related Function Documentation . . . 1117
10.362.5.1 operator<< . . . 1117
10.363 gdcm::VMToLength< T > Struct Template Reference . . . 1118
10.364 gdcm::VR Class Reference . . . 1118
10.364.1 Detailed Description . . . 1120
10.364.2 Member Enumeration Documentation . . . 1120
10.364.2.1 VRType . . . 1120
10.364.3 Constructor & Destructor Documentation . . . 1121
10.364.3.1 VR() . . . 1121
10.364.4 Member Function Documentation . . . 1121
10.364.4.1 CanDisplay() . . . 1121
10.364.4.2 Compatible() . . . 1121
10.364.4.3 GetLength() [1/2] . . . 1121
10.364.4.4 GetLength() [2/2] . . . 1122
10.364.4.5 GetSize() . . . 1122
10.364.4.6 GetSizeof() . . . 1122
10.364.4.7 GetVRString() . . . 1122
10.364.4.8 GetVRStringFromFile() . . . 1122
10.364.4.9 GetVRType() . . . 1122
10.364.4.10 GetVRTypeFromFile() . . . 1122
10.364.4.11 IsASCII() . . . 1122

10.364.4.12 IsASCII2() . . . 1123
10.364.4.13 IsBinary() . . . 1123
10.364.4.14 IsBinary2() . . . 1123
10.364.4.15 IsDual() . . . 1123
10.364.4.16 IsSwap() . . . 1123
10.364.4.17 IsValid() [1/2] . . . 1123
10.364.4.18 IsValid() [2/2] . . . 1123
10.364.4.19 IsVRFile() . . . 1123
10.364.4.20 operator VRType() . . . 1124
10.364.4.21 Read() . . . 1124
10.364.4.22 Write() . . . 1124
10.364.5 Friends And Related Function Documentation . . . 1124
10.364.5.1 operator<< . . . 1124
10.365 gdcm::VR16ExplicitDataElement Class Reference . . . 1124
10.365.1 Detailed Description . . . 1126
10.365.2 Member Function Documentation . . . 1126
10.365.2.1 GetLength() . . . 1126
10.365.2.2 Read() . . . 1126
10.365.2.3 ReadPreValue() . . . 1126
10.365.2.4 ReadValue() . . . 1126
10.365.2.5 ReadWithLength() . . . 1126
10.366 gdcm::VRToEncoding< T > Struct Template Reference . . . 1127
10.367 gdcm::VRToType< T > Struct Template Reference . . . 1127
10.367.1 Detailed Description . . . 1127
10.368 gdcm::VRVLSize< T > Class Template Reference . . . 1128
10.369 gdcm::VRVLSize< 0 > Class Template Reference . . . 1128
10.369.1 Member Function Documentation . . . 1128
10.369.1.1 Read() . . . 1128

10.369.1.2 Write() . . . 1128
10.370 gdcm::VRVLSize< 1 > Class Template Reference . . . 1128
10.370.1 Member Function Documentation . . . 1129
10.370.1.1 Read() . . . 1129
10.370.1.2 Write() . . . 1129
10.371 vtkGDCMImageReader Class Reference . . . 1129
10.371.1 Detailed Description . . . 1132
10.371.2 Constructor & Destructor Documentation . . . 1132
10.371.2.1 vtkGDCMImageReader() . . . 1132
10.371.2.2 ~vtkGDCMImageReader() . . . 1132
10.371.3 Member Function Documentation . . . 1132
10.371.3.1 CanReadFile() . . . 1132
10.371.3.2 ExecuteData() . . . 1132
10.371.3.3 ExecuteInformation() . . . 1132
10.371.3.4 FillMedicalImageInformation() . . . 1133
10.371.3.5 GetDescriptiveName() . . . 1133
10.371.3.6 GetFileExtensions() . . . 1133
10.371.3.7 GetIconImage() . . . 1133
10.371.3.8 GetOverlay() . . . 1133
10.371.3.9 LoadSingleFile() . . . 1133
10.371.3.10 New() . . . 1133
10.371.3.11 PrintSelf() . . . 1134
10.371.3.12 RequestDataCompat() . . . 1134
10.371.3.13 RequestInformationCompat() . . . 1134
10.371.3.14 SetCurve() . . . 1134
10.371.3.15 SetFileNames() . . . 1134
10.371.3.16 SetFilePattern() . . . 1134
10.371.3.17 SetFilePrefix() . . . 1134

10.371.3.18 SetMedicalImageProperties() 1135
10.371.3.19 vtkBooleanMacro() [1/5] 1135
10.371.3.20 vtkBooleanMacro() [2/5] 1135
10.371.3.21 vtkBooleanMacro() [3/5] 1135
10.371.3.22 vtkBooleanMacro() [4/5] 1135
10.371.3.23 vtkBooleanMacro() [5/5] 1135
10.371.3.24 vtkGetMacro() [1/11] 1135
10.371.3.25 vtkGetMacro() [2/11] 1136
10.371.3.26 vtkGetMacro() [3/11] 1136
10.371.3.27 vtkGetMacro() [4/11] 1136
10.371.3.28 vtkGetMacro() [5/11] 1136
10.371.3.29 vtkGetMacro() [6/11] 1136
10.371.3.30 vtkGetMacro() [7/11] 1136
10.371.3.31 vtkGetMacro() [8/11] 1136
10.371.3.32 vtkGetMacro() [9/11] 1137
10.371.3.33 vtkGetMacro() [10/11] 1137
10.371.3.34 vtkGetMacro() [11/11] 1137
10.371.3.35 vtkGetObjectMacro() [1/4] 1137
10.371.3.36 vtkGetObjectMacro() [2/4] 1137
10.371.3.37 vtkGetObjectMacro() [3/4] 1137
10.371.3.38 vtkGetObjectMacro() [4/4] 1137
10.371.3.39 vtkGetStringMacro() [1/2] 1138
10.371.3.40 vtkGetStringMacro() [2/2] 1138
10.371.3.41 vtkGetVector3Macro() 1138
10.371.3.42 vtkGetVector6Macro() 1138
10.371.3.43 vtkSetMacro() [1/4] 1138
10.371.3.44 vtkSetMacro() [2/4] 1138
10.371.3.45 vtkSetMacro() [3/4] 1138

10.371.3.46 vtkSetMacro() [4/4] . . . 1139
10.371.3.47 vtkSetVector6Macro() . . . 1139
10.371.3.48 vtkTypeRevisionMacro() . . . 1139
10.371.4 Member Data Documentation . . . 1139
10.371.4.1 ApplyInverseVideo . . . 1139
10.371.4.2 ApplyLookupTable . . . 1139
10.371.4.3 ApplyPlanarConfiguration . . . 1139
10.371.4.4 ApplyShiftScale . . . 1139
10.371.4.5 ApplyYBRToRGB . . . 1139
10.371.4.6 Curve . . . 1140
10.371.4.7 DirectionCosines . . . 1140
10.371.4.8 FileNames . . . 1140
10.371.4.9 ForceRescale . . . 1140
10.371.4.10 IconDataScalarType . . . 1140
10.371.4.11 IconImageDataExtent . . . 1140
10.371.4.12 IconNumberOfScalarComponents . . . 1140
10.371.4.13 ImageFormat . . . 1140
10.371.4.14 ImageOrientationPatient . . . 1140
10.371.4.15 ImagePositionPatient . . . 1140
10.371.4.16 LoadIconImage . . . 1141
10.371.4.17 LoadOverlays . . . 1141
10.371.4.18 LossyFlag . . . 1141
10.371.4.19 MedicalImageProperties . . . 1141
10.371.4.20 NumberOfIconImages . . . 1141
10.371.4.21 NumberOfOverlays . . . 1141
10.371.4.22 PlanarConfiguration . . . 1141
10.371.4.23 Scale . . . 1141
10.371.4.24 Shift . . . 1141

10.372vtkGDCMImageReader2 Class Reference . . . 1142
10.372.1Detailed Description . . . 1144
10.372.2Constructor & Destructor Documentation . . . 1144
10.372.2.1vtkGDCMImageReader2() . . . 1144
10.372.2.2~vtkGDCMImageReader2() . . . 1144
10.372.3Member Function Documentation . . . 1144
10.372.3.1CanReadFile() . . . 1144
10.372.3.2FillMedicalImageInformation() . . . 1145
10.372.3.3GetDescriptiveName() . . . 1145
10.372.3.4GetFileExtensions() . . . 1145
10.372.3.5GetIconImage() . . . 1145
10.372.3.6GetIconImagePort() . . . 1145
10.372.3.7GetOverlay() . . . 1145
10.372.3.8GetOverlayPort() . . . 1145
10.372.3.9LoadSingleFile() . . . 1145
10.372.3.10New() . . . 1146
10.372.3.11PrintSelf() . . . 1146
10.372.3.12ProcessRequest() . . . 1146
10.372.3.13RequestData() . . . 1146
10.372.3.14RequestDataCompat() . . . 1146
10.372.3.15RequestInformation() . . . 1146
10.372.3.16RequestInformationCompat() . . . 1146
10.372.3.17SetCurve() . . . 1147
10.372.3.18SetFilePattern() . . . 1147
10.372.3.19SetFilePrefix() . . . 1147
10.372.3.20SetMedicalImageProperties() . . . 1147
10.372.3.21vtkBooleanMacro() [1/5] . . . 1147
10.372.3.22vtkBooleanMacro() [2/5] . . . 1147

10.372.3.23 vtkBooleanMacro() [3/5] . . . 1147
10.372.3.24 vtkBooleanMacro() [4/5] . . . 1147
10.372.3.25 vtkBooleanMacro() [5/5] . . . 1148
10.372.3.26 vtkGetMacro() [1/11] . . . 1148
10.372.3.27 vtkGetMacro() [2/11] . . . 1148
10.372.3.28 vtkGetMacro() [3/11] . . . 1148
10.372.3.29 vtkGetMacro() [4/11] . . . 1148
10.372.3.30 vtkGetMacro() [5/11] . . . 1148
10.372.3.31 vtkGetMacro() [6/11] . . . 1148
10.372.3.32 vtkGetMacro() [7/11] . . . 1149
10.372.3.33 vtkGetMacro() [8/11] . . . 1149
10.372.3.34 vtkGetMacro() [9/11] . . . 1149
10.372.3.35 vtkGetMacro() [10/11] . . . 1149
10.372.3.36 vtkGetMacro() [11/11] . . . 1149
10.372.3.37 vtkGetObjectMacro() [1/2] . . . 1149
10.372.3.38 vtkGetObjectMacro() [2/2] . . . 1149
10.372.3.39 vtkGetStringMacro() [1/2] . . . 1150
10.372.3.40 vtkGetStringMacro() [2/2] . . . 1150
10.372.3.41 vtkGetVector3Macro() . . . 1150
10.372.3.42 vtkGetVector6Macro() . . . 1150
10.372.3.43 vtkSetMacro() [1/4] . . . 1150
10.372.3.44 vtkSetMacro() [2/4] . . . 1150
10.372.3.45 vtkSetMacro() [3/4] . . . 1150
10.372.3.46 vtkSetMacro() [4/4] . . . 1151
10.372.3.47 vtkSetVector6Macro() . . . 1151
10.372.3.48 vtkTypeRevisionMacro() . . . 1151
10.372.4 Member Data Documentation . . . 1151
10.372.4.1 ApplyInverseVideo . . . 1151

10.372.4.2 ApplyLookupTable . . . 1151
10.372.4.3 ApplyPlanarConfiguration . . . 1151
10.372.4.4 ApplyShiftScale . . . 1151
10.372.4.5 ApplyYBRToRGB . . . 1151
10.372.4.6 Curve . . . 1152
10.372.4.7 DirectionCosines . . . 1152
10.372.4.8 ForceRescale . . . 1152
10.372.4.9 IconDataScalarType . . . 1152
10.372.4.10 IconImageDataExtent . . . 1152
10.372.4.11 IconNumberOfScalarComponents . . . 1152
10.372.4.12 ImageFormat . . . 1152
10.372.4.13 ImageOrientationPatient . . . 1152
10.372.4.14 ImagePositionPatient . . . 1152
10.372.4.15 LoadIconImage . . . 1152
10.372.4.16 LoadOverlays . . . 1153
10.372.4.17 LossyFlag . . . 1153
10.372.4.18 NumberOfIconImages . . . 1153
10.372.4.19 NumberOfOverlays . . . 1153
10.372.4.20 PlanarConfiguration . . . 1153
10.372.4.21 Scale . . . 1153
10.372.4.22 Shift . . . 1153
10.373 vtkGDCMImageWriter Class Reference . . . 1154
10.373.1 Detailed Description . . . 1156
10.373.2 Member Enumeration Documentation . . . 1156
10.373.2.1 CompressionTypes . . . 1156
10.373.3 Constructor & Destructor Documentation . . . 1156
10.373.3.1 vtkGDCMImageWriter() . . . 1156
10.373.3.2 ~vtkGDCMImageWriter() . . . 1156

10.373.4 Member Function Documentation . . . 1156
10.373.4.1 GetDescriptiveName() . . . 1156
10.373.4.2 GetFileExtensions() . . . 1156
10.373.4.3 GetFileName() . . . 1157
10.373.4.4 New() . . . 1157
10.373.4.5 PrintSelf() . . . 1157
10.373.4.6 SetDirectionCosines() . . . 1157
10.373.4.7 SetDirectionCosinesFromImageOrientationPatient() . . . 1157
10.373.4.8 SetFileNames() . . . 1157
10.373.4.9 SetMedicalImageProperties() . . . 1158
10.373.4.10 vtkBooleanMacro() [1/2] . . . 1158
10.373.4.11 vtkBooleanMacro() [2/2] . . . 1158
10.373.4.12 vtkGetMacro() [1/7] . . . 1158
10.373.4.13 vtkGetMacro() [2/7] . . . 1158
10.373.4.14 vtkGetMacro() [3/7] . . . 1158
10.373.4.15 vtkGetMacro() [4/7] . . . 1159
10.373.4.16 vtkGetMacro() [5/7] . . . 1159
10.373.4.17 vtkGetMacro() [6/7] . . . 1159
10.373.4.18 vtkGetMacro() [7/7] . . . 1159
10.373.4.19 vtkGetObjectMacro() [1/3] . . . 1159
10.373.4.20 vtkGetObjectMacro() [2/3] . . . 1159
10.373.4.21 vtkGetObjectMacro() [3/3] . . . 1159
10.373.4.22 vtkGetStringMacro() [1/2] . . . 1160
10.373.4.23 vtkGetStringMacro() [2/2] . . . 1160
10.373.4.24 vtkSetMacro() [1/7] . . . 1160
10.373.4.25 vtkSetMacro() [2/7] . . . 1160
10.373.4.26 vtkSetMacro() [3/7] . . . 1160
10.373.4.27 vtkSetMacro() [4/7] . . . 1160

10.373.4.28 vtkSetMacro() [5/7] . . . 1160
10.373.4.29 vtkSetMacro() [6/7] . . . 1161
10.373.4.30 vtkSetMacro() [7/7] . . . 1161
10.373.4.31 vtkSetStringMacro() [1/2] . . . 1161
10.373.4.32 vtkSetStringMacro() [2/2] . . . 1161
10.373.4.33 vtkTypeRevisionMacro() . . . 1161
10.373.4.34 Write() . . . 1161
10.373.4.35 WriteGDCMData() . . . 1161
10.373.4.36 WriteSlice() . . . 1162
10.374 vtkGDCMMedicalImageProperties Class Reference . . . 1162
10.374.1 Constructor & Destructor Documentation . . . 1163
10.374.1.1 vtkGDCMMedicalImageProperties() . . . 1163
10.374.1.2 ~vtkGDCMMedicalImageProperties() . . . 1163
10.374.2 Member Function Documentation . . . 1163
10.374.2.1 Clear() . . . 1163
10.374.2.2 GetFile() . . . 1164
10.374.2.3 New() . . . 1164
10.374.2.4 PrintSelf() . . . 1164
10.374.2.5 PushBackFile() . . . 1164
10.374.2.6 vtkTypeRevisionMacro() . . . 1164
10.374.3 Friends And Related Function Documentation . . . 1164
10.374.3.1 vtkGDCMImageReader . . . 1164
10.374.3.2 vtkGDCMImageReader2 . . . 1164
10.374.3.3 vtkGDCMImageWriter . . . 1164
10.375 vtkGDCMPolyDataReader Class Reference . . . 1165
10.375.1 Detailed Description . . . 1166
10.375.2 Constructor & Destructor Documentation . . . 1166
10.375.2.1 vtkGDCMPolyDataReader() . . . 1166

10.375.2.2 ~vtkGDCMPolyDataReader() . . . 1166
10.375.3 Member Function Documentation . . . 1166
10.375.3.1 FillMedicalImageInformation() . . . 1166
10.375.3.2 New() . . . 1167
10.375.3.3 PrintSelf() . . . 1167
10.375.3.4 RequestData() . . . 1167
10.375.3.5 RequestData_HemodynamicWaveformStorage() . . . 1167
10.375.3.6 RequestData_RTStructureSetStorage() . . . 1167
10.375.3.7 RequestInformation() . . . 1167
10.375.3.8 RequestInformation_HemodynamicWaveformStorage() . . . 1167
10.375.3.9 RequestInformation_RTStructureSetStorage() . . . 1168
10.375.3.10 vtkGetObjectMacro() [1/2] . . . 1168
10.375.3.11 vtkGetObjectMacro() [2/2] . . . 1168
10.375.3.12 vtkGetStringMacro() . . . 1168
10.375.3.13 vtkSetStringMacro() . . . 1168
10.375.3.14 vtkTypeRevisionMacro() . . . 1168
10.375.4 Member Data Documentation . . . 1168
10.375.4.1 FileName . . . 1168
10.375.4.2 MedicalImageProperties . . . 1169
10.375.4.3 RTStructSetProperties . . . 1169
10.376 vtkGDCMPolyDataWriter Class Reference . . . 1169
10.376.1 Detailed Description . . . 1171
10.376.2 Constructor & Destructor Documentation . . . 1171
10.376.2.1 vtkGDCMPolyDataWriter() . . . 1171
10.376.2.2 ~vtkGDCMPolyDataWriter() . . . 1171
10.376.3 Member Function Documentation . . . 1171
10.376.3.1 InitializeRTStructSet() . . . 1171
10.376.3.2 New() . . . 1171

10.376.3.3 PrintSelf() . . . 1171
10.376.3.4 SetMedicalImageProperties() . . . 1172
10.376.3.5 SetNumberOfInputPorts() . . . 1172
10.376.3.6 SetRTStructSetProperties() . . . 1172
10.376.3.7 vtkTypeRevisionMacro() . . . 1172
10.376.3.8 WriteData() . . . 1172
10.376.3.9 WriteRTSTRUCTData() . . . 1172
10.376.3.10 WriteRTSTRUCTInfo() . . . 1173
10.376.4 Member Data Documentation . . . 1173
10.376.4.1 MedicalImageProperties . . . 1173
10.376.4.2 RTStructSetProperties . . . 1173
10.377 vtkGDCMTesting Class Reference . . . 1173
10.377.1 Detailed Description . . . 1174
10.377.2 Member Typedef Documentation . . . 1175
10.377.2.1 MD5MetaImagesType . . . 1175
10.377.3 Constructor & Destructor Documentation . . . 1175
10.377.3.1 vtkGDCMTesting() . . . 1175
10.377.3.2 ~vtkGDCMTesting() . . . 1175
10.377.4 Member Function Documentation . . . 1175
10.377.4.1 GetGDCMDataRoot() . . . 1175
10.377.4.2 GetMD5MetaImage() . . . 1175
10.377.4.3 GetMHDMD5FromFile() . . . 1175
10.377.4.4 GetNumberOfMD5MetaImages() . . . 1176
10.377.4.5 GetRAWMD5FromFile() . . . 1176
10.377.4.6 GetVTKDataRoot() . . . 1176
10.377.4.7 New() . . . 1176
10.377.4.8 PrintSelf() . . . 1176
10.377.4.9 vtkTypeRevisionMacro() . . . 1176

10.378 vtkGDCMThreadedImageReader Class Reference . . . 1177
10.378.1 Constructor & Destructor Documentation . . . 1178
10.378.1.1 vtkGDCMThreadedImageReader() . . . 1178
10.378.1.2 ~vtkGDCMThreadedImageReader() . . . 1178
10.378.2 Member Function Documentation . . . 1178
10.378.2.1 ExecuteData() . . . 1178
10.378.2.2 ExecuteInformation() . . . 1178
10.378.2.3 New() . . . 1179
10.378.2.4 PrintSelf() . . . 1179
10.378.2.5 ReadFiles() . . . 1179
10.378.2.6 RequestDataCompat() . . . 1179
10.378.2.7 vtkBooleanMacro() . . . 1179
10.378.2.8 vtkGetMacro() . . . 1179
10.378.2.9 vtkSetMacro() [1/3] . . . 1179
10.378.2.10 vtkSetMacro() [2/3] . . . 1180
10.378.2.11 vtkSetMacro() [3/3] . . . 1180
10.378.2.12 vtkTypeRevisionMacro() . . . 1180
10.379 vtkGDCMThreadedImageReader2 Class Reference . . . 1180
10.379.1 Constructor & Destructor Documentation . . . 1182
10.379.1.1 vtkGDCMThreadedImageReader2() . . . 1182
10.379.1.2 ~vtkGDCMThreadedImageReader2() . . . 1182
10.379.2 Member Function Documentation . . . 1182
10.379.2.1 GetFileName() . . . 1182
10.379.2.2 New() . . . 1182
10.379.2.3 PrintSelf() . . . 1182
10.379.2.4 RequestInformation() . . . 1183
10.379.2.5 SetFileName() . . . 1183
10.379.2.6 SetFileNames() . . . 1183

10.379.2.7 SplitExtent() . . . 1183
10.379.2.8 ThreadedRequestData() . . . 1183
10.379.2.9 vtkBooleanMacro() [1/3] . . . 1183
10.379.2.10 vtkBooleanMacro() [2/3] . . . 1184
10.379.2.11 vtkBooleanMacro() [3/3] . . . 1184
10.379.2.12 vtkGetMacro() [1/8] . . . 1184
10.379.2.13 vtkGetMacro() [2/8] . . . 1184
10.379.2.14 vtkGetMacro() [3/8] . . . 1184
10.379.2.15 vtkGetMacro() [4/8] . . . 1184
10.379.2.16 vtkGetMacro() [5/8] . . . 1184
10.379.2.17 vtkGetMacro() [6/8] . . . 1185
10.379.2.18 vtkGetMacro() [7/8] . . . 1185
10.379.2.19 vtkGetMacro() [8/8] . . . 1185
10.379.2.20 vtkGetObjectMacro() . . . 1185
10.379.2.21 vtkGetVector3Macro() [1/2] . . . 1185
10.379.2.22 vtkGetVector3Macro() [2/2] . . . 1185
10.379.2.23 vtkGetVector6Macro() . . . 1185
10.379.2.24 vtkSetMacro() [1/7] . . . 1186
10.379.2.25 vtkSetMacro() [2/7] . . . 1186
10.379.2.26 vtkSetMacro() [3/7] . . . 1186
10.379.2.27 vtkSetMacro() [4/7] . . . 1186
10.379.2.28 vtkSetMacro() [5/7] . . . 1186
10.379.2.29 vtkSetMacro() [6/7] . . . 1186
10.379.2.30 vtkSetMacro() [7/7] . . . 1186
10.379.2.31 vtkSetVector3Macro() [1/2] . . . 1187
10.379.2.32 vtkSetVector3Macro() [2/2] . . . 1187
10.379.2.33 vtkSetVector6Macro() . . . 1187
10.379.2.34 vtkTypeRevisionMacro() . . . 1187

10.380 vtkImageColorViewer Class Reference . . . 1187
10.380.1 Detailed Description . . . 1190
10.380.2 Member Enumeration Documentation . . . 1190
10.380.2.1 anonymous enum . . . 1190
10.380.3 Constructor & Destructor Documentation . . . 1190
10.380.3.1 vtkImageColorViewer() . . . 1190
10.380.3.2 ~vtkImageColorViewer() . . . 1190
10.380.4 Member Function Documentation . . . 1191
10.380.4.1 AddInput() . . . 1191
10.380.4.2 AddInputConnection() . . . 1191
10.380.4.3 GetColorLevel() . . . 1191
10.380.4.4 GetColorWindow() . . . 1191
10.380.4.5 GetInput() . . . 1191
10.380.4.6 GetOffScreenRendering() . . . 1191
10.380.4.7 GetOverlayVisibility() . . . 1191
10.380.4.8 GetPosition() . . . 1191
10.380.4.9 GetSize() . . . 1191
10.380.4.10 GetSliceMax() . . . 1192
10.380.4.11 GetSliceMin() . . . 1192
10.380.4.12 GetSliceRange() [1/3] . . . 1192
10.380.4.13 GetSliceRange() [2/3] . . . 1192
10.380.4.14 GetSliceRange() [3/3] . . . 1192
10.380.4.15 GetWindowName() . . . 1192
10.380.4.16 InstallPipeline() . . . 1192
10.380.4.17 New() . . . 1192
10.380.4.18 PrintSelf() . . . 1193
10.380.4.19 Render() . . . 1193
10.380.4.20 SetColorLevel() . . . 1193

10.380.4.23 SetColorWindow() . . . 1193
10.380.4.24 SetDisplayId() . . . 1193
10.380.4.25 SetInput() . . . 1193
10.380.4.26 SetInputConnection() . . . 1193
10.380.4.27 SetOffScreenRendering() . . . 1194
10.380.4.28 SetOverlayVisibility() . . . 1194
10.380.4.29 SetParentId() . . . 1194
10.380.4.30 SetPosition() [1/2] . . . 1194
10.380.4.31 SetPosition() [2/2] . . . 1194
10.380.4.32 SetRenderer() . . . 1194
10.380.4.33 SetRenderWindow() . . . 1194
10.380.4.34 SetSize() [1/2] . . . 1195
10.380.4.35 SetSize() [2/2] . . . 1195
10.380.4.36 SetSlice() . . . 1195
10.380.4.37 SetSliceOrientation() . . . 1195
10.380.4.38 SetSliceOrientationToXY() . . . 1195
10.380.4.39 SetSliceOrientationToXZ() . . . 1195
10.380.4.40 SetSliceOrientationToYZ() . . . 1196
10.380.4.41 SetupInteractor() . . . 1196
10.380.4.42 SetWindowId() . . . 1196
10.380.4.43 UnInstallPipeline() . . . 1196
10.380.4.44 UpdateDisplayExtent() . . . 1196
10.380.4.45 UpdateOrientation() . . . 1196
10.380.4.46 VTK_LEGACY() [1/4] . . . 1196
10.380.4.47 VTK_LEGACY() [2/4] . . . 1196
10.380.4.48 VTK_LEGACY() [3/4] . . . 1197
10.380.4.49 VTK_LEGACY() [4/4] . . . 1197
10.380.4.50 vtkBooleanMacro() . . . 1197

10.380.4.4 vtkGetMacro() [1/2] . . . 1197
10.380.4.5 vtkGetMacro() [2/2] . . . 1197
10.380.4.6 vtkGetObjectMacro() [1/5] . . . 1197
10.380.4.7 vtkGetObjectMacro() [2/5] . . . 1197
10.380.4.8 vtkGetObjectMacro() [3/5] . . . 1198
10.380.4.9 vtkGetObjectMacro() [4/5] . . . 1198
10.380.4.10 vtkGetObjectMacro() [5/5] . . . 1198
10.380.4.11 vtkTypeRevisionMacro() . . . 1198
10.380.5 Friends And Related Function Documentation . . . 1198
10.380.5.1 vtkImageColorViewerCallback . . . 1198
10.380.6 Member Data Documentation . . . 1198
10.380.6.1 FirstRender . . . 1198
10.380.6.2 ImageActor . . . 1198
10.380.6.3 Interactor . . . 1199
10.380.6.4 InteractorStyle . . . 1199
10.380.6.5 OverlayImageActor . . . 1199
10.380.6.6 Renderer . . . 1199
10.380.6.7 RenderWindow . . . 1199
10.380.6.8 Slice . . . 1199
10.380.6.9 SliceOrientation . . . 1199
10.380.6.10 WindowLevel . . . 1199
10.381 vtkImageMapToColors16 Class Reference . . . 1200
10.381.1 Constructor & Destructor Documentation . . . 1201
10.381.1.1 vtkImageMapToColors16() . . . 1201
10.381.1.2 ~vtkImageMapToColors16() . . . 1201
10.381.2 Member Function Documentation . . . 1201
10.381.2.1 GetMTime() . . . 1201
10.381.2.2 New() . . . 1202

10.381.2.3 PrintSelf() . . . 1202
10.381.2.4 RequestData() . . . 1202
10.381.2.5 RequestInformation() . . . 1202
10.381.2.6 SetLookupTable() . . . 1202
10.381.2.7 SetOutputFormatToLuminance() . . . 1202
10.381.2.8 SetOutputFormatToLuminanceAlpha() . . . 1202
10.381.2.9 SetOutputFormatToRGB() . . . 1202
10.381.2.10 SetOutputFormatToRGBA() . . . 1203
10.381.2.11 ThreadedRequestData() . . . 1203
10.381.2.12 vtkBooleanMacro() . . . 1203
10.381.2.13 vtkGetMacro() [1/3] . . . 1203
10.381.2.14 vtkGetMacro() [2/3] . . . 1203
10.381.2.15 vtkGetMacro() [3/3] . . . 1203
10.381.2.16 vtkGetObjectMacro() . . . 1203
10.381.2.17 vtkSetMacro() [1/3] . . . 1204
10.381.2.18 vtkSetMacro() [2/3] . . . 1204
10.381.2.19 vtkSetMacro() [3/3] . . . 1204
10.381.2.20 vtkTypeRevisionMacro() . . . 1204
10.381.3 Member Data Documentation . . . 1204
10.381.3.1 ActiveComponent . . . 1204
10.381.3.2 DataWasPassed . . . 1204
10.381.3.3 LookupTable . . . 1204
10.381.3.4 OutputFormat . . . 1204
10.381.3.5 PassAlphaToOutput . . . 1205
10.382 vtkImageMapToWindowLevelColors2 Class Reference . . . 1205
10.382.1 Constructor & Destructor Documentation . . . 1206
10.382.1.1 vtkImageMapToWindowLevelColors2() . . . 1206
10.382.1.2 ~vtkImageMapToWindowLevelColors2() . . . 1206

10.382.2 Member Function Documentation . . . 1206
10.382.2.1 New() . . . 1206
10.382.2.2 PrintSelf() . . . 1207
10.382.2.3 RequestData() . . . 1207
10.382.2.4 RequestInformation() . . . 1207
10.382.2.5 ThreadedRequestData() . . . 1207
10.382.2.6 vtkGetMacro() [1/2] . . . 1207
10.382.2.7 vtkGetMacro() [2/2] . . . 1207
10.382.2.8 vtkSetMacro() [1/2] . . . 1208
10.382.2.9 vtkSetMacro() [2/2] . . . 1208
10.382.2.10 vtkTypeRevisionMacro() . . . 1208
10.382.3 Member Data Documentation . . . 1208
10.382.3.1 Level . . . 1208
10.382.3.2 Window . . . 1208
10.383 vtkImagePlanarComponentsToComponents Class Reference . . . 1209
10.383.1 Constructor & Destructor Documentation . . . 1210
10.383.1.1 vtkImagePlanarComponentsToComponents() . . . 1210
10.383.1.2 ~vtkImagePlanarComponentsToComponents() . . . 1210
10.383.2 Member Function Documentation . . . 1210
10.383.2.1 New() . . . 1210
10.383.2.2 PrintSelf() . . . 1210
10.383.2.3 RequestData() . . . 1210
10.383.2.4 vtkTypeRevisionMacro() . . . 1210
10.384 vtkImageRGBToYBR Class Reference . . . 1211
10.384.1 Constructor & Destructor Documentation . . . 1212
10.384.1.1 vtkImageRGBToYBR() . . . 1212
10.384.1.2 ~vtkImageRGBToYBR() . . . 1212
10.384.2 Member Function Documentation . . . 1212

10.384.2.1 New() . . . 1212
10.384.2.2 PrintSelf() . . . 1212
10.384.2.3 ThreadedExecute() . . . 1212
10.384.2.4 vtkTypeRevisionMacro() . . . 1212
10.385 vtkImageYBRToRGB Class Reference . . . 1213
10.385.1 Constructor & Destructor Documentation . . . 1214
10.385.1.1 vtkImageYBRToRGB() . . . 1214
10.385.1.2 ~vtkImageYBRToRGB() . . . 1214
10.385.2 Member Function Documentation . . . 1214
10.385.2.1 New() . . . 1214
10.385.2.2 PrintSelf() . . . 1214
10.385.2.3 ThreadedExecute() . . . 1214
10.385.2.4 vtkTypeRevisionMacro() . . . 1214
10.386 vtkLookupTable16 Class Reference . . . 1215
10.386.1 Constructor & Destructor Documentation . . . 1216
10.386.1.1 vtkLookupTable16() . . . 1216
10.386.1.2 ~vtkLookupTable16() . . . 1216
10.386.2 Member Function Documentation . . . 1216
10.386.2.1 Build() . . . 1216
10.386.2.2 GetPointer() . . . 1216
10.386.2.3 MapScalarsThroughTable2() . . . 1216
10.386.2.4 New() . . . 1217
10.386.2.5 PrintSelf() . . . 1217
10.386.2.6 SetNumberOfTableValues() . . . 1217
10.386.2.7 vtkTypeRevisionMacro() . . . 1217
10.386.2.8 WritePointer() . . . 1217
10.386.3 Member Data Documentation . . . 1217
10.386.3.1 Table16 . . . 1217

10.387 vtkRTStructSetProperties Class Reference . . . 1218
10.387.1 Detailed Description . . . 1220
10.387.2 Constructor & Destructor Documentation . . . 1220
10.387.2.1 vtkRTStructSetProperties() . . . 1220
10.387.2.2 ~vtkRTStructSetProperties() . . . 1220
10.387.3 Member Function Documentation . . . 1220
10.387.3.1 AddContourReferencedFrameOfReference() . . . 1220
10.387.3.2 AddReferencedFrameOfReference() . . . 1220
10.387.3.3 AddStructureSetROI() . . . 1221
10.387.3.4 AddStructureSetROIObservation() . . . 1221
10.387.3.5 Clear() . . . 1221
10.387.3.6 DeepCopy() . . . 1221
10.387.3.7 GetContourReferencedFrameOfReferenceClassUID() . . . 1221
10.387.3.8 GetContourReferencedFrameOfReferenceInstanceUID() . . . 1221
10.387.3.9 GetNumberOfContourReferencedFrameOfReferences() [1/2] . . . 1221
10.387.3.10 GetNumberOfContourReferencedFrameOfReferences() [2/2] . . . 1222
10.387.3.11 GetNumberOfReferencedFrameOfReferences() . . . 1222
10.387.3.12 GetNumberOfStructureSetROIs() . . . 1222
10.387.3.13 GetReferencedFrameOfReferenceClassUID() . . . 1222
10.387.3.14 GetReferencedFrameOfReferenceInstanceUID() . . . 1222
10.387.3.15 GetStructureSetObservationNumber() . . . 1222
10.387.3.16 GetStructureSetROIDescription() . . . 1222
10.387.3.17 GetStructureSetROIGenerationAlgorithm() . . . 1222
10.387.3.18 GetStructureSetROIName() . . . 1223
10.387.3.19 GetStructureSetROINumber() . . . 1223
10.387.3.20 GetStructureSetROIObservationLabel() . . . 1223
10.387.3.21 GetStructureSetROIRefFrameRefUID() . . . 1223
10.387.3.22 GetStructureSetRTROIInterpretedType() . . . 1223

10.387.3.23 New() . . . 1223
10.387.3.24 PrintSelf() . . . 1223
10.387.3.25 vtkGetStringMacro() [1/9] . . . 1224
10.387.3.26 vtkGetStringMacro() [2/9] . . . 1224
10.387.3.27 vtkGetStringMacro() [3/9] . . . 1224
10.387.3.28 vtkGetStringMacro() [4/9] . . . 1224
10.387.3.29 vtkGetStringMacro() [5/9] . . . 1224
10.387.3.30 vtkGetStringMacro() [6/9] . . . 1224
10.387.3.31 vtkGetStringMacro() [7/9] . . . 1224
10.387.3.32 vtkGetStringMacro() [8/9] . . . 1224
10.387.3.33 vtkGetStringMacro() [9/9] . . . 1225
10.387.3.34 vtkSetStringMacro() [1/9] . . . 1225
10.387.3.35 vtkSetStringMacro() [2/9] . . . 1225
10.387.3.36 vtkSetStringMacro() [3/9] . . . 1225
10.387.3.37 vtkSetStringMacro() [4/9] . . . 1225
10.387.3.38 vtkSetStringMacro() [5/9] . . . 1225
10.387.3.39 vtkSetStringMacro() [6/9] . . . 1225
10.387.3.40 vtkSetStringMacro() [7/9] . . . 1225
10.387.3.41 vtkSetStringMacro() [8/9] . . . 1226
10.387.3.42 vtkSetStringMacro() [9/9] . . . 1226
10.387.3.43 vtkTypeRevisionMacro() . . . 1226
10.387.4 Member Data Documentation . . . 1226
10.387.4.1 Internals . . . 1226
10.387.4.2 ReferenceFrameOfReferenceUID . . . 1226
10.387.4.3 ReferenceSeriesInstanceUID . . . 1226
10.387.4.4 SeriesInstanceUID . . . 1226
10.387.4.5 SOPInstanceUID . . . 1226
10.387.4.6 StructureSetDate . . . 1227

10.387.4.7 StructureSetLabel . . . 1227
10.387.4.8 StructureSetName . . . 1227
10.387.4.9 StructureSetTime . . . 1227
10.387.4.10 StudyInstanceUID . . . 1227
10.388 gdcm::Waveform Class Reference . . . 1227
10.388.1 Detailed Description . . . 1227
10.388.2 Constructor & Destructor Documentation . . . 1228
10.388.2.1 Waveform() . . . 1228
10.389 gdcm::WLMFindQuery Class Reference . . . 1228
10.389.1 Detailed Description . . . 1229
10.389.2 Constructor & Destructor Documentation . . . 1229
10.389.2.1 WLMFindQuery() . . . 1229
10.389.3 Member Function Documentation . . . 1230
10.389.3.1 GetAbstractSyntaxUID() . . . 1230
10.389.3.2 GetTagListByLevel() . . . 1230
10.389.3.3 GetValidDataSet() . . . 1230
10.389.3.4 InitializeDataSet() . . . 1230
10.389.3.5 ValidateQuery() . . . 1230
10.389.4 Friends And Related Function Documentation . . . 1231
10.389.4.1 QueryFactory . . . 1231
10.390 gdcm::Writer Class Reference . . . 1231
10.390.1 Detailed Description . . . 1233
10.390.2 Constructor & Destructor Documentation . . . 1234
10.390.2.1 Writer() . . . 1234
10.390.2.2 ~Writer() . . . 1234
10.390.3 Member Function Documentation . . . 1234
10.390.3.1 CheckFileMetaInformationOff() . . . 1234
10.390.3.2 CheckFileMetaInformationOn() . . . 1234

10.390.3.3 GetCheckFileMetaInformation() . . . 1234
10.390.3.4 GetFile() . . . 1235
10.390.3.5 GetStreamPtr() . . . 1235
10.390.3.6 SetCheckFileMetaInformation() . . . 1235
10.390.3.7 SetFile() . . . 1235
10.390.3.8 SetFileName() . . . 1236
10.390.3.9 SetStream() . . . 1236
10.390.3.10 SetWriteDataSetOnly() . . . 1236
10.390.3.11 Write() . . . 1236
10.390.4 Friends And Related Function Documentation . . . 1237
10.390.4.1 StreamImageWriter . . . 1237
10.390.5 Member Data Documentation . . . 1237
10.390.5.1 Ofstream . . . 1237
10.390.5.2 Stream . . . 1237
10.391 gdcm::XMLDictReader Class Reference . . . 1237
10.391.1 Detailed Description . . . 1238
10.391.2 Constructor & Destructor Documentation . . . 1238
10.391.2.1 XMLDictReader() . . . 1238
10.391.2.2 ~XMLDictReader() . . . 1239
10.391.3 Member Function Documentation . . . 1239
10.391.3.1 CharacterDataHandler() . . . 1239
10.391.3.2 EndElement() . . . 1239
10.391.3.3 GetDict() . . . 1239
10.391.3.4 HandleDescription() . . . 1239
10.391.3.5 HandleEntry() . . . 1239
10.391.3.6 StartElement() . . . 1239
10.392 gdcm::XMLPrinter Class Reference . . . 1240
10.392.1 Member Enumeration Documentation . . . 1241

10.392.1.1 PrintStyles . . . 1241
10.392.2 Constructor & Destructor Documentation . . . 1241
10.392.2.1 XMLPrinter() . . . 1241
10.392.2.2 ~XMLPrinter() . . . 1241
10.392.3 Member Function Documentation . . . 1241
10.392.3.1 GetPrintStyle() . . . 1241
10.392.3.2 HandleBulkData() . . . 1241
10.392.3.3 Print() . . . 1242
10.392.3.4 PrintDataElement() . . . 1242
10.392.3.5 PrintDataSet() . . . 1242
10.392.3.6 PrintSQ() . . . 1242
10.392.3.7 SetFile() . . . 1242
10.392.3.8 SetStyle() . . . 1242
10.392.4 Member Data Documentation . . . 1242
10.392.4.1 F . . . 1242
10.392.4.2 PrintStyle . . . 1243
10.393 gdcm::XMLPrivateDictReader Class Reference . . . 1243
10.393.1 Detailed Description . . . 1244
10.393.2 Constructor & Destructor Documentation . . . 1244
10.393.2.1 XMLPrivateDictReader() . . . 1244
10.393.2.2 ~XMLPrivateDictReader() . . . 1244
10.393.3 Member Function Documentation . . . 1244
10.393.3.1 CharacterDataHandler() . . . 1244
10.393.3.2 EndElement() . . . 1245
10.393.3.3 GetPrivateDict() . . . 1245
10.393.3.4 HandleDescription() . . . 1245
10.393.3.5 HandleEntry() . . . 1245
10.393.3.6 StartElement() . . . 1245

11 File Documentation 1247

11.1 gdcmAAbortPDU.h File Reference . . . 1247

11.2 gdcmAAssociateACPDU.h File Reference . . . 1248

11.3 gdcmAAssociateRJPDU.h File Reference . . . 1248

11.4 gdcmAAssociateRQPDU.h File Reference . . . 1249

11.5 gdcmAbstractSyntax.h File Reference . . . 1250

11.6 gdcmAnonymizeEvent.h File Reference . . . 1251

11.7 gdcmAnonymizer.h File Reference . . . 1252

11.8 gdcmApplicationContext.h File Reference . . . 1253

11.9 gdcmApplicationEntity.h File Reference . . . 1254

11.10 gdcmAReleaseRPPDU.h File Reference . . . 1254

11.11 gdcmAReleaseRQPDU.h File Reference . . . 1255

11.12 gdcmARTIMTimer.h File Reference . . . 1256

11.13 gdcmASN1.h File Reference . . . 1257

11.14 gdcmAsynchronousOperationsWindowSub.h File Reference . . . 1258

11.15 gdcmAttribute.h File Reference . . . 1258

11.16 gdcmAudioCodec.h File Reference . . . 1260

11.17 gdcmBase64.h File Reference . . . 1260

11.18 gdcmBaseCompositeMessage.h File Reference . . . 1261

11.19 gdcmBaseNormalizedMessage.h File Reference . . . 1262

11.20 gdcmBasePDU.h File Reference . . . 1263

11.21 gdcmBaseQuery.h File Reference . . . 1264

11.22 gdcmBaseRootQuery.h File Reference . . . 1265

11.23 gdcmBasicOffsetTable.h File Reference . . . 1266

11.24 gdcmBitmap.h File Reference . . . 1267

11.25 gdcmBitmapToBitmapFilter.h File Reference . . . 1268

11.26 gdcmBoxRegion.h File Reference . . . 1269

11.27 gdcmByteBuffer.h File Reference . . . 1269

11.28gdcmByteSwap.h File Reference 1271
11.29gdcmByteSwapFilter.h File Reference 1271
11.30gdcmByteValue.h File Reference 1272
11.31gdcmCAPICryptoFactory.h File Reference 1273
11.32gdcmCAPICryptographicMessageSyntax.h File Reference 1274
11.33gdcmCEchoMessages.h File Reference 1274
11.34gdcmCFindMessages.h File Reference 1275
11.35gdcmCMoveMessages.h File Reference 1276
11.36gdcmCodec.h File Reference 1277
11.37gdcmCoder.h File Reference 1278
11.38gdcmCodeString.h File Reference 1280
11.39gdcmCommand.h File Reference 1281
11.40gdcmCommandDataSet.h File Reference 1282
11.41gdcmCompositeMessageFactory.h File Reference 1283
11.42gdcmCompositeNetworkFunctions.h File Reference 1283
11.43gdcmConstCharWrapper.h File Reference 1284
11.44gdcmCP246ExplicitDataElement.h File Reference 1284
11.45gdcmCryptoFactory.h File Reference 1285
11.46gdcmCryptographicMessageSyntax.h File Reference 1286
11.47gdcmCSAElement.h File Reference 1287
11.48gdcmCSAHeader.h File Reference 1288
11.49gdcmCSAHeaderDict.h File Reference 1289
11.50gdcmCSAHeaderDictEntry.h File Reference 1290
11.51gdcmCStoreMessages.h File Reference 1291
11.52gdcmCurve.h File Reference 1292
11.53gdcmDataElement.h File Reference 1293
11.54gdcmDataEvent.h File Reference 1295
11.55gdcmDataSet.h File Reference 1296

11.56gdcmDataSetEvent.h File Reference . . . 1297
11.57gdcmDataSetHelper.h File Reference . . . 1297
11.58gdcmDecoder.h File Reference . . . 1298
11.59gdcmDefinedTerms.h File Reference . . . 1300
11.60gdcmDeflateStream.h File Reference . . . 1300
11.61gdcmDefs.h File Reference . . . 1301
11.62gdcmDeltaEncodingCodec.h File Reference . . . 1302
11.63gdcmDICOMDIR.h File Reference . . . 1302
11.64gdcmDICOMDIRGenerator.h File Reference . . . 1303
11.65gdcmDict.h File Reference . . . 1304
11.66gdcmDictConverter.h File Reference . . . 1305
11.67gdcmDictEntry.h File Reference . . . 1306
11.68gdcmDictPrinter.h File Reference . . . 1307
11.69gdcmDicts.h File Reference . . . 1308
11.70gdcmDIMSE.h File Reference . . . 1309
11.71gdcmDirectionCosines.h File Reference . . . 1309
11.72gdcmDirectory.h File Reference . . . 1310
11.73gdcmDirectoryHelper.h File Reference . . . 1311
11.74gdcmDummyValueGenerator.h File Reference . . . 1312
11.75gdcmDumper.h File Reference . . . 1312
11.76gdcmElement.h File Reference . . . 1313
11.76.1 Macro Definition Documentation . . . 1315
11.76.1.1 VRDS16ILLEGAL . . . 1315
11.77gdcmEncapsulatedDocument.h File Reference . . . 1315
11.78gdcmEnumeratedValues.h File Reference . . . 1316
11.79gdcmEvent.h File Reference . . . 1316
11.79.1 Macro Definition Documentation . . . 1318
11.79.1.1 gdcmEventMacro . . . 1318

11.80gdcmException.h File Reference 1318
11.81gdcmExplicitDataElement.h File Reference 1319
11.82gdcmExplicitImplicitDataElement.h File Reference 1320
11.83gdcmFiducials.h File Reference 1320
11.84gdcmFile.h File Reference 1321
11.85gdcmFileAnonymizer.h File Reference 1322
11.86gdcmFileChangeTransferSyntax.h File Reference 1323
11.87gdcmFileDecompressLookupTable.h File Reference 1323
11.88gdcmFileDerivation.h File Reference 1324
11.89gdcmFileExplicitFilter.h File Reference 1325
11.90gdcmFileMetaInformation.h File Reference 1326
11.91gdcmFilename.h File Reference 1327
11.92gdcmFileNameEvent.h File Reference 1327
11.93gdcmFilenameGenerator.h File Reference 1328
11.94gdcmFileSet.h File Reference 1329
11.95gdcmFileStreamer.h File Reference 1330
11.96gdcmFindPatientRootQuery.h File Reference 1331
11.97gdcmFindStudyRootQuery.h File Reference 1332
11.98gdcmFragment.h File Reference 1332
11.99gdcmGlobal.h File Reference 1334
11.100gdcmGroupDict.h File Reference 1335
11.101gdcmIconImage.h File Reference 1335
11.102gdcmIconImageFilter.h File Reference 1336
11.103gdcmIconImageGenerator.h File Reference 1337
11.104gdcmImage.h File Reference 1338
11.105gdcmImageApplyLookupTable.h File Reference 1339
11.106gdcmImageChangePhotometricInterpretation.h File Reference 1340
11.107gdcmImageChangePlanarConfiguration.h File Reference 1340

11.108gdcmImageChangeTransferSyntax.h File Reference . . . 1341
11.109gdcmImageCodec.h File Reference . . . 1342
11.110gdcmImageConverter.h File Reference . . . 1343
11.111gdcmImageFragmentSplitter.h File Reference . . . 1343
11.112gdcmImageHelper.h File Reference . . . 1344
11.113gdcmImageReader.h File Reference . . . 1345
11.114gdcmImageRegionReader.h File Reference . . . 1346
11.115gdcmImageToImageFilter.h File Reference . . . 1347
11.116gdcmImageWriter.h File Reference . . . 1348
11.117gdcmImplementationClassUIDSub.h File Reference . . . 1348
11.118gdcmImplementationUIDSub.h File Reference . . . 1349
11.119gdcmImplementationVersionNameSub.h File Reference . . . 1350
11.120gdcmImplicitDataElement.h File Reference . . . 1351
11.121gdcmIOD.h File Reference . . . 1352
11.122gdcmIODEntry.h File Reference . . . 1353
11.123gdcmIODs.h File Reference . . . 1355
11.124gdcmIPPSorter.h File Reference . . . 1356
11.125gdcmItem.h File Reference . . . 1357
11.126gdcmJPEG12Codec.h File Reference . . . 1359
11.127gdcmJPEG16Codec.h File Reference . . . 1359
11.128gdcmJPEG2000Codec.h File Reference . . . 1360
11.129gdcmJPEG8Codec.h File Reference . . . 1361
11.130gdcmJPEGCodec.h File Reference . . . 1362
11.131gdcmJPEGLSCodec.h File Reference . . . 1364
11.132gdcmJSON.h File Reference . . . 1364
11.133gdcmKAKADUCodec.h File Reference . . . 1365
11.134gdcmLegacyMacro.h File Reference . . . 1366
11.134.1Macro Definition Documentation . . . 1367

11.134.1.1 GDCM_LEGACY . . . 1367
11.134.1.2 GDCM_LEGACY_BODY . . . 1367
11.134.1.3 GDCM_LEGACY_REPLACED_BODY . . . 1367
11.135 gdcmLO.h File Reference . . . 1367
11.136 gdcmLookupTable.h File Reference . . . 1368
11.137 gdcmMacro.h File Reference . . . 1369
11.138 gdcmMacroEntry.h File Reference . . . 1371
11.138.1 Macro Definition Documentation . . . 1372
11.138.1.1 GDCMMACROENTRY_H . . . 1372
11.139 gdcmMacros.h File Reference . . . 1372
11.140 gdcmMaximumLengthSub.h File Reference . . . 1374
11.141 gdcmMD5.h File Reference . . . 1375
11.142 gdcmMediaStorage.h File Reference . . . 1376
11.143 gdcmMeshPrimitive.h File Reference . . . 1377
11.144 gdcmModalityPerformedProcedureStepCreateQuery.h File Reference . . . 1378
11.145 gdcmModalityPerformedProcedureStepSetQuery.h File Reference . . . 1379
11.146 gdcmModule.h File Reference . . . 1379
11.147 gdcmModuleEntry.h File Reference . . . 1381
11.148 gdcmModules.h File Reference . . . 1383
11.149 gdcmMovePatientRootQuery.h File Reference . . . 1384
11.150 gdcmMoveStudyRootQuery.h File Reference . . . 1385
11.151 gdcmNActionMessages.h File Reference . . . 1385
11.152 gdcmNCreateMessages.h File Reference . . . 1386
11.153 gdcmNDeleteMessages.h File Reference . . . 1387
11.154 gdcmNestedModuleEntries.h File Reference . . . 1387
11.155 gdcmNetworkEvents.h File Reference . . . 1389
11.156 gdcmNetworkStateID.h File Reference . . . 1390
11.157 gdcmNEventReportMessages.h File Reference . . . 1391

11.158gdcmNGetMessages.h File Reference . . . 1391
11.159gdcmNormalizedMessageFactory.h File Reference . . . 1392
11.160gdcmNormalizedNetworkFunctions.h File Reference . . . 1393
11.161gdcmNSetMessages.h File Reference . . . 1393
11.162gdcmObject.h File Reference . . . 1394
11.163gdcmOpenSSLCryptoFactory.h File Reference . . . 1395
11.164gdcmOpenSSLCryptographicMessageSyntax.h File Reference . . . 1396
11.165gdcmOpenSSLP7CryptoFactory.h File Reference . . . 1397
11.166gdcmOpenSSLP7CryptographicMessageSyntax.h File Reference . . . 1397
11.167gdcmOrientation.h File Reference . . . 1399
11.168gdcmOverlay.h File Reference . . . 1399
11.169gdcmParseException.h File Reference . . . 1400
11.170gdcmParser.h File Reference . . . 1402
11.171gdcmPatient.h File Reference . . . 1402
11.172gdcmPDataTFPDU.h File Reference . . . 1403
11.173gdcmPDBElement.h File Reference . . . 1404
11.174gdcmPDBHeader.h File Reference . . . 1405
11.175gdcmPDFCodec.h File Reference . . . 1406
11.176gdcmPDUFactory.h File Reference . . . 1407
11.177gdcmPersonName.h File Reference . . . 1407
11.178gdcmPGXCodec.h File Reference . . . 1408
11.179gdcmPhotometricInterpretation.h File Reference . . . 1409
11.180gdcmPixelFormat.h File Reference . . . 1410
11.181gdcmPixmap.h File Reference . . . 1411
11.182gdcmPixmapReader.h File Reference . . . 1412
11.183gdcmPixmapToPixmapFilter.h File Reference . . . 1413
11.184gdcmPixmapWriter.h File Reference . . . 1413
11.185gdcmPNMCodec.h File Reference . . . 1415

11.186gdcmPreamble.h File Reference 1415
11.187gdcmPresentationContext.h File Reference 1417
11.188gdcmPresentationContextAC.h File Reference 1418
11.189gdcmPresentationContextGenerator.h File Reference 1419
11.190gdcmPresentationContextRQ.h File Reference 1419
11.191gdcmPresentationDataValue.h File Reference 1420
11.192gdcmPrinter.h File Reference 1421
11.193gdcmPrivateTag.h File Reference 1423
11.194gdcmProgressEvent.h File Reference 1424
11.195gdcmPVRGCodec.h File Reference 1425
11.196gdcmPythonFilter.h File Reference 1425
11.197gdcmQueryBase.h File Reference 1426
11.198gdcmQueryFactory.h File Reference 1427
11.199gdcmQueryImage.h File Reference 1428
11.200gdcmQueryPatient.h File Reference 1429
11.201gdcmQuerySeries.h File Reference 1430
11.202gdcmQueryStudy.h File Reference 1431
11.203gdcmRAWCodec.h File Reference 1432
11.204gdcmReader.h File Reference 1432
11.205gdcmRegion.h File Reference 1434
11.206gdcmRescaler.h File Reference 1435
11.207gdcmRLECodec.h File Reference 1436
11.208gdcmRoleSelectionSub.h File Reference 1436
11.209gdcmScanner.h File Reference 1437
11.210gdcmSegment.h File Reference 1438
11.211gdcmSegmentedPaletteColorLookupTable.h File Reference 1439
11.212gdcmSegmentHelper.h File Reference 1440
11.213gdcmSegmentReader.h File Reference 1441

11.214 gdcmSegmentWriter.h File Reference . . . 1442
11.215 gdcmSequenceOfFragments.h File Reference . . . 1443
11.216 gdcmSequenceOfItems.h File Reference . . . 1444
11.217 gdcmSerieHelper.h File Reference . . . 1444
11.218 gdcmSeries.h File Reference . . . 1446
11.219 gdcmServiceClassApplicationInformation.h File Reference . . . 1447
11.220 gdcmServiceClassUser.h File Reference . . . 1448
11.221 gdcmSHA1.h File Reference . . . 1448
11.222 gdcmSimpleSubjectWatcher.h File Reference . . . 1449
11.223 gdcmSmartPointer.h File Reference . . . 1450
11.224 gdcmSOPClassExtendedNegociationSub.h File Reference . . . 1451
11.225 gdcmSOPClassUIDToIOD.h File Reference . . . 1452
11.226 gdcmSorter.h File Reference . . . 1453
11.227 gdcmSpacing.h File Reference . . . 1455
11.228 gdcmSpectroscopy.h File Reference . . . 1455
11.229 gdcmSplitMosaicFilter.h File Reference . . . 1456
11.230 gdcmStaticAssert.h File Reference . . . 1456
11.230.1 Macro Definition Documentation . . . 1457
11.230.1.1 GDCM_DO_JOIN . . . 1457
11.230.1.2 GDCM_DO_JOIN2 . . . 1457
11.230.1.3 GDCM_JOIN . . . 1457
11.230.1.4 GDCM_STATIC_ASSERT . . . 1458
11.231 gdcmStreamImageReader.h File Reference . . . 1458
11.232 gdcmStreamImageWriter.h File Reference . . . 1459
11.233 gdcmStrictScanner.h File Reference . . . 1459
11.234 gdcmString.h File Reference . . . 1460
11.235 gdcmStringFilter.h File Reference . . . 1462
11.236 gdcmStudy.h File Reference . . . 1462

11.237gdcmSubject.h File Reference . . . 1463
11.238gdcmSurface.h File Reference . . . 1464
11.239gdcmSurfaceHelper.h File Reference . . . 1465
11.240gdcmSurfaceReader.h File Reference . . . 1466
11.241gdcmSurfaceWriter.h File Reference . . . 1467
11.242gdcmSwapCode.h File Reference . . . 1468
11.243gdcmSwapper.h File Reference . . . 1469
11.244gdcmSystem.h File Reference . . . 1469
11.245gdcmTable.h File Reference . . . 1470
11.246gdcmTableEntry.h File Reference . . . 1471
11.247gdcmTableReader.h File Reference . . . 1473
11.248gdcmTag.h File Reference . . . 1474
11.249gdcmTagPath.h File Reference . . . 1475
11.250gdcmTagToVR.h File Reference . . . 1475
11.251gdcmTerminal.h File Reference . . . 1476
11.252gdcmTestDriver.h File Reference . . . 1477
11.253gdcmTesting.h File Reference . . . 1478
11.254gdcmTrace.h File Reference . . . 1479
11.254.1Macro Definition Documentation . . . 1480
11.254.1.1GDCM_FUNCTION . . . 1480
11.254.1.2gdcmAssertAlwaysMacro . . . 1480
11.254.1.3gdcmAssertMacro . . . 1480
11.254.1.4gdcmDebugMacro . . . 1481
11.254.1.5gdcmErrorMacro . . . 1481
11.254.1.6gdcmWarningMacro . . . 1483
11.255gdcmTransferSyntax.h File Reference . . . 1483
11.256gdcmTransferSyntaxSub.h File Reference . . . 1485
11.257gdcmType.h File Reference . . . 1486

11.258gdcmTypes.h File Reference 1487
11.259gdcmUIDGenerator.h File Reference 1487
11.260gdcmUIDs.h File Reference 1488
11.261gdcmULAction.h File Reference 1489
11.262gdcmULActionAA.h File Reference 1490
11.263gdcmULActionAE.h File Reference 1491
11.264gdcmULActionAR.h File Reference 1491
11.265gdcmULActionDT.h File Reference 1492
11.266gdcmULBasicCallback.h File Reference 1493
11.267gdcmULConnection.h File Reference 1493
11.268gdcmULConnectionCallback.h File Reference 1494
11.269gdcmULConnectionInfo.h File Reference 1495
11.270gdcmULConnectionManager.h File Reference 1497
11.271gdcmULEvent.h File Reference 1497
11.272gdcmULTransitionTable.h File Reference 1499
11.273gdcmULWritingCallback.h File Reference 1500
11.274gdcmUNExplicitDataElement.h File Reference 1500
11.275gdcmUNExplicitImplicitDataElement.h File Reference 1501
11.276gdcmUnpacker12Bits.h File Reference 1502
11.277gdcmUsage.h File Reference 1502
11.278gdcmUserInformation.h File Reference 1504
11.279gdcmUUIDGenerator.h File Reference 1505
11.280gdcmValidate.h File Reference 1505
11.281gdcmValue.h File Reference 1506
11.282gdcmValueIO.h File Reference 1507
11.283gdcmVersion.h File Reference 1507
11.284gdcmVL.h File Reference 1508
11.285gdcmVM.h File Reference 1509

11.285.1 Macro Definition Documentation . . . 1510
11.285.1.1 TYPETOLENGTH . . . 1510
11.286 gdcmVR.h File Reference . . . 1511
11.286.1 Macro Definition Documentation . . . 1512
11.286.1.1 TYPETOENCODING . . . 1512
11.286.1.2 VRTypeTemplateCase . . . 1513
11.287 gdcmVR16ExplicitDataElement.h File Reference . . . 1513
11.288 gdcmWaveform.h File Reference . . . 1514
11.289 gdcmWin32.h File Reference . . . 1514
11.289.1 Macro Definition Documentation . . . 1515
11.289.1.1 GDCM_EXPORT . . . 1515
11.290 gdcmWLMFindQuery.h File Reference . . . 1515
11.291 gdcmWriter.h File Reference . . . 1516
11.292 gdcmXMLDictReader.h File Reference . . . 1517
11.293 gdcmXMLPrinter.h File Reference . . . 1517
11.294 gdcmXMLPrivateDictReader.h File Reference . . . 1518
11.295 README.txt File Reference . . . 1519
11.296 TestsList.txt File Reference . . . 1519
11.297 vtkGDCMImageReader.h File Reference . . . 1519
11.297.1 Macro Definition Documentation . . . 1520
11.297.1.1 VTK_CMYK . . . 1520
11.297.1.2 VTK_INVERSE_LUMINANCE . . . 1520
11.297.1.3 VTK_LOOKUP_TABLE . . . 1520
11.297.1.4 VTK_YBR . . . 1520
11.298 vtkGDCMImageReader2.h File Reference . . . 1520
11.298.1 Macro Definition Documentation . . . 1521
11.298.1.1 VTK_CMYK . . . 1521
11.298.1.2 VTK_INVERSE_LUMINANCE . . . 1521

11.298.1.3 VTK_LOOKUP_TABLE . . . 1521
11.298.1.4 VTK_YBR . . . 1521
11.299 vtkGDCMImageWriter.h File Reference . . . 1522
11.300 vtkGDCMMedicalImageProperties.h File Reference . . . 1522
11.301 vtkGDCMPolyDataReader.h File Reference . . . 1523
11.302 vtkGDCMPolyDataWriter.h File Reference . . . 1523
11.303 vtkGDCMTesting.h File Reference . . . 1524
11.304 vtkGDCMThreadedImageReader.h File Reference . . . 1524
11.305 vtkGDCMThreadedImageReader2.h File Reference . . . 1525
11.306 vtkImageColorViewer.h File Reference . . . 1526
11.307 vtkImageMapToColors16.h File Reference . . . 1526
11.308 vtkImageMapToWindowLevelColors2.h File Reference . . . 1527
11.309 vtkImagePlanarComponentsToComponents.h File Reference . . . 1527
11.310 vtkImageRGBToYBR.h File Reference . . . 1528
11.311 vtkImageYBRToRGB.h File Reference . . . 1528
11.312 vtkLookupTable16.h File Reference . . . 1529
11.313 vtkRTStructSetProperties.h File Reference . . . 1529

12 Example Documentation **1531**

12.1 AWTMedical3.java . . . 1531
12.2 BasicAnonymizer.cs . . . 1535
12.3 BasicImageAnonymizer.cs . . . 1536
12.4 CastConvertPhilips.py . . . 1538
12.5 ChangePrivateTags.cxx . . . 1540
12.6 ChangeSequenceUltrasound.cxx . . . 1541
12.7 CheckBigEndianBug.cxx . . . 1542
12.8 ClinicalTrialAnnotate.cxx . . . 1544
12.9 ClinicalTrialIdentificationWorkflow.cs . . . 1545

12.10CompressImage.cxx . . . 1548
12.11CompressLossyJPEG.cs . . . 1549
12.12Compute3DSpacing.cxx . . . 1550
12.13Convert16BitsTo8Bits.cxx . . . 1551
12.14ConvertMPL.py . . . 1552
12.15ConvertMultiFrameToSingleFrame.cxx . . . 1553
12.16ConvertNumpy.py . . . 1555
12.17ConvertPIL.py . . . 1555
12.18ConvertRGBToLuminance.cxx . . . 1556
12.19ConvertSingleBitTo8Bits.cxx . . . 1557
12.20ConvertToQImage.cxx . . . 1558
12.21CreateARGBImage.cxx . . . 1560
12.22CreateCMYKImage.cxx . . . 1561
12.23CreateFakePET.cxx . . . 1562
12.24CreateFakeRTDOSE.cxx . . . 1564
12.25CreateJPIPDataSet.cxx . . . 1566
12.26CreateRAWStorage.py . . . 1567
12.27csa2img.cxx . . . 1569
12.28CStoreQtProgress.cxx . . . 1571
12.29DecompressImage.cs . . . 1573
12.30DecompressImage.java . . . 1574
12.31DecompressImage.py . . . 1575
12.32DecompressImageMultiframe.cs . . . 1576
12.33DecompressJPEGFile.cs . . . 1578
12.34DecompressPixmap.java . . . 1579
12.35DiffFile.cxx . . . 1580
12.36DiscriminateVolume.cxx . . . 1581
12.37DumbAnonymizer.py . . . 1585

12.38DumpADAC.cxx . . . 1586
12.39DumpCSA.cs . . . 1591
12.40DumpExamCard.cxx . . . 1592
12.41DumpGEMSMovieGroup.cxx . . . 1599
12.42DumpImageHeaderInfo.cxx . . . 1605
12.43DumpPhilipsECHO.cxx . . . 1607
12.44DumpToshibaDTI.cxx . . . 1613
12.45DumpToSQLITE3.cxx . . . 1614
12.46DuplicatePCDE.cxx . . . 1616
12.47ELSCINT1WaveToText.cxx . . . 1618
12.48EncapsulateFileInRawData.cxx . . . 1620
12.49ExtractEncapsulatedFile.cs . . . 1621
12.50ExtractEncryptedContent.cxx . . . 1622
12.51ExtractIconFromFile.cxx . . . 1623
12.52ExtractImageRegion.cs . . . 1625
12.53ExtractImageRegion.java . . . 1626
12.54ExtractImageRegionWithLUT.cs . . . 1627
12.55Extracting_All_Resolution.cxx . . . 1629
12.56ExtractOneFrame.cs . . . 1634
12.57Fake_Image_Using_Stream_Image_Writer.cxx . . . 1636
12.58FileAnonymize.cs . . . 1639
12.59FileAnonymize.java . . . 1639
12.60FileChangeTS.cs . . . 1640
12.61FileChangeTSLossy.cs . . . 1643
12.62FileStreaming.cs . . . 1645
12.63FindAllPatientName.py . . . 1646
12.64FixBrokenJ2K.cxx . . . 1647
12.65FixCommaBug.py . . . 1649

12.66FixJAIBugJPEGLS.cxx . . . 1649
12.67FixOrientation.cxx . . . 1652
12.68gdcmorthoplanes.cxx . . . 1654
12.69gdcmreslice.cxx . . . 1660
12.70gdcmrtionplan.cxx . . . 1662
12.71gdcmrtplan.cxx . . . 1666
12.72gdcmscene.cxx . . . 1670
12.73gdcmtexture.cxx . . . 1672
12.74gdcmvolume.cxx . . . 1674
12.75GenAllVR.cxx . . . 1675
12.76GenerateDICOMDIR.cs . . . 1677
12.77GenerateRTSTRUCT.cxx . . . 1678
12.78GenerateStandardSOPClasses.cxx . . . 1681
12.79GenFakeIdentifyFile.cxx . . . 1682
12.80GenFakeImage.cxx . . . 1685
12.81GenLongSeqs.cxx . . . 1686
12.82GenSeqs.cxx . . . 1688
12.83GetArray.cs . . . 1689
12.84GetJPEGSamplePrecision.cxx . . . 1690
12.85GetPortionCSAHeader.py . . . 1692
12.86GetSequenceUltrasound.cxx . . . 1693
12.87GetSubSequenceData.cxx . . . 1694
12.88headsq2dcm.py . . . 1697
12.89HelloActiviz.cs . . . 1698
12.90HelloActiviz2.cs . . . 1699
12.91HelloActiviz3.cs . . . 1700
12.92HelloActiviz4.cs . . . 1701
12.93HelloActiviz5.cs . . . 1702

12.94HelloSimple.java . . . 1703
12.95HelloVizWorld.cxx . . . 1704
12.96HelloVTKWorld.cs . . . 1705
12.97HelloVTKWorld.java . . . 1706
12.98HelloVTKWorld2.cs . . . 1707
12.99HelloWorld.cxx . . . 1708
12.100HelloWorld.py . . . 1709
12.101U22tomultisc.cxx . . . 1710
12.102LargeVRDSExplicit.cxx . . . 1711
12.103MagnifyFile.cxx . . . 1713
12.104MakeTemplate.cxx . . . 1714
12.105ManipulateFile.cs . . . 1715
12.106ManipulateFile.py . . . 1716
12.107ManipulateSequence.py . . . 1717
12.108MergeFile.py . . . 1718
12.109MergeTwoFiles.cxx . . . 1719
12.110MetaImageMD5Activiz.cs . . . 1720
12.111MIPViewer.java . . . 1722
12.112MpegVideoInfo.cs . . . 1724
12.113MPRViewer.java . . . 1729
12.114MPRViewer2.java . . . 1731
12.115MrProtocol.cxx . . . 1735
12.116NewSequence.cs . . . 1742
12.117NewSequence.py . . . 1743
12.118offscreenimage.cxx . . . 1744
12.119PatchFile.cxx . . . 1745
12.120PhilipsPrivateRescaleInterceptSlope.py . . . 1747
12.121PlaySound.py . . . 1747

12.128msct_rgb1.cxx . . . 1748
12.129PrivateDict.py . . . 1752
12.130PublicDict.cxx . . . 1752
12.131QIDO-RS.cxx . . . 1753
12.132ReadAndDumpDICOMDIR.cxx . . . 1754
12.133ReadAndDumpDICOMDIR.py . . . 1757
12.134ReadAndPrintAttributes.cxx . . . 1759
12.135ReadExplicitLengthSQIVR.cxx . . . 1761
12.136ReadFiles.java . . . 1761
12.137ReadGEMSSDO.cxx . . . 1763
12.138ReadMultiTimesException.cxx . . . 1765
12.139ReadSeriesIntoVTK.java . . . 1766
12.140ReadUTF8QtDir.cxx . . . 1767
12.141RefCounting.cs . . . 1769
12.142ReformatFile.cs . . . 1769
12.143RemovePrivateTags.py . . . 1771
12.144RescaleImage.cs . . . 1771
12.145reslicesphere.cxx . . . 1772
12.146ReWriteSCAsMR.py . . . 1780
12.147rle2img.cxx . . . 1781
12.148rstructapp.cxx . . . 1784
12.149ScanDirectory.cs . . . 1785
12.150ScanDirectory.java . . . 1787
12.151ScanDirectory.py . . . 1790
12.152SendFileSCU.cs . . . 1791
12.153SimplePrint.cs . . . 1792
12.154SimplePrintPatientName.cs . . . 1793
12.155SimpleScanner.cxx . . . 1794

12.156SortImage.cxx . . . 1795
12.157SortImage.py . . . 1797
12.158SortImage2.cs . . . 1797
12.159StandardizeFiles.cs . . . 1798
12.160StreamImageReaderTest.cxx . . . 1800
12.161TestByteSwap.cxx . . . 1803
12.162TestReader.cxx . . . 1805
12.163TestReader.py . . . 1806
12.164threadgdcm.cxx . . . 1807
12.165TraverseModules.cxx . . . 1810
12.166uid_unique.cxx . . . 1812
12.167VolumeSorter.cxx . . . 1812
12.168WriteBuffer.py . . . 1815

Index 1817

Chapter 1

GDCM Documentation

This is the developpers documentation.

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

`http://gdcm.sourceforge.net/2.6/gdcm-2.6.6.pdf`

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

`http://gdcm.sourceforge.net/2.6/gdcm-2.6.6-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 differenciate 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

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 73
gdcm::SegmentHelper 79
gdcm::terminal
Class for Terminal 79

Chapter 6

Hierarchical Index

6.1 Class Hierarchy

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

gdcm::network::AbstractSyntax . . . 99
gdcm::network::ApplicationContext . . . 112
gdcm::ApplicationEntity . . . 114
gdcm::network::ARTIMTimer . . . 121
gdcm::ASN1 . . . 122
gdcm::network::AsynchronousOperationsWindowSub . . . 124
gdcm::Attribute< Group, Element, TVR, TVM > . . . 125
gdcm::Attribute< Group, Element, TVR, VM::VM1 > . . . 133
gdcm::Attribute< Group, Element, TVR, VM::VM1_n > . . . 141
gdcm::Attribute< Group, Element, TVR, VM::VM1_3 > . . . 139
gdcm::Attribute< Group, Element, TVR, VM::VM1_8 > . . . 140
gdcm::Attribute< Group, Element, TVR, VM::VM2_n > . . . 148
gdcm::Attribute< Group, Element, TVR, VM::VM2_2n > . . . 147
gdcm::Attribute< Group, Element, TVR, VM::VM3_n > . . . 151
gdcm::Attribute< Group, Element, TVR, VM::VM3_3n > . . . 149
gdcm::Base64 . . . 155
gdcm::network::BaseCompositeMessage . . . 157
gdcm::network::CEchoRQ . . . 213
gdcm::network::CEchoRSP . . . 215
gdcm::network::CFindCancelRQ . . . 217
gdcm::network::CFindRQ . . . 218
gdcm::network::CFindRSP . . . 220
gdcm::network::CMoveCancelRq . . . 221
gdcm::network::CMoveRQ . . . 222
gdcm::network::CMoveRSP . . . 224
gdcm::network::CStoreRQ . . . 269
gdcm::network::CStoreRSP . . . 271
gdcm::network::BaseNormalizedMessage . . . 159
gdcm::network::NActionRQ . . . 633
gdcm::network::NActionRSP . . . 635

gdcm::network::NCreateRQ . . . 636
gdcm::network::NCreateRSP . . . 637
gdcm::network::NDeleteRQ . . . 639
gdcm::network::NDeleteRSP . . . 640
gdcm::network::NEventReportRQ . . . 644
gdcm::network::NEventReportRSP . . . 646
gdcm::network::NGetRQ . . . 647
gdcm::network::NGetRSP . . . 648
gdcm::network::NSetRQ . . . 655
gdcm::network::NSetRSP . . . 656
gdcm::network::BasePDU . . . 161
gdcm::network::AAbortPDU . . . 83
gdcm::network::AAssociateACPDU . . . 85
gdcm::network::AAssociateRJPDU . . . 89
gdcm::network::AAssociateRQPDU . . . 91
gdcm::network::AReleaseRPPDU . . . 116
gdcm::network::AReleaseRQPDU . . . 118
gdcm::network::PDataTFPDU . . . 688
std::basic_string< Char >
std::string
gdcm::String< TDelimiter, TMaxLength, TPadChar > . . . 914
gdcm::SegmentHelper::BasicCodedEntry . . . 173
gdcm::BitmapToBitmapFilter . . . 190
gdcm::PixmapToPixmapFilter . . . 728
gdcm::ImageToImageFilter . . . 511
gdcm::ImageApplyLookupTable . . . 465
gdcm::ImageChangePhotometricInterpretation . . . 468
gdcm::ImageChangePlanarConfiguration . . . 472
gdcm::ImageChangeTransferSyntax . . . 477
gdcm::ImageFragmentSplitter . . . 494
gdcm::ByteBuffer . . . 197
gdcm::ByteSwap< T > . . . 199
gdcm::ByteSwapFilter . . . 200
gdcm::network::CFind . . . 216
gdcm::Coder . . . 226
gdcm::Codec . . . 225
gdcm::AudioCodec . . . 152
gdcm::ImageCodec . . . 482
gdcm::DeltaEncodingCodec . . . 314
gdcm::JPEG2000Codec . . . 548
gdcm::JPEGCodec . . . 557
gdcm::JPEG12Codec . . . 542
gdcm::JPEG16Codec . . . 545
gdcm::JPEG8Codec . . . 554
gdcm::JPEGLSCodec . . . 564
gdcm::KAKADUCodec . . . 572
gdcm::PGXCodec . . . 705
gdcm::PNMCodec . . . 734
gdcm::PVRGCodec . . . 770
gdcm::RAWCodec . . . 788
gdcm::RLECodec . . . 806
gdcm::PDFCodec . . . 697
gdcm::CodeString . . . 228

gdcm::network::CompositeMessageFactory . . . 237
gdcm::CompositeNetworkFunctions . . . 238
gdcm::ConstCharWrapper . . . 242
gdcm::CryptoFactory . . . 245
  gdcm::CAPICryptoFactory . . . 209
  gdcm::OpenSSLCryptoFactory . . . 661
  gdcm::OpenSSLP7CryptoFactory . . . 665
gdcm::CryptographicMessageSyntax . . . 248
  gdcm::CAPICryptographicMessageSyntax . . . 210
  gdcm::OpenSSLCryptographicMessageSyntax . . . 662
  gdcm::OpenSSLP7CryptographicMessageSyntax . . . 667
gdcm::CSAElement . . . 251
gdcm::CSAHeader . . . 258
gdcm::CSAHeaderDict . . . 263
gdcm::CSAHeaderDictEntry . . . 266
gdcm::DataElement . . . 277
  gdcm::CP246ExplicitDataElement . . . 243
  gdcm::ExplicitDataElement . . . 385
  gdcm::ExplicitImplicitDataElement . . . 388
  gdcm::Fragment . . . 444
    gdcm::BasicOffsetTable . . . 176
  gdcm::ImplicitDataElement . . . 520
  gdcm::Item . . . 536
  gdcm::UNExplicitDataElement . . . 1090
  gdcm::UNExplicitImplicitDataElement . . . 1092
  gdcm::VR16ExplicitDataElement . . . 1124
gdcm::DataSet . . . 294
  gdcm::CommandDataSet . . . 234
  gdcm::FileMetaInformation . . . 411
gdcm::DataSetHelper . . . 307
gdcm::Decoder . . . 308
  gdcm::Codec . . . 225
gdcm::DefinedTerms . . . 310
gdcm::Defs . . . 311
gdcm::DICOMDIR . . . 317
gdcm::DICOMDIRGenerator . . . 317
gdcm::Dict . . . 321
gdcm::DictConverter . . . 324
gdcm::DictEntry . . . 328
gdcm::Dicts . . . 335
gdcm::network::DIMSE . . . 338
gdcm::DirectionCosines . . . 340
gdcm::Directory . . . 343
gdcm::DirectoryHelper . . . 347
gdcm::DummyValueGenerator . . . 349
gdcm::Element< TVR, TVM > . . . 351
gdcm::Element< TVR, VM::VM1_n > . . . 357
  gdcm::Element< TVR, VM::VM1_2 > . . . 356
  gdcm::Element< TVR, VM::VM2_n > . . . 363
    gdcm::Element< TVR, VM::VM2_2n > . . . 362
  gdcm::Element< TVR, VM::VM3_n > . . . 367
    gdcm::Element< TVR, VM::VM3_3n > . . . 365

gdcm::Element< VR::AS, VM::VM5 > . . . 368
gdcm::Element< VR::OB, VM::VM1_n > . . . 351
  gdcm::Element< VR::OB, VM::VM1 > . . . 369
gdcm::Element< VR::OW, VM::VM1_n > . . . 351
  gdcm::Element< VR::OW, VM::VM1 > . . . 370
gdcm::ElementDisableCombinations< TVR, TVM > . . . 372
gdcm::ElementDisableCombinations< VR::OB, VM::VM1_n > . . . 373
gdcm::ElementDisableCombinations< VR::OW, VM::VM1_n > . . . 373
gdcm::EncapsulatedDocument . . . 373
gdcm::EncodingImplementation< T > . . . 374
gdcm::EncodingImplementation< VR::VRASCII > . . . 374
gdcm::EncodingImplementation< VR::VRBINARY > . . . 376
gdcm::EnumeratedValues . . . 379
gdcm::Event . . . 380
  gdcm::AnyEvent . . . 110
    gdcm::AbortEvent . . . 97
    gdcm::AnonymizeEvent . . . 101
    gdcm::DataEvent . . . 290
    gdcm::DataSetEvent . . . 305
    gdcm::EndEvent . . . 378
    gdcm::ExitEvent . . . 384
    gdcm::FileNameEvent . . . 422
    gdcm::InitializeEvent . . . 522
    gdcm::IterationEvent . . . 541
    gdcm::ModifiedEvent . . . 617
    gdcm::ProgressEvent . . . 766
    gdcm::StartEvent . . . 893
    gdcm::UserEvent . . . 1098
  gdcm::NoEvent . . . 650
std::exception
  gdcm::CSAHeaderDictException . . . 269
  gdcm::DataElementException . . . 290
  gdcm::Exception . . . 382
    gdcm::ParseException . . . 682
gdcm::Fiducials . . . 390
gdcm::FileDerivation . . . 404
gdcm::FileExplicitFilter . . . 408
gdcm::Filename . . . 419
gdcm::FilenameGenerator . . . 425
gdcm::FileSet . . . 429
gdcm::Global . . . 447
gdcm::GroupDict . . . 450
gdcm::IconImageFilter . . . 452
gdcm::IconImageGenerator . . . 455
gdcm::ignore_char . . . 459
gdcm::ImageConverter . . . 493
gdcm::ImageHelper . . . 497
gdcm::network::ImplementationClassUIDSub . . . 516
gdcm::network::ImplementationUIDSub . . . 517
gdcm::network::ImplementationVersionNameSub . . . 518
gdcm::IOD . . . 524
gdcm::IODEntry . . . 526
gdcm::IODs . . . 529

gdcm::JSON . . . 570
gdcm::Scanner::ltstr . . . 585
gdcm::StrictScanner::ltstr . . . 585
gdcm::Macro . . . 586
gdcm::Macros . . . 589
gdcm::network::MaximumLengthSub . . . 591
gdcm::MD5 . . . 592
gdcm::MediaStorage . . . 594
gdcm::Module . . . 618
gdcm::ModuleEntry . . . 621
  gdcm::NestedModuleEntries . . . 641
gdcm::Modules . . . 625
gdcm::network::NormalizedMessageFactory . . . 651
gdcm::NormalizedNetworkFunctions . . . 652
gdcm::Object . . . 657
  gdcm::BaseQuery . . . 164
    gdcm::BaseRootQuery . . . 168
      gdcm::FindPatientRootQuery . . . 438
      gdcm::FindStudyRootQuery . . . 441
      gdcm::MovePatientRootQuery . . . 627
      gdcm::MoveStudyRootQuery . . . 630
      gdcm::WLMFindQuery . . . 1228
    gdcm::ModalityPerformedProcedureStepCreateQuery . . . 612
    gdcm::ModalityPerformedProcedureStepSetQuery . . . 615
  gdcm::Bitmap . . . 178
    gdcm::Pixmap . . . 720
      gdcm::Image . . . 460
  gdcm::Curve . . . 272
  gdcm::File . . . 391
    gdcm::FileWithName . . . 436
  gdcm::LookupTable . . . 578
    gdcm::SegmentedPaletteColorLookupTable . . . 830
  gdcm::MeshPrimitive . . . 607
  gdcm::Overlay . . . 673
  gdcm::Segment . . . 822
  gdcm::Subject . . . 923
    gdcm::Anonymizer . . . 104
    gdcm::Command . . . 232
      gdcm::MemberCommand< T > . . . 603
      gdcm::SimpleMemberCommand< T > . . . 869
    gdcm::FileAnonymizer . . . 395
    gdcm::FileChangeTransferSyntax . . . 399
    gdcm::FileDecompressLookupTable . . . 402
    gdcm::FileStreamer . . . 431
    gdcm::network::ULConnectionManager . . . 1078
    gdcm::Scanner . . . 814
    gdcm::ServiceClassUser . . . 860
    gdcm::StrictScanner . . . 906
  gdcm::Surface . . . 926
  gdcm::Value . . . 1104
    gdcm::ByteValue . . . 202
    gdcm::SequenceOfFragments . . . 838
    gdcm::SequenceOfItems . . . 845

gdcm::Orientation . . . 670
gdcm::Parser . . . 684
gdcm::Patient . . . 687
gdcm::PDBElement . . . 691
gdcm::PDBHeader . . . 694
gdcm::network::PDUFactory . . . 699
gdcm::PersonName . . . 702
gdcm::PhotometricInterpretation . . . 707
gdcm::PixelFormat . . . 712
gdcm::Preamble . . . 737
gdcm::PresentationContext . . . 741
gdcm::network::PresentationContextAC . . . 745
gdcm::PresentationContextGenerator . . . 747
gdcm::network::PresentationContextRQ . . . 750
gdcm::network::PresentationDataValue . . . 753
gdcm::Printer . . . 757
gdcm::DictPrinter . . . 333
gdcm::Dumper . . . 350
gdcm::PrivateDict . . . 761
gdcm::PythonFilter . . . 773
gdcm::QueryBase . . . 774
gdcm::QueryImage . . . 779
gdcm::QueryPatient . . . 781
gdcm::QuerySeries . . . 783
gdcm::QueryStudy . . . 785
gdcm::QueryFactory . . . 777
gdcm::Reader . . . 791
gdcm::PixmapReader . . . 724
gdcm::ImageReader . . . 503
gdcm::ImageRegionReader . . . 507
gdcm::SegmentReader . . . 832
gdcm::SurfaceReader . . . 941
gdcm::RealWorldValueMappingContent . . . 798
gdcm::Region . . . 799
gdcm::BoxRegion . . . 193
gdcm::Rescaler . . . 801
gdcm::network::RoleSelectionSub . . . 811
gdcm::SerieHelper::Rule . . . 813
gdcm::SerieHelper . . . 854
gdcm::Series . . . 858
gdcm::network::ServiceClassApplicationInformation . . . 859
gdcm::SHA1 . . . 867
gdcm::SimpleSubjectWatcher . . . 873
gdcm::SmartPointer< ObjectType > . . . 876
gdcm::SmartPointer< gdcm::Bitmap > . . . 876
gdcm::SmartPointer< gdcm::File > . . . 876
gdcm::SmartPointer< gdcm::Image > . . . 876
gdcm::SmartPointer< gdcm::MemberCommand > . . . 876
gdcm::SmartPointer< gdcm::MeshPrimitive > . . . 876
gdcm::SmartPointer< gdcm::Pixmap > . . . 876
gdcm::SmartPointer< gdcm::SimpleMemberCommand > . . . 876
gdcm::SmartPointer< gdcm::Subject > . . . 876
gdcm::SmartPointer< LookupTable > . . . 876

gdcm::SmartPointer< Segment > . . . 876
gdcm::SmartPointer< Surface > . . . 876
gdcm::SmartPointer< Value > . . . 876
gdcm::network::SOPClassExtendedNegociationSub . . . 879
gdcm::SOPClassUIDToIOD . . . 881
gdcm::Sorter . . . 883
gdcm::IPPSorter . . . 532
gdcm::Spacing . . . 887
gdcm::Spectroscopy . . . 890
gdcm::SplitMosaicFilter . . . 891
gdcm::static_assert_test< x > . . . 894
gdcm::STATIC_ASSERTION_FAILURE< x > . . . 894
gdcm::STATIC_ASSERTION_FAILURE< true > . . . 895
gdcm::StreamImageReader . . . 895
gdcm::StreamImageWriter . . . 899
String<'\\', 64 >
gdcm::LO . . . 575
gdcm::StringFilter . . . 919
gdcm::Study . . . 922
gdcm::SurfaceHelper . . . 938
gdcm::SwapCode . . . 947
gdcm::SwapperDoOp . . . 949
gdcm::SwapperNoOp . . . 950
gdcm::System . . . 951
gdcm::Table . . . 957
gdcm::TableEntry . . . 959
gdcm::TableReader . . . 960
gdcm::XMLDictReader . . . 1237
gdcm::XMLPrivateDictReader . . . 1243
gdcm::network::TableRow . . . 964
gdcm::Tag . . . 965
gdcm::PrivateTag . . . 763
gdcm::TagPath . . . 974
gdcm::Testing . . . 976
gdcm::Trace . . . 983
gdcm::TransferSyntax . . . 988
gdcm::network::TransferSyntaxSub . . . 994
gdcm::network::Transition . . . 996
gdcm::Type . . . 998
gdcm::UI . . . 1000
gdcm::UIDGenerator . . . 1001
gdcm::UIDs . . . 1004
gdcm::network::ULAction . . . 1029
gdcm::network::ULActionAA1 . . . 1032
gdcm::network::ULActionAA2 . . . 1033
gdcm::network::ULActionAA3 . . . 1035
gdcm::network::ULActionAA4 . . . 1036
gdcm::network::ULActionAA5 . . . 1037
gdcm::network::ULActionAA6 . . . 1038
gdcm::network::ULActionAA7 . . . 1040
gdcm::network::ULActionAA8 . . . 1041
gdcm::network::ULActionAE1 . . . 1042
gdcm::network::ULActionAE2 . . . 1043

gdcm::network::ULActionAE3 . . . 1045
gdcm::network::ULActionAE4 . . . 1046
gdcm::network::ULActionAE5 . . . 1047
gdcm::network::ULActionAE6 . . . 1048
gdcm::network::ULActionAE7 . . . 1050
gdcm::network::ULActionAE8 . . . 1051
gdcm::network::ULActionAR1 . . . 1052
gdcm::network::ULActionAR10 . . . 1053
gdcm::network::ULActionAR2 . . . 1055
gdcm::network::ULActionAR3 . . . 1056
gdcm::network::ULActionAR4 . . . 1057
gdcm::network::ULActionAR5 . . . 1058
gdcm::network::ULActionAR6 . . . 1060
gdcm::network::ULActionAR7 . . . 1061
gdcm::network::ULActionAR8 . . . 1062
gdcm::network::ULActionAR9 . . . 1063
gdcm::network::ULActionDT1 . . . 1065
gdcm::network::ULActionDT2 . . . 1066
gdcm::network::ULConnection . . . 1069
gdcm::network::ULConnectionCallback . . . 1074
gdcm::network::ULBasicCallback . . . 1067
gdcm::network::ULWritingCallback . . . 1088
gdcm::network::ULConnectionInfo . . . 1076
gdcm::network::ULEvent . . . 1085
gdcm::network::ULTransitionTable . . . 1086
gdcm::Unpacker12Bits . . . 1094
gdcm::Usage . . . 1095
gdcm::network::UserInformation . . . 1099
gdcm::UUIDGenerator . . . 1101
gdcm::Validate . . . 1102
gdcm::ValueIO< TDE, TSwap, TType > . . . 1106
gdcm::Version . . . 1107
gdcm::VL . . . 1109
gdcm::VM . . . 1113
gdcm::VMToLength< T > . . . 1118
gdcm::VR . . . 1118
gdcm::VRToEncoding< T > . . . 1127
gdcm::VRToType< T > . . . 1127
gdcm::VRToType< TVR > . . . 1127
gdcm::VRVLSize< T > . . . 1128
gdcm::VRVLSize< 0 > . . . 1128
gdcm::VRVLSize< 1 > . . . 1128
vtkImageAlgorithm
vtkImagePlanarComponentsToComponents . . . 1209
vtkImageMapToColors
vtkImageMapToWindowLevelColors2 . . . 1205
vtkImageWriter
vtkGDCMImageWriter . . . 1154
vtkLookupTable
vtkLookupTable16 . . . 1215
vtkMedicalImageProperties
vtkGDCMMedicalImageProperties . . . 1162
vtkMedicalImageReader2
vtkGDCMImageReader . . . 1129

vtkGDCMThreadedImageReader 1177
vtkGDCMImageReader2 1142
vtkObject
vtkGDCMTesting 1173
vtkImageColorViewer 1187
vtkRTStructSetProperties 1218
vtkPolyDataAlgorithm
vtkGDCMPolyDataReader 1165
vtkPolyDataWriter
vtkGDCMPolyDataWriter 1169
vtkThreadedImageAlgorithm
vtkGDCMThreadedImageReader2 1180
vtkImageMapToColors16 1200
vtkImageRGBToYBR 1211
vtkImageYBRToRGB 1213
gdcm::Waveform 1227
gdcm::Writer 1231
gdcm::PixmapWriter 730
gdcm::ImageWriter 513
gdcm::SegmentWriter 835
gdcm::SurfaceWriter 944
gdcm::XMLPrinter 1240

Chapter 7

Class Index

7.1 Class List

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

gdcm::network::AAbortPDU
AAbortPDU . . . 83
gdcm::network::AAssociateACPDU
AAssociateACPDU . . . 85
gdcm::network::AAssociateRJPDU
AAssociateRJPDU . . . 89
gdcm::network::AAssociateRQPDU
AAssociateRQPDU . . . 91
gdcm::AbortEvent . . . 97
gdcm::network::AbstractSyntax
AbstractSyntax . . . 99
gdcm::AnonymizeEvent
AnonymizeEvent . . . 101
gdcm::Anonymizer
Anonymizer . . . 104
gdcm::AnyEvent . . . 110
gdcm::network::ApplicationContext
ApplicationContext . . . 112
gdcm::ApplicationEntity
ApplicationEntity . . . 114
gdcm::network::AReleaseRPPDU
AReleaseRPPDU . . . 116
gdcm::network::AReleaseRQPDU
AReleaseRQPDU . . . 118
gdcm::network::ARTIMTimer
ARTIMTimer . . . 121
gdcm::ASN1
Class for ASN1 . . . 122
gdcm::network::AsynchronousOperationsWindowSub
AsynchronousOperationsWindowSub . . . 124

gdcm::Attribute< Group, Element, TVR, TVM >
Attribute class This class use template metaprograming tricks to let the user know when the template instanciation does not match the public dictionary . . . 125
gdcm::Attribute< Group, Element, TVR, VM::VM1 > . . . 133
gdcm::Attribute< Group, Element, TVR, VM::VM1_3 > . . . 139
gdcm::Attribute< Group, Element, TVR, VM::VM1_8 > . . . 140
gdcm::Attribute< Group, Element, TVR, VM::VM1_n > . . . 141
gdcm::Attribute< Group, Element, TVR, VM::VM2_2n > . . . 147
gdcm::Attribute< Group, Element, TVR, VM::VM2_n > . . . 148
gdcm::Attribute< Group, Element, TVR, VM::VM3_3n > . . . 149
gdcm::Attribute< Group, Element, TVR, VM::VM3_n > . . . 151
gdcm::AudioCodec
AudioCodec . . . 152
gdcm::Base64
Class for Base64 . . . 155
gdcm::network::BaseCompositeMessage
BaseCompositeMessage . . . 157
gdcm::network::BaseNormalizedMessage
BaseNormalizedMessage . . . 159
gdcm::network::BasePDU
BasePDU . . . 161
gdcm::BaseQuery
BaseQuery . . . 164
gdcm::BaseRootQuery
BaseRootQuery . . . 168
gdcm::SegmentHelper::BasicCodedEntry
This structure defines a basic coded entry with all of its attributes . . . 173
gdcm::BasicOffsetTable
Class to represent a BasicOffsetTable . . . 176
gdcm::Bitmap
Bitmap class . . . 178
gdcm::BitmapToBitmapFilter
BitmapToBitmapFilter class . . . 190
gdcm::BoxRegion
Class for manipulation box region . . . 193
gdcm::ByteBuffer
ByteBuffer . . . 197
gdcm::ByteSwap< T >
ByteSwap . . . 199
gdcm::ByteSwapFilter
ByteSwapFilter . . . 200
gdcm::ByteValue
Class to represent binary value (array of bytes) . . . 202
gdcm::CAPICryptoFactory . . . 209
gdcm::CAPICryptographicMessageSyntax . . . 210
gdcm::network::CEchoRQ
CEchoRQ . . . 213
gdcm::network::CEchoRSP
CEchoRSP this file defines the messages for the cecho action . . . 215
gdcm::network::CFind . . . 216
gdcm::network::CFindCancelRQ
CFindCancelRQ this file defines the messages for the cfind action . . . 217
gdcm::network::CFindRQ
CFindRQ . . . 218

gdcm::network::CFindRSP
CFindRSP this file defines the messages for the cfind action . . . 220
gdcm::network::CMoveCancelRq . . . 221
gdcm::network::CMoveRQ
CMoveRQ . . . 222
gdcm::network::CMoveRSP
CMoveRSP this file defines the messages for the cmove action . . . 224
gdcm::Codec
Codec class . . . 225
gdcm::Coder
Coder . . . 226
gdcm::CodeString
CodeString . . . 228
gdcm::Command
Command superclass for callback/observer methods . . . 232
gdcm::CommandDataSet
Class to represent a Command DataSet . . . 234
gdcm::network::CompositeMessageFactory
CompositeMessageFactory . . . 237
gdcm::CompositeNetworkFunctions
Composite Network Functions . . . 238
gdcm::ConstCharWrapper
Do not use me . . . 242
gdcm::CP246ExplicitDataElement
Class to read/write a DataElement as CP246Explicit Data Element . . . 243
gdcm::CryptoFactory
Class to do handle the crypto factory . . . 245
gdcm::CryptographicMessageSyntax . . . 248
gdcm::CSAElement
Class to represent a CSA Element . . . 251
gdcm::CSAHeader
Class for CSAHeader . . . 258
gdcm::CSAHeaderDict
Class to represent a map of CSAHeaderDictEntry . . . 263
gdcm::CSAHeaderDictEntry
Class to represent an Entry in the Dict . . . 266
gdcm::CSAHeaderDictException . . . 269
gdcm::network::CStoreRQ
CStoreRQ . . . 269
gdcm::network::CStoreRSP
CStoreRSP this file defines the messages for the cecho action . . . 271
gdcm::Curve
Curve class to handle element 50xx,3000 Curve Data . . . 272
gdcm::DataElement
Class to represent a Data Element either Implicit or Explicit . . . 277
gdcm::DataElementException . . . 290
gdcm::DataEvent
DataEvent . . . 290
gdcm::DataSet
Class to represent a Data Set (which contains Data Elements) . . . 294
gdcm::DataSetEvent
DataSetEvent . . . 305
gdcm::DataSetHelper
DataSetHelper (internal class, not intended for user level) . . . 307

gdcm::Decoder
Decoder . . . 308
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 . . . 310
gdcm::Defs
FIXME I do not like the name 'Defs' . . . 311
gdcm::DeltaEncodingCodec
DeltaEncodingCodec compression used by some private vendor . . . 314
gdcm::DICOMDIR
DICOMDIR class . . . 317
gdcm::DICOMDIRGenerator
DICOMDIRGenerator class . . . 317
gdcm::Dict
Class to represent a map of DictEntry . . . 321
gdcm::DictConverter
Class to convert a .dic file into something else: . . . 324
gdcm::DictEntry
Class to represent an Entry in the Dict . . . 328
gdcm::DictPrinter
DictPrinter class . . . 333
gdcm::Dicts
Class to manipulate the sum of knowledge (all the dict user load) . . . 335
gdcm::network::DIMSE
DIMSE . . . 338
gdcm::DirectionCosines
Class to handle DirectionCosines . . . 340
gdcm::Directory
Class for manipulation directories . . . 343
gdcm::DirectoryHelper
DirectoryHelper . . . 347
gdcm::DummyValueGenerator
Class for generating dummy value . . . 349
gdcm::Dumper
Codec class . . . 350
gdcm::Element< TVR, TVM >
Element class . . . 351
gdcm::Element< TVR, VM::VM1_2 > . . . 356
gdcm::Element< TVR, VM::VM1_n > . . . 357
gdcm::Element< TVR, VM::VM2_2n > . . . 362
gdcm::Element< TVR, VM::VM2_n > . . . 363
gdcm::Element< TVR, VM::VM3_3n > . . . 365
gdcm::Element< TVR, VM::VM3_n > . . . 367
gdcm::Element< VR::AS, VM::VM5 > . . . 368
gdcm::Element< VR::OB, VM::VM1 > . . . 369
gdcm::Element< VR::OW, VM::VM1 > . . . 370

gdcm::ElementDisableCombinations< TVR, TVM >
A class which is used to produce compile errors for an invalid combination of template parameters . 372
gdcm::ElementDisableCombinations< VR::OB, VM::VM1_n > . . . 373
gdcm::ElementDisableCombinations< VR::OW, VM::VM1_n > . . . 373
gdcm::EncapsulatedDocument
EncapsulatedDocument . . . 373
gdcm::EncodingImplementation< T >
EncodingImplementation . . . 374
gdcm::EncodingImplementation< VR::VRASCII > . . . 374
gdcm::EncodingImplementation< VR::VRBINARY > . . . 376
gdcm::EndEvent . . . 378
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: . . . 379
gdcm::Event
Superclass for callback/observer methods . . . 380
gdcm::Exception
Exception . . . 382
gdcm::ExitEvent . . . 384
gdcm::ExplicitDataElement
Class to read/write a DataElement as Explicit Data Element . . . 385
gdcm::ExplicitImplicitDataElement
Class to read/write a DataElement as ExplicitImplicit Data Element . . . 388
gdcm::Fiducials
Fiducials . . . 390
gdcm::File
DICOM File . . . 391
gdcm::FileAnonymizer
FileAnonymizer . . . 395
gdcm::FileChangeTransferSyntax
FileChangeTransferSyntax . . . 399
gdcm::FileDecompressLookupTable
FileDecompressLookupTable class . . . 402
gdcm::FileDerivation
FileDerivation class . . . 404
gdcm::FileExplicitFilter
FileExplicitFilter class . . . 408
gdcm::FileMetaInformation
Class to represent a File Meta Information . . . 411
gdcm::Filename
Class to manipulate file name's . . . 419
gdcm::FileNameEvent
FileNameEvent . . . 422
gdcm::FilenameGenerator
FilenameGenerator . . . 425
gdcm::FileSet . . . 429
gdcm::FileStreamer
FileStreamer . . . 431
gdcm::FileWithName
FileWithName . . . 436
gdcm::FindPatientRootQuery
PatientRootQuery . . . 438

gdcm::FindStudyRootQuery
FindStudyRootQuery . . . 441
gdcm::Fragment
Class to represent a Fragment . . . 444
gdcm::Global
Global . . . 447
gdcm::GroupDict
Class to represent the mapping from group number to its abbreviation and name . . . 450
gdcm::IconImageFilter
IconImageFilter . . . 452
gdcm::IconImageGenerator
IconImageGenerator . . . 455
gdcm::ignore_char . . . 459
gdcm::Image
Image . . . 460
gdcm::ImageApplyLookupTable
ImageApplyLookupTable class . . . 465
gdcm::ImageChangePhotometricInterpretation
ImageChangePhotometricInterpretation class . . . 468
gdcm::ImageChangePlanarConfiguration
ImageChangePlanarConfiguration class . . . 472
gdcm::ImageChangeTransferSyntax
ImageChangeTransferSyntax class . . . 477
gdcm::ImageCodec
ImageCodec . . . 482
gdcm::ImageConverter
Image Converter . . . 493
gdcm::ImageFragmentSplitter
ImageFragmentSplitter class . . . 494
gdcm::ImageHelper
ImageHelper (internal class, not intended for user level) . . . 497
gdcm::ImageReader
ImageReader . . . 503
gdcm::ImageRegionReader
ImageRegionReader . . . 507
gdcm::ImageToImageFilter
ImageToImageFilter class . . . 511
gdcm::ImageWriter
ImageWriter . . . 513
gdcm::network::ImplementationClassUIDSub
ImplementationClassUIDSub . . . 516
gdcm::network::ImplementationUIDSub
ImplementationUIDSub . . . 517
gdcm::network::ImplementationVersionNameSub
ImplementationVersionNameSub . . . 518
gdcm::ImplicitDataElement
Class to represent an *Implicit VR* Data Element . . . 520
gdcm::InitializeEvent . . . 522
gdcm::IOD
Class for representing a IOD . . . 524
gdcm::IODEntry
Class for representing a IODEntry . . . 526
gdcm::IODs
Class for representing a IODs . . . 529

gdcm::IPPSorter
IPPSorter . . . 532
gdcm::Item
Class to represent an Item . . . 536
gdcm::IterationEvent . . . 541
gdcm::JPEG12Codec
Class to do JPEG 12bits (lossy & lossless) . . . 542
gdcm::JPEG16Codec
Class to do JPEG 16bits (lossless) . . . 545
gdcm::JPEG2000Codec
Class to do JPEG 2000 . . . 548
gdcm::JPEG8Codec
Class to do JPEG 8bits (lossy & lossless) . . . 554
gdcm::JPEGCodec
JPEG codec . . . 557
gdcm::JPEGLSCodec
JPEG-LS . . . 564
gdcm::JSON . . . 570
gdcm::KAKADUCodec
KAKADUCodec . . . 572
gdcm::LO
LO . . . 575
gdcm::LookupTable
LookupTable class . . . 578
gdcm::Scanner::ltstr . . . 585
gdcm::StrictScanner::ltstr . . . 585
gdcm::Macro
Class for representing a Macro . . . 586
gdcm::Macros
Class for representing a Modules . . . 589
gdcm::network::MaximumLengthSub
MaximumLengthSub . . . 591
gdcm::MD5
Class for MD5 . . . 592
gdcm::MediaStorage
MediaStorage . . . 594
gdcm::MemberCommand< T >
Command subclass that calls a pointer to a member function . . . 603
gdcm::MeshPrimitive
This class defines surface mesh primitives . . . 607
gdcm::ModalityPerformedProcedureStepCreateQuery
ModalityPerformedProcedureStepCreateQuery . . . 612
gdcm::ModalityPerformedProcedureStepSetQuery
ModalityPerformedProcedureStepSetQuery . . . 615
gdcm::ModifiedEvent . . . 617
gdcm::Module
Class for representing a Module . . . 618
gdcm::ModuleEntry
Class for representing a ModuleEntry . . . 621
gdcm::Modules
Class for representing a Modules . . . 625
gdcm::MovePatientRootQuery
MovePatientRootQuery . . . 627

gdcm::MoveStudyRootQuery
MoveStudyRootQuery . . . 630
gdcm::network::NActionRQ
NActionRQ . . . 633
gdcm::network::NActionRSP
NActionRSP this file defines the messages for the NAction action . . . 635
gdcm::network::NCreateRQ
NCreateRQ . . . 636
gdcm::network::NCreateRSP
NCreateRSP this file defines the messages for the ncreate action . . . 637
gdcm::network::NDeleteRQ
NDeleteRQ . . . 639
gdcm::network::NDeleteRSP
NDeleteRSP this file defines the messages for the ndelete action . . . 640
gdcm::NestedModuleEntries
Class for representing a NestedModuleEntries . . . 641
gdcm::network::NEventReportRQ
NEventReportRQ . . . 644
gdcm::network::NEventReportRSP
NEventReportRSP this file defines the messages for the neventreport action . . . 646
gdcm::network::NGetRQ
NGetRQ . . . 647
gdcm::network::NGetRSP
NGetRSP this file defines the messages for the nget action . . . 648
gdcm::NoEvent . . . 650
gdcm::network::NormalizedMessageFactory . . . 651
gdcm::NormalizedNetworkFunctions
Normalized Network Functions . . . 652
gdcm::network::NSetRQ
NSetRQ . . . 655
gdcm::network::NSetRSP
NSetRSP this file defines the messages for the nset action . . . 656
gdcm::Object
Object . . . 657
gdcm::OpenSSLCryptoFactory . . . 661
gdcm::OpenSSLCryptographicMessageSyntax . . . 662
gdcm::OpenSSLP7CryptoFactory . . . 665
gdcm::OpenSSLP7CryptographicMessageSyntax . . . 667
gdcm::Orientation
Class to handle Orientation . . . 670
gdcm::Overlay
Overlay class . . . 673
gdcm::ParseException
ParseException Standard exception handling object . . . 682
gdcm::Parser
Parser ala XML_Parser from expat (SAX) . . . 684
gdcm::Patient
See PS 3.3 - 2007 DICOM MODEL OF THE REAL-WORLD, p 54 . . . 687
gdcm::network::PDataTFPDU
PDataTFPDU . . . 688
gdcm::PDBElement
Class to represent a PDB Element . . . 691
gdcm::PDBHeader
Class for PDBHeader . . . 694

gdcm::PDFCodec
PDFCodec class . . . 697
gdcm::network::PDUFactory
PDUFactory basically, given an initial byte, construct the . . . 699
gdcm::PersonName
PersonName class . . . 702
gdcm::PGXCodec
Class to do PGX . . . 705
gdcm::PhotometricInterpretation
Class to represent an PhotometricInterpretation . . . 707
gdcm::PixelFormat
PixelFormat . . . 712
gdcm::Pixmap
Pixmap class . . . 720
gdcm::PixmapReader
PixmapReader . . . 724
gdcm::PixmapToPixmapFilter
PixmapToPixmapFilter class . . . 728
gdcm::PixmapWriter
PixmapWriter . . . 730
gdcm::PNMCodec
Class to do PNM . . . 734
gdcm::Preamble
DICOM Preamble (Part 10) . . . 737
gdcm::PresentationContext
PresentationContext . . . 741
gdcm::network::PresentationContextAC
PresentationContextAC . . . 745
gdcm::PresentationContextGenerator
PresentationContextGenerator . . . 747
gdcm::network::PresentationContextRQ
PresentationContextRQ . . . 750
gdcm::network::PresentationDataValue
PresentationDataValue . . . 753
gdcm::Printer
Printer class . . . 757
gdcm::PrivateDict
Private Dict . . . 761
gdcm::PrivateTag
Class to represent a Private DICOM Data Element (Attribute) Tag (Group, Element, Owner) . . . 763
gdcm::ProgressEvent
ProgressEvent . . . 766
gdcm::PVRGCodec
PVRGCodec . . . 770
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 . . . 773
gdcm::QueryBase
QueryBase . . . 774
gdcm::QueryFactory
QueryFactory.h . . . 777
gdcm::QueryImage
QueryImage . . . 779

gdcm::QueryPatient
QueryPatient . . . 781
gdcm::QuerySeries
QuerySeries . . . 783
gdcm::QueryStudy
QueryStudy.h . . . 785
gdcm::RAWCodec
RAWCodec class . . . 788
gdcm::Reader
Reader ala DOM (Document Object Model) . . . 791
gdcm::RealWorldValueMappingContent . . . 798
gdcm::Region
Class for manipulation region . . . 799
gdcm::Rescaler
Rescale class . . . 801
gdcm::RLECodec
Class to do RLE . . . 806
gdcm::network::RoleSelectionSub
RoleSelectionSub . . . 811
gdcm::SerieHelper::Rule . . . 813
gdcm::Scanner
Scanner . . . 814
gdcm::Segment
This class defines a segment . . . 822
gdcm::SegmentedPaletteColorLookupTable
SegmentedPaletteColorLookupTable class . . . 830
gdcm::SegmentReader
This class defines a segment reader . . . 832
gdcm::SegmentWriter
This class defines a segment writer . . . 835
gdcm::SequenceOfFragments
Class to represent a Sequence Of Fragments . . . 838
gdcm::SequenceOfItems
Class to represent a Sequence Of Items . . . 845
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 disapear soon, you've been warned . . . 854
gdcm::Series
Series . . . 858
gdcm::network::ServiceClassApplicationInformation . . . 859
gdcm::ServiceClassUser
ServiceClassUser . . . 860
gdcm::SHA1
Class for SHA1 . . . 867
gdcm::SimpleMemberCommand< T >
Command subclass that calls a pointer to a member function . . . 869
gdcm::SimpleSubjectWatcher
SimpleSubjectWatcher . . . 873
gdcm::SmartPointer< ObjectType >
Class for Smart Pointer . . . 876
gdcm::network::SOPClassExtendedNegociationSub
SOPClassExtendedNegociationSub . . . 879
gdcm::SOPClassUIDToIOD
Class convert a class SOP Class UID into IOD . . . 881

gdcm::Sorter
Sorter . . . 883
gdcm::Spacing
Class for Spacing . . . 887
gdcm::Spectroscopy
Spectroscopy class . . . 890
gdcm::SplitMosaicFilter
SplitMosaicFilter class . . . 891
gdcm::StartEvent . . . 893
gdcm::static_assert_test< x > . . . 894
gdcm::STATIC_ASSERTION_FAILURE< x > . . . 894
gdcm::STATIC_ASSERTION_FAILURE< true > . . . 895
gdcm::StreamImageReader
StreamImageReader . . . 895
gdcm::StreamImageWriter
StreamImageReader . . . 899
gdcm::StrictScanner
StrictScanner . . . 906
gdcm::String< TDelimiter, TMaxLength, TPadChar >
String . . . 914
gdcm::StringFilter
StringFilter . . . 919
gdcm::Study
Study . . . 922
gdcm::Subject
Subject . . . 923
gdcm::Surface
This class defines a SURFACE IE . . . 926
gdcm::SurfaceHelper
SurfaceHelper . . . 938
gdcm::SurfaceReader
This class defines a SURFACE IE reader . . . 941
gdcm::SurfaceWriter
This class defines a SURFACE IE writer . . . 944
gdcm::SwapCode
SwapCode representation . . . 947
gdcm::SwapperDoOp . . . 949
gdcm::SwapperNoOp . . . 950
gdcm::System
Class to do system operation . . . 951
gdcm::Table
Table . . . 957
gdcm::TableEntry
TableEntry . . . 959
gdcm::TableReader
Class for representing a TableReader . . . 960
gdcm::network::TableRow . . . 964
gdcm::Tag
Class to represent a DICOM Data Element (Attribute) Tag (Group, Element) . . . 965
gdcm::TagPath
Class to handle a path of tag . . . 974
gdcm::Testing
Class for testing . . . 976

gdcm::Trace
Trace . . . 983
gdcm::TransferSyntax
Class to manipulate Transfer Syntax . . . 988
gdcm::network::TransferSyntaxSub
TransferSyntaxSub . . . 994
gdcm::network::Transition . . . 996
gdcm::Type
Type . . . 998
gdcm::UI . . . 1000
gdcm::UIDGenerator
Class for generating unique UID . . . 1001
gdcm::UIDs
All known uids . . . 1004
gdcm::network::ULAction
ULAction . . . 1029
gdcm::network::ULActionAA1 . . . 1032
gdcm::network::ULActionAA2 . . . 1033
gdcm::network::ULActionAA3 . . . 1035
gdcm::network::ULActionAA4 . . . 1036
gdcm::network::ULActionAA5 . . . 1037
gdcm::network::ULActionAA6 . . . 1038
gdcm::network::ULActionAA7 . . . 1040
gdcm::network::ULActionAA8 . . . 1041
gdcm::network::ULActionAE1 . . . 1042
gdcm::network::ULActionAE2 . . . 1043
gdcm::network::ULActionAE3 . . . 1045
gdcm::network::ULActionAE4 . . . 1046
gdcm::network::ULActionAE5 . . . 1047
gdcm::network::ULActionAE6 . . . 1048
gdcm::network::ULActionAE7 . . . 1050
gdcm::network::ULActionAE8 . . . 1051
gdcm::network::ULActionAR1 . . . 1052
gdcm::network::ULActionAR10 . . . 1053
gdcm::network::ULActionAR2 . . . 1055
gdcm::network::ULActionAR3 . . . 1056
gdcm::network::ULActionAR4 . . . 1057
gdcm::network::ULActionAR5 . . . 1058
gdcm::network::ULActionAR6 . . . 1060
gdcm::network::ULActionAR7 . . . 1061
gdcm::network::ULActionAR8 . . . 1062
gdcm::network::ULActionAR9 . . . 1063
gdcm::network::ULActionDT1 . . . 1065
gdcm::network::ULActionDT2 . . . 1066
gdcm::network::ULBasicCallback
ULBasicCallback . . . 1067
gdcm::network::ULConnection
ULConnection . . . 1069
gdcm::network::ULConnectionCallback . . . 1074
gdcm::network::ULConnectionInfo
ULConnectionInfo . . . 1076
gdcm::network::ULConnectionManager
ULConnectionManager . . . 1078

gdcm::network::ULEvent
ULEvent . . . 1085
gdcm::network::ULTransitionTable
ULTransitionTable The transition table of all the ULEvents, new ULActions, and ULStates . . . 1086
gdcm::network::ULWritingCallback . . . 1088
gdcm::UNExplicitDataElement
Class to read/write a DataElement as UNExplicit Data Element . . . 1090
gdcm::UNExplicitImplicitDataElement
Class to read/write a DataElement as ExplicitImplicit Data Element . . . 1092
gdcm::Unpacker12Bits
Pack/Unpack 12 bits pixel into 16bits . . . 1094
gdcm::Usage
Usage . . . 1095
gdcm::UserEvent . . . 1098
gdcm::network::UserInformation
UserInformation . . . 1099
gdcm::UUIDGenerator
Class for generating unique UUID . . . 1101
gdcm::Validate
Validate class . . . 1102
gdcm::Value
Class to represent the value of a Data Element . . . 1104
gdcm::ValueIO< TDE, TSwap, TType >
Class to dispatch template calls . . . 1106
gdcm::Version
Major/minor and build version . . . 1107
gdcm::VL
Value Length . . . 1109
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 . . . 1113
gdcm::VMToLength< T > . . . 1118
gdcm::VR
VR class . . . 1118
gdcm::VR16ExplicitDataElement
Class to read/write a DataElement as Explicit Data Element . . . 1124
gdcm::VRToEncoding< T > . . . 1127
gdcm::VRToType< T > . . . 1127
gdcm::VRVLSize< T > . . . 1128
gdcm::VRVLSize< 0 > . . . 1128
gdcm::VRVLSize< 1 > . . . 1128
vtkGDCMImageReader . . . 1129
vtkGDCMImageReader2 . . . 1142
vtkGDCMImageWriter . . . 1154
vtkGDCMMedicalImageProperties . . . 1162
vtkGDCMPolyDataReader . . . 1165
vtkGDCMPolyDataWriter . . . 1169
vtkGDCMTesting . . . 1173
vtkGDCMThreadedImageReader . . . 1177
vtkGDCMThreadedImageReader2 . . . 1180
vtkImageColorViewer . . . 1187
vtkImageMapToColors16 . . . 1200
vtkImageMapToWindowLevelColors2 . . . 1205
vtkImagePlanarComponentsToComponents . . . 1209

vtkImageRGBToYBR 1211
vtkImageYBRToRGB 1213
vtkLookupTable16 1215
vtkRTStructSetProperties 1218
gdcm::Waveform
Waveform class 1227
gdcm::WLMFindQuery
PatientRootQuery 1228
gdcm::Writer
Writer ala DOM (Document Object Model) 1231
gdcm::XMLDictReader
Class for representing a XMLDictReader 1237
gdcm::XMLPrinter 1240
gdcm::XMLPrivateDictReader
Class for representing a XMLPrivateDictReader 1243

Chapter 8

File Index

8.1 File List

Here is a list of all files with brief descriptions:

gdcmAAbortPDU.h . . . 1247
gdcmAAssociateACPDU.h . . . 1248
gdcmAAssociateRJPDU.h . . . 1248
gdcmAAssociateRQPDU.h . . . 1249
gdcmAbstractSyntax.h . . . 1250
gdcmAnonymizeEvent.h . . . 1251
gdcmAnonymizer.h . . . 1252
gdcmApplicationContext.h . . . 1253
gdcmApplicationEntity.h . . . 1254
gdcmAReleaseRPPDU.h . . . 1254
gdcmAReleaseRQPDU.h . . . 1255
gdcmARTIMTimer.h . . . 1256
gdcmASN1.h . . . 1257
gdcmAsynchronousOperationsWindowSub.h . . . 1258
gdcmAttribute.h . . . 1258
gdcmAudioCodec.h . . . 1260
gdcmBase64.h . . . 1260
gdcmBaseCompositeMessage.h . . . 1261
gdcmBaseNormalizedMessage.h . . . 1262
gdcmBasePDU.h . . . 1263
gdcmBaseQuery.h . . . 1264
gdcmBaseRootQuery.h . . . 1265
gdcmBasicOffsetTable.h . . . 1266
gdcmBitmap.h . . . 1267
gdcmBitmapToBitmapFilter.h . . . 1268
gdcmBoxRegion.h . . . 1269
gdcmByteBuffer.h . . . 1269
gdcmByteSwap.h . . . 1271
gdcmByteSwapFilter.h . . . 1271
gdcmByteValue.h . . . 1272
gdcmCAPICryptoFactory.h . . . 1273

gdcmCAPICryptographicMessageSyntax.h . . . 1274
gdcmCEchoMessages.h . . . 1274
gdcmCFindMessages.h . . . 1275
gdcmCMoveMessages.h . . . 1276
gdcmCodec.h . . . 1277
gdcmCoder.h . . . 1278
gdcmCodeString.h . . . 1280
gdcmCommand.h . . . 1281
gdcmCommandDataSet.h . . . 1282
gdcmCompositeMessageFactory.h . . . 1283
gdcmCompositeNetworkFunctions.h . . . 1283
gdcmConstCharWrapper.h . . . 1284
gdcmCP246ExplicitDataElement.h . . . 1284
gdcmCryptoFactory.h . . . 1285
gdcmCryptographicMessageSyntax.h . . . 1286
gdcmCSAElement.h . . . 1287
gdcmCSAHeader.h . . . 1288
gdcmCSAHeaderDict.h . . . 1289
gdcmCSAHeaderDictEntry.h . . . 1290
gdcmCStoreMessages.h . . . 1291
gdcmCurve.h . . . 1292
gdcmDataElement.h . . . 1293
gdcmDataEvent.h . . . 1295
gdcmDataSet.h . . . 1296
gdcmDataSetEvent.h . . . 1297
gdcmDataSetHelper.h . . . 1297
gdcmDecoder.h . . . 1298
gdcmDefinedTerms.h . . . 1300
gdcmDeflateStream.h . . . 1300
gdcmDefs.h . . . 1301
gdcmDeltaEncodingCodec.h . . . 1302
gdcmDICOMDIR.h . . . 1302
gdcmDICOMDIRGenerator.h . . . 1303
gdcmDict.h . . . 1304
gdcmDictConverter.h . . . 1305
gdcmDictEntry.h . . . 1306
gdcmDictPrinter.h . . . 1307
gdcmDicts.h . . . 1308
gdcmDIMSE.h . . . 1309
gdcmDirectionCosines.h . . . 1309
gdcmDirectory.h . . . 1310
gdcmDirectoryHelper.h . . . 1311
gdcmDummyValueGenerator.h . . . 1312
gdcmDumper.h . . . 1312
gdcmElement.h . . . 1313
gdcmEncapsulatedDocument.h . . . 1315
gdcmEnumeratedValues.h . . . 1316
gdcmEvent.h . . . 1316
gdcmException.h . . . 1318
gdcmExplicitDataElement.h . . . 1319
gdcmExplicitImplicitDataElement.h . . . 1320
gdcmFiducials.h . . . 1320
gdcmFile.h . . . 1321
gdcmFileAnonymizer.h . . . 1322

gdcmFileChangeTransferSyntax.h . . . 1323
gdcmFileDecompressLookupTable.h . . . 1323
gdcmFileDerivation.h . . . 1324
gdcmFileExplicitFilter.h . . . 1325
gdcmFileMetaInformation.h . . . 1326
gdcmFilename.h . . . 1327
gdcmFileNameEvent.h . . . 1327
gdcmFilenameGenerator.h . . . 1328
gdcmFileSet.h . . . 1329
gdcmFileStreamer.h . . . 1330
gdcmFindPatientRootQuery.h . . . 1331
gdcmFindStudyRootQuery.h . . . 1332
gdcmFragment.h . . . 1332
gdcmGlobal.h . . . 1334
gdcmGroupDict.h . . . 1335
gdcmIconImage.h . . . 1335
gdcmIconImageFilter.h . . . 1336
gdcmIconImageGenerator.h . . . 1337
gdcmImage.h . . . 1338
gdcmImageApplyLookupTable.h . . . 1339
gdcmImageChangePhotometricInterpretation.h . . . 1340
gdcmImageChangePlanarConfiguration.h . . . 1340
gdcmImageChangeTransferSyntax.h . . . 1341
gdcmImageCodec.h . . . 1342
gdcmImageConverter.h . . . 1343
gdcmImageFragmentSplitter.h . . . 1343
gdcmImageHelper.h . . . 1344
gdcmImageReader.h . . . 1345
gdcmImageRegionReader.h . . . 1346
gdcmImageToImageFilter.h . . . 1347
gdcmImageWriter.h . . . 1348
gdcmImplementationClassUIDSub.h . . . 1348
gdcmImplementationUIDSub.h . . . 1349
gdcmImplementationVersionNameSub.h . . . 1350
gdcmImplicitDataElement.h . . . 1351
gdcmIOD.h . . . 1352
gdcmIODEntry.h . . . 1353
gdcmIODs.h . . . 1355
gdcmIPPSorter.h . . . 1356
gdcmItem.h . . . 1357
gdcmJPEG12Codec.h . . . 1359
gdcmJPEG16Codec.h . . . 1359
gdcmJPEG2000Codec.h . . . 1360
gdcmJPEG8Codec.h . . . 1361
gdcmJPEGCodec.h . . . 1362
gdcmJPEGLSCodec.h . . . 1364
gdcmJSON.h . . . 1364
gdcmKAKADUCodec.h . . . 1365
gdcmLegacyMacro.h . . . 1366
gdcmLO.h . . . 1367
gdcmLookupTable.h . . . 1368
gdcmMacro.h . . . 1369
gdcmMacroEntry.h . . . 1371
gdcmMacros.h . . . 1372

gdcmMaximumLengthSub.h . . . 1374
gdcmMD5.h . . . 1375
gdcmMediaStorage.h . . . 1376
gdcmMeshPrimitive.h . . . 1377
gdcmModalityPerformedProcedureStepCreateQuery.h . . . 1378
gdcmModalityPerformedProcedureStepSetQuery.h . . . 1379
gdcmModule.h . . . 1379
gdcmModuleEntry.h . . . 1381
gdcmModules.h . . . 1383
gdcmMovePatientRootQuery.h . . . 1384
gdcmMoveStudyRootQuery.h . . . 1385
gdcmNActionMessages.h . . . 1385
gdcmNCreateMessages.h . . . 1386
gdcmNDeleteMessages.h . . . 1387
gdcmNestedModuleEntries.h . . . 1387
gdcmNetworkEvents.h . . . 1389
gdcmNetworkStateID.h . . . 1390
gdcmNEventReportMessages.h . . . 1391
gdcmNGetMessages.h . . . 1391
gdcmNormalizedMessageFactory.h . . . 1392
gdcmNormalizedNetworkFunctions.h . . . 1393
gdcmNSetMessages.h . . . 1393
gdcmObject.h . . . 1394
gdcmOpenSSLCryptoFactory.h . . . 1395
gdcmOpenSSLCryptographicMessageSyntax.h . . . 1396
gdcmOpenSSLP7CryptoFactory.h . . . 1397
gdcmOpenSSLP7CryptographicMessageSyntax.h . . . 1397
gdcmOrientation.h . . . 1399
gdcmOverlay.h . . . 1399
gdcmParseException.h . . . 1400
gdcmParser.h . . . 1402
gdcmPatient.h . . . 1402
gdcmPDataTFPDU.h . . . 1403
gdcmPDBElement.h . . . 1404
gdcmPDBHeader.h . . . 1405
gdcmPDFCodec.h . . . 1406
gdcmPDUFactory.h . . . 1407
gdcmPersonName.h . . . 1407
gdcmPGXCodec.h . . . 1408
gdcmPhotometricInterpretation.h . . . 1409
gdcmPixelFormat.h . . . 1410
gdcmPixmap.h . . . 1411
gdcmPixmapReader.h . . . 1412
gdcmPixmapToPixmapFilter.h . . . 1413
gdcmPixmapWriter.h . . . 1413
gdcmPNMCodec.h . . . 1415
gdcmPreamble.h . . . 1415
gdcmPresentationContext.h . . . 1417
gdcmPresentationContextAC.h . . . 1418
gdcmPresentationContextGenerator.h . . . 1419
gdcmPresentationContextRQ.h . . . 1419
gdcmPresentationDataValue.h . . . 1420
gdcmPrinter.h . . . 1421
gdcmPrivateTag.h . . . 1423

gdcmProgressEvent.h . . . 1424
gdcmPVRGCodec.h . . . 1425
gdcmPythonFilter.h . . . 1425
gdcmQueryBase.h . . . 1426
gdcmQueryFactory.h . . . 1427
gdcmQueryImage.h . . . 1428
gdcmQueryPatient.h . . . 1429
gdcmQuerySeries.h . . . 1430
gdcmQueryStudy.h . . . 1431
gdcmRAWCodec.h . . . 1432
gdcmReader.h . . . 1432
gdcmRegion.h . . . 1434
gdcmRescaler.h . . . 1435
gdcmRLECodec.h . . . 1436
gdcmRoleSelectionSub.h . . . 1436
gdcmScanner.h . . . 1437
gdcmSegment.h . . . 1438
gdcmSegmentedPaletteColorLookupTable.h . . . 1439
gdcmSegmentHelper.h . . . 1440
gdcmSegmentReader.h . . . 1441
gdcmSegmentWriter.h . . . 1442
gdcmSequenceOfFragments.h . . . 1443
gdcmSequenceOfItems.h . . . 1444
gdcmSerieHelper.h . . . 1444
gdcmSeries.h . . . 1446
gdcmServiceClassApplicationInformation.h . . . 1447
gdcmServiceClassUser.h . . . 1448
gdcmSHA1.h . . . 1448
gdcmSimpleSubjectWatcher.h . . . 1449
gdcmSmartPointer.h . . . 1450
gdcmSOPClassExtendedNegociationSub.h . . . 1451
gdcmSOPClassUIDToIOD.h . . . 1452
gdcmSorter.h . . . 1453
gdcmSpacing.h . . . 1455
gdcmSpectroscopy.h . . . 1455
gdcmSplitMosaicFilter.h . . . 1456
gdcmStaticAssert.h . . . 1456
gdcmStreamImageReader.h . . . 1458
gdcmStreamImageWriter.h . . . 1459
gdcmStrictScanner.h . . . 1459
gdcmString.h . . . 1460
gdcmStringFilter.h . . . 1462
gdcmStudy.h . . . 1462
gdcmSubject.h . . . 1463
gdcmSurface.h . . . 1464
gdcmSurfaceHelper.h . . . 1465
gdcmSurfaceReader.h . . . 1466
gdcmSurfaceWriter.h . . . 1467
gdcmSwapCode.h . . . 1468
gdcmSwapper.h . . . 1469
gdcmSystem.h . . . 1469
gdcmTable.h . . . 1470
gdcmTableEntry.h . . . 1471
gdcmTableReader.h . . . 1473

gdcmTag.h . . . 1474
gdcmTagPath.h . . . 1475
gdcmTagToVR.h . . . 1475
gdcmTerminal.h . . . 1476
gdcmTestDriver.h . . . 1477
gdcmTesting.h . . . 1478
gdcmTrace.h . . . 1479
gdcmTransferSyntax.h . . . 1483
gdcmTransferSyntaxSub.h . . . 1485
gdcmType.h . . . 1486
gdcmTypes.h . . . 1487
gdcmUIDGenerator.h . . . 1487
gdcmUIDs.h . . . 1488
gdcmULAction.h . . . 1489
gdcmULActionAA.h . . . 1490
gdcmULActionAE.h . . . 1491
gdcmULActionAR.h . . . 1491
gdcmULActionDT.h . . . 1492
gdcmULBasicCallback.h . . . 1493
gdcmULConnection.h . . . 1493
gdcmULConnectionCallback.h . . . 1494
gdcmULConnectionInfo.h . . . 1495
gdcmULConnectionManager.h . . . 1497
gdcmULEvent.h . . . 1497
gdcmULTransitionTable.h . . . 1499
gdcmULWritingCallback.h . . . 1500
gdcmUNExplicitDataElement.h . . . 1500
gdcmUNExplicitImplicitDataElement.h . . . 1501
gdcmUnpacker12Bits.h . . . 1502
gdcmUsage.h . . . 1502
gdcmUserInformation.h . . . 1504
gdcmUUIDGenerator.h . . . 1505
gdcmValidate.h . . . 1505
gdcmValue.h . . . 1506
gdcmValueIO.h . . . 1507
gdcmVersion.h . . . 1507
gdcmVL.h . . . 1508
gdcmVM.h . . . 1509
gdcmVR.h . . . 1511
gdcmVR16ExplicitDataElement.h . . . 1513
gdcmWaveform.h . . . 1514
gdcmWin32.h . . . 1514
gdcmWLMFindQuery.h . . . 1515
gdcmWriter.h . . . 1516
gdcmXMLDictReader.h . . . 1517
gdcmXMLPrinter.h . . . 1517
gdcmXMLPrivateDictReader.h . . . 1518
vtkGDCMImageReader.h . . . 1519
vtkGDCMImageReader2.h . . . 1520
vtkGDCMImageWriter.h . . . 1522
vtkGDCMMedicalImageProperties.h . . . 1522
vtkGDCMPolyDataReader.h . . . 1523
vtkGDCMPolyDataWriter.h . . . 1523
vtkGDCMTesting.h . . . 1524

vtkGDCMThreadedImageReader.h 1524
vtkGDCMThreadedImageReader2.h 1525
vtkImageColorViewer.h 1526
vtkImageMapToColors16.h 1526
vtkImageMapToWindowLevelColors2.h 1527
vtkImagePlanarComponentsToComponents.h 1527
vtkImageRGBToYBR.h 1528
vtkImageYBRToRGB.h 1528
vtkLookupTable16.h 1529
vtkRTStructSetProperties.h 1529

Chapter 9

Namespace Documentation

9.1 gdcm Namespace Reference

Namespaces

- network
- SegmentHelper
- 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 instanciation 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 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 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_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 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 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 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 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 disapear 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 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<'\\', 64, 0 > UIComp
- 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

- ignore_char const backslash ('\\')
- 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 Version &v)
- std::ostream & operator<< (std::ostream &_os, const NestedModuleEntries &_val)
- std::ostream & operator<< (std::ostream &os, const SwapCode &sc)
- std::ostream & operator<< (std::ostream &os, const FileSet &f)
- std::ostream & operator<< (std::ostream &os, const Region &r)
- std::ostream & operator<< (std::ostream &os, Event &e)

  Generic inserter operator for Event and its subclasses.
- std::ostream & operator<< (std::ostream &os, const PDBElement &val)
- std::ostream & operator<< (std::ostream &os, const CommandDataSet &val)
- std::ostream & operator<< (std::ostream &os, const Orientation &o)
- std::ostream & operator<< (std::ostream &_os, const IODs &_val)
- std::ostream & operator<< (std::ostream &_os, const Macros &_val)
- std::ostream & operator<< (std::ostream &_os, const Modules &_val)
- std::ostream & operator<< (std::ostream &_os, const Type &val)
- std::ostream & operator<< (std::ostream &_os, const ModuleEntry &_val)
- std::ostream & operator<< (std::ostream &_os, const GroupDict &_val)
- std::ostream & operator<< (std::ostream &os, const PrivateTag &val)
- std::ostream & operator<< (std::ostream &_os, const IOD &_val)
- std::ostream & operator<< (std::ostream &os, const File &val)
- std::ostream & operator<< (std::ostream &_os, const Usage &val)
- std::ostream & operator<< (std::ostream &os, const Sorter &s)
- std::ostream & operator<< (std::ostream &os, const CSAHeaderDictEntry &val)
- std::ostream & operator<< (std::ostream &os, const Preamble &val)
- std::ostream & operator<< (std::ostream &_os, const IODEntry &_val)
- std::ostream & operator<< (std::ostream &_os, const Macro &_val)
- std::ostream & operator<< (std::ostream &os, const CSAHeaderDict &val)
- std::ostream & operator<< (std::ostream &os, const Dicts &d)
- std::ostream & operator<< (std::ostream &os, const PDBHeader &d)
- std::ostream & operator<< (std::ostream &os, const CodeString &str)
- std::ostream & operator<< (std::ostream &os, const Directory &d)
- std::ostream & operator<< (std::ostream &_os, const Module &_val)
- std::ostream & operator<< (std::ostream &os, const PhotometricInterpretation &val)
- std::ostream & operator<< (std::ostream &os, const Global &g)
- std::ostream & operator<< (std::ostream &os, const Object &obj)
- std::ostream & operator<< (std::ostream &os, const BasicOffsetTable &val)
- std::ostream & operator<< (std::ostream &os, const DictEntry &val)
- std::ostream & operator<< (std::ostream &os, const VL &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 FileMetaInformation &val)
- std::ostream & operator<< (std::ostream &_os, const TransferSyntax &ts)
- std::ostream & operator<< (std::ostream &_os, const VM &_val)
- std::ostream & operator<< (std::ostream &os, const StrictScanner &s)
- std::ostream & operator<< (std::ostream &os, const Scanner &s)
- std::ostream & operator<< (std::ostream &os, const Dict &val)
- std::ostream & operator<< (std::ostream &_os, const MediaStorage &ms)
- std::ostream & operator<< (std::ostream &_os, const VR &val)
- std::ostream & operator<< (std::ostream &os, const Fragment &val)
- std::ostream & operator<< (std::ostream &os, const PixelFormat &pf)

- std::ostream & operator<< (std::ostream &_os, const UI &_val)
- std::ostream & operator<< (std::ostream &os, const DataElement &val)
- std::ostream & operator<< (std::ostream &_os, const Tag &_val)
- std::ostream & operator<< (std::ostream &os, const Item &val)
- std::ostream & operator<< (std::ostream &os, const DataSet &val)
- std::ostream & operator<< (std::ostream &os, const PrivateDict &val)
- std::ostream & operator<< (std::ostream &_os, const UIDs &uid)
- bool operator== (const CodeString &ref, const CodeString &cs)
- template<char TDelimiter, unsigned int TMaxLength, char TPadChar>
  std::istream & operator>> (std::istream &is, String< TDelimiter, TMaxLength, TPadChar > &ms)
- std::istream & operator>> (std::istream &in, ignore_char const &ic)
- std::istream & operator>> (std::istream &_is, Tag &_val)
- template<typename Float >
  std::string to_string (Float data)
- TYPETOENCODING (SQ, VRBINARY, unsigned char) TYPETOENCODING(UN

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 UIComp

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

9.1.2.18 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 backslash()

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

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

9.1.4.2 GetVRFromTag()

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

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

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

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

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

9.1.4.5 operator<<() [1/55]

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

References gdcm::Version::Print().

9.1.4.6 operator<<() [2/55]

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

References gdcm::ModuleEntry::DataElementType, gdcm::ModuleEntry::DescriptionField, and gdcm::ModuleEntry::↩ Name.

9.1.4.7 operator<<() [3/55]

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

References gdcm::SwapCode::GetSwapCodeString().

9.1.4.8 operator<<() [4/55]

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

9.1.4.9 operator<<() [5/55]

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

References gdcm::Region::Print().

9.1.4.10 operator<<() [6/55]

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

Generic inserter operator for Event and its subclasses.

References gdcm::Event::Print().

9.1.4.11 operator<<() [7/55]

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

References gdcm::PDBElement::NameField, and gdcm::PDBElement::ValueField.

9.1.4.12 operator<<() [8/55]

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

References gdcm::DataSet::Print().

9.1.4.13 operator<<() [9/55]

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

References gdcm::Orientation::Print().

9.1.4.14 operator<<() [10/55]

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

9.1.4.15 operator<<() [11/55]

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

9.1.4.16 operator<<() [12/55]

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

9.1.4.17 operator<<() [13/55]

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

References gdcm::Type::GetTypeString().

9.1.4.18 operator<<() [14/55]

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

References gdcm::ModuleEntry::DataElementType, gdcm::ModuleEntry::DescriptionField, and gdcm::ModuleEntry::↩ Name.

9.1.4.19 operator<<() [15/55]

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

References gdcm::GroupDict::GetAbbreviation(), gdcm::GroupDict::GetName(), and gdcm::GroupDict::Size().

9.1.4.20 operator<<() [16/55]

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

9.1.4.21 operator<<() [17/55]

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

9.1.4.22 operator<<() [18/55]

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

References gdcm::File::GetHeader().

9.1.4.23 operator<<() [19/55]

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

References gdcm::Usage::GetUsageString().

9.1.4.24 operator<<() [20/55]

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

References gdcm::Sorter::Print().

9.1.4.25 operator<<() [21/55]

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

9.1.4.26 operator<<() [22/55]

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

9.1.4.27 operator<<() [23/55]

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

9.1.4.28 operator<<() [24/55]

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

9.1.4.29 operator<<() [25/55]

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

9.1.4.30 operator<<() [26/55]

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

9.1.4.31 operator<<() [27/55]

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

References gdcm::PDBHeader::Print().

9.1.4.32 operator<<() [28/55]

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

9.1.4.33 operator<<() [29/55]

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

References gdcm::Directory::Print().

Referenced by gdcm::CSAElement::CSAElement(), gdcm::CSAHeaderDict::CSAHeaderDict(), gdcm::CSAHeaderDictEntry::CSAHeaderDictEntry(), gdcm::Dict::Dict(), gdcm::DictEntry::DictEntry(), gdcm::VL::GetLength(), gdcm::TransferSyntax::GetString(), gdcm::IOD::IOD(), gdcm::IODEntry::IODEntry(), gdcm::IODs::IODs(), gdcm::Macro::Macro(), gdcm::Macros::Macros(), gdcm::Module::Module(), gdcm::Modules::Modules(), gdcm::NestedModuleEntries::NestedModuleEntries(), gdcm::MediaStorage::operator MSType(), gdcm::Type::operator TypeType(), gdcm::Usage::operator UsageType(), gdcm::VM::operator VMType(), gdcm::PDBElement::PDBElement(), gdcm::PhotometricInterpretation::PhotometricInterpretation(), gdcm::SwapCode::SwapCode(), gdcm::Tag::Tag(), gdcm::VR::Write(), gdcm::CommandDataSet::~CommandDataSet(), gdcm::GroupDict::~GroupDict(), and gdcm::ModuleEntry::~ModuleEntry().

9.1.4.34 operator<<() [30/55]

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

9.1.4.35 operator<<() [31/55]

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

References gdcm::PhotometricInterpretation::GetPIString().

9.1.4.36 operator<<() [32/55]

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

9.1.4.37 operator<<() [33/55]

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

References gdcm::Object::Print().

9.1.4.38 operator<<() [34/55]

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

References gdcm::DataElement::GetByteValue(), gdcm::DataElement::ValueField, and gdcm::DataElement::ValueLengthField.

Referenced by gdcm::BasicOffsetTable::BasicOffsetTable().

9.1.4.39 operator<<() [35/55]

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

9.1.4.40 operator<<() [36/55]

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

9.1.4.41 operator<<() [37/55]

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

References gdcm::CSAElement::DataField, gdcm::ByteValue::GetLength(), gdcm::ByteValue::GetPointer(), gdcm::CSAElement::KeyField, gdcm::CSAElement::NameField, gdcm::CSAElement::NoOfItemsField, gdcm::CSAElement::SyngoDTField, gdcm::CSAElement::ValueMultiplicityField, gdcm::VM::VM1, and gdcm::CSAElement::VRField.

9.1.4.42 operator<<() [38/55]

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

References gdcm::CSAHeader::Print().

9.1.4.43 operator<<() [39/55]

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

References gdcm::FileMetaInformation::GetPreamble(), and gdcm::DataSet::Print().

9.1.4.44 operator<<() [40/55]

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

References gdcm::TransferSyntax::GetTSString().

9.1.4.45 operator<<() [41/55]

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

References gdcm::VM::GetVMString().

9.1.4.46 operator<<() [42/55]

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

References gdcm::StrictScanner::Print().

9.1.4.47 operator<<() [43/55]

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

References gdcm::Scanner::Print().

9.1.4.48 operator<<() [44/55]

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

9.1.4.49 operator<<() [45/55]

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

References gdcm::MediaStorage::GetMSString().

9.1.4.50 operator<<() [46/55]

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

References gdcm::VR::GetVRString().

9.1.4.51 operator<<() [47/55]

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

References gdcm::DataElement::TagField, gdcm::DataElement::ValueField, and gdcm::DataElement::ValueLength↩Field.

Referenced by gdcm::Fragment::Fragment().

9.1.4.52 operator<<() [48/55]

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

References gdcm::PixelFormat::Print().

9.1.4.53 operator<<() [49/55]

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

References gdcm::UI::Internal.

9.1.4.54 operator<<() [50/55]

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

References gdcm::Object::Print(), gdcm::DataElement::TagField, gdcm::DataElement::ValueField, gdcm::Data↩Element::ValueLengthField, and gdcm::DataElement::VRField.

Referenced by gdcm::DataElement::DataElement().

9.1.4.55 operator<<() [51/55]

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

9.1.4.56 operator<<() [52/55]

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

References gdcm::DataSet::Print(), gdcm::DataElement::TagField, and gdcm::DataElement::ValueLengthField.

Referenced by gdcm::Item::Item().

9.1.4.57 operator<<() [53/55]

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

References gdcm::DataSet::Begin(), and gdcm::DataSet::Print().

Referenced by gdcm::DataSet::InsertDataElement().

9.1.4.58 operator<<() [54/55]

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

9.1.4.59 operator<<() [55/55]

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

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

9.1.4.60 operator==()

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

Examples:

DumpPhilipsECHO.cxx.

Referenced by gdcm::Value::~Value().

9.1.4.61 operator>>() [1/3]

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

Referenced by gdcm::Tag::Tag().

9.1.4.62 operator>>() [2/3]

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

References gdcm::ignore_char::m_char.

9.1.4.63 operator>>() [3/3]

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

References gdcm::Tag::SetElement(), and gdcm::Tag::SetGroup().

9.1.4.64 to_string()

```
template<typename Float >
std::string gdcm::to_string (
            Float data )
```

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

9.1.4.65 TYPETOENCODING()

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

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	

Enumerator

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	
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

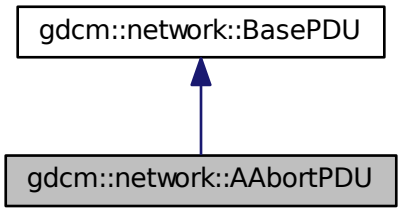
Class Documentation

10.1 gdcm::network::AAbortPDU Class Reference

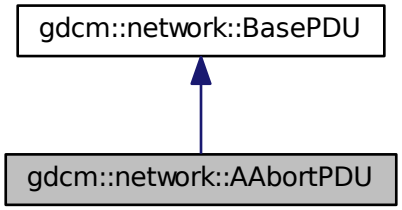
AAbortPDU.

`#include <gdcmAAbortPDU.h>`

Inheritance diagram for gdcm::network::AAbortPDU:



Collaboration diagram for gdcm::network::AAbortPDU:



Public Member Functions

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

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], [virtual]
```

Implements gdcm::network::BasePDU.

10.1.3.2 Print()

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

Implements gdcm::network::BasePDU.

10.1.3.3 Read()

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

Implements gdcm::network::BasePDU.

10.1.3.7 Write()

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

Implements gdcm::network::BasePDU.

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

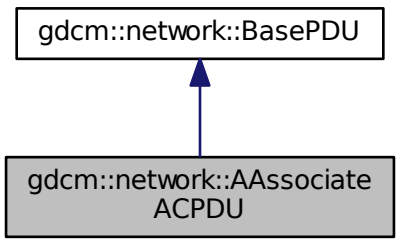
- gdcmAAbortPDU.h

10.2 gdcm::network::AAssociateACPDU Class Reference

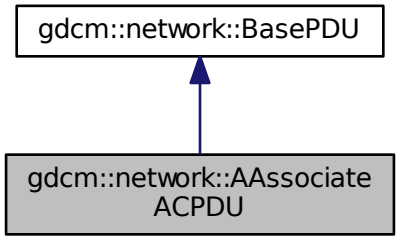
AAssociateACPDU.

```
#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
- void Print (std::ostream &os) const
- std::istream & Read (std::istream &is)
- SizeType Size () const
- const std::ostream & Write (std::ostream &os) const

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]
```

References Print(), and Size().

10.2.4.5 InitFromRQ()

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

Referenced by IsLastFragment().

10.2.4.6 IsLastFragment()

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

Implements gdcm::network::BasePDU.

References InitFromRQ().

10.2.4.7 Print()

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

Implements gdcm::network::BasePDU.

Referenced by GetUserInformation().

10.2.4.8 Read()

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

Implements gdcm::network::BasePDU.

Referenced by GetUserInformation().

10.2.4.12 Write()

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

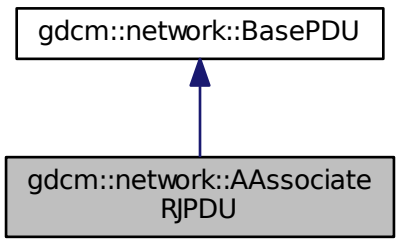
- gdcmAAssociateACPDU.h

10.3 gdcm::network::AAssociateRJPDU Class Reference

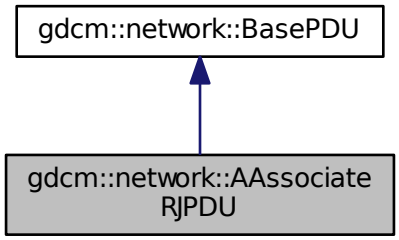
AAssociateRJPDU.

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

Inheritance diagram for gdcm::network::AAssociateRJPDU:



Collaboration diagram for gdcm::network::AAssociateRJPDU:



Public Member Functions

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

10.3.1 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], [virtual]
```

Implements gdcm::network::BasePDU.

10.3.3.2 Print()

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

Implements gdcm::network::BasePDU.

10.3.3.3 Read()

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

Implements gdcm::network::BasePDU.

10.3.3.4 Size()

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

Implements gdcm::network::BasePDU.

10.3.3.5 Write()

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

Implements gdcm::network::BasePDU.

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

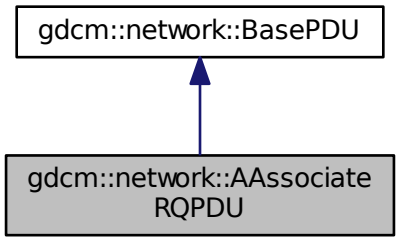
- gdcmAAssociateRJPDU.h

10.4 gdcm::network::AAssociateRQPDU Class Reference

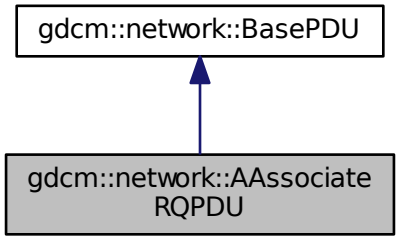
AAssociateRQPDU.

```
#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
- void Print (std::ostream &os) const
- std::istream & Read (std::istream &is)
- 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
- const std::ostream & Write (std::ostream &os) const

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::PresentationContext↩
ArrayType
```

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]
```

References SetCallingAETitle().

10.4.4.3 GetCallingAETitle()

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

References IsAETitleValid(), and Print().

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
```

Referenced by GetPresentationContexts().

10.4.4.7 GetPresentationContextByID()

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

Referenced by GetPresentationContexts().

10.4.4.8 GetPresentationContexts()

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

References GetPresentationContextByAbstractSyntax(), and GetPresentationContextByID().

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]
```

References SetUserInformation().

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

Referenced by GetCallingAETitle().

10.4.4.12 IsLastFragment()

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

Implements gdcm::network::BasePDU.

10.4.4.13 Print()

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

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

Implements gdcm::network::BasePDU.

Referenced by GetCallingAETitle().

10.4.4.14 Read()

```
std::istream& gdcm::network::AAssociateRQPDU::Read (
            std::istream & is )  [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.

Referenced by GetCalledAETitle().

10.4.4.17 SetUserInformation()

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

Referenced by GetUserInformation().

10.4.4.18 Size()

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

Implements gdcm::network::BasePDU.

10.4.4.19 Write()

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

Implements gdcm::network::BasePDU.

10.4.5 Friends And Related Function Documentation

10.4.5.1 AAssociateACPDU

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

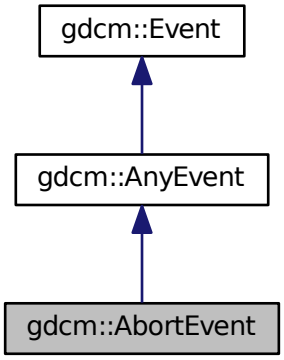
The documentation for this class was generated from the following file:

- gdcmAAssociateRQPDU.h

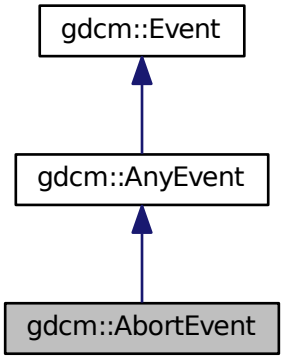
10.5 gdcm::AbortEvent Class Reference

```
#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
```

Referenced by operator==().

10.6.3.2 GetName()

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

References Print(), SetNameFromUID(), and Size().

10.6.3.3 operator==()

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

References GetAsDataElement().

10.6.3.4 Print()

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

Referenced by GetName().

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 )
```

Referenced by GetName().

10.6.3.8 Size()

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

Referenced by GetName().

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:

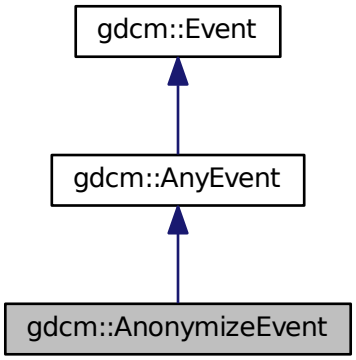
- gdcmAbstractSyntax.h

10.7 gdcm::AnonymizeEvent Class Reference

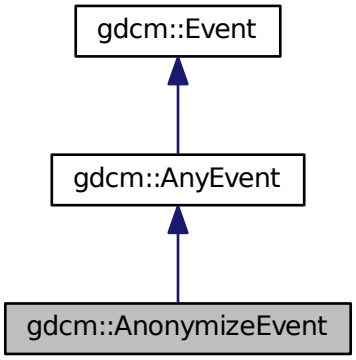
AnonymizeEvent.

`#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 (Tag const &tag=0)
- AnonymizeEvent (const Self &s)
- virtual ~AnonymizeEvent ()
- virtual bool CheckEvent (const ::gdcm::Event *e) const
- virtual const char * GetEventName () const
- Tag const & GetTag () const
- virtual ::gdcm::Event * MakeObject () const
- void SetTag (const Tag &t)

10.7.1 Detailed Description

AnonymizeEvent.

Special type of event triggered during the Anonymization process

See also

Anonymizer

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()

```
virtual gdcm::AnonymizeEvent::~AnonymizeEvent ( )  [inline], [virtual]
```

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()

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

10.7.4.2 GetEventName()

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

Return the StringName associated with the event.

Implements gdcm::Event.

10.7.4.3 GetTag()

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

10.7.4.4 MakeObject()

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

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

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

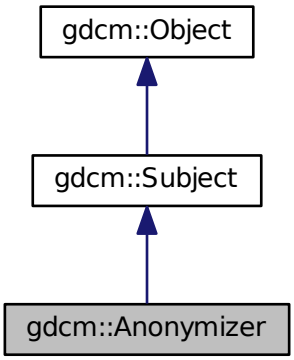
- gdcmAnonymizeEvent.h

10.8 gdcm::Anonymizer Class Reference

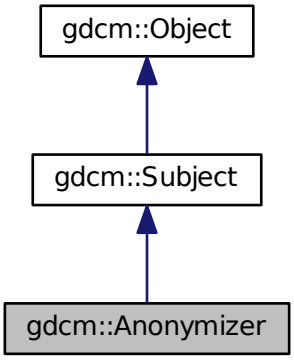
Anonymizer.

```
#include <gdcmAnonymizer.h>
```

Inheritance diagram for gdcm::Anonymizer:



Collaboration diagram for gdcm::Anonymizer:



Public Member Functions

- Anonymizer ()
- ~Anonymizer ()
- bool BasicApplicationLevelConfidentialityProfile (bool deidentify=true)
- bool Empty (Tag const &t)
- const CryptographicMessageSyntax * GetCryptographicMessageSyntax () const
- File & GetFile ()
- bool Remove (Tag const &t)
- 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 (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:

ClinicalTrialAnnotate.cxx, CreateJPIPDataSet.cxx, and EncapsulateFileInRawData.cxx.

10.8.2 Constructor & Destructor Documentation

10.8.2.1 Anonymizer()

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

10.8.2.2 ~Anonymizer()

```
gdcm::Anonymizer::~Anonymizer ( )
```

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

10.8.3.3 CanEmptyTag()

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

10.8.3.4 ClearInternalUIDs()

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

Clear the internal mapping of real UIDs to generated UIDs

Warning

the mapping is definitely lost

10.8.3.5 Empty()

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

Make Tag t empty (if not found tag will be created) Warning: does not handle SQ element

Examples:

CreateJPIPDataSet.cxx.

10.8.3.6 GetBasicApplicationLevelConfidentialityProfileAttributes()

```
static std::vector<Tag> gdcm::Anonymizer::GetBasicApplicationLevelConfidentialityProfileAttributes
( )  [static]
```

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

Examples:

GenFakeIdentifyFile.cxx, and TraverseModules.cxx.

10.8.3.7 GetCryptographicMessageSyntax()

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

10.8.3.8 GetFile()

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

10.8.3.9 New()

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

for wrapped language: instantiate a reference counted object

10.8.3.10 RecurseDataSet()

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

10.8.3.11 Remove()

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

remove a tag (even a SQ can be removed) Return code is false when tag t cannot be found

10.8.3.12 RemoveGroupLength()

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

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

Examples:

ClinicalTrialAnnotate.cxx.

10.8.3.13 RemovePrivateTags()

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

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

Examples:

ClinicalTrialAnnotate.cxx.

10.8.3.14 RemoveRetired()

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

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

10.8.3.15 Replace() [1/2]

```
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, and EncapsulateFileInRawData.cxx.

10.8.3.16 Replace() [2/2]

```
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

10.8.3.17 SetCryptographicMessageSyntax()

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

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

10.8.3.18 SetFile()

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

Set/Get File.

Examples:

ClinicalTrialAnnotate.cxx, CreateJPIPDataSet.cxx, and EncapsulateFileInRawData.cxx.

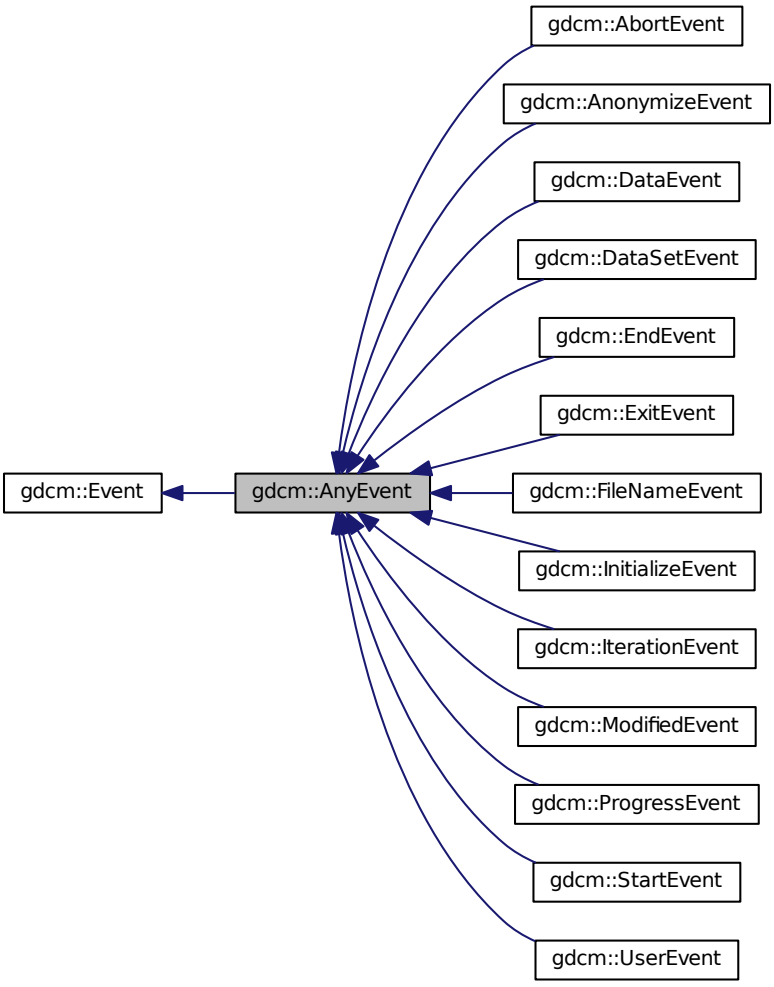
The documentation for this class was generated from the following file:

- gdcmAnonymizer.h

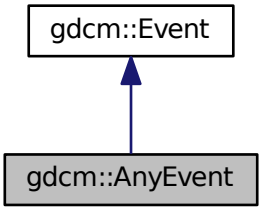
10.9 gdcm::AnyEvent Class Reference

```
#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]
```

References Print(), and Size().

10.10.3.2 Print()

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

Referenced by GetName().

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
```

Referenced by GetName().

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:

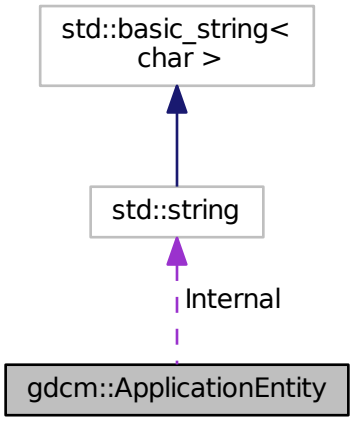
- gdcmApplicationContext.h

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:

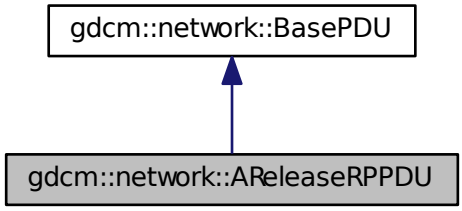
- gdcmApplicationEntity.h

10.12 gdcm::network::AReleaseRPPDU Class Reference

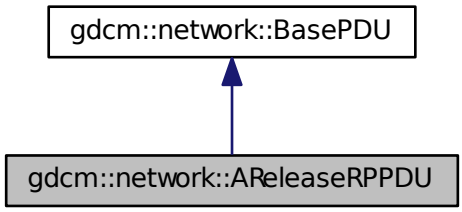
AReleaseRPPDU.

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

Inheritance diagram for gdcm::network::AReleaseRPPDU:



Collaboration diagram for gdcm::network::AReleaseRPPDU:



Public Member Functions

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

10.12.1 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], [virtual]
```

Implements gdcm::network::BasePDU.

10.12.3.2 Print()

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

Implements gdcm::network::BasePDU.

10.12.3.3 Read()

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

Implements gdcm::network::BasePDU.

10.12.3.4 Size()

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

Implements gdcm::network::BasePDU.

10.12.3.5 Write()

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

Implements gdcm::network::BasePDU.

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

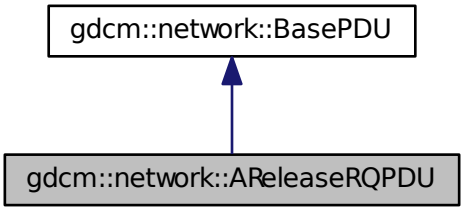
- gdcmAReleaseRPPDU.h

10.13 gdcm::network::AReleaseRQPDU Class Reference

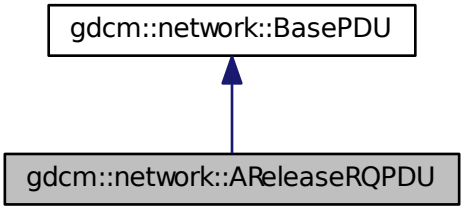
AReleaseRQPDU.

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

Inheritance diagram for gdcm::network::AReleaseRQPDU:



Collaboration diagram for gdcm::network::AReleaseRQPDU:



Public Member Functions

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

10.13.1 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], [virtual]
```

Implements gdcm::network::BasePDU.

10.13.3.2 Print()

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

Implements gdcm::network::BasePDU.

10.13.3.3 Read()

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

Implements gdcm::network::BasePDU.

10.13.3.4 Size()

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

Implements gdcm::network::BasePDU.

10.13.3.5 Write()

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

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()

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

10.15.2.2 ~ASN1()

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

10.15.3 Member Function Documentation

10.15.3.1 ParseDump()

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

10.15.3.2 ParseDumpFile()

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

10.15.3.3 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:

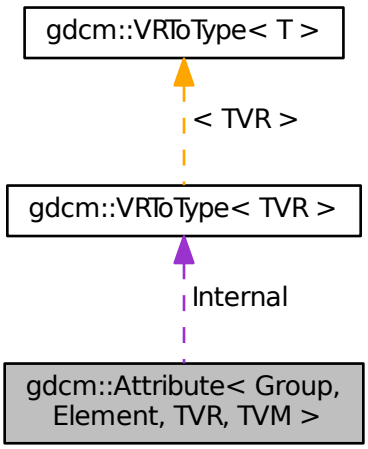
- gdcmAsynchronousOperationsWindowSub.h

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 instanciation 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::VRType)(TagToType< Group, Element >::VRType)))
- GDCM_STATIC_ASSERT (((VM::VMType) TVM &(VM::VMType)(TagToType< Group, Element >::VMType)))
- GDCM_STATIC_ASSERT (((((VR::VRType) TVR &VR::VR_VM1) &&((VM::VMType) TVM==VM::VM1))||!((VR::VRType) TVR &VR::VR_VM1)))
- 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, int 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 instanciation 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, Extracting_All_Resolution.cxx, Fake_Image_Using_Stream_Image_Writer.cxx, FixOrientation.cxx, gdcmrtionplan.cxx, gdcmrtplan.cxx, GenFakeIdentifyFile.cxx, GetSequenceUltrasound.cxx, HelloWorld.cxx, LargeVRDSExplicit.cxx, PatchFile.cxx, pmsct_rgb1.cxx, ReadAndPrintAttributes.cxx, rle2img.cxx, SortImage.cxx, StreamImageReaderTest.cxx, and VolumeSorter.cxx.

10.17.2 Member Typedef Documentation

10.17.2.1 ArrayType

```
template<uint16_t Group, uint16_t Element, int 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, int 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, int 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))  )
```

10.17.4.2 GDCM_STATIC_ASSERT() [2/3]

```
template<uint16_t Group, uint16_t Element, int 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, int 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))  )
```

10.17.4.4 GetAsDataElement()

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

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

10.17.4.5 GetDictVM()

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

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

10.17.4.8 GetTag()

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

10.17.4.9 GetValue() [1/2]

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

10.17.4.10 GetValue() [2/2]

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

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

10.17.4.16 operator==()

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

References gdcm::DataSet::GetDataElement().

10.17.4.21 SetByteValue()

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

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

10.17.4.24 SetFromDataSet()

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

References gdcm::DataSet::FindDataElement(), gdcm::DataSet::GetDataElement(), and gdcm::DataElement::Is↩ Empty().

10.17.4.25 SetValue()

```
template<uint16_t Group, uint16_t Element, int 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]
```

10.17.4.26 SetValues()

```
template<uint16_t Group, uint16_t Element, int 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:

LargeVRDSExplicit.cxx.

10.17.5 Member Data Documentation

10.17.5.1 Internal

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

Referenced by gdcm::Attribute< Group, Element, TVR, VM::VM3_3n >::GetVM().

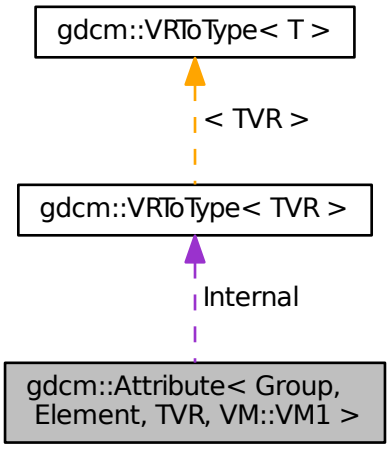
The documentation for this class was generated from the following file:

- 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 (VMToLength< VM::VM1 >::Length==1)
- GDCM_STATIC_ASSERT (((VR::VRType) TVR &(VR::VRType)(TagToType< Group, Element >::VRType)))
- GDCM_STATIC_ASSERT (((VM::VMType) VM::VM1 &(VM::VMType)(TagToType< Group, Element >::VM↩ Type)))
- GDCM_STATIC_ASSERT (((((VR::VRType) TVR &VR::VR_VM1) &&((VM::VMType) VM::VM1==VM::V↩ M1))||!((VR::VRType) TVR &VR::VR_VM1)))
- 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, int 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, int 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, int TVR>
gdcm::Attribute< Group, Element, TVR, VM::VM1 >::GDCM_STATIC_ASSERT (
            VMToLength< VM::VM1 >::Length  = 1 )
```

10.18.3.2 GDCM_STATIC_ASSERT() [2/4]

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

10.18.3.3 GDCM_STATIC_ASSERT() [3/4]

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

10.18.3.4 GDCM_STATIC_ASSERT() [4/4]

```
template<uint16_t Group, uint16_t Element, int 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))  )
```

10.18.3.5 GetAsDataElement()

```
template<uint16_t Group, uint16_t Element, int 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, int 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, int 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, int 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, int 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, int 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, int 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, int 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, int 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, int 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, int 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, int 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 >::GetValues().

10.18.3.17 operator==()

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

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

10.18.3.23 SetFromDataSet()

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

References gdcm::DataSet::FindDataElement(), gdcm::DataSet::GetDataElement(), and gdcm::DataElement::Is↩ Empty().

10.18.3.24 SetValue()

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

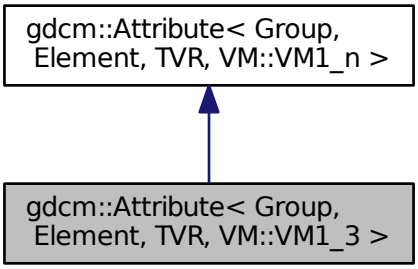
The documentation for this class was generated from the following file:

- gdcmAttribute.h

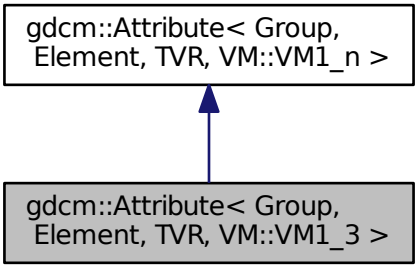
10.19 gdcm::Attribute< Group, Element, TVR, VM::VM1_3 > Class Template Reference

`#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, int TVR>
VM gdcm::Attribute< Group, Element, TVR, VM::VM1_3 >::GetVM ( ) const  [inline]
```

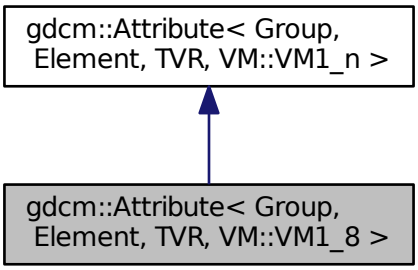
The documentation for this class was generated from the following file:

- gdcmAttribute.h

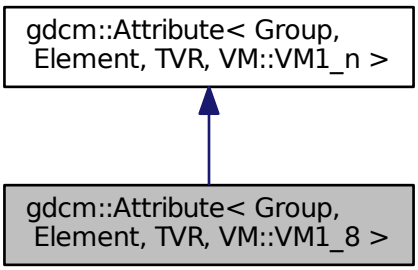
10.20 gdcm::Attribute< Group, Element, TVR, VM::VM1_8 > Class Template Reference

```
#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, int TVR>
VM gdcm::Attribute< Group, Element, TVR, VM::VM1_8 >::GetVM ( ) const  [inline]
```

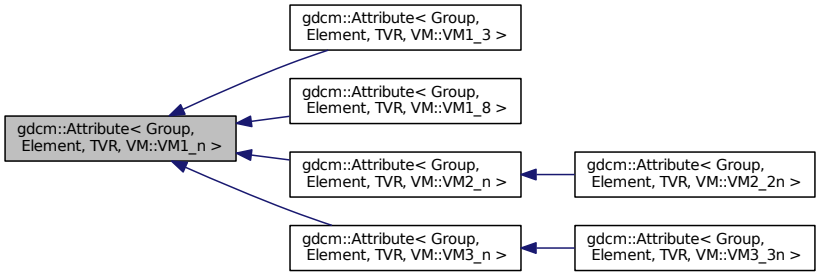
The documentation for this class was generated from the following file:

- 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::VRType)(TagToType< Group, Element >::VRType)))
- GDCM_STATIC_ASSERT ((VM::VM1_n &(VM::VMType)(TagToType< Group, Element >::VMType)))
- GDCM_STATIC_ASSERT (((((VR::VRType) TVR &VR::VR_VM1) &&((VM::VMType) TagToType< Group, Element >::VMType==VM::VM1))||!((VR::VRType) TVR &VR::VR_VM1)))
- 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 (unsigned int idx, ArrayType v)
- void SetValue (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, int 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, int TVR>
gdcm::Attribute< Group, Element, TVR, VM::VM1_n >::Attribute ( )  [inline], [explicit]
```

10.21.2.2 ~Attribute()

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

10.21.3.2 GDCM_STATIC_ASSERT() [2/3]

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

10.21.3.3 GDCM_STATIC_ASSERT() [3/3]

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

10.21.3.4 GetAsDataElement()

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

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

10.21.3.20 SetFromDataSet()

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

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

10.21.3.21 SetNumberOfValues()

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

10.21.3.23 SetValue() [2/2]

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

References SetValue().

Referenced by SetValue().

10.21.3.24 SetValues()

```
template<uint16_t Group, uint16_t Element, int TVR>
void gdcm::Attribute< Group, Element, TVR, VM::VM1_n >::SetValues (
            const ArrayType * array,
            unsigned int numel,
            bool own = false )  [inline]
```

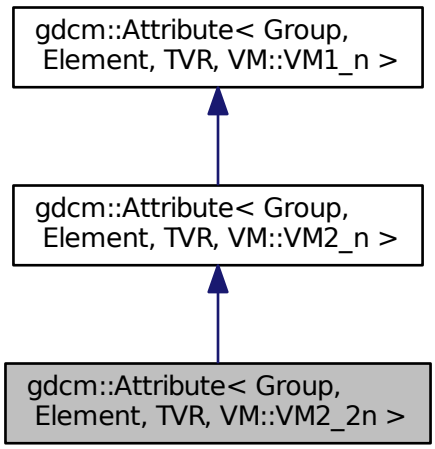
The documentation for this class was generated from the following file:

- gdcmAttribute.h

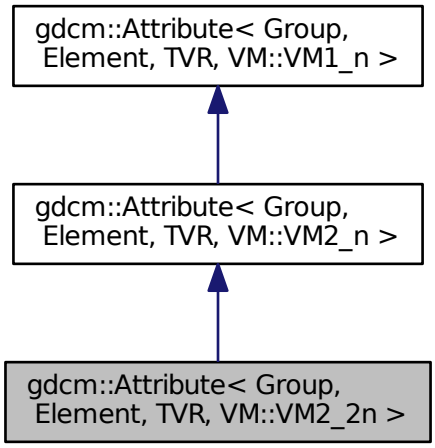
10.22 gdcm::Attribute< Group, Element, TVR, VM::VM2_2n > Class Template Reference

`#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, int TVR>
static VM gdcm::Attribute< Group, Element, TVR, VM::VM2_2n >::GetVM ( )  [inline], [static]
```

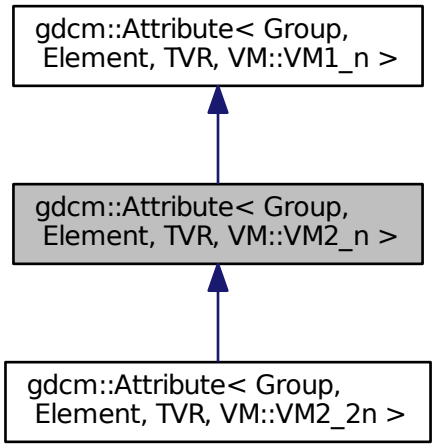
The documentation for this class was generated from the following file:

- gdcmAttribute.h

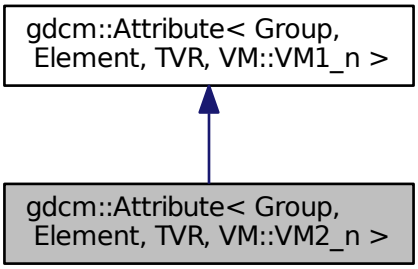
10.23 gdcm::Attribute< Group, Element, TVR, VM::VM2_n > Class Template Reference

```
#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, int TVR>
VM gdcm::Attribute< Group, Element, TVR, VM::VM2_n >::GetVM ( ) const  [inline]
```

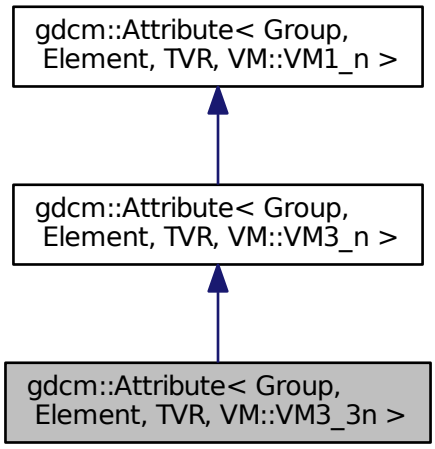
The documentation for this class was generated from the following file:

- gdcmAttribute.h

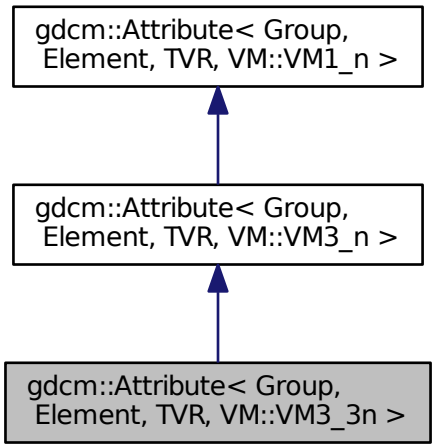
10.24 gdcm::Attribute< Group, Element, TVR, VM::VM3_3n > Class Template Reference

```
#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, int TVR>
static VM gdcm::Attribute< Group, Element, TVR, VM::VM3_3n >::GetVM ( )  [inline], [static]
```

References gdcm::Attribute< Group, Element, TVR, TVM >::Internal.

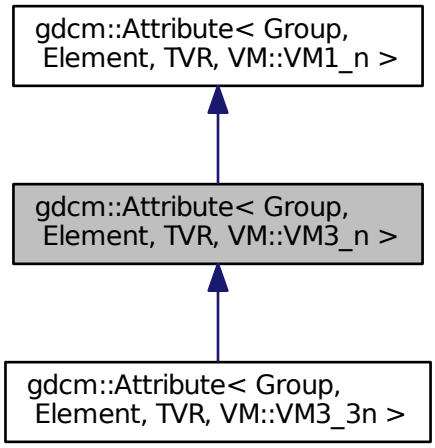
The documentation for this class was generated from the following file:

- gdcmAttribute.h

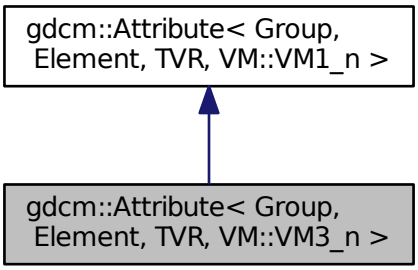
10.25 gdcm::Attribute< Group, Element, TVR, VM::VM3_n > Class Template Reference

```
#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, int 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:

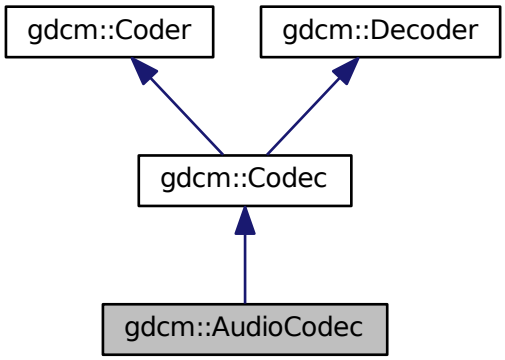
- gdcmAttribute.h

10.26 gdcm::AudioCodec Class Reference

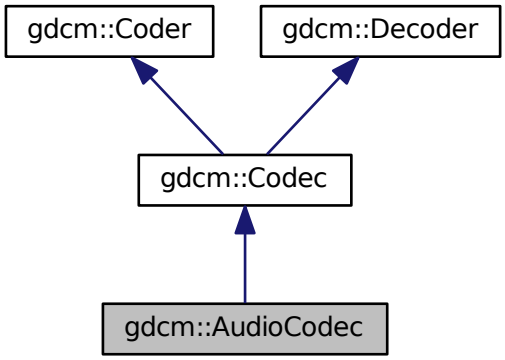
AudioCodec.

```
#include <gdcmAudioCodec.h>
```

Inheritance diagram for gdcm::AudioCodec:



Collaboration diagram for gdcm::AudioCodec:



Public Member Functions

- AudioCodec ()
- ~AudioCodec ()
- bool CanCode (TransferSyntax const &) const

  Return whether this coder support this transfer syntax (can code it)
- bool CanDecode (TransferSyntax const &) const

  Return whether this decoder support this transfer syntax (can decode it)
- bool Decode (DataElement const &is, DataElement &os)

  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 ( )
```

10.26.3 Member Function Documentation

10.26.3.1 CanCode()

```
bool gdcm::AudioCodec::CanCode (
            TransferSyntax const &  ) const  [inline], [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], [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 &  )  [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>
```

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 Member Function Documentation

10.27.2.1 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.

10.27.2.2 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

10.27.2.3 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.

10.27.2.4 GetEncodeLength()

```
static size_t gdcm::Base64::GetEncodeLength (
            const char * src,
            size_t srclen )  [static]
```

Call this function to obtain the required buffer size

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

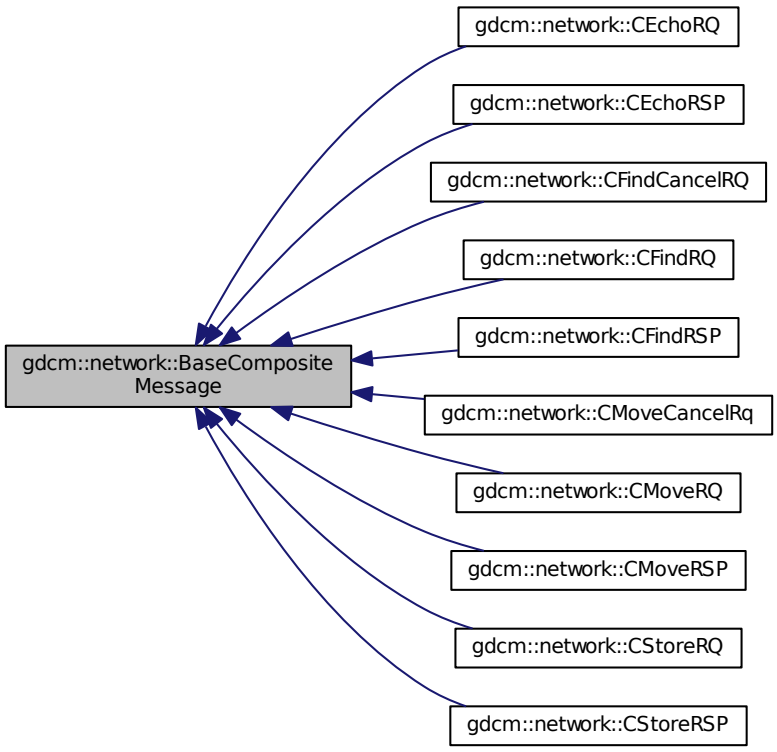
- gdcmBase64.h

10.28 gdcm::network::BaseCompositeMessage Class Reference

BaseCompositeMessage.

`#include <gdcmBaseCompositeMessage.h>`

Inheritance diagram for gdcm::network::BaseCompositeMessage:



Public Member Functions

- virtual ~BaseCompositeMessage ()
- 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 ( )  [inline], [virtual]
```

References ConstructPDV().

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::CMoveRQ, gdcm::network::CFindRQ, and gdcm::network::CEchoRQ.

Referenced by ~BaseCompositeMessage().

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

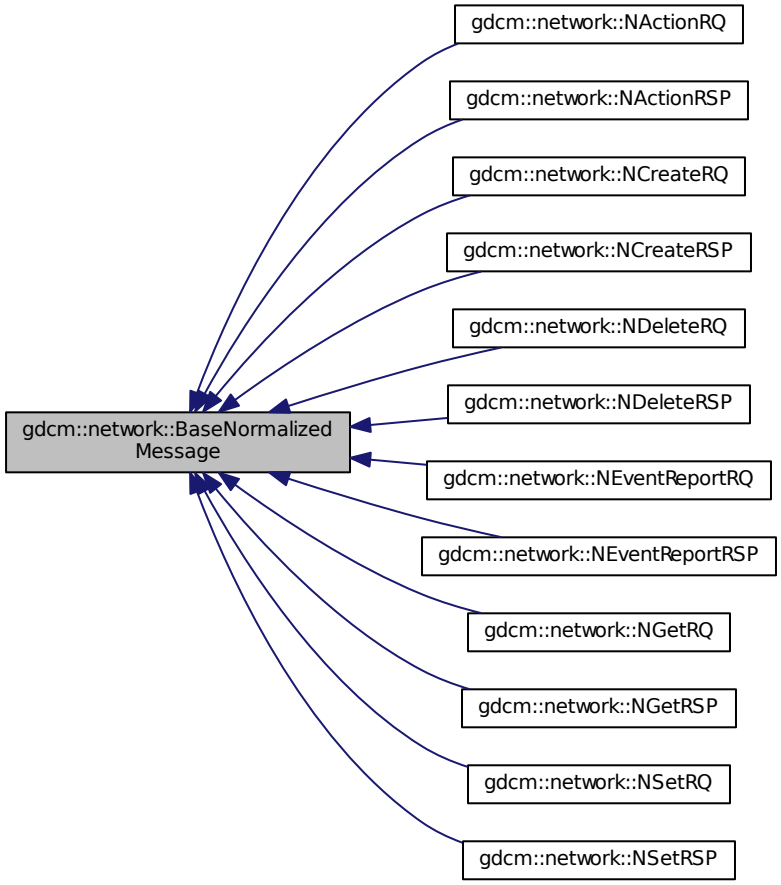
- gdcmBaseCompositeMessage.h

10.29 gdcm::network::BaseNormalizedMessage Class Reference

BaseNormalizedMessage.

```
#include <gdcmBaseNormalizedMessage.h>
```

Inheritance diagram for gdcm::network::BaseNormalizedMessage:



Public Member Functions

- virtual ~BaseNormalizedMessage ()
- virtual std::vector< PresentationDataValue > ConstructPDV (const ULConnection &inConnection, const Base↩ Query *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 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, gdcmNormalized↩ MessageFactory.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 ( )  [inline], [virtual]
```

References ConstructPDV().

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.

Referenced by ~BaseNormalizedMessage().

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

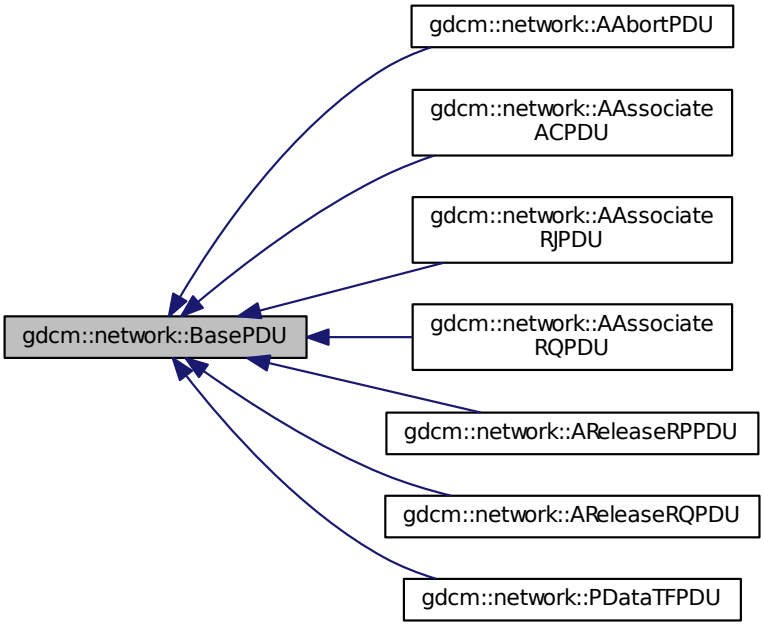
- gdcmBaseNormalizedMessage.h

10.30 gdcm::network::BasePDU Class Reference

BasePDU.

`#include <gdcmBasePDU.h>`

Inheritance diagram for gdcm::network::BasePDU:



Public Member Functions

- virtual ~BasePDU ()
- 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 ( )  [inline], [virtual]
```

References IsLastFragment(), Print(), Read(), Size(), and Write().

10.30.3 Member Function Documentation

10.30.3.1 IsLastFragment()

```
virtual bool gdcm::network::BasePDU::IsLastFragment ( ) const  [pure virtual]
```

Implemented in gdcm::network::AAssociateRQPDU, gdcm::network::AAssociateACPDU, gdcm::network::PDataTFPDU, gdcm::network::AAbortPDU, gdcm::network::AAssociateRJPDU, gdcm::network::AReleaseRPPDU, and gdcm::network::AReleaseRQPDU.

Referenced by ~BasePDU().

10.30.3.2 Print()

```
virtual void gdcm::network::BasePDU::Print (
            std::ostream & os ) const  [pure virtual]
```

Implemented in gdcm::network::AAssociateRQPDU, gdcm::network::AAssociateACPDU, gdcm::network::PDataTFPDU, gdcm::network::AAbortPDU, gdcm::network::AReleaseRPPDU, gdcm::network::AReleaseRQPDU, and gdcm::network::AAssociateRJPDU.

Referenced by ~BasePDU().

10.30.3.3 Read()

```
virtual std::istream& gdcm::network::BasePDU::Read (
            std::istream & is )  [pure virtual]
```

Implemented in gdcm::network::AAssociateACPDU, gdcm::network::AAssociateRQPDU, gdcm::network::PDataTF↩PDU, gdcm::network::AAssociateRJPDU, gdcm::network::AReleaseRPPDU, gdcm::network::AReleaseRQPDU, and gdcm::network::AAbortPDU.

Referenced by ~BasePDU().

10.30.3.4 Size()

```
virtual size_t gdcm::network::BasePDU::Size ( ) const  [pure virtual]
```

Implemented in gdcm::network::AAssociateACPDU, gdcm::network::AAssociateRQPDU, gdcm::network::PDataTFP↩DU, gdcm::network::AAbortPDU, gdcm::network::AAssociateRJPDU, gdcm::network::AReleaseRPPDU, and gdcm↩::network::AReleaseRQPDU.

Referenced by ~BasePDU().

10.30.3.5 Write()

```
virtual const std::ostream& gdcm::network::BasePDU::Write (
            std::ostream & os ) const  [pure virtual]
```

Implemented in gdcm::network::AAssociateACPDU, gdcm::network::AAssociateRQPDU, gdcm::network::PDataTF↩PDU, gdcm::network::AAssociateRJPDU, gdcm::network::AReleaseRPPDU, gdcm::network::AReleaseRQPDU, and gdcm::network::AAbortPDU.

Referenced by ~BasePDU().

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

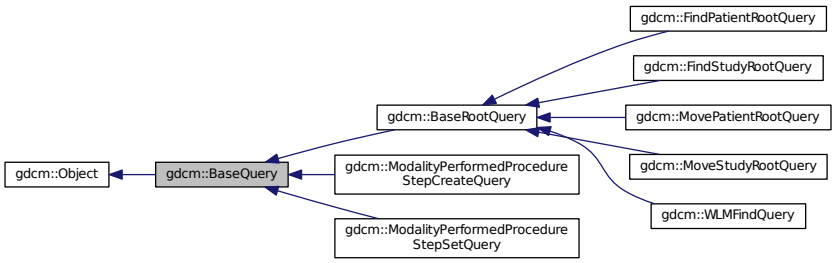
- gdcmBasePDU.h

10.31 gdcm::BaseQuery Class Reference

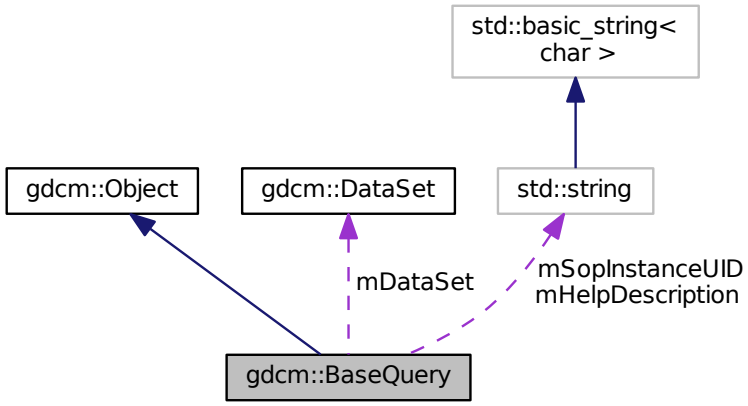
BaseQuery.

`#include <gdcmBaseQuery.h>`

Inheritance diagram for gdcm::BaseQuery:



Collaboration diagram for gdcm::BaseQuery:



Public Member Functions

- virtual ~BaseQuery ()
- void AddQueryDataSet (const DataSet &ds)
- virtual UIDs::TSName GetAbstractSyntaxUID () const =0
- DataSet const & GetQueryDataSet () const

  Set/Get the internal representation of the query as a DataSet.

- DataSet & GetQueryDataSet ()
- std::string GetSOPInstanceUID () const
- void Print (std::ostream &os) const
- void SetSearchParameter (const Tag &inTag, const std::string &inValue)
- void SetSearchParameter (const std::string &inKeyword, 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 mHelpDescription
- 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()

```
virtual gdcm::BaseQuery::~BaseQuery ( )  [virtual]
```

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::FindStudyRootQuery, gdcm::MovePatientRootQuery, gdcm::MoveStudyRootQuery, gdcm::↩ WLMFindQuery, gdcm::FindPatientRootQuery, gdcm::ModalityPerformedProcedureStepCreateQuery, and gdcm::↩ ModalityPerformedProcedureStepSetQuery.

10.31.3.3 GetQueryDataSet() [1/2]

```
DataSet const& gdcm::BaseQuery::GetQueryDataSet ( ) const
```

Set/Get the internal representation of the query as a DataSet.

10.31.3.4 GetQueryDataSet() [2/2]

```
DataSet& gdcm::BaseQuery::GetQueryDataSet ( )
```

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  [virtual]
```

Reimplemented from gdcm::Object.

10.31.3.7 SetSearchParameter() [1/3]

```
void gdcm::BaseQuery::SetSearchParameter (
            const Tag & inTag,
            const DictEntry & inDictEntry,
            const std::string & inValue )  [protected]
```

10.31.3.8 SetSearchParameter() [2/3]

```
void gdcm::BaseQuery::SetSearchParameter (
            const Tag & inTag,
            const std::string & inValue )
```

10.31.3.9 SetSearchParameter() [3/3]

```
void gdcm::BaseQuery::SetSearchParameter (
            const std::string & inKeyword,
            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::BaseRootQuery, gdcm::FindStudyRootQuery, gdcm::MovePatientRootQuery, gdcm::MoveStudyRootQuery, gdcm::WLMFindQuery, gdcm::FindPatientRootQuery, gdcm::ModalityPerformedProcedureStepCreateQuery, and gdcm::ModalityPerformedProcedureStepSetQuery.

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 mHelpDescription

```
std::string gdcm::BaseQuery::mHelpDescription  [protected]
```

10.31.5.3 mSopInstanceUID

```
std::string gdcm::BaseQuery::mSopInstanceUID  [protected]
```

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

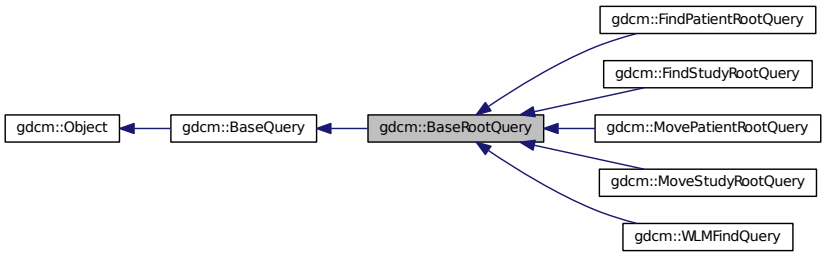
- gdcmBaseQuery.h

10.32 gdcm::BaseRootQuery Class Reference

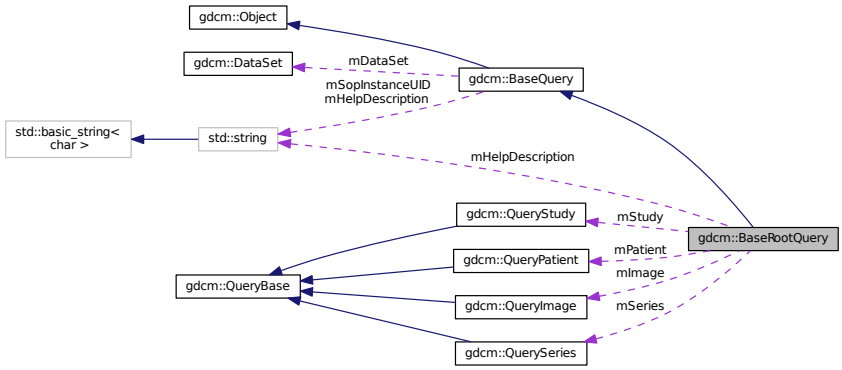
BaseRootQuery.

```
#include <gdcmBaseRootQuery.h>
```

Inheritance diagram for gdcm::BaseRootQuery:



Collaboration diagram for gdcm::BaseRootQuery:



Public Member Functions

- virtual ~BaseRootQuery ()
- EQueryLevel GetQueryLevelFromQueryRoot (ERootType roottype)
- virtual std::vector< Tag > GetTagListByLevel (const EQueryLevel &inQueryLevel)=0
- virtual void InitializeDataSet (const EQueryLevel &inQueryLevel)=0
- virtual bool ValidateQuery (bool inStrict=true) const =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()

```
virtual gdcm::BaseRootQuery::~BaseRootQuery ( )  [virtual]
```

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::WLMFindQuery, gdcm::FindPatientRootQuery, gdcm::FindStudyRootQuery, gdcm::MovePatientRootQuery, and gdcm::MoveStudyRootQuery.

10.32.3.7 ValidateQuery()

```
virtual bool gdcm::BaseRootQuery::ValidateQuery (
            bool inStrict = true ) const  [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::FindStudyRootQuery, gdcm::MovePatientRootQuery, gdcm::MoveStudyRootQuery, gdcm::WLMFindQuery, and gdcm::FindPatientRootQuery.

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:

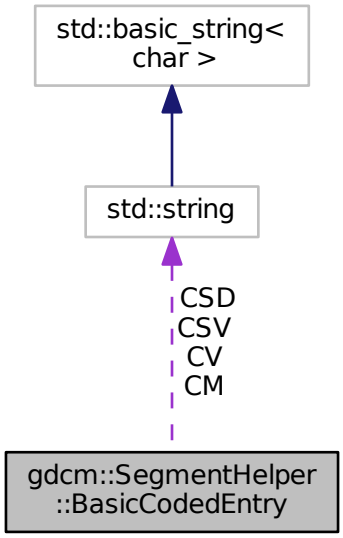
- gdcmBaseRootQuery.h

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 attibutes 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.

References IsEmpty().

10.33.3 Member Function Documentation

10.33.3.1 IsEmpty()

```
bool gdcm::SegmentHelper::BasicCodedEntry::IsEmpty (
            const bool checkOptionalAttributes = false ) const
```

Check if each attibutes of the basic coded entry is defined.

Parameters

checkOptionalAttributes	Check also type 1C attributes.

Referenced by BasicCodedEntry().

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:

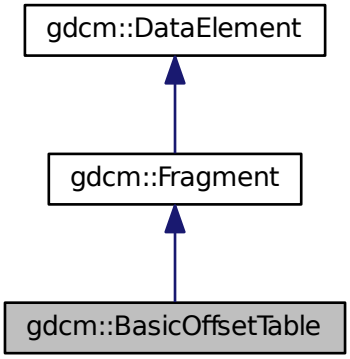
- gdcmSegmentHelper.h

10.34 gdcm::BasicOffsetTable Class Reference

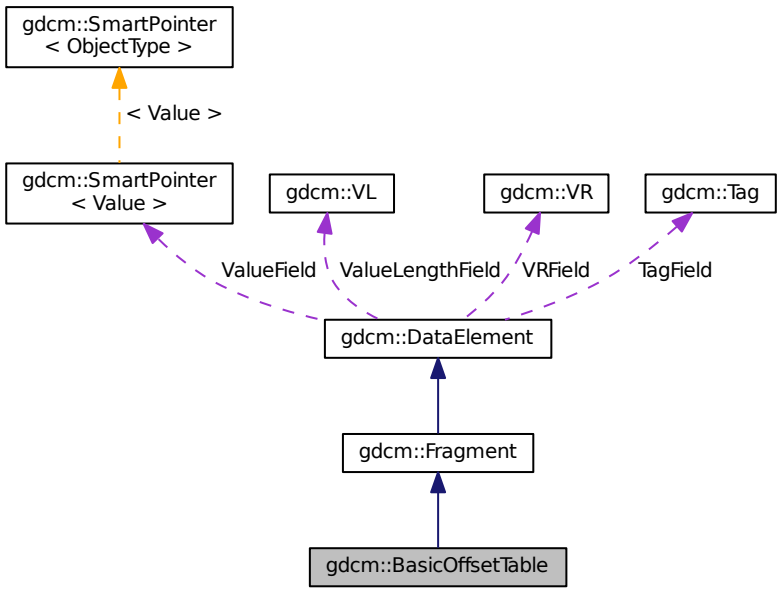
Class to represent a BasicOffsetTable.

`#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]
```

References gdcm::operator<<().

10.34.3 Member Function Documentation

10.34.3.1 Read()

```
template<typename TSwap >
std::istream& gdcm::BasicOffsetTable::Read (
            std::istream & is )  [inline]
```

References gdcmDebugMacro.

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:

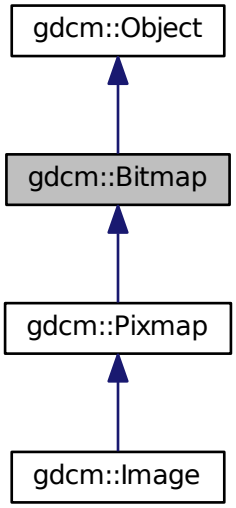
- gdcmBasicOffsetTable.h

10.35 gdcm::Bitmap Class Reference

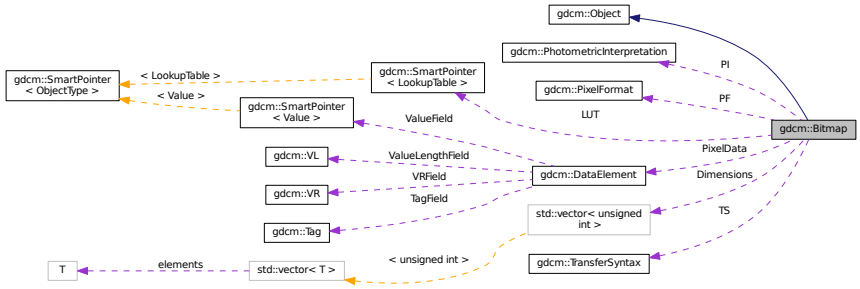
Bitmap class.

`#include <gdcmBitmap.h>`

Inheritance diagram for gdcm::Bitmap:



Collaboration diagram for gdcm::Bitmap:



Public Member Functions

- Bitmap ()

- ~Bitmap ()
- virtual bool AreOverlaysInPixelData () const
- void Clear ()
- bool GetBuffer (char *buffer) const

  Acces the raw data.
- unsigned long GetBufferLength () const
- unsigned int GetColumns () const
- const DataElement & GetDataElement () const
- DataElement & GetDataElement ()
- 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)...
- const LookupTable & GetLUT () const
- LookupTable & GetLUT ()
- bool GetNeedByteSwap () const
- 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
- const PixelFormat & GetPixelFormat () const

  Get/Set PixelFormat.
- PixelFormat & GetPixelFormat ()
- 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
- 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.

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 ( )
```

10.35.4 Member Function Documentation

10.35.4.1 AreOverlaysInPixelData()

```
virtual bool gdcm::Bitmap::AreOverlaysInPixelData ( ) const  [inline], [virtual]
```

Reimplemented in gdcm::Pixmap.

References gdcm::terminal::dim.

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
```

Acces the raw data.

Examples:

ConvertToQImage.cxx, 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:

ConvertToQImage.cxx, GenFakeImage.cxx, GetSubSequenceData.cxx, PatchFile.cxx, ReadMultiTimes↩Exception.cxx, and threadgdcm.cxx.

10.35.4.7 GetColumns()

```
unsigned int gdcm::Bitmap::GetColumns ( ) const  [inline]
```

10.35.4.8 GetDataElement() [1/2]

```
const DataElement& gdcm::Bitmap::GetDataElement ( ) const  [inline]
```

Examples:

ExtractIconFromFile.cxx.

10.35.4.9 GetDataElement() [2/2]

```
DataElement& gdcm::Bitmap::GetDataElement ( )  [inline]
```

10.35.4.10 GetDimension()

```
unsigned int gdcm::Bitmap::GetDimension (
            unsigned int idx ) const
```

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]

```
const LookupTable& gdcm::Bitmap::GetLUT ( ) const  [inline]
```

Examples:

ExtractIconFromFile.cxx.

10.35.4.13 GetLUT() [2/2]

```
LookupTable& gdcm::Bitmap::GetLUT ( )  [inline]
```

10.35.4.14 GetNeedByteSwap()

```
bool gdcm::Bitmap::GetNeedByteSwap ( ) const  [inline]
```

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:

HelloVizWorld.cxx, and threadgdcm.cxx.

10.35.4.16 GetPhotometricInterpretation()

```
const PhotometricInterpretation& gdcm::Bitmap::GetPhotometricInterpretation ( ) const
```

return the photometric interpretation

Examples:

ConvertToQImage.cxx, ExtractIconFromFile.cxx, and HelloVizWorld.cxx.

10.35.4.17 GetPixelFormat() [1/2]

```
const PixelFormat& gdcm::Bitmap::GetPixelFormat ( ) const  [inline]
```

Get/Set PixelFormat.

Examples:

ConvertToQImage.cxx, ExtractIconFromFile.cxx, FixJAIBugJPEGLS.cxx, GenFakeImage.cxx, GetJPEGSample↩ Precision.cxx, and threadgdcm.cxx.

10.35.4.18 GetPixelFormat() [2/2]

```
PixelFormat& gdcm::Bitmap::GetPixelFormat ( )  [inline]
```

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  [virtual]
```

Reimplemented from gdcm::Object.

Reimplemented in gdcm::Image, and gdcm::Pixmap.

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:

CreateARGBImage.cxx, CreateCMYKImage.cxx, csa2img.cxx, GenFakeImage.cxx, GetSubSequenceData.cxx, and iU22tomultisc.cxx.

10.35.4.28 SetDimension()

```
void gdcm::Bitmap::SetDimension (
            unsigned int idx,
            unsigned int dim )
```

Examples:

csa2img.cxx, GenFakeImage.cxx, GetSubSequenceData.cxx, and iU22tomultisc.cxx.

10.35.4.29 SetDimensions()

```
void gdcm::Bitmap::SetDimensions (
            const unsigned int dims[3] )
```

Examples:

CreateARGBImage.cxx, and CreateCMYKImage.cxx.

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, csa2img.cxx, FileChangeTS.cs, FileChangeTSLossy.cs, GenFakeImage.cxx, GetSubSequenceData.cxx, and iU22tomultisc.cxx.

10.35.4.34 SetPhotometricInterpretation()

```
void gdcm::Bitmap::SetPhotometricInterpretation (
            PhotometricInterpretation const & pi )
```

Examples:

CreateARGBImage.cxx, CreateCMYKImage.cxx, csa2img.cxx, GenFakeImage.cxx, GetSubSequenceData.cxx, and iU22tomultisc.cxx.

10.35.4.35 SetPixelFormat()

```
void gdcm::Bitmap::SetPixelFormat (
            PixelFormat const & pf )  [inline]
```

Examples:

CreateARGBImage.cxx, CreateCMYKImage.cxx, 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:

CreateARGBImage.cxx, CreateCMYKImage.cxx, and MergeTwoFiles.cxx.

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]
```

10.35.5 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:

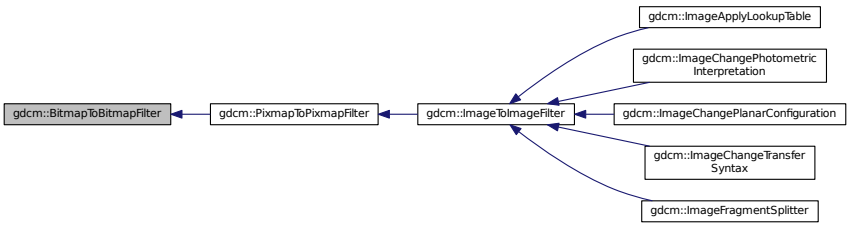
- gdcmBitmap.h

10.36 gdcm::BitmapToBitmapFilter Class Reference

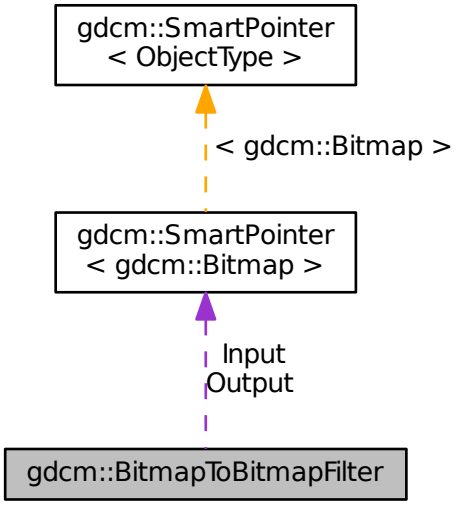
BitmapToBitmapFilter class.

```
#include <gdcmBitmapToBitmapFilter.h>
```

Inheritance diagram for gdcm::BitmapToBitmapFilter:



Collaboration diagram for gdcm::BitmapToBitmapFilter:



Public Member Functions

- BitmapToBitmapFilter ()
- ~BitmapToBitmapFilter ()
- 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 ( )  [inline]
```

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:

CompressImage.cxx.

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:

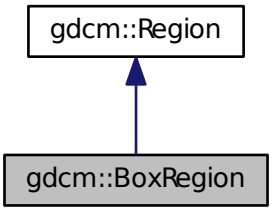
- gdcmBitmapToBitmapFilter.h

10.37 gdcm::BoxRegion Class Reference

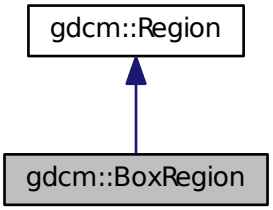
Class for manipulation box region.

```
#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 ()
- size_t Area () const

  compute the area
- Region * Clone () const
- BoxRegion ComputeBoundingBox ()

  Return the Axis-Aligned minimum bounding box for all regions.
- bool Empty () const

  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

  return whether this is valid domain
- void operator= (const BoxRegion &)
- void Print (std::ostream &os=std::cout) const

  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)

10.37.2 Constructor & Destructor Documentation

10.37.2.1 BoxRegion() [1/2]

```
gdcm::BoxRegion::BoxRegion ( )
```

10.37.2.2 ~BoxRegion()

```
gdcm::BoxRegion::~BoxRegion ( )
```

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  [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  [virtual]
```

Implements gdcm::Region.

10.37.3.4 ComputeBoundingBox()

```
BoxRegion gdcm::BoxRegion::ComputeBoundingBox ( )  [virtual]
```

Return the Axis-Aligned minimum bounding box for all regions.

Implements gdcm::Region.

10.37.3.5 Empty()

```
bool gdcm::BoxRegion::Empty ( ) const  [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  [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  [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.

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.

Detailled 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 swaping (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 (DataSet &ds)
- ~ByteSwapFilter ()
- bool ByteSwap ()
- 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()

```
gdcm::ByteSwapFilter::ByteSwapFilter (
            DataSet & ds )  [inline]
```

10.40.2.2 ~ByteSwapFilter()

```
gdcm::ByteSwapFilter::~ByteSwapFilter ( )
```

10.40.3 Member Function Documentation

10.40.3.1 ByteSwap()

```
bool gdcm::ByteSwapFilter::ByteSwap ( )
```

10.40.3.2 SetByteSwapTag()

```
void gdcm::ByteSwapFilter::SetByteSwapTag (
            bool b )  [inline]
```

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

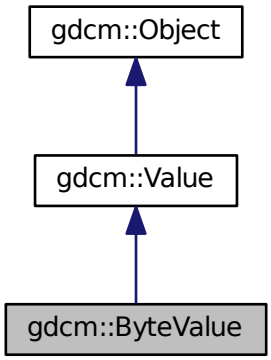
- gdcmByteSwapFilter.h

10.41 gdcm::ByteValue Class Reference

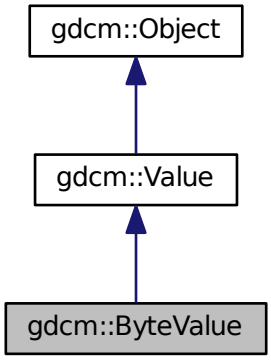
Class to represent binary value (array of bytes)

`#include <gdcmByteValue.h>`

Inheritance diagram for gdcm::ByteValue:



Collaboration diagram for gdcm::ByteValue:



Public Member Functions

- ByteValue (const char *array=0, VL const &vl=0)
- ByteValue (std::vector< char > &v)
- ~ByteValue ()
- void Append (ByteValue const &bv)
- void Clear ()
- VL ComputeLength () const
- void Fill (char c)
- bool GetBuffer (char *buffer, unsigned long length) const
- VL GetLength () const
- const char * GetPointer () 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
- 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 , typename TType >
  std::istream & Read (std::istream &is, bool readvalues=true)
- template<typename TSwap >
  std::istream & Read (std::istream &is)
- void SetLength (VL vl)
- 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
- void SetLengthOnly (VL vl)

10.41.1 Detailed Description

Class to represent binary value (array of bytes)

Note

Examples:

DumpADAC.cxx, DumpExamCard.cxx, DumpGEMSMovieGroup.cxx, DumpImageHeaderInfo.cxx, DumpPhilips↩ ECHO.cxx, DumpToshibaDTI.cxx, DuplicatePCDE.cxx, ELSCINT1WaveToText.cxx, ExtractEncryptedContent.cxx, ExtractIconFromFile.cxx, FixBrokenJ2K.cxx, FixJAIBugJPEGLS.cxx, GetSubSequenceData.cxx, MrProtocol.cxx, PatchFile.cxx, pmsct_rgb1.cxx, ReadExplicitLengthSQIVR.cxx, and rle2img.cxx.

10.41.2 Constructor & Destructor Documentation

10.41.2.1 ByteValue() [1/2]

```
gdcm::ByteValue::ByteValue (
            const char * array = 0,
            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]
```

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], [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:

FixJAIBugJPEGLS.cxx.

10.41.3.6 GetLength()

```
VL gdcm::ByteValue::GetLength ( ) const  [inline], [virtual]
```

Implements gdcm::Value.

Examples:

DumpADAC.cxx, DumpExamCard.cxx, DumpGEMSMovieGroup.cxx, DumpImageHeaderInfo.cxx, DumpPhilipsECHO.cxx, DumpToshibaDTI.cxx, ELSCINT1WaveToText.cxx, ExtractEncryptedContent.cxx, ExtractIconFromFile.cxx, FixBrokenJ2K.cxx, FixJAIBugJPEGLS.cxx, GetSubSequenceData.cxx, MrProtocol.cxx, PatchFile.cxx, pmsct_rgb1.cxx, ReadExplicitLengthSQIVR.cxx, ReadGEMSSDO.cxx, and rle2img.cxx.

Referenced by gdcm::operator<<(), gdcm::SequenceOfFragments::ReadValue(), gdcm::Element< VR::OB, VM::VM1_n >::Set(), gdcm::Element< TVR, VM::VM1_n >::Set(), 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 >::SetByteValueNoSwap(), gdcm::Attribute< Group, Element, TVR, VM::VM1 >::SetByteValueNoSwap(), gdcm::Element< VR::OB, VM::VM1_n >::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, DumpToshibaDTI.cxx, ELSCINT1WaveToText.cxx, ExtractEncryptedContent.cxx, ExtractIconFromFile.cxx, FixBrokenJ2K.cxx, GetSubSequenceData.cxx, MrProtocol.cxx, pmsct_rgb1.cxx, ReadExplicitLengthSQIVR.cxx, ReadGEMSSDO.cxx, and rle2img.cxx.

Referenced by gdcm::operator<<(), gdcm::SequenceOfFragments::ReadValue(), gdcm::Element< VR::OB, VM::VM1_n >::Set(), gdcm::Element< TVR, VM::VM1_n >::Set(), 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 >::SetByteValueNoSwap(), gdcm::Attribute< Group, Element, TVR, VM::VM1 >::SetByteValueNoSwap(), gdcm::Element< VR::OB, VM::VM1_n >::SetNoSwap(), and gdcm::Element< TVR, VM::VM1_n >::SetNoSwap().

10.41.3.8 IsEmpty()

```
bool gdcm::ByteValue::IsEmpty ( ) const  [inline]
```

10.41.3.9 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...

10.41.3.10 operator const std::vector< char > &()

```
gdcm::ByteValue::operator const std::vector< char > & ( ) const  [inline]
```

10.41.3.11 operator=()

```
ByteValue& gdcm::ByteValue::operator= (
            const ByteValue & val )  [inline]
```

10.41.3.12 operator==() [1/2]

```
bool gdcm::ByteValue::operator== (
            const ByteValue & val ) const  [inline]
```

10.41.3.13 operator==() [2/2]

```
bool gdcm::ByteValue::operator== (
            const Value & val ) const  [inline], [virtual]
```

Implements gdcm::Value.

10.41.3.14 Print()

```
void gdcm::ByteValue::Print (
            std::ostream & os ) const  [inline], [protected], [virtual]
```

Reimplemented from gdcm::Object.

10.41.3.15 PrintASCII()

```
void gdcm::ByteValue::PrintASCII (
            std::ostream & os,
            VL maxlength ) const
```

10.41.3.16 PrintASCIIXML()

```
void gdcm::ByteValue::PrintASCIIXML (
            std::ostream & os ) const
```

10.41.3.17 PrintGroupLength()

```
void gdcm::ByteValue::PrintGroupLength (
            std::ostream & os )  [inline]
```

10.41.3.18 PrintHex()

```
void gdcm::ByteValue::PrintHex (
            std::ostream & os,
            VL maxlength ) const
```

10.41.3.19 PrintHexXML()

```
void gdcm::ByteValue::PrintHexXML (
            std::ostream & os ) const
```

10.41.3.20 PrintPNXML()

```
void gdcm::ByteValue::PrintPNXML (
            std::ostream & os ) const
```

To Print Values in Native DICOM format

10.41.3.21 Read() [1/2]

```
template<typename TSwap , typename TType >
std::istream& gdcm::ByteValue::Read (
            std::istream & is,
            bool readvalues = true )  [inline]
```

10.41.3.22 Read() [2/2]

```
template<typename TSwap >
std::istream& gdcm::ByteValue::Read (
            std::istream & is )  [inline]
```

10.41.3.23 SetLength()

```
void gdcm::ByteValue::SetLength (
            VL vl )  [virtual]
```

Implements gdcm::Value.

10.41.3.24 SetLengthOnly()

```
void gdcm::ByteValue::SetLengthOnly (
            VL vl )  [inline], [protected], [virtual]
```

Reimplemented from gdcm::Value.

10.41.3.25 Write() [1/2]

```
template<typename TSwap , typename TType >
std::ostream const& gdcm::ByteValue::Write (
            std::ostream & os ) const  [inline]
```

Referenced by gdcm::Fragment::Write().

10.41.3.26 Write() [2/2]

```
template<typename TSwap >
std::ostream const& gdcm::ByteValue::Write (
            std::ostream & os ) const  [inline]
```

10.41.3.27 WriteBuffer()

```
bool gdcm::ByteValue::WriteBuffer (
            std::ostream & os ) const  [inline]
```

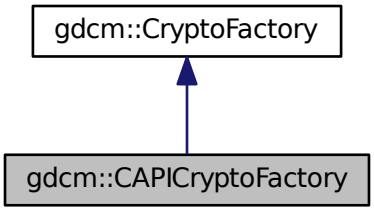
The documentation for this class was generated from the following file:

- gdcmByteValue.h

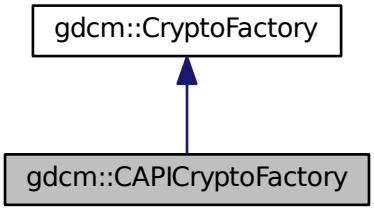
10.42 gdcm::CAPICryptoFactory Class Reference

`#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.

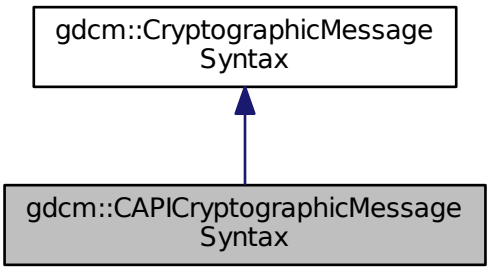
The documentation for this class was generated from the following file:

- gdcmCAPICryptoFactory.h

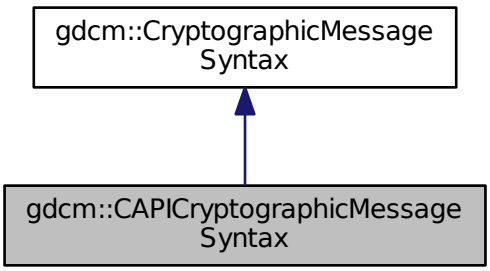
10.43 gdcm::CAPICryptographicMessageSyntax Class Reference

```
#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 )
```

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:

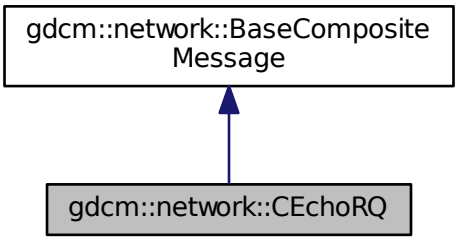
- gdcmCAPICryptographicMessageSyntax.h

10.44 gdcm::network::CEchoRQ Class Reference

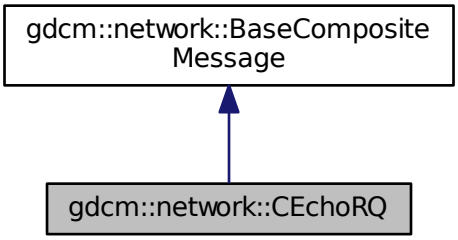
CEchoRQ.

```
#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 BaseRoot↩ Query *inRootQuery)

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 )  [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:

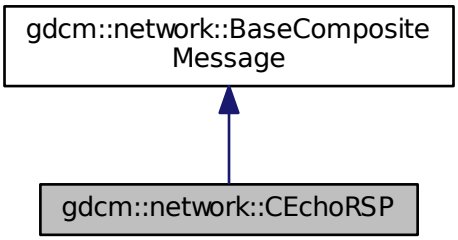
- gdcmCEchoMessages.h
- gdcmDIMSE.h

10.45 gdcm::network::CEchoRSP Class Reference

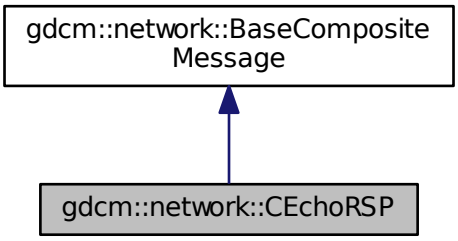
CEchoRSP this file defines the messages for the cecho action.

```
#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:

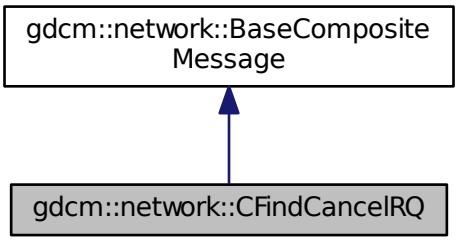
- gdcmDIMSE.h

10.47 gdcm::network::CFindCancelRQ Class Reference

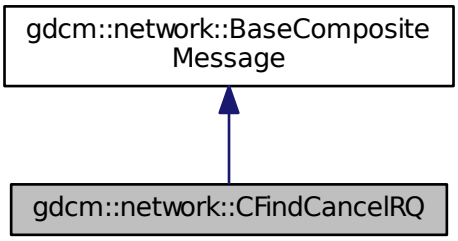
CFindCancelRQ this file defines the messages for the cfind action.

```
#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:

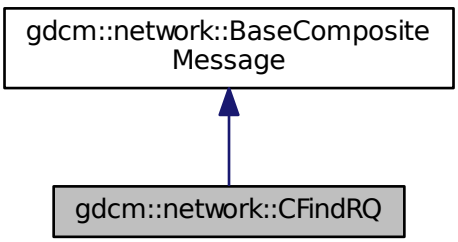
- gdcmCFindMessages.h

10.48 gdcm::network::CFindRQ Class Reference

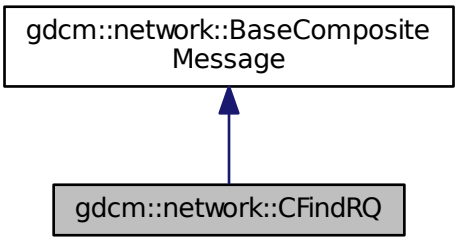
CFindRQ.

```
#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 BaseRoot↩ Query *inRootQuery)

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 )  [virtual]
```

Implements gdcm::network::BaseCompositeMessage.

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

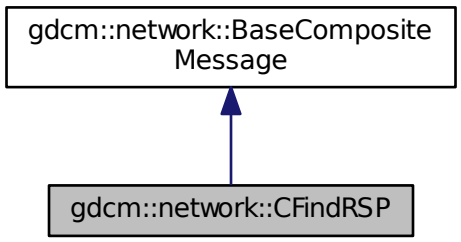
- gdcmCFindMessages.h

10.49 gdcm::network::CFindRSP Class Reference

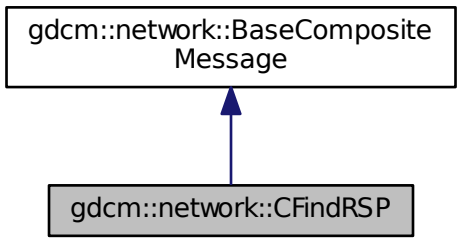
CFindRSP this file defines the messages for the cfind action.

`#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 )
```

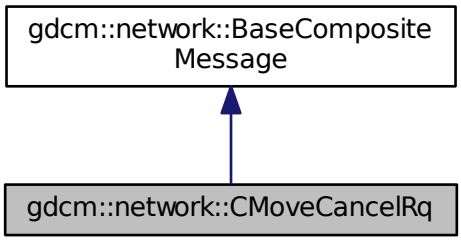
The documentation for this class was generated from the following file:

- gdcmCFindMessages.h

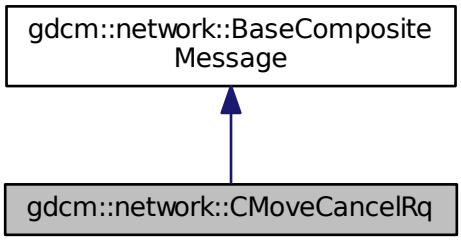
10.50 gdcm::network::CMoveCancelRq Class Reference

```
#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)

10.50.1 Member Function Documentation

10.50.1.1 ConstructPDVByDataSet()

```
std::vector<PresentationDataValue> gdcm::network::CMoveCancelRq::ConstructPDVByDataSet (
            const DataSet * inDataSet )
```

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

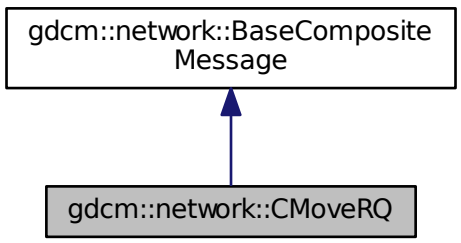
- gdcmCMoveMessages.h

10.51 gdcm::network::CMoveRQ Class Reference

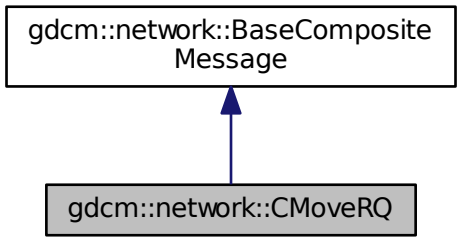
CMoveRQ.

```
#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 BaseRoot↩ Query *inRootQuery)

10.51.1 Detailed Description

CMoveRQ.

this file defines the messages for the cmove action

10.51.2 Member Function Documentation

10.51.2.1 ConstructPDV()

```
std::vector<PresentationDataValue> gdcm::network::CMoveRQ::ConstructPDV (
            const ULConnection & inConnection,
            const BaseRootQuery * inRootQuery )  [virtual]
```

Implements gdcm::network::BaseCompositeMessage.

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

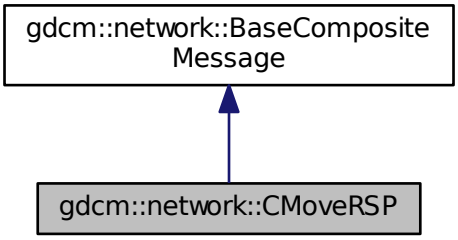
- gdcmCMoveMessages.h

10.52 gdcm::network::CMoveRSP Class Reference

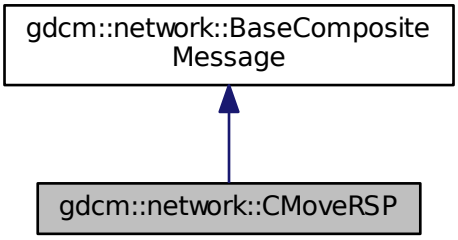
CMoveRSP this file defines the messages for the cmove action.

`#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)

10.52.1 Detailed Description

CMoveRSP this file defines the messages for the cmove action.

10.52.2 Member Function Documentation

10.52.2.1 ConstructPDVByDataSet()

```
std::vector<PresentationDataValue> gdcm::network::CMoveRSP::ConstructPDVByDataSet (
            const DataSet * inDataSet )
```

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

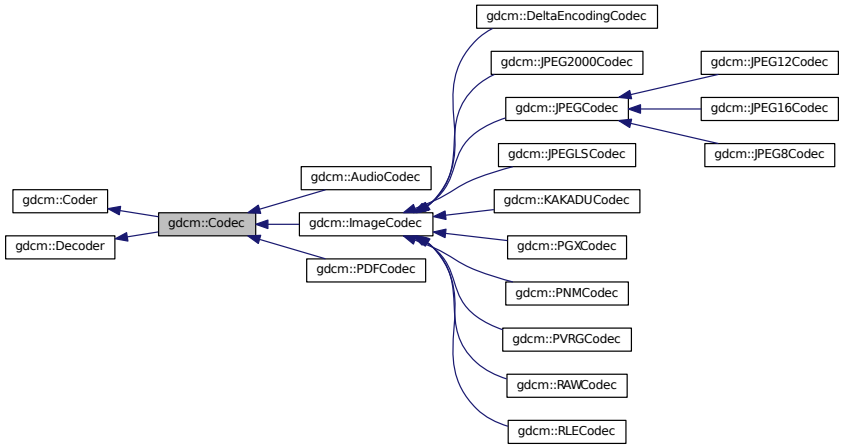
- gdcmCMoveMessages.h

10.53 gdcm::Codec Class Reference

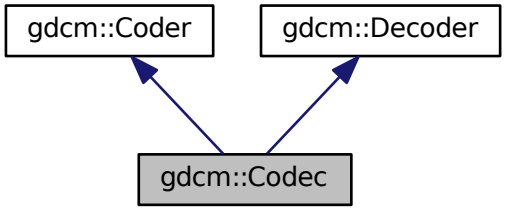
Codec class.

`#include <gdcmCodec.h>`

Inheritance diagram for gdcm::Codec:



Collaboration diagram for gdcm::Codec:



Additional Inherited Members

10.53.1 Detailed Description

Codec class.

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

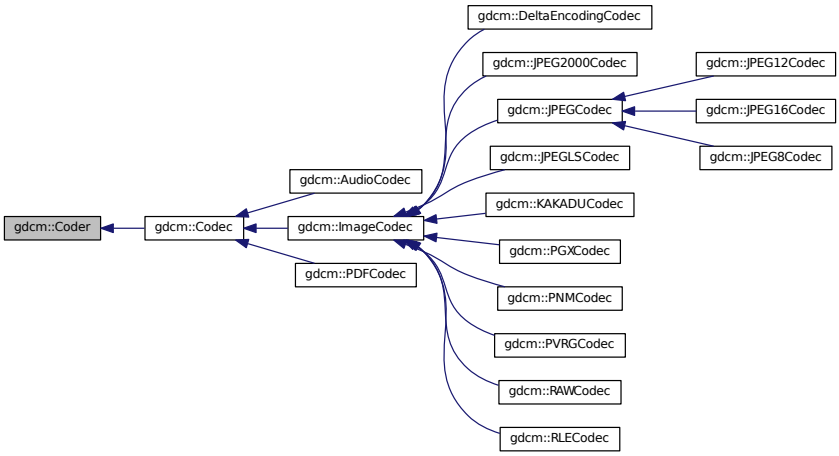
- gdcmCodec.h

10.54 gdcm::Coder Class Reference

Coder.

`#include <gdcmCoder.h>`

Inheritance diagram for gdcm::Coder:



Public Member Functions

- virtual ~Coder ()
- 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)

10.54.1 Detailed Description

Coder.

10.54.2 Constructor & Destructor Documentation

10.54.2.1 ~Coder()

```
virtual gdcm::Coder::~Coder ( )  [inline], [virtual]
```

10.54.3 Member Function Documentation

10.54.3.1 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::JPEGCodec, gdcm::RLECodec, gdcm::PVRGCodec, gdcm::JPEG2000Codec, gdcm::JPEGLSCodec, gdcm::ImageCodec, gdcm::PNMCodec, gdcm::PGXCodec, gdcm::KAKADUCodec, gdcm::RAWCodec, gdcm::AudioCodec, and gdcm::PDFCodec.

10.54.3.2 Code()

```
virtual bool gdcm::Coder::Code (
            DataElement const & in_,
            DataElement & out_ )  [inline], [virtual]
```

Code.

Reimplemented in gdcm::JPEGCodec, gdcm::RLECodec, gdcm::JPEGLSCodec, gdcm::PVRGCodec, gdcm::JPEG2000Codec, gdcm::KAKADUCodec, and gdcm::RAWCodec.

10.54.3.3 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

10.55 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 value_type *s)
- CodeString (const value_type *s, size_type n)
- CodeString (const InternalClass &s, size_type pos=0, size_type n=InternalClass::npos)
- 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)

10.55.1 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.

10.55.2 Member Typedef Documentation

10.55.2.1 const_iterator

```
typedef InternalClass::const_iterator gdcm::CodeString::const_iterator
```

10.55.2.2 const_reference

```
typedef InternalClass::const_reference gdcm::CodeString::const_reference
```

10.55.2.3 const_reverse_iterator

```
typedef InternalClass::const_reverse_iterator gdcm::CodeString::const_reverse_iterator
```

10.55.2.4 difference_type

```
typedef InternalClass::difference_type gdcm::CodeString::difference_type
```

10.55.2.5 iterator

```
typedef InternalClass::iterator gdcm::CodeString::iterator
```

10.55.2.6 pointer

```
typedef InternalClass::pointer gdcm::CodeString::pointer
```

10.55.2.7 reference

```
typedef InternalClass::reference gdcm::CodeString::reference
```

10.55.2.8 reverse_iterator

```
typedef InternalClass::reverse_iterator gdcm::CodeString::reverse_iterator
```

10.55.2.9 size_type

```
typedef InternalClass::size_type gdcm::CodeString::size_type
```

10.55.2.10 value_type

```
typedef InternalClass::value_type gdcm::CodeString::value_type
```

10.55.3 Constructor & Destructor Documentation

10.55.3.1 CodeString() [1/4]

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

CodeString constructors.

10.55.3.2 CodeString() [2/4]

```
gdcm::CodeString::CodeString (
            const value_type * s )  [inline]
```

10.55.3.3 CodeString() [3/4]

```
gdcm::CodeString::CodeString (
            const value_type * s,
            size_type n )  [inline]
```

10.55.3.4 CodeString() [4/4]

```
gdcm::CodeString::CodeString (
            const InternalClass & s,
            size_type pos = 0,
            size_type n = InternalClass::npos )  [inline]
```

10.55.4 Member Function Documentation

10.55.4.1 GetAsString()

```
std::string gdcm::CodeString::GetAsString ( ) const  [inline]
```

Return the full code string as std::string.

10.55.4.2 IsValid()

```
bool gdcm::CodeString::IsValid ( ) const
```

Check if CodeString obj is correct..

10.55.4.3 Size()

```
size_type gdcm::CodeString::Size ( ) const  [inline]
```

Return the size of the string.

10.55.4.4 TrimInternal()

```
std::string gdcm::CodeString::TrimInternal ( ) const  [inline], [protected]
```

10.55.5 Friends And Related Function Documentation

10.55.5.1 operator"!=

```
bool operator!= (
            const CodeString & ref,
            const CodeString & cs )  [friend]
```

10.55.5.2 operator<<

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

10.55.5.3 operator==

```
bool operator== (
            const CodeString & ref,
            const CodeString & cs )  [friend]
```

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

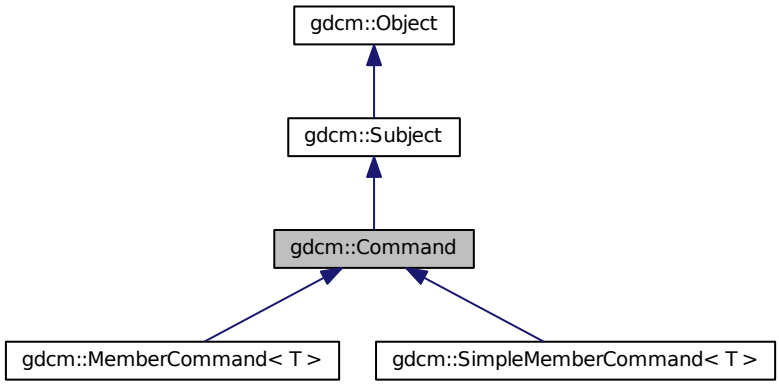
- gdcmCodeString.h

10.56 gdcm::Command Class Reference

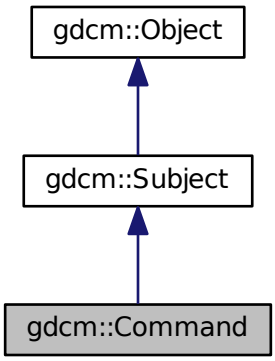
Command superclass for callback/observer methods.

```
#include <gdcmCommand.h>
```

Inheritance diagram for gdcm::Command:



Collaboration diagram for gdcm::Command:



Public Member Functions

- virtual void Execute (Subject *caller, const Event &event)=0

  Abstract method that defines the action to be taken by the command.
- virtual void Execute (const Subject *caller, const Event &event)=0

Protected Member Functions

- Command ()
- ~Command ()

10.56.1 Detailed Description

Command superclass for callback/observer methods.

See also

Subject

10.56.2 Constructor & Destructor Documentation

10.56.2.1 Command()

```
gdcm::Command::Command ( )  [protected]
```

10.56.2.2 ~Command()

```
gdcm::Command::~Command ( )  [protected]
```

10.56.3 Member Function Documentation

10.56.3.1 Execute() [1/2]

```
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 >.

10.56.3.2 Execute() [2/2]

```
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 >.

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

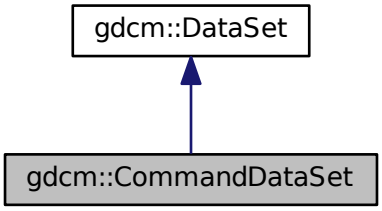
- gdcmCommand.h

10.57 gdcm::CommandDataSet Class Reference

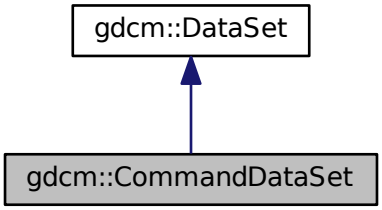
Class to represent a Command DataSet.

`#include <gdcmCommandDataSet.h>`

Inheritance diagram for gdcm::CommandDataSet:



Collaboration diagram for gdcm::CommandDataSet:



Public Member Functions

- CommandDataSet ()
- ~CommandDataSet ()
- 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

10.57.1 Detailed Description

Class to represent a Command DataSet.

See also

DataSet

10.57.2 Constructor & Destructor Documentation

10.57.2.1 CommandDataSet()

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

10.57.2.2 ~CommandDataSet()

```
gdcm::CommandDataSet::~CommandDataSet ( )  [inline]
```

References gdcm::operator<<().

10.57.3 Member Function Documentation

10.57.3.1 Insert()

```
void gdcm::CommandDataSet::Insert (
            const DataElement & de )  [inline]
```

References gdcmErrorMacro, gdcm::Tag::GetGroup(), and gdcm::DataElement::GetTag().

10.57.3.2 Read()

```
std::istream& gdcm::CommandDataSet::Read (
            std::istream & is )
```

Read.

10.57.3.3 Replace()

```
void gdcm::CommandDataSet::Replace (
            const DataElement & de )  [inline]
```

References gdcm::DataElement::GetTag().

10.57.3.4 Write()

```
std::ostream& gdcm::CommandDataSet::Write (
            std::ostream & os ) const
```

Write.

10.57.4 Friends And Related Function Documentation

10.57.4.1 operator<<

```
std::ostream& operator<< (
            std::ostream & _os,
            const CommandDataSet & _val )  [friend]
```

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

- gdcmCommandDataSet.h

10.58 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)

10.58.1 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).

10.58.2 Member Function Documentation

10.58.2.1 ConstructCEchoRQ()

```
static std::vector<PresentationDataValue> gdcm::network::CompositeMessageFactory::ConstructCEchoRQ (
            const ULConnection & inConnection )  [static]
```

10.58.2.2 ConstructCFindRQ()

```
static std::vector<PresentationDataValue> gdcm::network::CompositeMessageFactory::ConstructCFindRQ (
            const ULConnection & inConnection,
            const BaseRootQuery * inRootQuery )  [static]
```

10.58.2.3 ConstructCMoveRQ()

```
static std::vector<PresentationDataValue> gdcm::network::CompositeMessageFactory::ConstructC↩
MoveRQ (
             const ULConnection & inConnection,
             const BaseRootQuery * inRootQuery )  [static]
```

10.58.2.4 ConstructCStoreRQ()

```
static std::vector<PresentationDataValue> gdcm::network::CompositeMessageFactory::ConstructC↩
StoreRQ (
             const ULConnection & inConnection,
             const File & file,
             bool writeDataSet = true )  [static]
```

10.58.2.5 ConstructCStoreRSP()

```
static std::vector<PresentationDataValue> gdcm::network::CompositeMessageFactory::ConstructC↩
StoreRSP (
             const DataSet * inDataSet,
             const BasePDU * inPC )  [static]
```

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

- gdcmCompositeMessageFactory.h

10.59 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=NULL, const char *call=NULL)
- static bool CFind (const char *remote, uint16_t portno, const BaseRootQuery *query, std::vector< DataSet > &retDataSets, const char *aetitle=NULL, const char *call=NULL)
- static bool CMove (const char *remote, uint16_t portno, const BaseRootQuery *query, uint16_t portscp, const char *aetitle=NULL, const char *call=NULL, const char *outputdir=NULL)
- static BaseRootQuery * ConstructQuery (ERootType inRootType, EQueryLevel inQueryLevel, const DataSet &queryds, EQueryType queryType=eFind)
- static BaseRootQuery * ConstructQuery (ERootType inRootType, EQueryLevel inQueryLevel, const KeyValue↩PairArrayType &keys, EQueryType queryType=eFind)
- static bool CStore (const char *remote, uint16_t portno, const Directory::FilenamesType &filenames, const char *aetitle=NULL, const char *call=NULL)

10.59.1 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)

10.59.2 Member Typedef Documentation

10.59.2.1 KeyValuePairArrayType

```
typedef std::vector< KeyValuePairType > gdcm::CompositeNetworkFunctions::KeyValuePairArrayType
```

10.59.2.2 KeyValuePairType

```
typedef std::pair<Tag, std::string> gdcm::CompositeNetworkFunctions::KeyValuePairType
```

10.59.3 Member Function Documentation

10.59.3.1 CEcho()

```
static bool gdcm::CompositeNetworkFunctions::CEcho (
            const char * remote,
            uint16_t portno,
            const char * aetitle = NULL,
            const char * call = NULL )  [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.

10.59.3.2 CFind()

```
static bool gdcm::CompositeNetworkFunctions::CFind (
            const char * remote,
            uint16_t portno,
            const BaseRootQuery * query,
            std::vector< DataSet > & retDataSets,
            const char * aetitle = NULL,
            const char * call = NULL )  [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.

10.59.3.3 CMove()

```
static bool gdcm::CompositeNetworkFunctions::CMove (
            const char * remote,
            uint16_t portno,
            const BaseRootQuery * query,
            uint16_t portscp,
            const char * aetitle = NULL,
            const char * call = NULL,
            const char * outputdir = NULL )  [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.

10.59.3.4 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)

10.59.3.5 ConstructQuery() [2/2]

```
static BaseRootQuery* gdcm::CompositeNetworkFunctions::ConstructQuery (
            ERootType inRootType,
            EQueryLevel inQueryLevel,
            const KeyValuePairArrayType & keys,
            EQueryType queryType = eFind )  [static]
```

Deprecated

10.59.3.6 CStore()

```
static bool gdcm::CompositeNetworkFunctions::CStore (
            const char * remote,
            uint16_t portno,
            const Directory::FilenamesType & filenames,
            const char * aetitle = NULL,
            const char * call = NULL )  [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

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

- gdcmCompositeNetworkFunctions.h

10.60 gdcm::ConstCharWrapper Class Reference

Do not use me.

```
#include <gdcmConstCharWrapper.h>
```

Public Member Functions

- ConstCharWrapper (const char *i=0)
- operator const char * () const

10.60.1 Detailed Description

Do not use me.

10.60.2 Constructor & Destructor Documentation

10.60.2.1 ConstCharWrapper()

```
gdcm::ConstCharWrapper::ConstCharWrapper (
            const char * i = 0 )  [inline]
```

10.60.3 Member Function Documentation

10.60.3.1 operator const char *()

```
gdcm::ConstCharWrapper::operator const char * ( ) const  [inline]
```

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

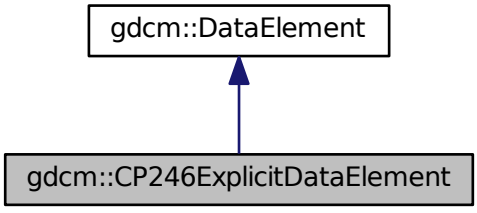
- gdcmConstCharWrapper.h

10.61 gdcm::CP246ExplicitDataElement Class Reference

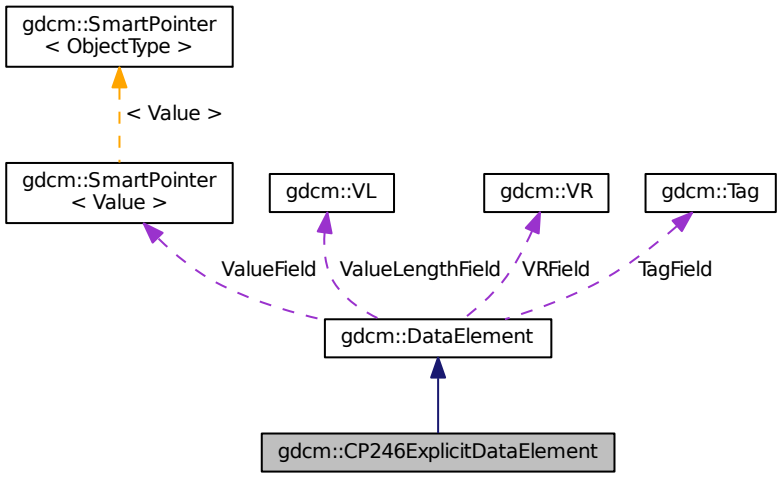
Class to read/write a DataElement as CP246Explicit Data Element.

```
#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

10.61.1 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

10.61.2 Member Function Documentation

10.61.2.1 GetLength()

```
VL gdcm::CP246ExplicitDataElement::GetLength ( ) const
```

10.61.2.2 Read()

```
template<typename TSwap >
std::istream& gdcm::CP246ExplicitDataElement::Read (
             std::istream & is )
```

10.61.2.3 ReadPreValue()

```
template<typename TSwap >
std::istream& gdcm::CP246ExplicitDataElement::ReadPreValue (
             std::istream & is )
```

10.61.2.4 ReadValue()

```
template<typename TSwap >
std::istream& gdcm::CP246ExplicitDataElement::ReadValue (
             std::istream & is,
             bool readvalues = true )
```

10.61.2.5 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:

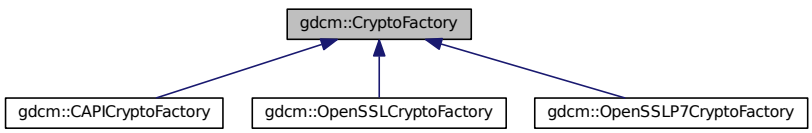
- gdcmCP246ExplicitDataElement.h

10.62 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 (CryptoLib id)
- CryptoFactory ()
- ~CryptoFactory ()

10.62.1 Detailed Description

Class to do handle the crypto factory.

GDCM needs to access in a platform independant 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.

10.62.2 Member Enumeration Documentation

10.62.2.1 CryptoLib

`enum gdcm::CryptoFactory::CryptoLib`

Enumerator

DEFAULT	
OPENSSL	
CAPI	
OPENSSLP7	

10.62.3 Constructor & Destructor Documentation

10.62.3.1 CryptoFactory() [1/2]

```
gdcm::CryptoFactory::CryptoFactory (
            CryptoLib id )  [inline], [protected]
```

References gdcmErrorMacro.

10.62.3.2 CryptoFactory() [2/2]

```
gdcm::CryptoFactory::CryptoFactory ( )  [inline], [protected]
```

10.62.3.3 ~CryptoFactory()

```
gdcm::CryptoFactory::~CryptoFactory ( )  [inline], [protected]
```

10.62.4 Member Function Documentation

10.62.4.1 CreateCMSProvider()

```
virtual CryptographicMessageSyntax* gdcm::CryptoFactory::CreateCMSProvider ( )  [pure virtual]
```

Implemented in gdcm::OpenSSLCryptoFactory, gdcm::OpenSSLP7CryptoFactory, and gdcm::CAPICryptoFactory.

10.62.4.2 GetFactoryInstance()

```
static CryptoFactory* gdcm::CryptoFactory::GetFactoryInstance (
            CryptoLib id = DEFAULT )  [static]
```

Examples:

BasicAnonymizer.cs, and ClinicalTrialIdentificationWorkflow.cs.

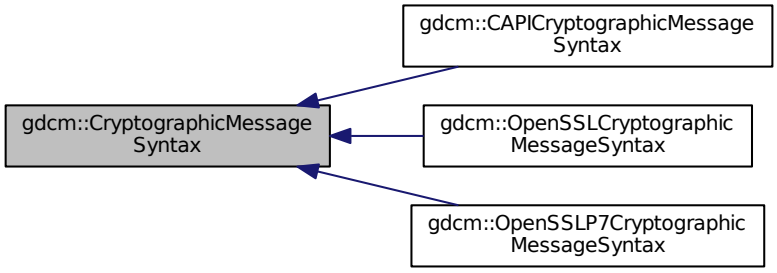
The documentation for this class was generated from the following file:

- gdcmCryptoFactory.h

10.63 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 ()
- virtual ~CryptographicMessageSyntax ()
- 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
- 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

10.63.1 Member Enumeration Documentation

10.63.1.1 CipherTypes

`enum gdcm::CryptographicMessageSyntax::CipherTypes`

Enumerator

DES3_CIPHER	
AES128_CIPHER	
AES192_CIPHER	
AES256_CIPHER	

10.63.2 Constructor & Destructor Documentation

10.63.2.1 CryptographicMessageSyntax()

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

10.63.2.2 ~CryptographicMessageSyntax()

```
virtual gdcm::CryptographicMessageSyntax::~CryptographicMessageSyntax ( )  [inline], [virtual]
```

10.63.3 Member Function Documentation

10.63.3.1 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::OpenSSLP7CryptographicMessageSyntax, gdcm::CAPICryptographicMessageSyntax, and gdcm::OpenSSLCryptographicMessageSyntax.

10.63.3.2 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::OpenSSLP7CryptographicMessageSyntax, gdcm::CAPICryptographicMessageSyntax, and gdcm::OpenSSLCryptographicMessageSyntax.

10.63.3.3 GetCipherType()

```
virtual CipherTypes gdcm::CryptographicMessageSyntax::GetCipherType ( ) const  [pure virtual]
```

Implemented in gdcm::OpenSSLP7CryptographicMessageSyntax, gdcm::CAPICryptographicMessageSyntax, and gdcm::OpenSSLCryptographicMessageSyntax.

10.63.3.4 ParseCertificateFile()

```
virtual bool gdcm::CryptographicMessageSyntax::ParseCertificateFile (
            const char * filename )  [pure virtual]
```

Implemented in gdcm::OpenSSLP7CryptographicMessageSyntax, gdcm::CAPICryptographicMessageSyntax, and gdcm::OpenSSLCryptographicMessageSyntax.

10.63.3.5 ParseKeyFile()

```
virtual bool gdcm::CryptographicMessageSyntax::ParseKeyFile (
            const char * filename )  [pure virtual]
```

Implemented in gdcm::OpenSSLP7CryptographicMessageSyntax, gdcm::CAPICryptographicMessageSyntax, and gdcm::OpenSSLCryptographicMessageSyntax.

10.63.3.6 SetCipherType()

```
virtual void gdcm::CryptographicMessageSyntax::SetCipherType (
            CipherTypes type )  [pure virtual]
```

Implemented in gdcm::OpenSSLP7CryptographicMessageSyntax, and gdcm::OpenSSLCryptographicMessageSyntax.

10.63.3.7 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:

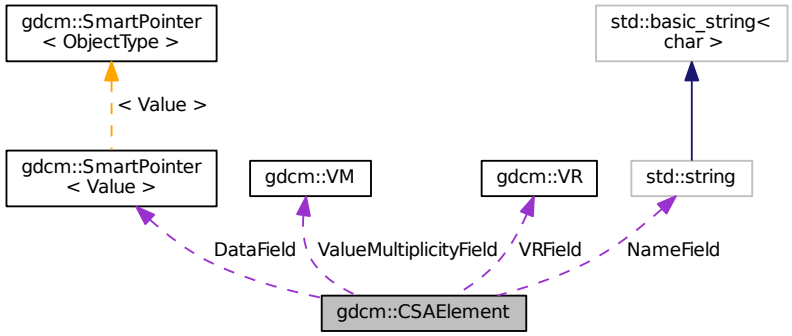
- gdcmCryptographicMessageSyntax.h

10.64 gdcm::CSAElement Class Reference

Class to represent a CSA Element.

`#include <gdcmCSAElement.h>`

Collaboration diagram for gdcm::CSAElement:



Public Member Functions

- CSAElement (unsigned int kf=0)
- CSAElement (const CSAElement &_val)
- 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 const & GetValue () const

  Set/Get Value (bytes array, SQ of items, SQ of fragments):
- Value & GetValue ()
- 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)

- 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)

10.64.1 Detailed Description

Class to represent a CSA Element.

See also

CSAHeader

Examples:

csa2img.cxx, and MrProtocol.cxx.

10.64.2 Member Typedef Documentation

10.64.2.1 DataPtr

```
typedef SmartPointer<Value> gdcm::CSAElement::DataPtr  [protected]
```

10.64.3 Constructor & Destructor Documentation

10.64.3.1 CSAElement() [1/2]

```
gdcm::CSAElement::CSAElement (
            unsigned int kf = 0 )  [inline]
```

References gdcm::operator<<().

10.64.3.2 CSAElement() [2/2]

```
gdcm::CSAElement::CSAElement (
            const CSAElement & _val )  [inline]
```

10.64.4 Member Function Documentation

10.64.4.1 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:

MrProtocol.cxx.

10.64.4.2 GetKey()

```
unsigned int gdcm::CSAElement::GetKey ( ) const  [inline]
```

Set/Get Key.

Referenced by operator<().

10.64.4.3 GetName()

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

Set/Get Name.

10.64.4.4 GetNoOfItems()

```
unsigned int gdcm::CSAElement::GetNoOfItems ( ) const  [inline]
```

Set/Get NoOfItems.

10.64.4.5 GetSyngoDT()

```
unsigned int gdcm::CSAElement::GetSyngoDT ( ) const  [inline]
```

Set/Get SyngoDT.

10.64.4.6 GetValue() [1/2]

```
Value const& gdcm::CSAElement::GetValue ( ) const  [inline]
```

Set/Get Value (bytes array, SQ of items, SQ of fragments):

Examples:

csa2img.cxx.

10.64.4.7 GetValue() [2/2]

```
Value& gdcm::CSAElement::GetValue ( )  [inline]
```

10.64.4.8 GetVM()

```
const VM& gdcm::CSAElement::GetVM ( ) const  [inline]
```

Set/Get VM.

10.64.4.9 GetVR()

```
VR const& gdcm::CSAElement::GetVR ( ) const  [inline]
```

Set/Get VR.

10.64.4.10 IsEmpty()

```
bool gdcm::CSAElement::IsEmpty ( ) const  [inline]
```

Check if CSA Element is empty.

Examples:

csa2img.cxx.

10.64.4.11 operator<()

```
bool gdcm::CSAElement::operator< (
            const CSAElement & de ) const  [inline]
```

References GetKey().

10.64.4.12 operator=()

```
CSAElement& gdcm::CSAElement::operator= (
            const CSAElement & de )  [inline]
```

References DataField, KeyField, NameField, NoOfItemsField, SyngoDTField, ValueMultiplicityField, and VRField.

10.64.4.13 operator==()

```
bool gdcm::CSAElement::operator== (
            const CSAElement & de ) const  [inline]
```

References KeyField, NameField, SyngoDTField, ValueMultiplicityField, and VRField.

10.64.4.14 SetByteValue()

```
void gdcm::CSAElement::SetByteValue (
            const char * array,
            VL length )  [inline]
```

Set.

10.64.4.15 SetKey()

```
void gdcm::CSAElement::SetKey (
            unsigned int key )  [inline]
```

10.64.4.16 SetName()

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

10.64.4.17 SetNoOfItems()

```
void gdcm::CSAElement::SetNoOfItems (
            unsigned int items )  [inline]
```

10.64.4.18 SetSyngoDT()

```
void gdcm::CSAElement::SetSyngoDT (
            unsigned int syngodt )  [inline]
```

10.64.4.19 SetValue()

```
void gdcm::CSAElement::SetValue (
            Value const & vl )  [inline]
```

10.64.4.20 SetVM()

```
void gdcm::CSAElement::SetVM (
            const VM & vm )  [inline]
```

10.64.4.21 SetVR()

```
void gdcm::CSAElement::SetVR (
            VR const & vr )  [inline]
```

10.64.5 Friends And Related Function Documentation

10.64.5.1 operator<<

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

10.64.6 Member Data Documentation

10.64.6.1 DataField

```
DataPtr gdcm::CSAElement::DataField  [protected]
```

Referenced by gdcm::operator<<(), and operator=().

10.64.6.2 KeyField

```
unsigned int gdcm::CSAElement::KeyField  [protected]
```

Referenced by gdcm::operator<<(), operator=(), and operator==().

10.64.6.3 NameField

```
std::string gdcm::CSAElement::NameField  [protected]
```

Referenced by gdcm::operator<<(), operator=(), and operator==().

10.64.6.4 NoOfItemsField

```
unsigned int gdcm::CSAElement::NoOfItemsField  [protected]
```

Referenced by gdcm::operator<<(), and operator=().

10.64.6.5 SyngoDTField

```
unsigned int gdcm::CSAElement::SyngoDTField  [protected]
```

Referenced by gdcm::operator<<(), operator=(), and operator==().

10.64.6.6 ValueMultiplicityField

```
VM gdcm::CSAElement::ValueMultiplicityField  [protected]
```

Referenced by gdcm::operator<<(), operator=(), and operator==().

10.64.6.7 VRField

```
VR gdcm::CSAElement::VRField  [protected]
```

Referenced by gdcm::operator<<(), operator=(), and operator==().

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

- gdcmCSAElement.h

10.65 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 ()
- 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 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)
- template<typename TSwap >
  std::istream & Read (std::istream &is)
- template<typename TSwap >
  const std::ostream & Write (std::ostream &os) const

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)

10.65.1 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/NO↩ MAGIC 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:

csa2img.cxx, and MrProtocol.cxx.

10.65.2 Member Enumeration Documentation

10.65.2.1 CSAHeaderType

`enum gdcm::CSAHeader::CSAHeaderType`

Divers format of CSAHeader as found 'in the wild'.

Enumerator

UNKNOWN	
SV10	
NOMAGIC	
DATASET_FORMAT	
INTERFILE	
ZEROED_OUT	

10.65.3 Constructor & Destructor Documentation

10.65.3.1 CSAHeader()

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

10.65.3.2 ~CSAHeader()

```
gdcm::CSAHeader::~CSAHeader ( )  [inline]
```

10.65.4 Member Function Documentation

10.65.4.1 FindCSAElementByName()

```
bool gdcm::CSAHeader::FindCSAElementByName (
            const char * name )
```

Return true if the CSA element matching 'name' is found or not

Warning

Case Sensitive

Examples:

csa2img.cxx, and MrProtocol.cxx.

10.65.4.2 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");

10.65.4.3 GetCSAEEnd()

```
const CSAElement& gdcm::CSAHeader::GetCSAEEnd ( ) const  [protected]
```

10.65.4.4 GetCSAElementByName()

```
const CSAElement& gdcm::CSAHeader::GetCSAElementByName (
            const char * name )
```

Return the CSAElement corresponding to name 'name'

Warning

Case Sensitive

Examples:

csa2img.cxx, and MrProtocol.cxx.

10.65.4.5 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,"SIEM↩ ENS CSA HEADER");

Examples:

csa2img.cxx, and PublicDict.cxx.

10.65.4.6 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,"SIEM↩ ENS CSA HEADER");

Examples:

MrProtocol.cxx.

10.65.4.7 GetDataSet()

```
const DataSet& gdcm::CSAHeader::GetDataSet ( ) const  [inline]
```

Return the DataSet output (use only if Format == DATASET_FORMAT)

10.65.4.8 GetFormat()

```
CSAHeaderType gdcm::CSAHeader::GetFormat ( ) const
```

return the format of the CSAHeader SV10 and NOMAGIC are equivalent.

10.65.4.9 GetInterfile()

```
const char* gdcm::CSAHeader::GetInterfile ( ) const  [inline]
```

Return the string output (use only if Format == Interfile)

10.65.4.10 LoadFromDataElement()

```
bool gdcm::CSAHeader::LoadFromDataElement (
            DataElement const & de )
```

Decode the CSAHeader from element 'de'.

Examples:

csa2img.cxx, and MrProtocol.cxx.

10.65.4.11 Print()

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

Print the CSAHeader (use only if Format == SV10 or NOMAGIC)

Examples:

csa2img.cxx.

Referenced by gdcm::operator<<().

10.65.4.12 Read()

```
template<typename TSwap >
std::istream& gdcm::CSAHeader::Read (
            std::istream & is )
```

10.65.4.13 Write()

```
template<typename TSwap >
const std::ostream& gdcm::CSAHeader::Write (
             std::ostream & os ) const
```

10.65.5 Friends And Related Function Documentation

10.65.5.1 operator<<

```
std::ostream& operator<< (
             std::ostream & _os,
             const CSAHeader & d )  [friend]
```

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

- gdcmCSAHeader.h

10.66 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 ()
- void AddCSAHeaderDictEntry (const CSAHeaderDictEntry &de)
- ConstIterator Begin () const
- ConstIterator End () const
- const CSAHeaderDictEntry & GetCSAHeaderDictEntry (const char *name) const
- bool IsEmpty () const

Protected Member Functions

- void LoadDefault ()

Friends

- class Dicts
- std::ostream & operator<< (std::ostream &_os, const CSAHeaderDict &_val)

10.66.1 Detailed Description

Class to represent a map of CSAHeaderDictEntry.

Examples:

MrProtocol.cxx.

10.66.2 Member Typedef Documentation

10.66.2.1 ConstIterator

```
typedef MapCSAHeaderDictEntry::const_iterator gdcm::CSAHeaderDict::ConstIterator
```

10.66.2.2 Iterator

```
typedef MapCSAHeaderDictEntry::iterator gdcm::CSAHeaderDict::Iterator
```

10.66.2.3 MapCSAHeaderDictEntry

```
typedef std::set<CSAHeaderDictEntry> gdcm::CSAHeaderDict::MapCSAHeaderDictEntry
```

10.66.3 Constructor & Destructor Documentation

10.66.3.1 CSAHeaderDict()

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

References gdcm::operator<<().

10.66.4 Member Function Documentation

10.66.4.1 AddCSAHeaderDictEntry()

```
void gdcm::CSAHeaderDict::AddCSAHeaderDictEntry (
            const CSAHeaderDictEntry & de )  [inline]
```

10.66.4.2 Begin()

```
ConstIterator gdcm::CSAHeaderDict::Begin ( ) const  [inline]
```

10.66.4.3 End()

```
ConstIterator gdcm::CSAHeaderDict::End ( ) const  [inline]
```

10.66.4.4 GetCSAHeaderDictEntry()

```
const CSAHeaderDictEntry& gdcm::CSAHeaderDict::GetCSAHeaderDictEntry (
            const char * name ) const  [inline]
```

Examples:

MrProtocol.cxx.

10.66.4.5 IsEmpty()

```
bool gdcm::CSAHeaderDict::IsEmpty ( ) const  [inline]
```

10.66.4.6 LoadDefault()

```
void gdcm::CSAHeaderDict::LoadDefault ( )  [protected]
```

10.66.5 Friends And Related Function Documentation

10.66.5.1 Dicts

```
friend class Dicts  [friend]
```

10.66.5.2 operator<<

```
std::ostream& operator<< (
            std::ostream & _os,
            const CSAHeaderDict & _val )  [friend]
```

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

- gdcmCSAHeaderDict.h

10.67 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)

10.67.1 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.

10.67.2 Constructor & Destructor Documentation

10.67.2.1 CSAHeaderDictEntry()

```
gdcm::CSAHeaderDictEntry::CSAHeaderDictEntry (
            const char * name = "",
            VR const & vr = VR::INVALID,
            VM const & vm = VM::VM0,
            const char * desc = "" )  [inline]
```

References gdcm::operator<<().

10.67.3 Member Function Documentation

10.67.3.1 GetDescription()

```
const char* gdcm::CSAHeaderDictEntry::GetDescription ( ) const  [inline]
```

Set/Get Description.

10.67.3.2 GetName()

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

Set/Get Name.

Referenced by operator<().

10.67.3.3 GetVM()

```
const VM& gdcm::CSAHeaderDictEntry::GetVM ( ) const  [inline]
```

Set/Get VM.

10.67.3.4 GetVR()

```
const VR& gdcm::CSAHeaderDictEntry::GetVR ( ) const  [inline]
```

Set/Get VR.

10.67.3.5 operator<()

```
bool gdcm::CSAHeaderDictEntry::operator< (
            const CSAHeaderDictEntry & entry ) const  [inline]
```

References GetName().

10.67.3.6 SetDescription()

```
void gdcm::CSAHeaderDictEntry::SetDescription (
            const char * desc )  [inline]
```

10.67.3.7 SetName()

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

10.67.3.8 SetVM()

```
void gdcm::CSAHeaderDictEntry::SetVM (
            VM const & vm )  [inline]
```

10.67.3.9 SetVR()

```
void gdcm::CSAHeaderDictEntry::SetVR (
            const VR & vr )  [inline]
```

10.67.4 Friends And Related Function Documentation

10.67.4.1 operator<<

```
std::ostream& operator<< (
            std::ostream & _os,
            const CSAHeaderDictEntry & _val )  [friend]
```

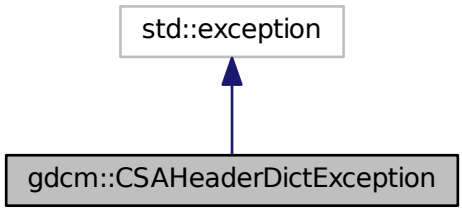
The documentation for this class was generated from the following file:

- gdcmCSAHeaderDictEntry.h

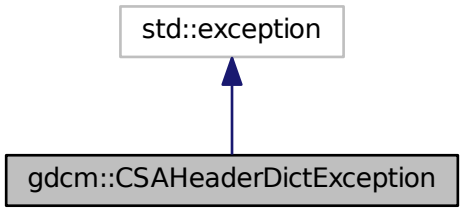
10.68 gdcm::CSAHeaderDictException Class Reference

`#include <gdcmCSAHeaderDict.h>`

Inheritance diagram for gdcm::CSAHeaderDictException:



Collaboration diagram for gdcm::CSAHeaderDictException:



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

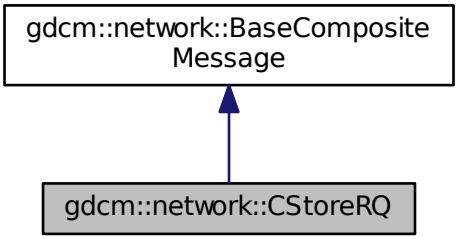
- gdcmCSAHeaderDict.h

10.69 gdcm::network::CStoreRQ Class Reference

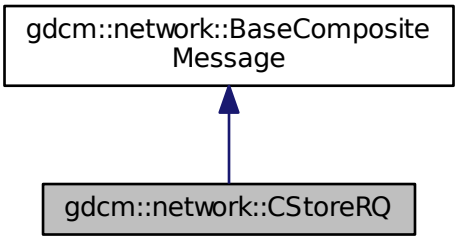
CStoreRQ.

`#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)

10.69.1 Detailed Description

CStoreRQ.

this file defines the messages for the cecho action

10.69.2 Member Function Documentation

10.69.2.1 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:

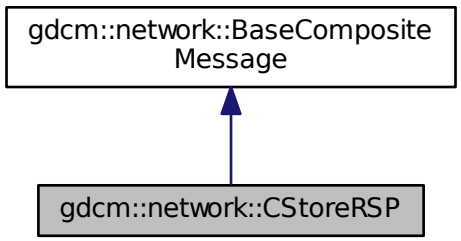
- gdcmCStoreMessages.h

10.70 gdcm::network::CStoreRSP Class Reference

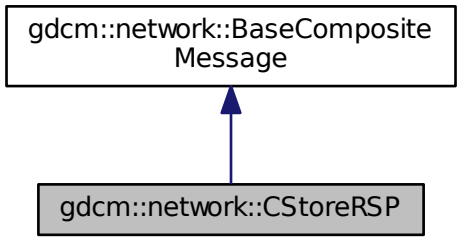
CStoreRSP this file defines the messages for the cecho action.

```
#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)

10.70.1 Detailed Description

CStoreRSP 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::CStoreRSP::ConstructPDV (
            const DataSet * inDataSet,
            const BasePDU * inPC )
```

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

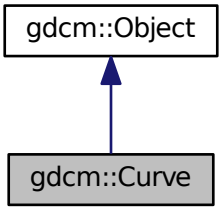
- gdcmCStoreMessages.h

10.71 gdcm::Curve Class Reference

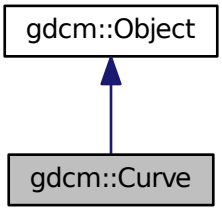
Curve class to handle element 50xx,3000 Curve Data.

```
#include <gdcmCurve.h>
```

Inheritance diagram for gdcm::Curve:



Collaboration diagram for gdcm::Curve:



Public Member Functions

- Curve ()
- Curve (Curve const &ov)
- ∼Curve ()
- 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
- 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

10.71.1 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

10.71.2 Constructor & Destructor Documentation

10.71.2.1 Curve() [1/2]

```
gdcm::Curve::Curve ( )
```

10.71.2.2 ~Curve()

```
gdcm::Curve::~Curve ( )
```

10.71.2.3 Curve() [2/2]

```
gdcm::Curve::Curve (
            Curve const & ov )
```

10.71.3 Member Function Documentation

10.71.3.1 Decode()

```
void gdcm::Curve::Decode (
            std::istream & is,
            std::ostream & os )
```

10.71.3.2 GetAsPoints()

```
void gdcm::Curve::GetAsPoints (
            float * array ) const
```

10.71.3.3 GetCurveDataDescriptor()

```
std::vector<unsigned short> const& gdcm::Curve::GetCurveDataDescriptor ( ) const
```

10.71.3.4 GetDataValueRepresentation()

```
unsigned short gdcm::Curve::GetDataValueRepresentation ( ) const
```

10.71.3.5 GetDimensions()

```
unsigned short gdcm::Curve::GetDimensions ( ) const
```

10.71.3.6 GetGroup()

```
unsigned short gdcm::Curve::GetGroup ( ) const
```

10.71.3.7 GetNumberOfCurves()

```
static unsigned int gdcm::Curve::GetNumberOfCurves (
            DataSet const & ds )  [static]
```

10.71.3.8 GetNumberOfPoints()

```
unsigned short gdcm::Curve::GetNumberOfPoints ( ) const
```

10.71.3.9 GetTypeOfData()

```
const char* gdcm::Curve::GetTypeOfData ( ) const
```

10.71.3.10 GetTypeOfDataDescription()

```
const char* gdcm::Curve::GetTypeOfDataDescription ( ) const
```

10.71.3.11 IsEmpty()

```
bool gdcm::Curve::IsEmpty ( ) const
```

10.71.3.12 Print()

```
void gdcm::Curve::Print (
            std::ostream &  ) const  [virtual]
```

Reimplemented from gdcm::Object.

10.71.3.13 SetCoordinateStartValue()

```
void gdcm::Curve::SetCoordinateStartValue (
            unsigned short v )
```

10.71.3.14 SetCoordinateStepValue()

```
void gdcm::Curve::SetCoordinateStepValue (
            unsigned short v )
```

10.71.3.15 SetCurve()

```
void gdcm::Curve::SetCurve (
            const char * array,
            unsigned int length )
```

10.71.3.16 SetCurveDataDescriptor()

```
void gdcm::Curve::SetCurveDataDescriptor (
            const uint16_t * values,
            size_t num )
```

10.71.3.17 SetCurveDescription()

```
void gdcm::Curve::SetCurveDescription (
            const char * curvedescription )
```

10.71.3.18 SetDataValueRepresentation()

```
void gdcm::Curve::SetDataValueRepresentation (
            unsigned short datavaluerepresentation )
```

10.71.3.19 SetDimensions()

```
void gdcm::Curve::SetDimensions (
            unsigned short dimensions )
```

10.71.3.20 SetGroup()

```
void gdcm::Curve::SetGroup (
            unsigned short group )
```

10.71.3.21 SetNumberOfPoints()

```
void gdcm::Curve::SetNumberOfPoints (
            unsigned short numberofpoints )
```

10.71.3.22 SetTypeOfData()

```
void gdcm::Curve::SetTypeOfData (
            const char * typeofdata )
```

10.71.3.23 Update()

```
void gdcm::Curve::Update (
            const DataElement & de )
```

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

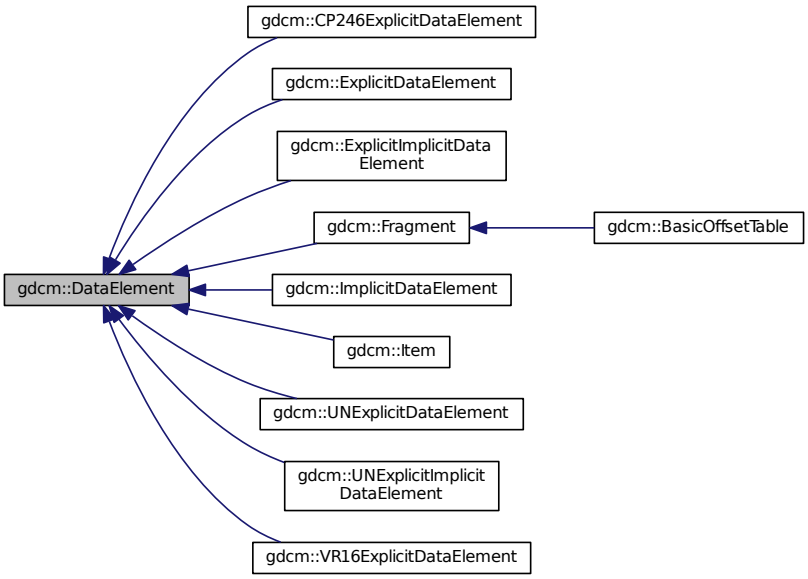
- gdcmCurve.h

10.72 gdcm::DataElement Class Reference

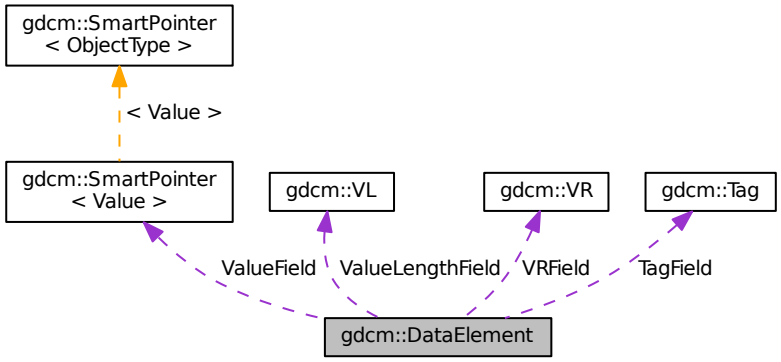
Class to represent a Data Element either Implicit or Explicit.

```
#include <gdcmDataElement.h>
```

Inheritance diagram for gdcm::DataElement:



Collaboration diagram for gdcm::DataElement:



Public Member Functions

- DataElement (const Tag &t=Tag(0), const VL &vl=0, const VR &vr=VR::INVALID)
- DataElement (const DataElement &_val)
- 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
- const SequenceOfFragments * GetSequenceOfFragments () const
- SequenceOfFragments * GetSequenceOfFragments ()
- const Tag & GetTag () const

  Get Tag.
- Tag & GetTag ()
- Value const & GetValue () const

  Set/Get Value (bytes array, SQ of items, SQ of fragments):
- Value & GetValue ()
- SmartPointer< SequenceOfItems > GetValueAsSQ () const
- const VL & GetVL () const

  Get VL.
- VL & GetVL ()
- 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 &de)
- 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)

10.72.1 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:

ChangePrivateTags.cxx, ChangeSequenceUltrasound.cxx, CreateARGBImage.cxx, CreateCMYKImage.cxx, csa2img.cxx, DiffFile.cxx, DumpADAC.cxx, DumpExamCard.cxx, DumpGEMSMovieGroup.cxx, DumpImage↩HeaderInfo.cxx, DumpPhilipsECHO.cxx, DumpToshibaDTI.cxx, DuplicatePCDE.cxx, ELSCINT1WaveToText.cxx, ExtractEncryptedContent.cxx, ExtractIconFromFile.cxx, Extracting_All_Resolution.cxx, Fake_Image_Using↩_Stream_Image_Writer.cxx, FixBrokenJ2K.cxx, FixJAIBugJPEGLS.cxx, gdcmrtionplan.cxx, gdcmrtplan.cxx, GenAllVR.cxx, GenFakeIdentifyFile.cxx, GenFakeImage.cxx, GenLongSeqs.cxx, GenSeqs.cxx, GetJPEG↩SamplePrecision.cxx, GetSequenceUltrasound.cxx, GetSubSequenceData.cxx, iU22tomultisc.cxx, LargeVRD↩SExplicit.cxx, pmsct_rgb1.cxx, ReadAndDumpDICOMDIR.cxx, ReadExplicitLengthSQIVR.cxx, ReadGEMSSD↩O.cxx, rle2img.cxx, and StreamImageReaderTest.cxx.

10.72.2 Member Typedef Documentation

10.72.2.1 ValuePtr

```
typedef SmartPointer<Value> gdcm::DataElement::ValuePtr  [protected]
```

10.72.3 Constructor & Destructor Documentation

10.72.3.1 DataElement() [1/2]

```
gdcm::DataElement::DataElement (
            const Tag & t = Tag(0),
            const VL & vl = 0,
            const VR & vr = VR::INVALID )  [inline]
```

References gdcm::operator<<().

10.72.3.2 DataElement() [2/2]

```
gdcm::DataElement::DataElement (
            const DataElement & _val )  [inline]
```

10.72.4 Member Function Documentation

10.72.4.1 Clear()

```
void gdcm::DataElement::Clear ( )  [inline]
```

Clear Data Element (make Value empty and invalidate Tag & VR)

10.72.4.2 Empty()

```
void gdcm::DataElement::Empty ( )  [inline]
```

Make Data Element empty (no Value)

10.72.4.3 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, DumpImageHeaderInfo.cxx, DumpPhilipsECHO.cxx, DumpToshibaDTI.cxx, DuplicatePCDE.cxx, ELSCINT1WaveToText.cxx, ExtractEncryptedContent.cxx, ExtractIconFromFile.cxx, FixBrokenJ2K.cxx, FixJAIBugJPEGLS.cxx, GetSubSequenceData.cxx, PatchFile.cxx, pmsct_rgb1.cxx, ReadExplicitLengthSQIVR.cxx, ReadGEMSSDO.cxx, and rle2img.cxx.

Referenced by gdcm::operator<<(), gdcm::Element< VR::OB, VM::VM1_n >::SetFromDataElement(), gdcm::Attribute< Group, Element, TVR, TVM >::SetFromDataElement(), gdcm::Attribute< Group, Element, TVR, VM::VM1 >::SetFromDataElement(), gdcm::Element< TVR, VM::VM1_n >::SetFromDataElement(), and gdcm::Attribute< Group, Element, TVR, VM::VM1_n >::SetFromDataElement().

10.72.4.4 GetLength()

```
template<typename TDE >
VL gdcm::DataElement::GetLength ( ) const  [inline]
```

10.72.4.5 GetSequenceOfFragments() [1/2]

```
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:

FixBrokenJ2K.cxx, FixJAIBugJPEGLS.cxx, and GetJPEGSamplePrecision.cxx.

10.72.4.6 GetSequenceOfFragments() [2/2]

```
SequenceOfFragments* gdcm::DataElement::GetSequenceOfFragments ( )
```

10.72.4.7 GetTag() [1/2]

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

Get Tag.

Examples:

DumpGEMSMovieGroup.cxx, DuplicatePCDE.cxx, pmsct_rgb1.cxx, and rle2img.cxx.

Referenced by gdcm::CommandDataSet::Insert(), gdcm::FileMetaInformation::Insert(), gdcm::DataSet::Insert(), operator<(), gdcm::SequenceOfItems::Read(), gdcm::SequenceOfFragments::ReadValue(), gdcm::CommandData↩Set::Replace(), gdcm::FileMetaInformation::Replace(), gdcm::Attribute< Group, Element, TVR, TVM >::SetFrom↩DataElement(), gdcm::Attribute< Group, Element, TVR, VM::VM1 >::SetFromDataElement(), and gdcm::Attribute< Group, Element, TVR, VM::VM1_n >::SetFromDataElement().

10.72.4.8 GetTag() [2/2]

```
Tag& gdcm::DataElement::GetTag ( )  [inline]
```

10.72.4.9 GetValue() [1/2]

```
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< VR::OB, VM::VM1_n >::SetFromDataElement(), and gdcm::Element< TVR, VM::VM1_n >::SetFromDataElement().

10.72.4.10 GetValue() [2/2]

```
Value& gdcm::DataElement::GetValue ( )  [inline]
```

10.72.4.11 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 simpliest 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, ExtractEncryptedContent.cxx, gdcmrtionplan.cxx, gdcmrtplan.cxx, GetSequenceUltrasound.cxx, GetSubSequenceData.cxx, LargeVRDSExplicit.cxx, and ReadAndDumpDICOMDIR.cxx.

10.72.4.12 GetVL() [1/2]

```
const VL& gdcm::DataElement::GetVL ( ) const  [inline]
```

Get VL.

Referenced by gdcm::DataSet::InsertDataElement(), gdcm::SequenceOfItems::Read(), and gdcm::SequenceOfFragments::ReadValue().

10.72.4.13 GetVL() [2/2]

```
VL& gdcm::DataElement::GetVL ( )  [inline]
```

10.72.4.14 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::Element< VR::OB, VM::VM1_n >::GetAsDataElement(), gdcm::Attribute< Group, Element, TVR, TVM >::GetAsDataElement(), gdcm::Attribute< Group, Element, TVR, VM::VM1 >::GetAsDataElement(), gdcm::Attribute< Group, Element, TVR, VM::VM1_n >::GetAsDataElement(), gdcm::Element< TVR, VM::VM1_n >::GetAsDataElement(), gdcm::Element< VR::OB, VM::VM1_n >::SetFromDataElement(), gdcm::Attribute< Group, Element, TVR, TVM >::SetFromDataElement(), gdcm::Attribute< Group, Element, TVR, VM::VM1 >::SetFromDataElement(), gdcm::Element< TVR, VM::VM1_n >::SetFromDataElement(), and gdcm::Attribute< Group, Element, TVR, VM::VM1_n >::SetFromDataElement().

10.72.4.15 IsEmpty()

```
bool gdcm::DataElement::IsEmpty ( ) const  [inline]
```

Check if Data Element is empty.

Examples:

DumpADAC.cxx, DumpExamCard.cxx, DumpGEMSMovieGroup.cxx, DumpImageHeaderInfo.cxx, DumpToshibaDTI.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(), gdcm::Attribute< Group, Element, TVR, VM::VM1_n >::SetFromDataElement(), gdcm::Attribute< Group, Element, TVR, TVM >::SetFromDataSet(), gdcm::Attribute< Group, Element, TVR, VM::VM1 >::SetFromDataSet(), and gdcm::Attribute< Group, Element, TVR, VM::VM1_n >::SetFromDataSet().

10.72.4.16 IsUndefinedLength()

```
bool gdcm::DataElement::IsUndefinedLength ( ) const  [inline]
```

return if Value Length if of undefined length

10.72.4.17 operator<()

```
bool gdcm::DataElement::operator< (
            const DataElement & de ) const  [inline]
```

References GetTag().

10.72.4.18 operator=()

```
DataElement& gdcm::DataElement::operator= (
            const DataElement & de )  [inline]
```

References TagField, ValueField, ValueLengthField, and VRField.

10.72.4.19 operator==()

```
bool gdcm::DataElement::operator== (
            const DataElement & de ) const  [inline]
```

References TagField, ValueField, ValueLengthField, and VRField.

10.72.4.20 Read()

```
template<typename TDE , typename TSwap >
std::istream& gdcm::DataElement::Read (
             std::istream & is )  [inline]
```

10.72.4.21 ReadOrSkip()

```
template<typename TDE , typename TSwap >
std::istream& gdcm::DataElement::ReadOrSkip (
             std::istream & is,
             std::set< Tag > const & skiptags )  [inline]
```

10.72.4.22 ReadPreValue()

```
template<typename TDE , typename TSwap >
std::istream& gdcm::DataElement::ReadPreValue (
             std::istream & is,
             std::set< Tag > const & skiptags )  [inline]
```

10.72.4.23 ReadValue()

```
template<typename TDE , typename TSwap >
std::istream& gdcm::DataElement::ReadValue (
             std::istream & is,
             std::set< Tag > const & skiptags )  [inline]
```

10.72.4.24 ReadValueWithLength()

```
template<typename TDE , typename TSwap >
std::istream& gdcm::DataElement::ReadValueWithLength (
             std::istream & is,
             VL & length,
             std::set< Tag > const & skiptags )  [inline]
```

10.72.4.25 ReadWithLength()

```
template<typename TDE , typename TSwap >
std::istream& gdcm::DataElement::ReadWithLength (
             std::istream & is,
             VL & length )  [inline]
```

10.72.4.26 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:

ChangePrivateTags.cxx, ChangeSequenceUltrasound.cxx, CreateARGBImage.cxx, CreateCMYKImage.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, GetSubSequenceData.cxx, iU22tomultisc.cxx, LargeVRDSExplicit.cxx, NewSequence.cs, rle2img.cxx, and StreamImageReaderTest.cxx.

Referenced by gdcm::Element< VR::OB, VM::VM1_n >::GetAsDataElement(), gdcm::Attribute< Group, Element, TVR, TVM >::GetAsDataElement(), gdcm::Attribute< Group, Element, TVR, VM::VM1 >::GetAsDataElement(), gdcm::Attribute< Group, Element, TVR, VM::VM1_n >::GetAsDataElement(), and gdcm::Element< TVR, VM::VM1_n >::GetAsDataElement().

10.72.4.27 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, GenAllVR.cxx, GenFakeIdentifyFile.cxx, and GetSubSequenceData.cxx.

10.72.4.28 SetValue()

```
void gdcm::DataElement::SetValue (
            Value const & vl )  [inline]
```

Warning

you need to set the ValueLengthField explicitly

Examples:

DuplicatePCDE.cxx, Extracting_All_Resolution.cxx, Fake_Image_Using_Stream_Image_Writer.cxx, FixBrokenJ2K.cxx, GenAllVR.cxx, GenFakeIdentifyFile.cxx, GenLongSeqs.cxx, and GenSeqs.cxx.

References gdcm::Value::GetLength().

10.72.4.29 SetValueFieldLength()

```
void gdcm::DataElement::SetValueFieldLength (
            VL vl,
            bool readvalues )  [protected]
```

10.72.4.30 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

10.72.4.31 SetVLToUndefined()

```
void gdcm::DataElement::SetVLToUndefined ( )
```

Examples:

Extracting_All_Resolution.cxx, Fake_Image_Using_Stream_Image_Writer.cxx, GenAllVR.cxx, GenFakeIdentify↩File.cxx, GenLongSeqs.cxx, GenSeqs.cxx, and NewSequence.cs.

10.72.4.32 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:

Extracting_All_Resolution.cxx, Fake_Image_Using_Stream_Image_Writer.cxx, FixBrokenJ2K.cxx, FixJAIBugJ↩PEGLS.cxx, GenAllVR.cxx, GenFakeIdentifyFile.cxx, GenLongSeqs.cxx, GenSeqs.cxx, GetSubSequenceData.↩cxx, iU22tomultisc.cxx, LargeVRDSExplicit.cxx, NewSequence.cs, rle2img.cxx, and StreamImageReaderTest.cxx.

References gdcm::VR::IsVRFile().

Referenced by gdcm::Element< VR::OB, VM::VM1_n >::GetAsDataElement(), gdcm::Attribute< Group, Element, T↩VR, TVM >::GetAsDataElement(), gdcm::Attribute< Group, Element, TVR, VM::VM1 >::GetAsDataElement(), gdcm↩::Attribute< Group, Element, TVR, VM::VM1_n >::GetAsDataElement(), and gdcm::Element< TVR, VM::VM1_n >::↩GetAsDataElement().

10.72.4.33 Write()

```
template<typename TDE , typename TSwap >
const std::ostream& gdcm::DataElement::Write (
            std::ostream & os ) const  [inline]
```

10.72.5 Friends And Related Function Documentation

10.72.5.1 operator<<

```
std::ostream& operator<< (
            std::ostream & _os,
            const DataElement & _val )  [friend]
```

10.72.6 Member Data Documentation

10.72.6.1 TagField

```
Tag gdcm::DataElement::TagField  [protected]
```

Referenced by gdcm::operator<<(), operator=(), and operator==().

10.72.6.2 ValueField

```
ValuePtr gdcm::DataElement::ValueField  [protected]
```

Referenced by gdcm::operator<<(), operator=(), and operator==().

10.72.6.3 ValueLengthField

```
VL gdcm::DataElement::ValueLengthField  [protected]
```

Referenced by gdcm::operator<<(), operator=(), and operator==().

10.72.6.4 VRField

```
VR gdcm::DataElement::VRField  [protected]
```

Referenced by gdcm::operator<<(), operator=(), and operator==().

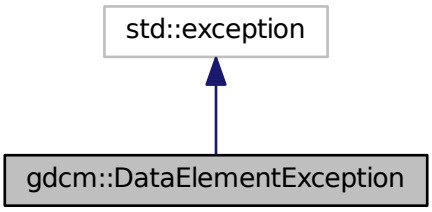
The documentation for this class was generated from the following file:

- gdcmDataElement.h

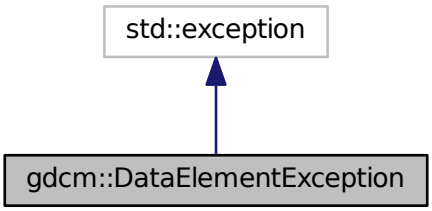
10.73 gdcm::DataElementException Class Reference

`#include <gdcmDataSet.h>`

Inheritance diagram for gdcm::DataElementException:



Collaboration diagram for gdcm::DataElementException:



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

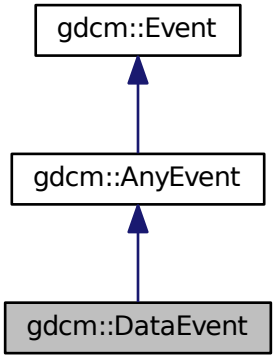
- gdcmDataSet.h

10.74 gdcm::DataEvent Class Reference

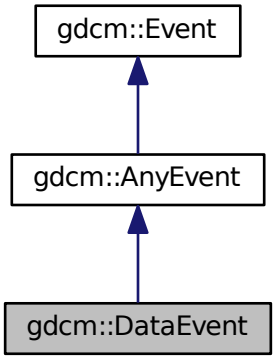
DataEvent.

`#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=0, size_t len=0)
- DataEvent (const Self &s)
- virtual ~DataEvent ()
- virtual bool CheckEvent (const ::gdcm::Event *e) const
- const char * GetData () const
- size_t GetDataLength () const
- virtual const char * GetEventName () const
- virtual ::gdcm::Event * MakeObject () const
- void SetData (const char *bytes, size_t len)

10.74.1 Detailed Description

DataEvent.

10.74.2 Member Typedef Documentation

10.74.2.1 Self

```
typedef DataEvent gdcm::DataEvent::Self
```

10.74.2.2 Superclass

```
typedef AnyEvent gdcm::DataEvent::Superclass
```

10.74.3 Constructor & Destructor Documentation

10.74.3.1 DataEvent() [1/2]

```
gdcm::DataEvent::DataEvent (
            const char * bytes = 0,
            size_t len = 0 )  [inline]
```

10.74.3.2 ~DataEvent()

```
virtual gdcm::DataEvent::~DataEvent ( )  [inline], [virtual]
```

10.74.3.3 DataEvent() [2/2]

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

10.74.4 Member Function Documentation

10.74.4.1 CheckEvent()

```
virtual bool gdcm::DataEvent::CheckEvent (
            const ::gdcm::Event * e ) const  [inline], [virtual]
```

10.74.4.2 GetData()

```
const char* gdcm::DataEvent::GetData ( ) const  [inline]
```

10.74.4.3 GetDataLength()

```
size_t gdcm::DataEvent::GetDataLength ( ) const  [inline]
```

10.74.4.4 GetEventName()

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

Return the StringName associated with the event.

Implements gdcm::Event.

10.74.4.5 MakeObject()

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

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

Implements gdcm::Event.

10.74.4.6 SetData()

```
void gdcm::DataEvent::SetData (
            const char * bytes,
            size_t len )  [inline]
```

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

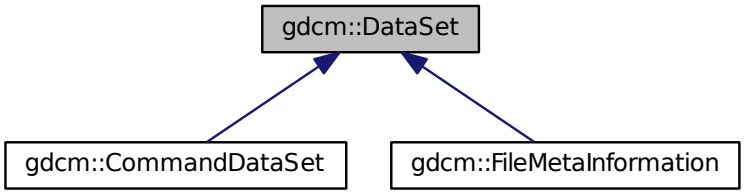
- gdcmDataEvent.h

10.75 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

- ConstIterator Begin () const
- Iterator Begin ()
- void Clear ()
- template<typename TDE >
  unsigned int ComputeGroupLength (Tag const &tag) const
- ConstIterator End () const
- Iterator End ()
- 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 Tag &t) const
- const DataElement & GetDataElement (const PrivateTag &t) const

  Return the dataelement.
- const DataElementSet & GetDES () const
- DataElementSet & GetDES ()
- template<typename TDE >
  VL GetLength () const

- MediaStorage GetMediaStorage () const
- std::string GetPrivateCreator (const Tag &t) const

  Return the private creator of the private tag 't':
- 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 &val)
- 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 &val)

10.75.1 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, CreateFakeRTDOSE.cxx, CreateJPIPDataSet.cxx, csa2img.cxx, DiffFile.cxx, DumpADAC.cxx, DumpExamCard.cxx, DumpGEMSMovieGroup.cxx, DumpImageHeaderInfo.cxx, DumpPhilipsECHO.cxx, DumpToshibaDTI.cxx, DuplicatePCDE.cxx, ELSCINT1WaveToText.cxx, ExtractEncryptedContent.cxx, Extracting_All_Resolution.cxx, Fake_Image_Using_Stream_Image_Writer.cxx, FixOrientation.cxx, gdcmrtionplan.cxx, gdcmrtplan.cxx, GenAllVR.cxx, GenFakeIdentifyFile.cxx, GenLongSeqs.cxx, GenSeqs.cxx, GetJPEGSamplePrecision.cxx, GetSequenceUltrasound.cxx, GetSubSequenceData.cxx, HelloWorld.cxx, iU22tomultisc.cxx, LargeVRDSExplicit.cxx, MergeTwoFiles.cxx, MrProtocol.cxx, PatchFile.cxx, pmsct_rgb1.cxx, ReadAndDumpDICOMDIR.cxx, ReadAndPrintAttributes.cxx, ReadExplicitLengthSQIVR.cxx, ReadGEMSSDO.cxx, rle2img.cxx, SortImage.cxx, StreamImageReaderTest.cxx, and VolumeSorter.cxx.

10.75.2 Member Typedef Documentation

10.75.2.1 ConstIterator

```
typedef DataElementSet::const_iterator gdcm::DataSet::ConstIterator
```

10.75.2.2 DataElementSet

```
typedef std::set<DataElement> gdcm::DataSet::DataElementSet
```

10.75.2.3 Iterator

```
typedef DataElementSet::iterator gdcm::DataSet::Iterator
```

10.75.2.4 SizeType

```
typedef DataElementSet::size_type gdcm::DataSet::SizeType
```

10.75.3 Member Function Documentation

10.75.3.1 Begin() [1/2]

```
ConstIterator gdcm::DataSet::Begin ( ) const  [inline]
```

Examples:

DiffFile.cxx, DumpGEMSMovieGroup.cxx, and DuplicatePCDE.cxx.

Referenced by gdcm::operator<<().

10.75.3.2 Begin() [2/2]

```
Iterator gdcm::DataSet::Begin ( )  [inline]
```

10.75.3.3 Clear()

```
void gdcm::DataSet::Clear ( )  [inline]
```

Referenced by gdcm::Item::Read().

10.75.3.4 ComputeDataElement()

```
Tag gdcm::DataSet::ComputeDataElement (
            const PrivateTag & t ) const  [protected]
```

10.75.3.5 ComputeGroupLength()

```
template<typename TDE >
unsigned int gdcm::DataSet::ComputeGroupLength (
            Tag const & tag ) const  [inline]
```

References gdcm::Tag::GetElement(), and gdcm::Tag::GetGroup().

10.75.3.6 End() [1/2]

```
ConstIterator gdcm::DataSet::End ( ) const  [inline]
```

Examples:

DiffFile.cxx, DumpGEMSMovieGroup.cxx, and DuplicatePCDE.cxx.

10.75.3.7 End() [2/2]

```
Iterator gdcm::DataSet::End ( )  [inline]
```

10.75.3.8 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, csa2img.cxx, DumpADAC.cxx, DumpExamCard.cxx, DumpGEMSMovieGroup.cxx, DumpImageHeaderInfo.cxx, DumpPhilipsECHO.cxx, DumpToshibaDTI.cxx, ELSCINT1WaveToText.cxx, ExtractEncryptedContent.cxx, gdcmrtionplan.cxx, gdcmrtplan.cxx, GetSequenceUltrasound.cxx, GetSubSequenceData.cxx, LargeVRDSExplicit.cxx, MrProtocol.cxx, pmsct_rgb1.cxx, ReadAndDumpDICOMDIR.cxx, ReadAndPrintAttributes.cxx, ReadGEMSSDO.cxx, and rle2img.cxx.

Referenced by gdcm::Attribute< Group, Element, TVR, TVM >::SetFromDataSet(), gdcm::Attribute< Group, Element, TVR, VM::VM1 >::SetFromDataSet(), and gdcm::Attribute< Group, Element, TVR, VM::VM1_n >::SetFromDataSet().

10.75.3.9 FindDataElement() [2/2]

```
bool gdcm::DataSet::FindDataElement (
            const Tag & t ) const  [inline]
```

10.75.3.10 FindNextDataElement()

```
const DataElement& gdcm::DataSet::FindNextDataElement (
            const Tag & t ) const  [inline]
```

Examples:

DuplicatePCDE.cxx.

10.75.3.11 GetDataElement() [1/2]

```
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, csa2img.cxx, DumpADAC.cxx, DumpExamCard.cxx, DumpGEMSMovieGroup.cxx, DumpImageHeaderInfo.cxx, DumpPhilipsECHO.cxx, DumpToshibaDTI.cxx, ELSCINT1WaveToText.cxx, ExtractEncryptedContent.cxx, FixBrokenJ2K.cxx, FixJAIBugJPEGLS.cxx, gdcmrtionplan.cxx, gdcmrtplan.cxx, GetJPEGSamplePrecision.cxx, GetSequenceUltrasound.cxx, GetSubSequenceData.cxx, iU22tomultisc.cxx, LargeVRDSExplicit.cxx, MrProtocol.cxx, PatchFile.cxx, pmsct_rgb1.cxx, ReadAndDumpDICOMDIR.cxx, ReadExplicitLengthSQIVR.cxx, ReadGEMSSDO.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 >::SetFromDataSet().

10.75.3.12 GetDataElement() [2/2]

```
const DataElement& gdcm::DataSet::GetDataElement (
            const PrivateTag & t ) const
```

Return the dataelement.

10.75.3.13 GetDEEnd()

```
const DataElement& gdcm::DataSet::GetDEEnd ( ) const  [protected]
```

10.75.3.14 GetDES() [1/2]

```
const DataElementSet& gdcm::DataSet::GetDES ( ) const  [inline]
```

Examples:

ReadAndDumpDICOMDIR.cxx.

10.75.3.15 GetDES() [2/2]

```
DataElementSet& gdcm::DataSet::GetDES ( )  [inline]
```

10.75.3.16 GetLength()

```
template<typename TDE >
VL gdcm::DataSet::GetLength ( ) const  [inline]
```

10.75.3.17 GetMediaStorage()

```
MediaStorage gdcm::DataSet::GetMediaStorage ( ) const
```

10.75.3.18 GetPrivateCreator()

```
std::string gdcm::DataSet::GetPrivateCreator (
            const Tag & t ) const
```

Return the private creator of the private tag 't':

Examples:

DuplicatePCDE.cxx.

10.75.3.19 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, DuplicatePCDE.cxx, Extracting_All_Resolution.cxx, Fake_Image_Using_Stream_↩ Image_Writer.cxx, GenAllVR.cxx, GenFakeIdentifyFile.cxx, GenLongSeqs.cxx, GenSeqs.cxx, NewSequence.cs, and StreamImageReaderTest.cxx.

References gdcmErrorMacro, gdcm::Tag::GetGroup(), and gdcm::DataElement::GetTag().

10.75.3.20 InsertDataElement()

```
void gdcm::DataSet::InsertDataElement (
            const DataElement & de )  [inline], [protected]
```

References gdcmWarningMacro, gdcm::Value::GetLength(), gdcm::DataElement::GetValue(), gdcm::DataElement::GetVL(), gdcm::DataElement::IsEmpty(), and gdcm::operator<<().

10.75.3.21 IsEmpty()

```
bool gdcm::DataSet::IsEmpty ( ) const  [inline]
```

Returns if the dataset is empty.

Referenced by gdcm::Item::Read().

10.75.3.22 operator()()

```
const DataElement& gdcm::DataSet::operator() (
            uint16_t group,
            uint16_t element ) const  [inline]
```

10.75.3.23 operator=()

```
DataSet& gdcm::DataSet::operator= (
            DataSet const & val )  [inline]
```

10.75.3.24 operator[]()

```
const DataElement& gdcm::DataSet::operator[] (
            const Tag & t ) const  [inline]
```

10.75.3.25 Print()

```
void gdcm::DataSet::Print (
            std::ostream & os,
            std::string const & indent = "" ) const  [inline]
```

Referenced by gdcm::operator<<().

10.75.3.26 Read()

```
template<typename TDE , typename TSwap >
std::istream& gdcm::DataSet::Read (
            std::istream & is )
```

10.75.3.27 ReadNested()

```
template<typename TDE , typename TSwap >
std::istream& gdcm::DataSet::ReadNested (
             std::istream & is )
```

10.75.3.28 ReadSelectedPrivateTags()

```
template<typename TDE , typename TSwap >
std::istream& gdcm::DataSet::ReadSelectedPrivateTags (
             std::istream & is,
             const std::set< PrivateTag > & tags,
             bool readvalues = true )
```

10.75.3.29 ReadSelectedPrivateTagsWithLength()

```
template<typename TDE , typename TSwap >
std::istream& gdcm::DataSet::ReadSelectedPrivateTagsWithLength (
             std::istream & is,
             const std::set< PrivateTag > & tags,
             VL & length,
             bool readvalues = true )
```

10.75.3.30 ReadSelectedTags()

```
template<typename TDE , typename TSwap >
std::istream& gdcm::DataSet::ReadSelectedTags (
             std::istream & is,
             const std::set< Tag > & tags,
             bool readvalues = true )
```

10.75.3.31 ReadSelectedTagsWithLength()

```
template<typename TDE , typename TSwap >
std::istream& gdcm::DataSet::ReadSelectedTagsWithLength (
             std::istream & is,
             const std::set< Tag > & tags,
             VL & length,
             bool readvalues = true )
```

10.75.3.32 ReadUpToTag()

```
template<typename TDE , typename TSwap >
std::istream& gdcm::DataSet::ReadUpToTag (
             std::istream & is,
             const Tag & t,
             std::set< Tag > const & skiptags )
```

10.75.3.33 ReadUpToTagWithLength()

```
template<typename TDE , typename TSwap >
std::istream& gdcm::DataSet::ReadUpToTagWithLength (
             std::istream & is,
             const Tag & t,
             std::set< Tag > const & skiptags,
             VL & length )
```

10.75.3.34 ReadWithLength()

```
template<typename TDE , typename TSwap >
std::istream& gdcm::DataSet::ReadWithLength (
             std::istream & is,
             VL & length )
```

10.75.3.35 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, pmsct_rgb1.cxx, ReformatFile.cs, rle2img.cxx, and StandardizeFiles.cs.

10.75.3.36 Replace()

```
void gdcm::DataSet::Replace (
             const DataElement & de )  [inline]
```

Replace a dataelement with another one.

Examples:

ChangePrivateTags.cxx, ChangeSequenceUltrasound.cxx, CreateFakeRTDOSE.cxx, FixBrokenJ2K.cxx, FixJ↩AIBugJPEGLS.cxx, FixOrientation.cxx, GenFakeIdentifyFile.cxx, GetSubSequenceData.cxx, HelloWorld.cxx, i↩U22tomultisc.cxx, LargeVRDSExplicit.cxx, PatchFile.cxx, pmsct_rgb1.cxx, and rle2img.cxx.

References gdcmAssertAlwaysMacro.

10.75.3.37 ReplaceEmpty()

```
void gdcm::DataSet::ReplaceEmpty (
            const DataElement & de )  [inline]
```

Only replace a DICOM attribute when it is missing or empty.

References gdcmAssertAlwaysMacro.

10.75.3.38 Size()

```
SizeType gdcm::DataSet::Size ( ) const  [inline]
```

Examples:

DumpGEMSMovieGroup.cxx.

Referenced by gdcm::SequenceOfItems::Read().

10.75.3.39 Write()

```
template<typename TDE , typename TSwap >
std::ostream const& gdcm::DataSet::Write (
            std::ostream & os ) const
```

10.75.4 Friends And Related Function Documentation

10.75.4.1 CSAHeader

```
friend class CSAHeader  [friend]
```

10.75.4.2 operator<<

```
std::ostream& operator<< (
            std::ostream & _os,
            const DataSet & val )  [friend]
```

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

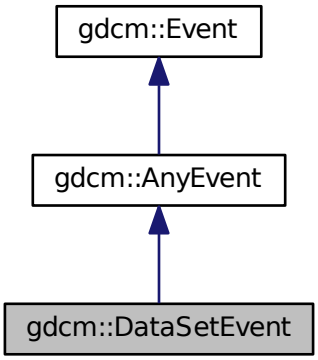
- gdcmDataSet.h

10.76 gdcm::DataSetEvent Class Reference

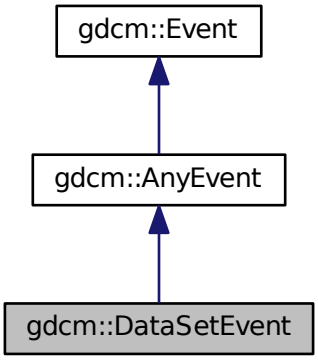
DataSetEvent.

`#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 (DataSet const *ds=NULL)
- DataSetEvent (const Self &s)
- virtual ~DataSetEvent ()
- virtual bool CheckEvent (const ::gdcm::Event *e) const
- DataSet const & GetDataSet () const
- virtual const char * GetEventName () const
- virtual ::gdcm::Event * MakeObject () const

10.76.1 Detailed Description

DataSetEvent.

Special type of event triggered during the DataSet store/move process

See also

10.76.2 Member Typedef Documentation

10.76.2.1 Self

```
typedef DataSetEvent gdcm::DataSetEvent::Self
```

10.76.2.2 Superclass

```
typedef AnyEvent gdcm::DataSetEvent::Superclass
```

10.76.3 Constructor & Destructor Documentation

10.76.3.1 DataSetEvent() [1/2]

```
gdcm::DataSetEvent::DataSetEvent (
            DataSet const * ds = NULL )  [inline]
```

10.76.3.2 ~DataSetEvent()

```
virtual gdcm::DataSetEvent::~DataSetEvent ( )  [inline], [virtual]
```

10.76.3.3 DataSetEvent() [2/2]

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

10.76.4 Member Function Documentation

10.76.4.1 CheckEvent()

```
virtual bool gdcm::DataSetEvent::CheckEvent (
            const ::gdcm::Event * e ) const  [inline], [virtual]
```

10.76.4.2 GetDataSet()

```
DataSet const& gdcm::DataSetEvent::GetDataSet ( ) const  [inline]
```

10.76.4.3 GetEventName()

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

Return the StringName associated with the event.

Implements gdcm::Event.

10.76.4.4 MakeObject()

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

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

Implements gdcm::Event.

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

- gdcmDataSetEvent.h

10.77 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)

10.77.1 Detailed Description

DataSetHelper (internal class, not intended for user level)

10.77.2 Member Function Documentation

10.77.2.1 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

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

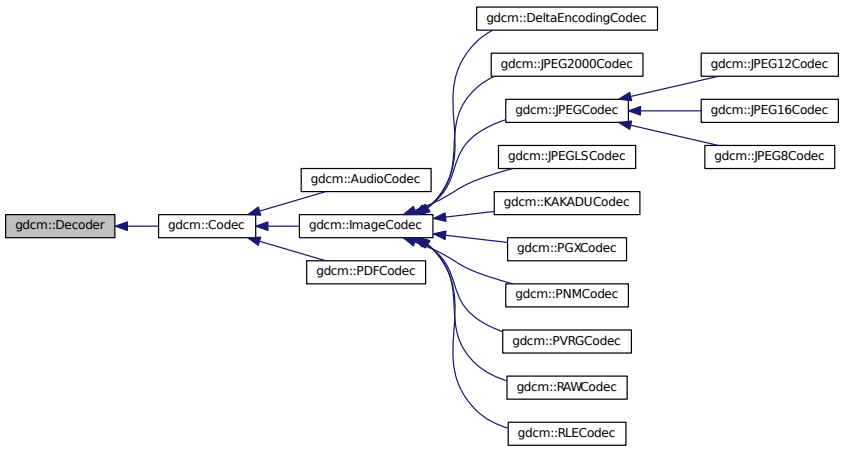
- gdcmDataSetHelper.h

10.78 gdcm::Decoder Class Reference

Decoder.

```
#include <gdcmDecoder.h>
```

Inheritance diagram for gdcm::Decoder:



Public Member Functions

- virtual ~Decoder ()
- 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 &)

10.78.1 Detailed Description

Decoder.

10.78.2 Constructor & Destructor Documentation

10.78.2.1 ~Decoder()

```
virtual gdcm::Decoder::~Decoder ( )  [inline], [virtual]
```

10.78.3 Member Function Documentation

10.78.3.1 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::JPEGCodec, gdcm::RLECodec, gdcm::PVRGCodec, gdcm::JPEG2000Codec, gdcm::ImageCodec, gdcm::JPEGLSCodec, gdcm::PNMCodec, gdcm::RAWCodec, gdcm::AudioCodec, gdcm::PDFCodec, gdcm::PGXCodec, and gdcm::KAKADUCodec.

10.78.3.2 Decode()

```
virtual bool gdcm::Decoder::Decode (
            DataElement const & ,
            DataElement &  )  [inline], [virtual]
```

Decode.

Reimplemented in gdcm::JPEGCodec, gdcm::RLECodec, gdcm::PVRGCodec, gdcm::JPEGLSCodec, gdcm::JPEG2000Codec, gdcm::ImageCodec, gdcm::DeltaEncodingCodec, gdcm::KAKADUCodec, gdcm::RAWCodec, gdcm::AudioCodec, and gdcm::PDFCodec.

10.78.3.3 DecodeByStreams()

```
virtual bool gdcm::Decoder::DecodeByStreams (
            std::istream & ,
            std::ostream &  )  [inline], [protected], [virtual]
```

Reimplemented in gdcm::JPEGCodec, gdcm::JPEG2000Codec, gdcm::RLECodec, gdcm::ImageCodec, gdcm::RAW↩ Codec, gdcm::JPEG12Codec, gdcm::JPEG16Codec, and gdcm::JPEG8Codec.

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

- gdcmDecoder.h

10.79 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 ()

10.79.1 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.

10.79.2 Constructor & Destructor Documentation

10.79.2.1 DefinedTerms()

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

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

- gdcmDefinedTerms.h

10.80 gdcm::Defs Class Reference

FIXME I do not like the name 'Defs'.

`#include <gdcmDefs.h>`

Public Member Functions

- Defs ()
- ~Defs ()
- const IOD & GetIODFromFile (const File &file) const
- const IODs & GetIODs () const
- IODs & GetIODs ()
- const Macros & GetMacros () const
- Macros & GetMacros ()
- const Modules & GetModules () const
- Modules & GetModules ()
- Type GetTypeFromTag (const File &file, const Tag &tag) const
- bool IsEmpty () const
- bool Verify (const File &file) const
- bool Verify (const DataSet &ds) 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

10.80.1 Detailed Description

FIXME I do not like the name 'Defs'.

Note

bla

Examples:

GenerateStandardSOPClasses.cxx, and TraverseModules.cxx.

10.80.2 Constructor & Destructor Documentation

10.80.2.1 Defs()

```
gdcm::Defs::Defs ( )
```

10.80.2.2 ~Defs()

```
gdcm::Defs::~Defs ( )
```

10.80.3 Member Function Documentation

10.80.3.1 GetIODFromFile()

```
const IOD& gdcm::Defs::GetIODFromFile (
            const File & file ) const
```

10.80.3.2 GetIODNameFromMediaStorage()

```
static const char* gdcm::Defs::GetIODNameFromMediaStorage (
            MediaStorage const & ms )  [static]
```

Examples:

GenerateStandardSOPClasses.cxx.

10.80.3.3 GetIODs() [1/2]

```
const IODs& gdcm::Defs::GetIODs ( ) const  [inline]
```

Examples:

TraverseModules.cxx.

10.80.3.4 GetIODs() [2/2]

```
IODs& gdcm::Defs::GetIODs ( )  [inline]
```

10.80.3.5 GetMacros() [1/2]

```
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 wraped within Modules. See gdcm::Module API directly

Examples:

TraverseModules.cxx.

10.80.3.6 GetMacros() [2/2]

```
Macros& gdcm::Defs::GetMacros ( )  [inline]
```

10.80.3.7 GetModules() [1/2]

```
const Modules& gdcm::Defs::GetModules ( ) const  [inline]
```

Examples:

TraverseModules.cxx.

10.80.3.8 GetModules() [2/2]

```
Modules& gdcm::Defs::GetModules ( )  [inline]
```

10.80.3.9 GetTypeFromTag()

```
Type gdcm::Defs::GetTypeFromTag (
            const File & file,
            const Tag & tag ) const
```

10.80.3.10 IsEmpty()

```
bool gdcm::Defs::IsEmpty ( ) const  [inline]
```

10.80.3.11 LoadDefaults()

```
void gdcm::Defs::LoadDefaults ( )  [protected]
```

10.80.3.12 LoadFromFile()

```
void gdcm::Defs::LoadFromFile (
            const char * filename )  [protected]
```

10.80.3.13 Verify() [1/2]

```
bool gdcm::Defs::Verify (
            const File & file ) const
```

10.80.3.14 Verify() [2/2]

```
bool gdcm::Defs::Verify (
            const DataSet & ds ) const
```

10.80.4 Friends And Related Function Documentation

10.80.4.1 Global

```
friend class Global  [friend]
```

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

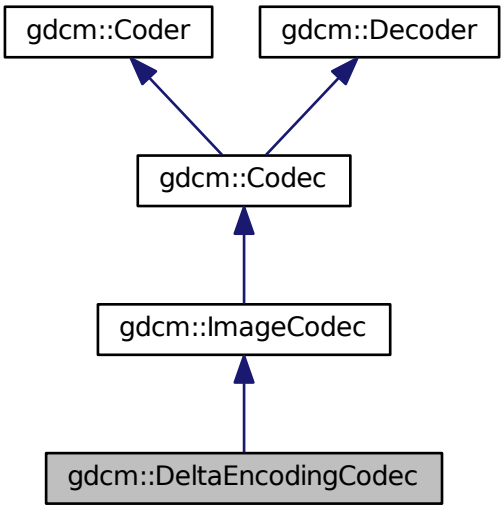
- gdcmDefs.h

10.81 gdcm::DeltaEncodingCodec Class Reference

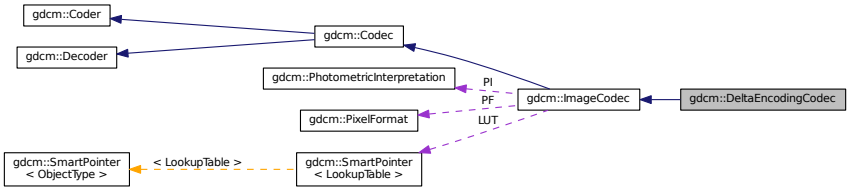
DeltaEncodingCodec compression used by some private vendor.

```
#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

10.81.1 Detailed Description

DeltaEncodingCodec compression used by some private vendor.

10.81.2 Constructor & Destructor Documentation

10.81.2.1 DeltaEncodingCodec()

```
gdcm::DeltaEncodingCodec::DeltaEncodingCodec ( )
```

10.81.2.2 ~DeltaEncodingCodec()

```
gdcm::DeltaEncodingCodec::~DeltaEncodingCodec ( )
```

10.81.3 Member Function Documentation

10.81.3.1 CanDecode()

```
bool gdcm::DeltaEncodingCodec::CanDecode (
            TransferSyntax const & ts )
```

10.81.3.2 Decode() [1/2]

```
bool gdcm::DeltaEncodingCodec::Decode (
            DataElement const & ,
            DataElement &  )  [virtual]
```

Decode.

Reimplemented from gdcm::Decoder.

10.81.3.3 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

10.82 gdcm::DICOMDIR Class Reference

DICOMDIR class.

`#include <gdcmDICOMDIR.h>`

Public Member Functions

- DICOMDIR ()
- DICOMDIR (const FileSet &fs)

10.82.1 Detailed Description

DICOMDIR class.

Structured for handling DICOMDIR

10.82.2 Constructor & Destructor Documentation

10.82.2.1 DICOMDIR() [1/2]

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

10.82.2.2 DICOMDIR() [2/2]

```
gdcm::DICOMDIR::DICOMDIR (
            const FileSet & fs )  [inline]
```

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

- gdcmDICOMDIR.h

10.83 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 ()

10.83.1 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

10.83.2 Member Typedef Documentation

10.83.2.1 FilenamesType

```
typedef Directory::FilenamesType gdcm::DICOMDIRGenerator::FilenamesType
```

10.83.2.2 FilenameType

```
typedef Directory::FilenameType gdcm::DICOMDIRGenerator::FilenameType
```

10.83.3 Constructor & Destructor Documentation

10.83.3.1 DICOMDIRGenerator()

```
gdcm::DICOMDIRGenerator::DICOMDIRGenerator ( )
```

10.83.3.2 ~DICOMDIRGenerator()

```
gdcm::DICOMDIRGenerator::~DICOMDIRGenerator ( )
```

10.83.4 Member Function Documentation

10.83.4.1 AddImageDirectoryRecord()

```
bool gdcm::DICOMDIRGenerator::AddImageDirectoryRecord ( )  [protected]
```

10.83.4.2 AddPatientDirectoryRecord()

```
bool gdcm::DICOMDIRGenerator::AddPatientDirectoryRecord ( )  [protected]
```

10.83.4.3 AddSeriesDirectoryRecord()

```
bool gdcm::DICOMDIRGenerator::AddSeriesDirectoryRecord ( )  [protected]
```

10.83.4.4 AddStudyDirectoryRecord()

```
bool gdcm::DICOMDIRGenerator::AddStudyDirectoryRecord ( )  [protected]
```

10.83.4.5 Generate()

```
bool gdcm::DICOMDIRGenerator::Generate ( )
```

Main function to generate the DICOMDIR.

10.83.4.6 GetFile()

```
File& gdcm::DICOMDIRGenerator::GetFile ( )
```

10.83.4.7 GetScanner()

```
Scanner& gdcm::DICOMDIRGenerator::GetScanner ( )  [protected]
```

10.83.4.8 SetDescriptor()

```
void gdcm::DICOMDIRGenerator::SetDescriptor (
            const char * d )
```

Set the File Set ID.

Warning

this need to be a valid VR::CS value

10.83.4.9 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.

10.83.4.10 SetFilenames()

```
void gdcm::DICOMDIRGenerator::SetFilenames (
            FilenamesType const & fns )
```

Set the list of filenames from which the DICOMDIR should be generated from.

10.83.4.11 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

10.84 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 ()
- 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

Protected Member Functions

- void LoadDefault ()

Friends

- class Dicts
- std::ostream & operator<< (std::ostream &_os, const Dict &_val)

10.84.1 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.

10.84.2 Member Typedef Documentation

10.84.2.1 ConstIterator

```
typedef MapDictEntry::const_iterator gdcm::Dict::ConstIterator
```

10.84.2.2 Iterator

```
typedef MapDictEntry::iterator gdcm::Dict::Iterator
```

10.84.2.3 MapDictEntry

```
typedef std::map<Tag, DictEntry> gdcm::Dict::MapDictEntry
```

10.84.3 Constructor & Destructor Documentation

10.84.3.1 Dict()

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

References gdcm::operator<<().

10.84.4 Member Function Documentation

10.84.4.1 AddDictEntry()

```
void gdcm::Dict::AddDictEntry (
            const Tag & tag,
            const DictEntry & de )  [inline]
```

10.84.4.2 Begin()

```
ConstIterator gdcm::Dict::Begin ( ) const  [inline]
```

Examples:

GenAllVR.cxx, and GenFakeIdentifyFile.cxx.

10.84.4.3 End()

```
ConstIterator gdcm::Dict::End ( ) const  [inline]
```

Examples:

GenAllVR.cxx, and GenFakeIdentifyFile.cxx.

10.84.4.4 GetDictEntry()

```
const DictEntry& gdcm::Dict::GetDictEntry (
            const Tag & tag ) const  [inline]
```

Examples:

GenFakeIdentifyFile.cxx, and PublicDict.cxx.

10.84.4.5 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.

10.84.4.6 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 garantee 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.

10.84.4.7 GetKeywordFromTag()

```
const char* gdcm::Dict::GetKeywordFromTag (
            Tag const & tag ) const  [inline]
```

Function to return the Keyword from a Tag.

10.84.4.8 IsEmpty()

```
bool gdcm::Dict::IsEmpty ( ) const  [inline]
```

Referenced by gdcm::Dicts::IsEmpty().

10.84.4.9 LoadDefault()

```
void gdcm::Dict::LoadDefault ( )  [protected]
```

10.84.5 Friends And Related Function Documentation

10.84.5.1 Dicts

```
friend class Dicts  [friend]
```

10.84.5.2 operator<<

```
std::ostream& operator<< (
            std::ostream & _os,
            const Dict & _val )  [friend]
```

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

- gdcmDict.h

10.85 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 ()

10.85.1 Detailed Description

Class to convert a .dic file into something else:

- CXX code : embeded dict into shared lib (DICT_DEFAULT)
- Debug mode (DICT_DEBUG)
- XML dict (DICT_XML)

  Note

10.85.2 Member Enumeration Documentation

10.85.2.1 OutputTypes

```
enum gdcm::DictConverter::OutputTypes
```

Enumerator

DICT_DEFAULT	
DICT_DEBUG	
DICT_XML	

10.85.3 Constructor & Destructor Documentation

10.85.3.1 DictConverter()

```
gdcm::DictConverter::DictConverter ( )
```

10.85.3.2 ~DictConverter()

```
gdcm::DictConverter::~DictConverter ( )
```

10.85.4 Member Function Documentation

10.85.4.1 AddGroupLength()

```
void gdcm::DictConverter::AddGroupLength ( )  [protected]
```

10.85.4.2 Convert()

```
void gdcm::DictConverter::Convert ( )
```

10.85.4.3 ConvertToCXX()

```
bool gdcm::DictConverter::ConvertToCXX (
            const char * raw,
            std::string & cxx )  [protected]
```

10.85.4.4 ConvertToXML()

```
bool gdcm::DictConverter::ConvertToXML (
            const char * raw,
            std::string & cxx )  [protected]
```

10.85.4.5 GetDictName()

```
const std::string& gdcm::DictConverter::GetDictName ( ) const
```

10.85.4.6 GetInputFilename()

```
const std::string& gdcm::DictConverter::GetInputFilename ( ) const
```

10.85.4.7 GetOutputFilename()

```
const std::string& gdcm::DictConverter::GetOutputFilename ( ) const
```

10.85.4.8 GetOutputType()

```
int gdcm::DictConverter::GetOutputType ( ) const  [inline]
```

10.85.4.9 Readuint16()

```
static bool gdcm::DictConverter::Readuint16 (
             const char * raw,
             uint16_t & ov )  [static]
```

10.85.4.10 ReadVM()

```
static bool gdcm::DictConverter::ReadVM (
             const char * raw,
             VM::VMType & type )  [static]
```

10.85.4.11 ReadVR()

```
static bool gdcm::DictConverter::ReadVR (
             const char * raw,
             VR::VRType & type )  [static]
```

10.85.4.12 SetDictName()

```
void gdcm::DictConverter::SetDictName (
             const char * name )
```

10.85.4.13 SetInputFileName()

```
void gdcm::DictConverter::SetInputFileName (
            const char * filename )
```

10.85.4.14 SetOutputFileName()

```
void gdcm::DictConverter::SetOutputFileName (
            const char * filename )
```

10.85.4.15 SetOutputType()

```
void gdcm::DictConverter::SetOutputType (
            int type )  [inline]
```

10.85.4.16 WriteFooter()

```
void gdcm::DictConverter::WriteFooter ( )  [protected]
```

10.85.4.17 WriteHeader()

```
void gdcm::DictConverter::WriteHeader ( )  [protected]
```

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

- gdcmDictConverter.h

10.86 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)

10.86.1 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.

10.86.2 Constructor & Destructor Documentation

10.86.2.1 DictEntry()

```
gdcm::DictEntry::DictEntry (
            const char * name = "",
            const char * keyword = "",
            VR const & vr = VR::INVALID,
            VM const & vm = VM::VM0,
            bool ret = false )  [inline]
```

References gdcm::operator<<().

10.86.3 Member Function Documentation

10.86.3.1 GetKeyword()

```
const char* gdcm::DictEntry::GetKeyword ( ) const  [inline]
```

same as GetName but without spaces...

10.86.3.2 GetName()

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

Set/Get Name.

Referenced by gdcm::PrivateDict::PrintXML().

10.86.3.3 GetRetired()

```
bool gdcm::DictEntry::GetRetired ( ) const  [inline]
```

Set/Get Retired flag.

Examples:

GenAllVR.cxx.

10.86.3.4 GetVM()

```
const VM& gdcm::DictEntry::GetVM ( ) const  [inline]
```

Set/Get VM.

Referenced by gdcm::PrivateDict::AddDictEntry(), and gdcm::PrivateDict::PrintXML().

10.86.3.5 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().

10.86.3.6 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)

10.86.3.7 SetElementXX()

```
void gdcm::DictEntry::SetElementXX (
            bool v )  [inline]
```

Set whether element is shared in multiple elements (Source Image IDs typically)

10.86.3.8 SetGroupXX()

```
void gdcm::DictEntry::SetGroupXX (
            bool v )  [inline]
```

Set whether element is shared in multiple groups (Curve/Overlay typically)

10.86.3.9 SetKeyword()

```
void gdcm::DictEntry::SetKeyword (
            const char * keyword )  [inline]
```

10.86.3.10 SetName()

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

10.86.3.11 SetRetired()

```
void gdcm::DictEntry::SetRetired (
            bool retired )  [inline]
```

10.86.3.12 SetVM()

```
void gdcm::DictEntry::SetVM (
            VM const & vm )  [inline]
```

10.86.3.13 SetVR()

```
void gdcm::DictEntry::SetVR (
            const VR & vr )  [inline]
```

Referenced by gdcm::PrivateDict::AddDictEntry().

10.86.4 Friends And Related Function Documentation

10.86.4.1 Dict

```
friend class Dict  [friend]
```

10.86.4.2 operator<<

```
std::ostream& operator<< (
            std::ostream & _os,
            const DictEntry & _val )  [friend]
```

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

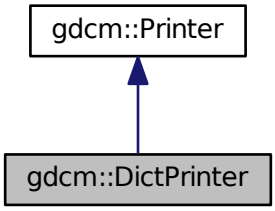
- gdcmDictEntry.h

10.87 gdcm::DictPrinter Class Reference

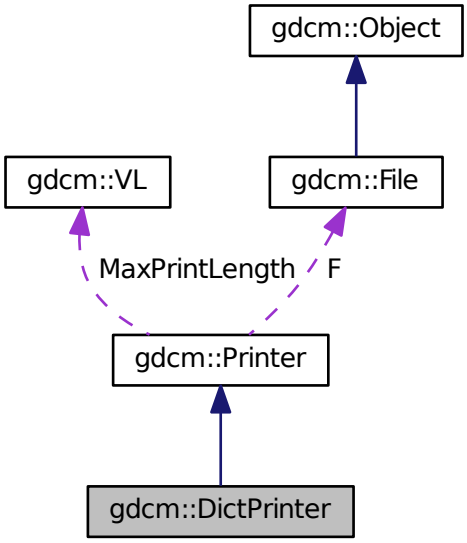
DictPrinter class.

```
#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

10.87.1 Detailed Description

DictPrinter class.

10.87.2 Constructor & Destructor Documentation

10.87.2.1 DictPrinter()

```
gdcm::DictPrinter::DictPrinter ( )
```

10.87.2.2 ~DictPrinter()

```
gdcm::DictPrinter::~DictPrinter ( )
```

10.87.3 Member Function Documentation

10.87.3.1 Print()

```
void gdcm::DictPrinter::Print (
            std::ostream & os )
```

10.87.3.2 PrintDataElement2()

```
void gdcm::DictPrinter::PrintDataElement2 (
            std::ostream & os,
            const DataSet & ds,
            const DataElement & ide )  [protected]
```

10.87.3.3 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

10.88 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 CSAHeaderDict & GetCSAHeaderDict () const
- const DictEntry & GetDictEntry (const Tag &tag, const char *owner=NULL) const

  NOT THREAD SAFE.
- const DictEntry & GetDictEntry (const PrivateTag &tag) const
- const PrivateDict & GetPrivateDict () const
- PrivateDict & GetPrivateDict ()
- const Dict & GetPublicDict () const
- bool IsEmpty () const

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)

10.88.1 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.

10.88.2 Member Enumeration Documentation

10.88.2.1 ConstructorType

```
enum gdcm::Dicts::ConstructorType  [protected]
```

Enumerator

PHILIPS	
GEMS	
SIEMENS	

10.88.3 Constructor & Destructor Documentation

10.88.3.1 Dicts()

```
gdcm::Dicts::Dicts ( )
```

10.88.3.2 ~Dicts()

```
gdcm::Dicts::~Dicts ( )
```

10.88.4 Member Function Documentation

10.88.4.1 GetConstructorString()

```
static const char* gdcm::Dicts::GetConstructorString (
            ConstructorType type )  [static], [protected]
```

10.88.4.2 GetCSAHeaderDict()

```
const CSAHeaderDict& gdcm::Dicts::GetCSAHeaderDict ( ) const
```

Examples:

MrProtocol.cxx.

10.88.4.3 GetDictEntry() [1/2]

```
const DictEntry& gdcm::Dicts::GetDictEntry (
            const Tag & tag,
            const char * owner = NULL ) const
```

NOT 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.

10.88.4.4 GetDictEntry() [2/2]

```
const DictEntry& gdcm::Dicts::GetDictEntry (
            const PrivateTag & tag ) const
```

10.88.4.5 GetPrivateDict() [1/2]

```
const PrivateDict& gdcm::Dicts::GetPrivateDict ( ) const
```

10.88.4.6 GetPrivateDict() [2/2]

```
PrivateDict& gdcm::Dicts::GetPrivateDict ( )
```

10.88.4.7 GetPublicDict()

```
const Dict& gdcm::Dicts::GetPublicDict ( ) const
```

Examples:

GenAllVR.cxx, GenFakeIdentifyFile.cxx, PublicDict.cxx, and ReadAndPrintAttributes.cxx.

10.88.4.8 IsEmpty()

```
bool gdcm::Dicts::IsEmpty ( ) const  [inline]
```

References gdcm::Dict::IsEmpty().

10.88.4.9 LoadDefaults()

```
void gdcm::Dicts::LoadDefaults ( )  [protected]
```

10.88.5 Friends And Related Function Documentation

10.88.5.1 Global

```
friend class Global  [friend]
```

10.88.5.2 operator<<

```
std::ostream& operator<< (
            std::ostream & _os,
            const Dicts & d )  [friend]
```

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

- gdcmDicts.h

10.89 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 }

10.89.1 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)

10.89.2 Member Enumeration Documentation

10.89.2.1 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	

Enumerator

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

10.90 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)

10.90.1 Detailed Description

class to handle DirectionCosines

Examples:

DiscriminateVolume.cxx.

10.90.2 Constructor & Destructor Documentation

10.90.2.1 DirectionCosines() [1/2]

```
gdcm::DirectionCosines::DirectionCosines ( )
```

10.90.2.2 DirectionCosines() [2/2]

```
gdcm::DirectionCosines::DirectionCosines (
            const double dircos[6] )
```

10.90.2.3 ~DirectionCosines()

```
gdcm::DirectionCosines::~DirectionCosines ( )
```

10.90.3 Member Function Documentation

10.90.3.1 ComputeDistAlongNormal()

```
double gdcm::DirectionCosines::ComputeDistAlongNormal (
            const double ipp[3] ) const
```

Compute the distance along the normal.

10.90.3.2 Cross()

```
void gdcm::DirectionCosines::Cross (
            double z[3] ) const
```

Compute Cross product.

10.90.3.3 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.

10.90.3.4 Dot()

```
double gdcm::DirectionCosines::Dot ( ) const
```

Compute Dot.

10.90.3.5 IsValid()

```
bool gdcm::DirectionCosines::IsValid ( ) const
```

Return whether or not this is a valid direction cosines.

10.90.3.6 Normalize()

```
void gdcm::DirectionCosines::Normalize ( )
```

Normalize in-place.

10.90.3.7 operator const double *()

```
gdcm::DirectionCosines::operator const double * ( ) const  [inline]
```

Make the class behave like a const double *.

10.90.3.8 Print()

```
void gdcm::DirectionCosines::Print (
            std::ostream &  ) const
```

Print.

10.90.3.9 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

10.91 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 ()
- ~Directory ()
- 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)

10.91.1 Detailed Description

Class for manipulation directories.

Note

This implementation provide a cross platform implementation for manipulating directores: 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:

DiscriminateVolume.cxx, DumpToSQLITE3.cxx, gdcmorthoplanes.cxx, GenerateRTSTRUCT.cxx, ReadUTF8Qt↩Dir.cxx, reslicesphere.cxx, SortImage.cxx, threadgdcm.cxx, and VolumeSorter.cxx.

10.91.2 Member Typedef Documentation

10.91.2.1 FilenamesType

```
typedef std::vector<FilenameType> gdcm::Directory::FilenamesType
```

Examples:

DiscriminateVolume.cxx.

10.91.2.2 FilenameType

```
typedef std::string gdcm::Directory::FilenameType
```

10.91.3 Constructor & Destructor Documentation

10.91.3.1 Directory()

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

10.91.3.2 ~Directory()

```
gdcm::Directory::~Directory ( )  [inline]
```

10.91.4 Member Function Documentation

10.91.4.1 Explore()

```
unsigned int gdcm::Directory::Explore (
            FilenameType const & name,
            bool recursive )  [protected]
```

Return number of file found when 'recursive'ly exploring directory `name`

10.91.4.2 GetDirectories()

```
FilenamesType const& gdcm::Directory::GetDirectories ( ) const  [inline]
```

Return the Directories traversed.

10.91.4.3 GetFilenames()

```
FilenamesType const& gdcm::Directory::GetFilenames ( ) const  [inline]
```

Set/Get the file names within the directory.

Examples:

DiscriminateVolume.cxx, DumpToSQLITE3.cxx, gdcmorthoplanes.cxx, GenerateRTSTRUCT.cxx, ReadUTF8Qt↩Dir.cxx, reslicesphere.cxx, SortImage.cxx, threadgdcm.cxx, and VolumeSorter.cxx.

10.91.4.4 GetToplevel()

```
FilenameType const& gdcm::Directory::GetToplevel ( ) const  [inline]
```

Get the name of the toplevel directory.

10.91.4.5 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:

DecompressImageMultiframe.cs, DiscriminateVolume.cxx, DumpToSQLITE3.cxx, gdcmorthoplanes.cxx, GenerateRTSTRUCT.cxx, ReadUTF8QtDir.cxx, reslicesphere.cxx, SortImage.cxx, threadgdcm.cxx, and VolumeSorter.cxx.

10.91.4.6 Print()

```
void gdcm::Directory::Print (
            std::ostream & os = std::cout ) const
```

Print.

Examples:

SortImage.cxx.

Referenced by gdcm::operator<<().

10.91.5 Friends And Related Function Documentation

10.91.5.1 operator<<

```
std::ostream& operator<< (
            std::ostream & _os,
            const Directory & d )  [friend]
```

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

- gdcmDirectory.h

10.92 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)

10.92.1 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

10.92.2 Member Function Documentation

10.92.2.1 GetCTImageSeriesUIDs()

```
static Directory::FilenamesType gdcm::DirectoryHelper::GetCTImageSeriesUIDs (
            const std::string & inDirectory )  [static]
```

10.92.2.2 GetFilenamesFromSeriesUIDs()

```
static Directory::FilenamesType gdcm::DirectoryHelper::GetFilenamesFromSeriesUIDs (
            const std::string & inDirectory,
            const std::string & inSeriesUID )  [static]
```

Examples:

GenerateRTSTRUCT.cxx.

10.92.2.3 GetFrameOfReference()

```
static std::string gdcm::DirectoryHelper::GetFrameOfReference (
            const std::vector< DataSet > & inDS )  [static]
```

10.92.2.4 GetMRImageSeriesUIDs()

```
static Directory::FilenamesType gdcm::DirectoryHelper::GetMRImageSeriesUIDs (
            const std::string & inDirectory )  [static]
```

10.92.2.5 GetRTStructSeriesUIDs()

```
static Directory::FilenamesType gdcm::DirectoryHelper::GetRTStructSeriesUIDs (
            const std::string & inDirectory )  [static]
```

Examples:

GenerateRTSTRUCT.cxx.

10.92.2.6 GetSeriesUIDsBySOPClassUID()

```
static Directory::FilenamesType gdcm::DirectoryHelper::GetSeriesUIDsBySOPClassUID (
            const std::string & inDirectory,
            const std::string & inSOPClassUID )  [static]
```

10.92.2.7 GetSOPClassUID()

```
static std::string gdcm::DirectoryHelper::GetSOPClassUID (
            const std::vector< DataSet > & inDS )  [static]
```

10.92.2.8 GetStringValueFromTag()

```
static std::string gdcm::DirectoryHelper::GetStringValueFromTag (
            const Tag & t,
            const DataSet & ds )  [static]
```

10.92.2.9 LoadImageFromFiles()

```
static std::vector<DataSet> gdcm::DirectoryHelper::LoadImageFromFiles (
            const std::string & inDirectory,
            const std::string & inSeriesUID )  [static]
```

10.92.2.10 RetrieveSOPInstanceUIDFromIndex()

```
static std::string gdcm::DirectoryHelper::RetrieveSOPInstanceUIDFromIndex (
            int inIndex,
            const std::vector< DataSet > & inDS )  [static]
```

10.92.2.11 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

10.93 gdcm::DummyValueGenerator Class Reference

Class for generating dummy value.

`#include <gdcmDummyValueGenerator.h>`

Static Public Member Functions

- static const char * Generate (const char *input)

10.93.1 Detailed Description

Class for generating dummy value.

See also

Anonymizer

10.93.2 Member Function Documentation

10.93.2.1 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:

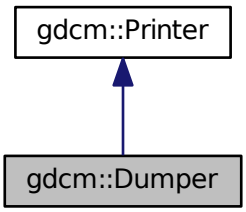
- gdcmDummyValueGenerator.h

10.94 gdcm::Dumper Class Reference

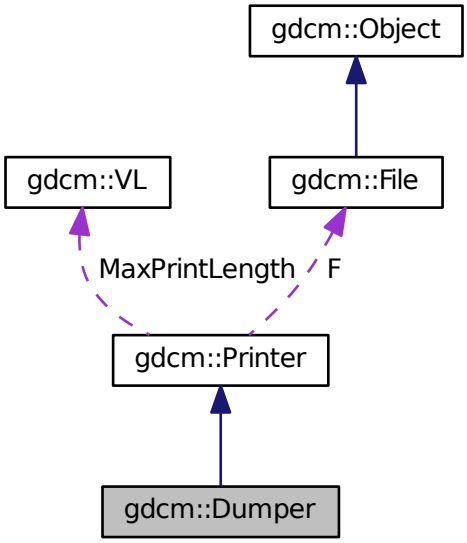
Codec class.

`#include <gdcmDumper.h>`

Inheritance diagram for gdcm::Dumper:



Collaboration diagram for gdcm::Dumper:



Public Member Functions

- Dumper ()
- ~Dumper ()

Additional Inherited Members

10.94.1 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

10.94.2 Constructor & Destructor Documentation

10.94.2.1 Dumper()

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

10.94.2.2 ~Dumper()

```
gdcm::Dumper::~Dumper ( )  [inline]
```

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

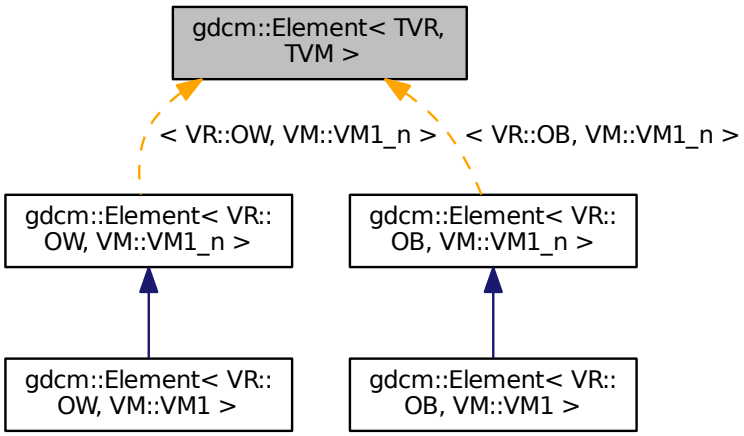
- gdcmDumper.h

10.95 gdcm::Element< TVR, TVM > Class Template Reference

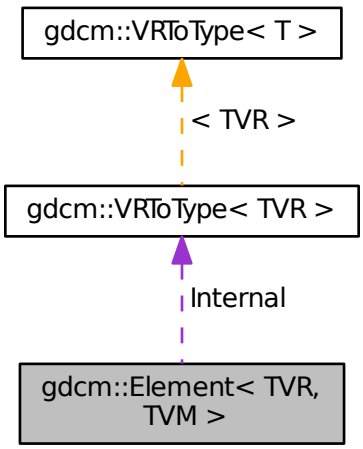
Element class.

```
#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
- const VRToType< TVR >::Type & GetValue (unsigned int idx=0) const
- VRToType< TVR >::Type & GetValue (unsigned int idx=0)
- 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)

10.95.1 Detailed Description

template<int TVR, int TVM>
class gdcm::Element< TVR, TVM >

Element class.

Note

TODO

Examples:

csa2img.cxx, DumpADAC.cxx, DumpGEMSMovieGroup.cxx, DumpPhilipsECHO.cxx, Extracting_All_Resolution.cxx, Fake_Image_Using_Stream_Image_Writer.cxx, GetSubSequenceData.cxx, and iU22tomultisc.cxx.

10.95.2 Member Typedef Documentation

10.95.2.1 Type

```
template<int TVR, int TVM>
typedef VRToType<TVR>::Type gdcm::Element< TVR, TVM >::Type
```

10.95.3 Member Function Documentation

10.95.3.1 GetAsDataElement()

```
template<int TVR, int TVM>
DataElement gdcm::Element< TVR, TVM >::GetAsDataElement ( ) const  [inline]
```

10.95.3.2 GetLength()

```
template<int TVR, int TVM>
unsigned long gdcm::Element< TVR, TVM >::GetLength ( ) const  [inline]
```

10.95.3.3 GetValue() [1/2]

```
template<int TVR, int TVM>
const VRToType<TVR>::Type& gdcm::Element< TVR, TVM >::GetValue (
            unsigned int idx = 0 ) const  [inline]
```

10.95.3.4 GetValue() [2/2]

```
template<int TVR, int TVM>
VRToType<TVR>::Type& gdcm::Element< TVR, TVM >::GetValue (
            unsigned int idx = 0 )  [inline]
```

10.95.3.5 GetValues()

```
template<int TVR, int TVM>
const VRToType<TVR>::Type* gdcm::Element< TVR, TVM >::GetValues ( ) const  [inline]
```

10.95.3.6 GetVM()

```
template<int TVR, int TVM>
static VM gdcm::Element< TVR, TVM >::GetVM ( )  [inline], [static]
```

10.95.3.7 GetVR()

```
template<int TVR, int TVM>
static VR gdcm::Element< TVR, TVM >::GetVR ( )  [inline], [static]
```

10.95.3.8 operator[]()

```
template<int TVR, int TVM>
VRToType<TVR>::Type gdcm::Element< TVR, TVM >::operator[] (
            unsigned int idx ) const  [inline]
```

10.95.3.9 Print()

```
template<int TVR, int TVM>
void gdcm::Element< TVR, TVM >::Print (
            std::ostream & _os ) const  [inline]
```

10.95.3.10 Read()

```
template<int TVR, int TVM>
void gdcm::Element< TVR, TVM >::Read (
             std::istream & _is )  [inline]
```

10.95.3.11 Set()

```
template<int TVR, int TVM>
void gdcm::Element< TVR, TVM >::Set (
             Value const & v )  [inline]
```

10.95.3.12 SetFromDataElement()

```
template<int TVR, int TVM>
void gdcm::Element< TVR, TVM >::SetFromDataElement (
             DataElement< TVR, TVM > const & de )  [inline]
```

10.95.3.13 SetNoSwap()

```
template<int TVR, int TVM>
void gdcm::Element< TVR, TVM >::SetNoSwap (
             Value const & v )  [inline], [protected]
```

10.95.3.14 SetValue()

```
template<int TVR, int TVM>
void gdcm::Element< TVR, TVM >::SetValue (
             typename VRToType< TVR >::Type v,
             unsigned int idx = 0 )  [inline]
```

10.95.3.15 Write()

```
template<int TVR, int TVM>
void gdcm::Element< TVR, TVM >::Write (
             std::ostream & _os ) const  [inline]
```

10.95.4 Member Data Documentation

10.95.4.1 Internal

```
template<int TVR, int TVM>
VRToType<TVR>::Type gdcm::Element< TVR, TVM >::Internal[VMToLength< TVM >::Length]
```

Referenced by gdcm::Element< TVR, VM::VM1_n >::operator=().

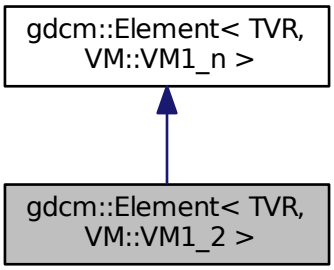
The documentation for this class was generated from the following file:

- gdcmElement.h

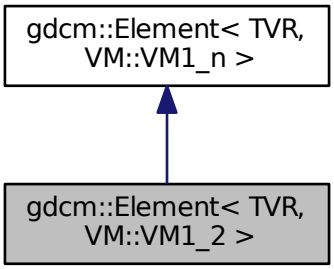
10.96 gdcm::Element< TVR, VM::VM1_2 > Class Template Reference

`#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

10.96.1 Member Typedef Documentation

10.96.1.1 Parent

```
template<int TVR>
typedef Element<TVR, VM::VM1_n> gdcm::Element< TVR, VM::VM1_2 >::Parent
```

10.96.2 Member Function Documentation

10.96.2.1 SetLength()

```
template<int TVR>
void gdcm::Element< TVR, VM::VM1_2 >::SetLength (
            int len )  [inline]
```

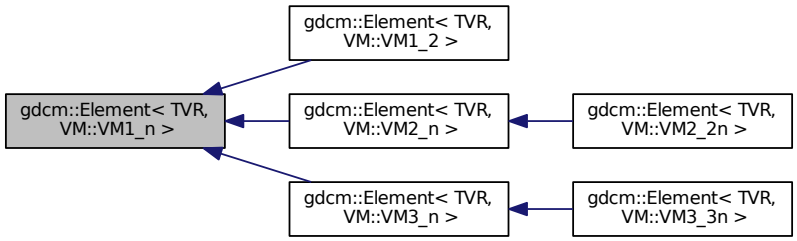
The documentation for this class was generated from the following file:

- gdcmElement.h

10.97 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
- const VRToType< TVR >::Type & GetValue (unsigned int idx=0) const
- VRToType< TVR >::Type & GetValue (unsigned int idx=0)
- 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)

10.97.1 Member Typedef Documentation

10.97.1.1 Type

```
template<int TVR>
typedef VRToType<TVR>::Type gdcm::Element< TVR, VM::VM1_n >::Type
```

10.97.2 Constructor & Destructor Documentation

10.97.2.1 Element() [1/2]

```
template<int TVR>
gdcm::Element< TVR, VM::VM1_n >::Element ( )  [inline], [explicit]
```

10.97.2.2 ~Element()

```
template<int TVR>
gdcm::Element< TVR, VM::VM1_n >::~Element ( )  [inline]
```

10.97.2.3 Element() [2/2]

```
template<int TVR>
gdcm::Element< TVR, VM::VM1_n >::Element (
            const Element< TVR, VM::VM1_n > & _val )  [inline]
```

10.97.3 Member Function Documentation

10.97.3.1 GetAsDataElement()

```
template<int TVR>
DataElement gdcm::Element< TVR, VM::VM1_n >::GetAsDataElement ( ) const  [inline]
```

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

10.97.3.2 GetLength()

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

10.97.3.3 GetValue() [1/2]

```
template<int TVR>
const VRToType<TVR>::Type& gdcm::Element< TVR, VM::VM1_n >::GetValue (
            unsigned int idx = 0 ) const  [inline]
```

10.97.3.4 GetValue() [2/2]

```
template<int TVR>
VRToType<TVR>::Type& gdcm::Element< TVR, VM::VM1_n >::GetValue (
            unsigned int idx = 0 )  [inline]
```

10.97.3.5 GetVM()

```
template<int TVR>
static VM gdcm::Element< TVR, VM::VM1_n >::GetVM ( )  [inline], [static]
```

10.97.3.6 GetVR()

```
template<int TVR>
static VR gdcm::Element< TVR, VM::VM1_n >::GetVR ( )  [inline], [static]
```

10.97.3.7 operator=()

```
template<int TVR>
Element& gdcm::Element< TVR, VM::VM1_n >::operator= (
             const Element< TVR, VM::VM1_n > & _val )  [inline]
```

References gdcm::Element< TVR, TVM >::Internal.

10.97.3.8 operator[]()

```
template<int TVR>
VRToType<TVR>::Type gdcm::Element< TVR, VM::VM1_n >::operator[] (
             unsigned int idx ) const  [inline]
```

10.97.3.9 Print()

```
template<int TVR>
void gdcm::Element< TVR, VM::VM1_n >::Print (
             std::ostream & _os ) const  [inline]
```

10.97.3.10 Read()

```
template<int TVR>
void gdcm::Element< TVR, VM::VM1_n >::Read (
             std::istream & _is )  [inline]
```

10.97.3.11 Set()

```
template<int TVR>
void gdcm::Element< TVR, VM::VM1_n >::Set (
             Value const & v )  [inline]
```

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

10.97.3.12 SetArray()

```
template<int TVR>
void gdcm::Element< TVR, VM::VM1_n >::SetArray (
             const Type * array,
             unsigned long len,
             bool save = false )  [inline]
```

10.97.3.13 SetFromDataElement()

```
template<int 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().

10.97.3.14 SetLength()

```
template<int TVR>
void gdcm::Element< TVR, VM::VM1_n >::SetLength (
            unsigned long len )  [inline]
```

10.97.3.15 SetNoSwap()

```
template<int TVR>
void gdcm::Element< TVR, VM::VM1_n >::SetNoSwap (
            Value const & v )  [inline], [protected]
```

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

10.97.3.16 SetValue()

```
template<int TVR>
void gdcm::Element< TVR, VM::VM1_n >::SetValue (
            typename VRToType< TVR >::Type v,
            unsigned int idx = 0 )  [inline]
```

10.97.3.17 Write()

```
template<int TVR>
void gdcm::Element< TVR, VM::VM1_n >::Write (
            std::ostream & _os ) const  [inline]
```

10.97.3.18 WriteASCII()

```
template<int TVR>
void gdcm::Element< TVR, VM::VM1_n >::WriteASCII (
            std::ostream & os ) const  [inline]
```

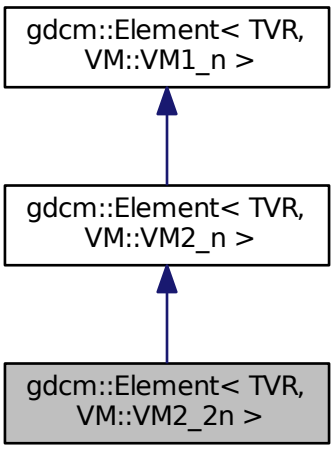
The documentation for this class was generated from the following file:

- gdcmElement.h

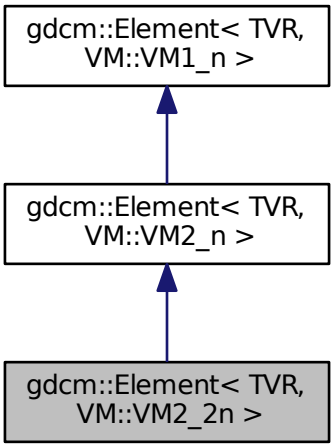
10.98 gdcm::Element< TVR, VM::VM2_2n > Class Template Reference

`#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

10.98.1 Member Typedef Documentation

10.98.1.1 Parent

```
template<int TVR>
typedef Element<TVR, VM::VM2_n> gdcm::Element< TVR, VM::VM2_2n >::Parent
```

10.98.2 Member Function Documentation

10.98.2.1 SetLength()

```
template<int TVR>
void gdcm::Element< TVR, VM::VM2_2n >::SetLength (
            int len )  [inline]
```

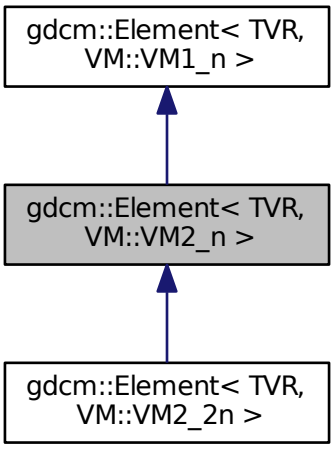
The documentation for this class was generated from the following file:

- gdcmElement.h

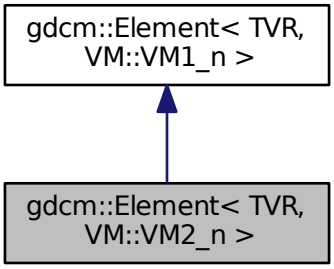
10.99 gdcm::Element< TVR, VM::VM2_n > Class Template Reference

```
#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

10.99.1 Member Typedef Documentation

10.99.1.1 Parent

```
template<int TVR>
typedef Element<TVR, VM::VM1_n> gdcm::Element< TVR, VM::VM2_n >::Parent
```

10.99.2 Member Function Documentation

10.99.2.1 SetLength()

```
template<int TVR>
void gdcm::Element< TVR, VM::VM2_n >::SetLength (
            int len )  [inline]
```

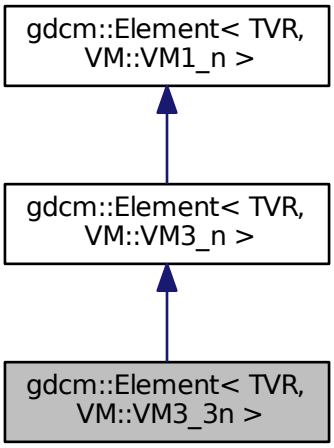
The documentation for this class was generated from the following file:

- gdcmElement.h

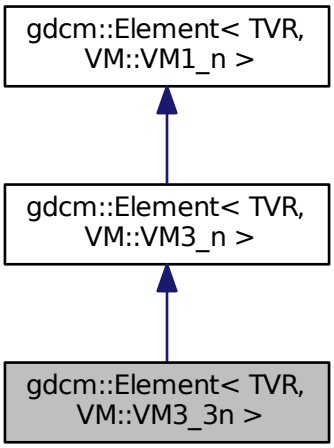
10.100 gdcm::Element< TVR, VM::VM3_3n > Class Template Reference

```
#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

10.100.1 Member Typedef Documentation

10.100.1.1 Parent

```
template<int TVR>
typedef Element<TVR, VM::VM3_n> gdcm::Element< TVR, VM::VM3_3n >::Parent
```

10.100.2 Member Function Documentation

10.100.2.1 SetLength()

```
template<int TVR>
void gdcm::Element< TVR, VM::VM3_3n >::SetLength (
            int len )  [inline]
```

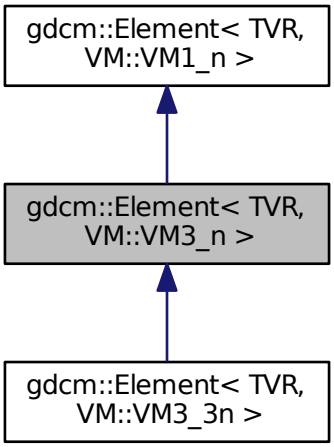
The documentation for this class was generated from the following file:

- gdcmElement.h

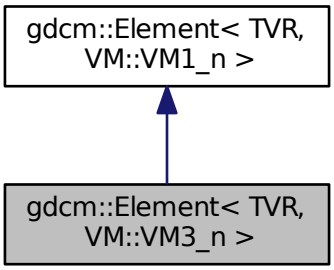
10.101 gdcm::Element< TVR, VM::VM3_n > Class Template Reference

`#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

10.101.1 Member Typedef Documentation

10.101.1.1 Parent

```
template<int TVR>
typedef Element<TVR, VM::VM1_n> gdcm::Element< TVR, VM::VM3_n >::Parent
```

10.101.2 Member Function Documentation

10.101.2.1 SetLength()

```
template<int 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

10.102 gdcm::Element< VR::AS, VM::VM5 > Class Template 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)]

10.102.1 Member Function Documentation

10.102.1.1 GetLength()

```
unsigned long gdcm::Element< VR::AS, VM::VM5 >::GetLength ( ) const  [inline]
```

10.102.1.2 Print()

```
void gdcm::Element< VR::AS, VM::VM5 >::Print (
            std::ostream & _os ) const  [inline]
```

10.102.2 Member Data Documentation

10.102.2.1 Internal

```
char gdcm::Element< VR::AS, VM::VM5 >::Internal[VMToLength< VM::VM5 >::Length *sizeof(VRToType<
VR::AS >::Type)]
```

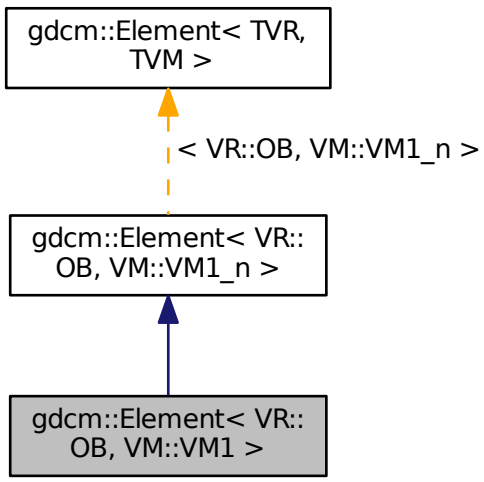
The documentation for this class was generated from the following file:

- gdcmElement.h

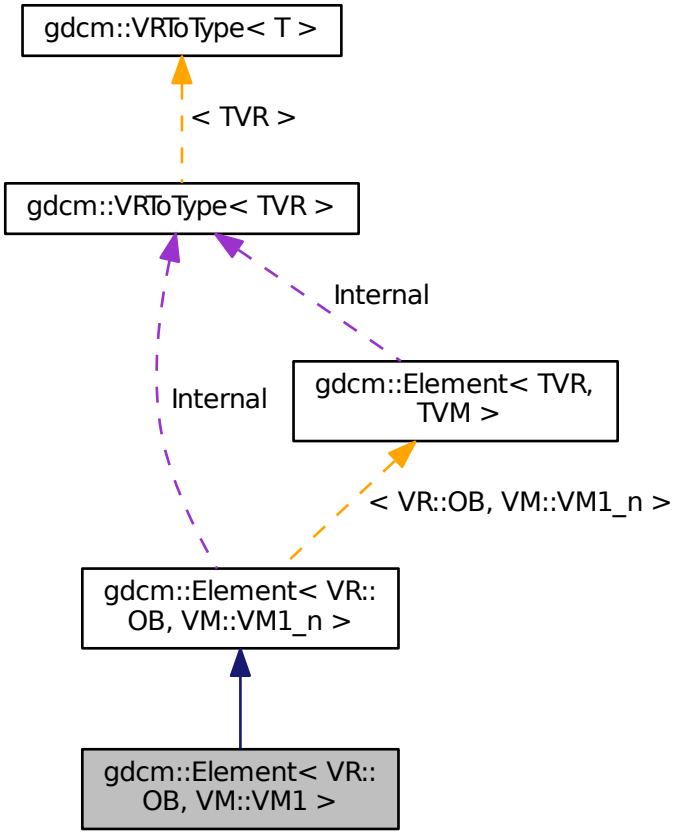
10.103 gdcm::Element< VR::OB, VM::VM1 > Class Template Reference

```
#include <gdcmElement.h>
```

Inheritance diagram for gdcm::Element< VR::OB, VM::VM1 >:



Collaboration diagram for gdcm::Element< VR::OB, VM::VM1 >:



Additional Inherited Members

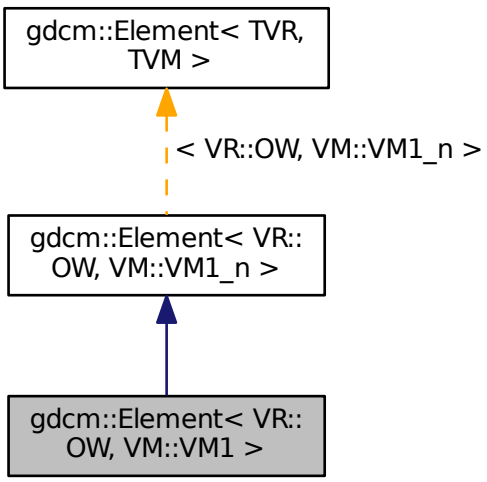
The documentation for this class was generated from the following file:

- gdcmElement.h

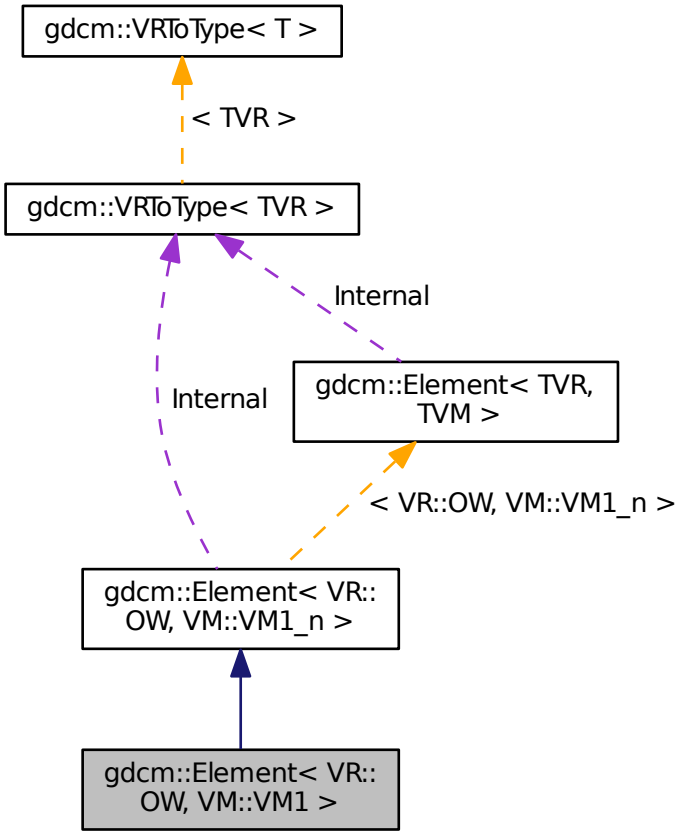
10.104 gdcm::Element< VR::OW, VM::VM1 > Class Template Reference

`#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

10.105 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>`

10.105.1 Detailed Description

template<int 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

10.106 gdcm::ElementDisableCombinations< VR::OB, VM::VM1_n > Class Template Reference

`#include <gdcmElement.h>`

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

- gdcmElement.h

10.107 gdcm::ElementDisableCombinations< VR::OW, VM::VM1_n > Class Template Reference

`#include <gdcmElement.h>`

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

- gdcmElement.h

10.108 gdcm::EncapsulatedDocument Class Reference

EncapsulatedDocument.

`#include <gdcmEncapsulatedDocument.h>`

Public Member Functions

- EncapsulatedDocument ()

10.108.1 Detailed Description

EncapsulatedDocument.

10.108.2 Constructor & Destructor Documentation

10.108.2.1 EncapsulatedDocument()

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

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

- gdcmEncapsulatedDocument.h

10.109 gdcm::EncodingImplementation< T > Class Template Reference

EncodingImplementation.

```
#include <gdcmElement.h>
```

10.109.1 Detailed Description

template<int T>
class gdcm::EncodingImplementation< T >

EncodingImplementation.

Note

TODO

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

- gdcmElement.h

10.110 gdcm::EncodingImplementation< VR::VRASCII > Class Template Reference

```
#include <gdcmElement.h>
```

Public Member Functions

- template<>
  void Write (const float *data, unsigned long length, std::ostream &_os)
- 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)

10.110.1 Member Function Documentation

10.110.1.1 Read()

```
template<typename T >
static void gdcm::EncodingImplementation< VR::VRASCII >::Read (
            T * data,
            unsigned long length,
            std::istream & _is )  [inline], [static]
```

10.110.1.2 ReadComputeLength()

```
template<typename T >
static void gdcm::EncodingImplementation< VR::VRASCII >::ReadComputeLength (
            T * data,
            unsigned int & length,
            std::istream & _is )  [inline], [static]
```

References gdcm::backslash().

10.110.1.3 ReadNoSwap()

```
template<typename T >
static void gdcm::EncodingImplementation< VR::VRASCII >::ReadNoSwap (
            T * data,
            unsigned long length,
            std::istream & _is )  [inline], [static]
```

10.110.1.4 Write() [1/3]

```
template<typename T >
static void gdcm::EncodingImplementation< VR::VRASCII >::Write (
             const T * data,
             unsigned long length,
             std::ostream & _os )  [inline], [static]
```

10.110.1.5 Write() [2/3]

```
template<>
void gdcm::EncodingImplementation< VR::VRASCII >::Write (
             const float * data,
             unsigned long length,
             std::ostream & _os )  [inline]
```

References gdcm::to_string().

10.110.1.6 Write() [3/3]

```
template<>
void gdcm::EncodingImplementation< VR::VRASCII >::Write (
             const double * data,
             unsigned long length,
             std::ostream & _os )  [inline]
```

References gdcm::to_string().

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

- gdcmElement.h

10.111 gdcm::EncodingImplementation< VR::VRBINARY > Class Template 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)

10.111.1 Member Function Documentation

10.111.1.1 Read()

```
template<typename T >
static void gdcm::EncodingImplementation< VR::VRBINARY >::Read (
             T * data,
             unsigned long length,
             std::istream & _is )  [inline], [static]
```

10.111.1.2 ReadComputeLength()

```
template<typename T >
static void gdcm::EncodingImplementation< VR::VRBINARY >::ReadComputeLength (
             T * data,
             unsigned int & length,
             std::istream & _is )  [inline], [static]
```

10.111.1.3 ReadNoSwap()

```
template<typename T >
static void gdcm::EncodingImplementation< VR::VRBINARY >::ReadNoSwap (
             T * data,
             unsigned long length,
             std::istream & _is )  [inline], [static]
```

10.111.1.4 Write()

```
template<typename T >
static void gdcm::EncodingImplementation< VR::VRBINARY >::Write (
             const T * data,
             unsigned long length,
             std::ostream & _os )  [inline], [static]
```

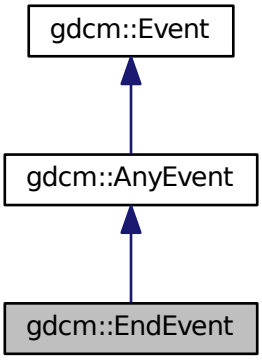
The documentation for this class was generated from the following file:

- gdcmElement.h

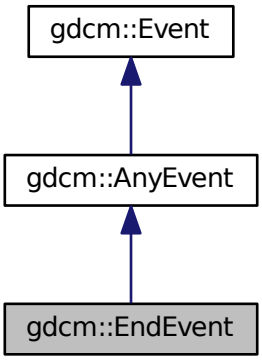
10.112 gdcm::EndEvent Class Reference

`#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

10.113 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 ()

10.113.1 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.

10.113.2 Constructor & Destructor Documentation

10.113.2.1 EnumeratedValues()

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

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

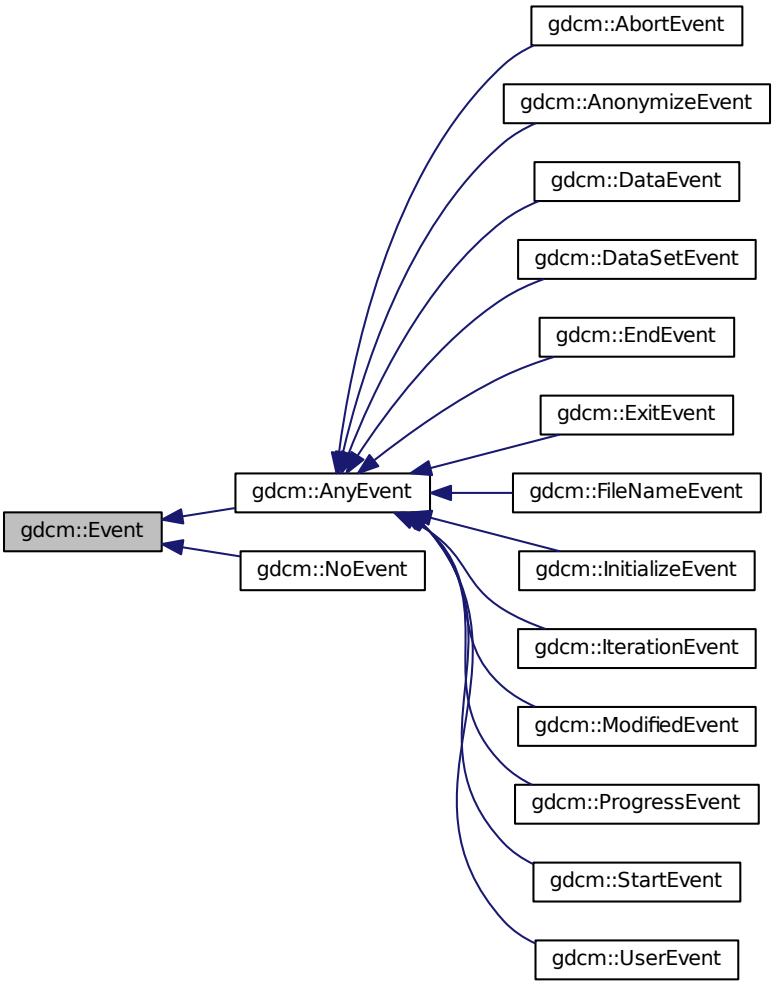
- gdcmEnumeratedValues.h

10.114 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 (void) const =0
- virtual Event * MakeObject () const =0
- virtual void Print (std::ostream &os) const

10.114.1 Detailed Description

superclass for callback/observer methods

See also

Command Subject

Examples:

SimpleScanner.cxx.

10.114.2 Constructor & Destructor Documentation

10.114.2.1 Event() [1/2]

```
gdcm::Event::Event ( )
```

10.114.2.2 Event() [2/2]

```
gdcm::Event::Event (
            const Event &  )
```

10.114.2.3 ~Event()

```
virtual gdcm::Event::~Event ( )  [virtual]
```

10.114.3 Member Function Documentation

10.114.3.1 CheckEvent()

```
virtual bool gdcm::Event::CheckEvent (
            const Event *  ) const  [pure virtual]
```

Check if given event matches or derives from this event.

10.114.3.2 GetEventName()

```
virtual const char* gdcm::Event::GetEventName (
            void  ) const  [pure virtual]
```

Return the StringName associated with the event.

Implemented in gdcm::FileNameEvent, gdcm::ProgressEvent, gdcm::DataSetEvent, gdcm::AnonymizeEvent, and gdcm::DataEvent.

10.114.3.3 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::FileNameEvent, gdcm::ProgressEvent, gdcm::DataSetEvent, gdcm::AnonymizeEvent, and gdcm::DataEvent.

10.114.3.4 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:

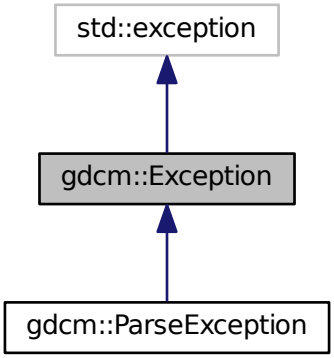
- gdcmEvent.h

10.115 gdcm::Exception Class Reference

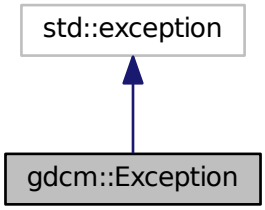
Exception.

```
#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="")
- virtual ~Exception () throw ()
- const char * GetDescription () const

  Return the Description.
- const char * what () const throw ()

  what implementation

10.115.1 Detailed Description

Exception.

Standard exception handling object.

Note

Its copy-constructor and assignment operator are generated by the compiler.

10.115.2 Constructor & Destructor Documentation

10.115.2.1 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.

10.115.2.2 ~Exception()

```
virtual gdcm::Exception::~Exception ( ) throw )   [inline], [virtual]
```

10.115.3 Member Function Documentation

10.115.3.1 GetDescription()

```
const char* gdcm::Exception::GetDescription ( ) const  [inline]
```

Return the Description.

Referenced by gdcm::SequenceOfItems::Read().

10.115.3.2 what()

```
const char* gdcm::Exception::what ( ) const throw )   [inline]
```

what implementation

Referenced by gdcm::SequenceOfFragments::ReadValue().

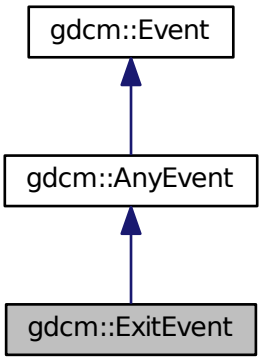
The documentation for this class was generated from the following file:

- gdcmException.h

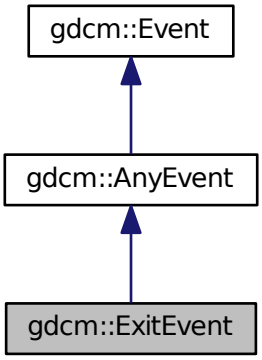
10.116 gdcm::ExitEvent Class Reference

```
#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:

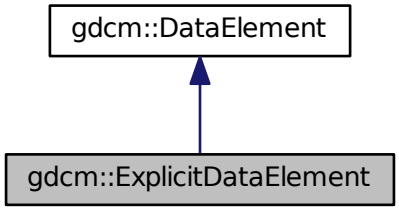
- gdcmEvent.h

10.117 gdcm::ExplicitDataElement Class Reference

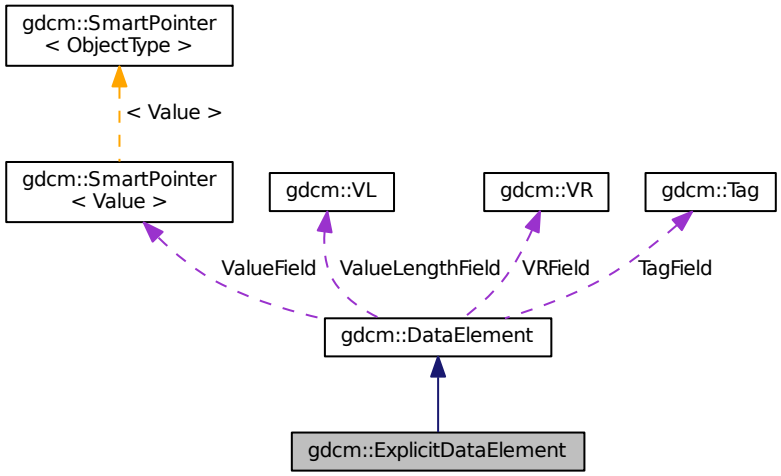
Class to read/write a DataElement as Explicit Data Element.

`#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

10.117.1 Detailed Description

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

Note

bla

10.117.2 Member Function Documentation

10.117.2.1 GetLength()

```
VL gdcm::ExplicitDataElement::GetLength ( ) const
```

10.117.2.2 Read()

```
template<typename TSwap >
std::istream& gdcm::ExplicitDataElement::Read (
             std::istream & is )
```

10.117.2.3 ReadPreValue()

```
template<typename TSwap >
std::istream& gdcm::ExplicitDataElement::ReadPreValue (
             std::istream & is )
```

10.117.2.4 ReadValue()

```
template<typename TSwap >
std::istream& gdcm::ExplicitDataElement::ReadValue (
             std::istream & is,
             bool readvalues = true )
```

10.117.2.5 ReadWithLength()

```
template<typename TSwap >
std::istream& gdcm::ExplicitDataElement::ReadWithLength (
             std::istream & is,
             VL & length )
```

10.117.2.6 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:

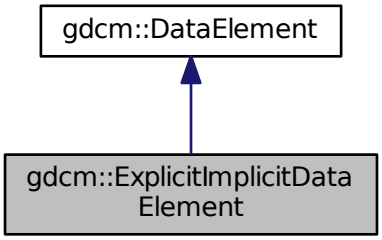
- gdcmExplicitDataElement.h

10.118 gdcm::ExplicitImplicitDataElement Class Reference

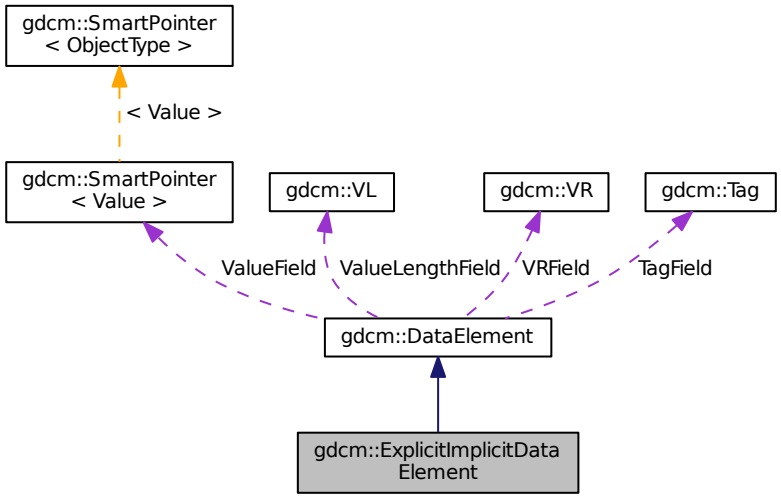
Class to read/write a DataElement as ExplicitImplicit Data Element.

`#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

10.118.1 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.

10.118.2 Member Function Documentation

10.118.2.1 GetLength()

```
VL gdcm::ExplicitImplicitDataElement::GetLength ( ) const
```

10.118.2.2 Read()

```
template<typename TSwap >
std::istream& gdcm::ExplicitImplicitDataElement::Read (
            std::istream & is )
```

10.118.2.3 ReadPreValue()

```
template<typename TSwap >
std::istream& gdcm::ExplicitImplicitDataElement::ReadPreValue (
            std::istream & is )
```

10.118.2.4 ReadValue()

```
template<typename TSwap >
std::istream& gdcm::ExplicitImplicitDataElement::ReadValue (
            std::istream & is,
            bool readvalues = true )
```

10.118.2.5 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

10.119 gdcm::Fiducials Class Reference

Fiducials.

```
#include <gdcmFiducials.h>
```

Public Member Functions

- Fiducials ()

10.119.1 Detailed Description

Fiducials.

10.119.2 Constructor & Destructor Documentation

10.119.2.1 Fiducials()

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

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

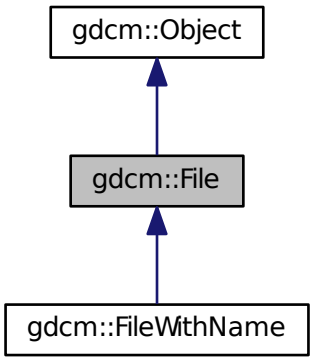
- gdcmFiducials.h

10.120 gdcm::File Class Reference

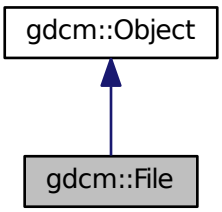
a DICOM File

`#include <gdcmFile.h>`

Inheritance diagram for gdcm::File:



Collaboration diagram for gdcm::File:



Public Member Functions

- File ()
- ~File ()
- const DataSet & GetDataSet () const

  Get Data Set.

- DataSet & GetDataSet ()

  Get Data Set.
- const FileMetaInformation & GetHeader () const

  Get File Meta Information.
- FileMetaInformation & GetHeader ()

  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

10.120.1 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, CreateFakeRTDOSE.cxx, CreateJPIPDataSet.cxx, DiffFile.cxx, DumpGEMSMovieGroup.cxx, DumpPhilipsECHO.cxx, DuplicatePCDE.cxx, EncapsulateFileInRawData.cxx, ExtractEncryptedContent.cxx, Extracting_All_Resolution.cxx, Fake_Image_Using_Stream_Image_Writer.cxx, FixBrokenJ2K.cxx, FixOrientation.cxx, GenAllVR.cxx, GenFakeIdentifyFile.cxx, GenFakeImage.cxx, GenLongSeqs.cxx, GenSeqs.cxx, GetJPEGSamplePrecision.cxx, GetSequenceUltrasound.cxx, GetSubSequenceData.cxx, HelloWorld.cxx, iU22tomultisc.cxx, LargeVRDSExplicit.cxx, MakeTemplate.cxx, PatchFile.cxx, QIDO-RS.cxx, ReadAndDumpDICOMDIR.cxx, ReadAndPrintAttributes.cxx, ReadGEMSSDO.cxx, and StreamImageReaderTest.cxx.

10.120.2 Constructor & Destructor Documentation

10.120.2.1 File()

```
gdcm::File::File ( )
```

10.120.2.2 ~File()

```
gdcm::File::~File ( )
```

10.120.3 Member Function Documentation

10.120.3.1 GetDataSet() [1/2]

```
const DataSet& gdcm::File::GetDataSet ( ) const  [inline]
```

Get Data Set.

Examples:

ChangePrivateTags.cxx, ChangeSequenceUltrasound.cxx, CreateFakeRTDOSE.cxx, CreateJPIPDataSet.cxx, csa2img.cxx, DiffFile.cxx, DumpADAC.cxx, DumpExamCard.cxx, DumpGEMSMovieGroup.cxx, DumpImageHeaderInfo.cxx, DumpPhilipsECHO.cxx, DumpToshibaDTI.cxx, DuplicatePCDE.cxx, ELSCINT1WaveToText.cxx, ExtractEncryptedContent.cxx, Extracting_All_Resolution.cxx, Fake_Image_Using_Stream_Image_Writer.cxx, FixBrokenJ2K.cxx, FixJAIBugJPEGLS.cxx, FixOrientation.cxx, gdcmrtionplan.cxx, gdcmrtplan.cxx, GenAllVR.cxx, GenFakeIdentifyFile.cxx, GenLongSeqs.cxx, GenSeqs.cxx, GetJPEGSamplePrecision.cxx, GetSequenceUltrasound.cxx, GetSubSequenceData.cxx, HelloWorld.cxx, iU22tomultisc.cxx, LargeVRDSExplicit.cxx, MergeTwoFiles.cxx, MrProtocol.cxx, PatchFile.cxx, pmsct_rgb1.cxx, QIDO-RS.cxx, ReadAndDumpDICOMDIR.cxx, ReadAndPrintAttributes.cxx, ReadExplicitLengthSQIVR.cxx, ReadGEMSSDO.cxx, rle2img.cxx, and StreamImageReaderTest.cxx.

10.120.3.2 GetDataSet() [2/2]

```
DataSet& gdcm::File::GetDataSet ( )  [inline]
```

Get Data Set.

10.120.3.3 GetHeader() [1/2]

```
const FileMetaInformation& gdcm::File::GetHeader ( ) const  [inline]
```

Get File Meta Information.

Examples:

CreateJPIPDataSet.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, pmsct_rgb1.cxx, QIDO-RS.cxx, ReadAndDumpDICOMDIR.cxx, rle2img.cxx, and StreamImageReaderTest.cxx.

Referenced by gdcm::operator<<().

10.120.3.4 GetHeader() [2/2]

```
FileMetaInformation& gdcm::File::GetHeader ( )  [inline]
```

Get File Meta Information.

10.120.3.5 Read()

```
std::istream& gdcm::File::Read (
            std::istream & is )
```

Read.

10.120.3.6 SetDataSet()

```
void gdcm::File::SetDataSet (
            const DataSet & ds )  [inline]
```

Set Data Set.

10.120.3.7 SetHeader()

```
void gdcm::File::SetHeader (
            const FileMetaInformation & fmi )  [inline]
```

Set File Meta Information.

10.120.3.8 Write()

```
std::ostream const& gdcm::File::Write (
            std::ostream & os ) const
```

Write.

10.120.4 Friends And Related Function Documentation

10.120.4.1 operator<<

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

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

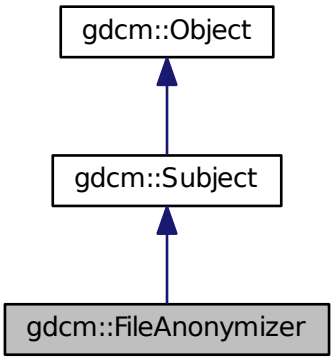
- gdcmFile.h

10.121 gdcm::FileAnonymizer Class Reference

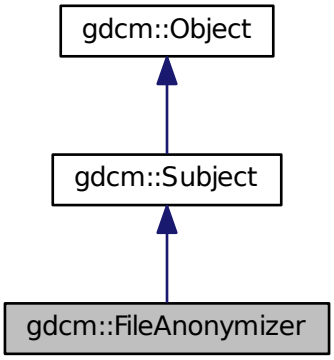
FileAnonymizer.

```
#include <gdcmFileAnonymizer.h>
```

Inheritance diagram for gdcm::FileAnonymizer:



Collaboration diagram for gdcm::FileAnonymizer:



Public Member Functions

- FileAnonymizer ()
- ~FileAnonymizer ()
- 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_str)
- void Replace (Tag const &t, const char *value_data, VL const &vl)
- 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

10.121.1 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:

MakeTemplate.cxx.

10.121.2 Constructor & Destructor Documentation

10.121.2.1 FileAnonymizer()

```
gdcm::FileAnonymizer::FileAnonymizer ( )
```

10.121.2.2 ~FileAnonymizer()

```
gdcm::FileAnonymizer::~FileAnonymizer ( )
```

10.121.3 Member Function Documentation

10.121.3.1 Empty()

```
void gdcm::FileAnonymizer::Empty (
            Tag const & t )
```

Make Tag t empty Warning: does not handle SQ element

Examples:

MakeTemplate.cxx.

10.121.3.2 Remove()

```
void gdcm::FileAnonymizer::Remove (
            Tag const & t )
```

remove a tag (even a SQ can be removed)

10.121.3.3 Replace() [1/2]

```
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 !

10.121.3.4 Replace() [2/2]

```
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

10.121.3.5 SetInputFileName()

```
void gdcm::FileAnonymizer::SetInputFileName (
            const char * filename_native )
```

Set input filename.

Examples:

FileAnonymize.cs, and MakeTemplate.cxx.

10.121.3.6 SetOutputFileName()

```
void gdcm::FileAnonymizer::SetOutputFileName (
            const char * filename_native )
```

Set output filename.

Examples:

MakeTemplate.cxx.

10.121.3.7 Write()

```
bool gdcm::FileAnonymizer::Write ( )
```

Write the output file.

Examples:

MakeTemplate.cxx.

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

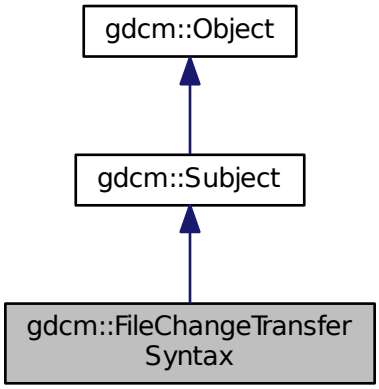
- gdcmFileAnonymizer.h

10.122 gdcm::FileChangeTransferSyntax Class Reference

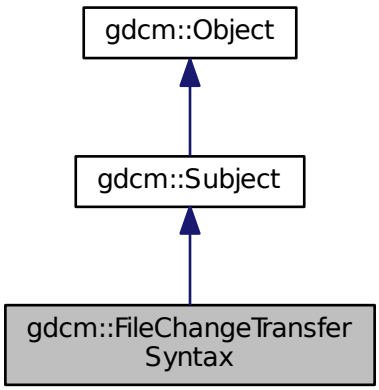
FileChangeTransferSyntax.

```
#include <gdcmFileChangeTransferSyntax.h>
```

Inheritance diagram for gdcm::FileChangeTransferSyntax:



Collaboration diagram for gdcm::FileChangeTransferSyntax:



Public Member Functions

- FileChangeTransferSyntax ()
- ~FileChangeTransferSyntax ()
- 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

10.122.1 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

10.122.2 Constructor & Destructor Documentation

10.122.2.1 FileChangeTransferSyntax()

```
gdcm::FileChangeTransferSyntax::FileChangeTransferSyntax ( )
```

10.122.2.2 ~FileChangeTransferSyntax()

```
gdcm::FileChangeTransferSyntax::~FileChangeTransferSyntax ( )
```

10.122.3 Member Function Documentation

10.122.3.1 Change()

```
bool gdcm::FileChangeTransferSyntax::Change ( )
```

Change the transfer syntax.

10.122.3.2 GetCodec()

```
ImageCodec* gdcm::FileChangeTransferSyntax::GetCodec ( )
```

Retrieve the actual codec (valid after calling SetTransferSyntax) Only advanced users should call this function.

10.122.3.3 New()

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

for wrapped language: instantiate a reference counted object

10.122.3.4 SetInputFileName()

```
void gdcm::FileChangeTransferSyntax::SetInputFileName (
            const char * filename_native )
```

Set input filename (raw DICOM)

10.122.3.5 SetOutputFileName()

```
void gdcm::FileChangeTransferSyntax::SetOutputFileName (
            const char * filename_native )
```

Set output filename (target compressed DICOM)

10.122.3.6 SetTransferSyntax()

```
void gdcm::FileChangeTransferSyntax::SetTransferSyntax (
            TransferSyntax const & ts )
```

Specify the Target Transfer Syntax.

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

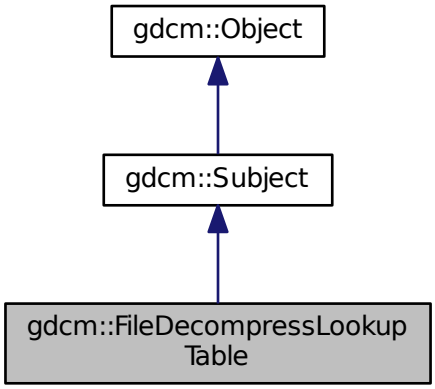
- gdcmFileChangeTransferSyntax.h

10.123 gdcm::FileDecompressLookupTable Class Reference

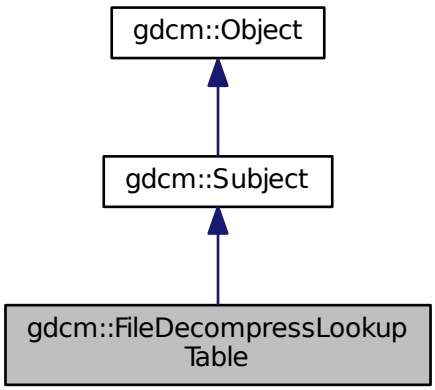
FileDecompressLookupTable class.

```
#include <gdcmFileDecompressLookupTable.h>
```

Inheritance diagram for gdcm::FileDecompressLookupTable:



Collaboration diagram for gdcm::FileDecompressLookupTable:



Public Member Functions

- FileDecompressLookupTable ()
- ~FileDecompressLookupTable ()
- bool Change ()

  Decompress.
- File & GetFile ()
- const Pixmap & GetPixmap () const
- Pixmap & GetPixmap ()
- void SetFile (const File &f)

  Set/Get File.
- void SetPixmap (Pixmap const &img)

Additional Inherited Members

10.123.1 Detailed Description

FileDecompressLookupTable class.

It decompress the segmented LUT into linearized one (only PALETTE_COLOR images) Output will be a PhotometricInterpretation=RGB image

10.123.2 Constructor & Destructor Documentation

10.123.2.1 FileDecompressLookupTable()

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

10.123.2.2 ~FileDecompressLookupTable()

```
gdcm::FileDecompressLookupTable::~FileDecompressLookupTable ( )  [inline]
```

10.123.3 Member Function Documentation

10.123.3.1 Change()

```
bool gdcm::FileDecompressLookupTable::Change ( )
```

Decompress.

10.123.3.2 GetFile()

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

10.123.3.3 GetPixmap() [1/2]

```
const Pixmap& gdcm::FileDecompressLookupTable::GetPixmap ( ) const  [inline]
```

10.123.3.4 GetPixmap() [2/2]

```
Pixmap& gdcm::FileDecompressLookupTable::GetPixmap ( )  [inline]
```

10.123.3.5 SetFile()

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

Set/Get File.

10.123.3.6 SetPixmap()

```
void gdcm::FileDecompressLookupTable::SetPixmap (
            Pixmap const & img )  [inline]
```

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

- gdcmFileDecompressLookupTable.h

10.124 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 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 ()

10.124.1 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 compulsary when creating a lossy derived image.

Examples:

GenFakeImage.cxx.

10.124.2 Constructor & Destructor Documentation

10.124.2.1 FileDerivation()

```
gdcm::FileDerivation::FileDerivation ( )
```

10.124.2.2 ~FileDerivation()

```
gdcm::FileDerivation::~FileDerivation ( )
```

10.124.3 Member Function Documentation

10.124.3.1 AddDerivationDescription()

```
bool gdcm::FileDerivation::AddDerivationDescription ( )  [protected]
```

10.124.3.2 AddPurposeOfReferenceCodeSequence()

```
bool gdcm::FileDerivation::AddPurposeOfReferenceCodeSequence (
            DataSet & ds )  [protected]
```

10.124.3.3 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:

GenFakeImage.cxx.

10.124.3.4 AddSourceImageSequence()

```
bool gdcm::FileDerivation::AddSourceImageSequence ( )  [protected]
```

10.124.3.5 Derive()

```
bool gdcm::FileDerivation::Derive ( )
```

Change.

Examples:

GenFakeImage.cxx.

10.124.3.6 GetFile() [1/2]

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

Examples:

GenFakeImage.cxx.

10.124.3.7 GetFile() [2/2]

```
const File& gdcm::FileDerivation::GetFile ( ) const  [inline]
```

10.124.3.8 SetDerivationCodeSequenceCodeValue()

```
void gdcm::FileDerivation::SetDerivationCodeSequenceCodeValue (
            unsigned int codevalue )
```

Specify the Derivation Code Sequence Code Value. Eg 113040.

Examples:

GenFakeImage.cxx.

10.124.3.9 SetDerivationDescription()

```
void gdcm::FileDerivation::SetDerivationDescription (
            const char * dd )
```

Specify the Derivation Description. Eg "lossy conversion".

10.124.3.10 SetFile()

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

Set/Get File.

Examples:

GenFakeImage.cxx.

10.124.3.11 SetPurposeOfReferenceCodeSequenceCodeValue()

```
void gdcm::FileDerivation::SetPurposeOfReferenceCodeSequenceCodeValue (
            unsigned int codevalue )
```

Specify the Purpose Of Reference Code Value. Eg. 121320.

Examples:

GenFakeImage.cxx.

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

- gdcmFileDerivation.h

10.125 gdcm::FileExplicitFilter Class Reference

FileExplicitFilter class.

`#include <gdcmFileExplicitFilter.h>`

Public Member Functions

- FileExplicitFilter ()
- ~FileExplicitFilter ()
- 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)

10.125.1 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 reprensentation, 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.

10.125.2 Constructor & Destructor Documentation

10.125.2.1 FileExplicitFilter()

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

10.125.2.2 ~FileExplicitFilter()

```
gdcm::FileExplicitFilter::~FileExplicitFilter ( )  [inline]
```

10.125.3 Member Function Documentation

10.125.3.1 Change()

```
bool gdcm::FileExplicitFilter::Change ( )
```

Set FMI Transfer Syntax.

Change

Examples:

GenAllVR.cxx, and LargeVRDSExplicit.cxx.

10.125.3.2 ChangeFMI()

```
bool gdcm::FileExplicitFilter::ChangeFMI ( )  [protected]
```

10.125.3.3 GetFile()

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

10.125.3.4 ProcessDataSet()

```
bool gdcm::FileExplicitFilter::ProcessDataSet (
            DataSet & ds,
            Dicts const & dicts )  [protected]
```

10.125.3.5 SetChangePrivateTags()

```
void gdcm::FileExplicitFilter::SetChangePrivateTags (
            bool b )  [inline]
```

Decide whether or not to VR'ify private tags.

10.125.3.6 SetFile()

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

Set/Get File.

Examples:

GenAllVR.cxx, and LargeVRDSExplicit.cxx.

10.125.3.7 SetRecomputeItemLength()

```
void gdcm::FileExplicitFilter::SetRecomputeItemLength (
            bool b )
```

By default set Sequence & Item length to Undefined to avoid recomputing length:

10.125.3.8 SetRecomputeSequenceLength()

```
void gdcm::FileExplicitFilter::SetRecomputeSequenceLength (
            bool b )
```

10.125.3.9 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:

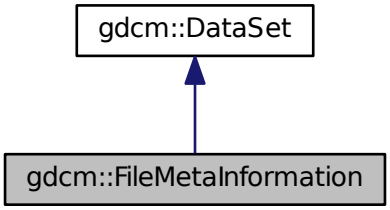
- gdcmFileExplicitFilter.h

10.126 gdcm::FileMetaInformation Class Reference

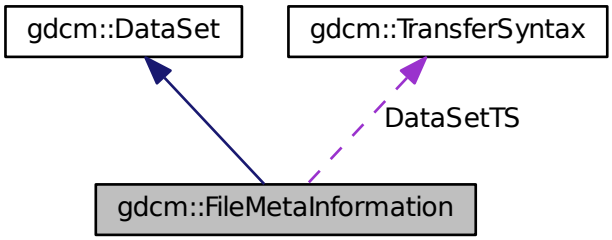
Class to represent a File Meta Information.

`#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
- const Preamble & GetPreamble () const

  Get Preamble.
- Preamble & GetPreamble ()
- void Insert (const DataElement &de)
- bool IsValid () const
- 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

10.126.1 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:

GenAllVR.cxx, GenFakeIdentifyFile.cxx, LargeVRDSExplicit.cxx, MakeTemplate.cxx, and ReadAndDumpDICOMDIR.cxx.

10.126.2 Constructor & Destructor Documentation

10.126.2.1 FileMetaInformation() [1/2]

```
gdcm::FileMetaInformation::FileMetaInformation ( )
```

10.126.2.2 ~FileMetaInformation()

```
gdcm::FileMetaInformation::~FileMetaInformation ( )
```

10.126.2.3 FileMetaInformation() [2/2]

```
gdcm::FileMetaInformation::FileMetaInformation (
            FileMetaInformation const & fmi )  [inline]
```

References DataSetMS, DataSetTS, and MetaInformationTS.

10.126.3 Member Function Documentation

10.126.3.1 AppendImplementationClassUID()

```
static void gdcm::FileMetaInformation::AppendImplementationClassUID (
            const char * imp )  [static]
```

10.126.3.2 ComputeDataSetMediaStorageSOPClass()

```
void gdcm::FileMetaInformation::ComputeDataSetMediaStorageSOPClass ( )  [protected]
```

10.126.3.3 ComputeDataSetTransferSyntax()

```
void gdcm::FileMetaInformation::ComputeDataSetTransferSyntax ( )  [protected]
```

10.126.3.4 Default()

```
void gdcm::FileMetaInformation::Default ( )  [protected]
```

10.126.3.5 FillFromDataSet()

```
void gdcm::FileMetaInformation::FillFromDataSet (
            DataSet const & ds )
```

Construct a FileMetaInformation from an already existing DataSet:

10.126.3.6 GetDataSetTransferSyntax()

```
const TransferSyntax& gdcm::FileMetaInformation::GetDataSetTransferSyntax ( ) const  [inline]
```

Examples:

GetJPEGSamplePrecision.cxx, and MergeTwoFiles.cxx.

10.126.3.7 GetFileMetaInformationVersion()

```
static const char* gdcm::FileMetaInformation::GetFileMetaInformationVersion ( )  [static], [protected]
```

10.126.3.8 GetFullLength()

```
VL gdcm::FileMetaInformation::GetFullLength ( ) const  [inline]
```

References gdcm::VL::GetLength().

10.126.3.9 GetGDCMImplementationClassUID()

```
static const char* gdcm::FileMetaInformation::GetGDCMImplementationClassUID ( )  [static], [protected]
```

10.126.3.10 GetGDCMImplementationVersionName()

```
static const char* gdcm::FileMetaInformation::GetGDCMImplementationVersionName ( )  [static],
[protected]
```

10.126.3.11 GetGDCMSourceApplicationEntityTitle()

```
static const char* gdcm::FileMetaInformation::GetGDCMSourceApplicationEntityTitle ( )  [static],
[protected]
```

10.126.3.12 GetImplementationClassUID()

```
static const char* gdcm::FileMetaInformation::GetImplementationClassUID ( )  [static]
```

10.126.3.13 GetImplementationVersionName()

```
static const char* gdcm::FileMetaInformation::GetImplementationVersionName ( )  [static]
```

10.126.3.14 GetMediaStorage()

```
MediaStorage gdcm::FileMetaInformation::GetMediaStorage ( ) const
```

10.126.3.15 GetMediaStorageAsString()

```
std::string gdcm::FileMetaInformation::GetMediaStorageAsString ( ) const
```

10.126.3.16 GetMetaInformationTS()

```
TransferSyntax::NegociatedType gdcm::FileMetaInformation::GetMetaInformationTS ( ) const  [inline]
```

10.126.3.17 GetPreamble() [1/2]

```
const Preamble& gdcm::FileMetaInformation::GetPreamble ( ) const  [inline]
```

Get Preamble.

Referenced by gdcm::operator<<().

10.126.3.18 GetPreamble() [2/2]

```
Preamble& gdcm::FileMetaInformation::GetPreamble ( )  [inline]
```

10.126.3.19 GetSourceApplicationEntityTitle()

```
static const char* gdcm::FileMetaInformation::GetSourceApplicationEntityTitle ( )  [static]
```

10.126.3.20 Insert()

```
void gdcm::FileMetaInformation::Insert (
            const DataElement & de )  [inline]
```

References gdcmErrorMacro, gdcm::Tag::GetGroup(), and gdcm::DataElement::GetTag().

10.126.3.21 IsValid()

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

10.126.3.22 Read()

```
std::istream& gdcm::FileMetaInformation::Read (
            std::istream & is )
```

Read.

10.126.3.23 ReadCompat()

```
std::istream& gdcm::FileMetaInformation::ReadCompat (
            std::istream & is )
```

10.126.3.24 ReadCompatInternal()

```
template<typename TSwap >
std::istream& gdcm::FileMetaInformation::ReadCompatInternal (
            std::istream & is )  [protected]
```

10.126.3.25 Replace()

```
void gdcm::FileMetaInformation::Replace (
            const DataElement & de )  [inline]
```

Examples:

LargeVRDSExplicit.cxx.

References gdcm::DataElement::GetTag().

10.126.3.26 SetDataSetTransferSyntax()

```
void gdcm::FileMetaInformation::SetDataSetTransferSyntax (
            const TransferSyntax & ts )
```

Examples:

CreateJPIPDataSet.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, pmsct_rgb1.cxx, QIDO-RS.cxx, rle2img.cxx, and StreamImageReaderTest.cxx.

10.126.3.27 SetImplementationClassUID()

```
static void gdcm::FileMetaInformation::SetImplementationClassUID (
            const char * imp )  [static]
```

Override the GDCM default values:

10.126.3.28 SetImplementationVersionName()

```
static void gdcm::FileMetaInformation::SetImplementationVersionName (
            const char * version )  [static]
```

10.126.3.29 SetPreamble()

```
void gdcm::FileMetaInformation::SetPreamble (
            const Preamble & p )  [inline]
```

10.126.3.30 SetSourceApplicationEntityTitle()

```
static void gdcm::FileMetaInformation::SetSourceApplicationEntityTitle (
            const char * title )  [static]
```

Examples:

ClinicalTrialIdentificationWorkflow.cs, FixJAIBugJPEGLS.cxx, GenerateDICOMDIR.cs, ReformatFile.cs, and StandardizeFiles.cs.

10.126.3.31 Write()

```
std::ostream& gdcm::FileMetaInformation::Write (
            std::ostream & os ) const
```

Write.

10.126.4 Friends And Related Function Documentation

10.126.4.1 operator<<

```
std::ostream& operator<< (
            std::ostream & _os,
            const FileMetaInformation & _val )  [friend]
```

10.126.5 Member Data Documentation

10.126.5.1 DataSetMS

```
MediaStorage::MSType gdcm::FileMetaInformation::DataSetMS  [protected]
```

Referenced by FileMetaInformation().

10.126.5.2 DataSetTS

```
TransferSyntax gdcm::FileMetaInformation::DataSetTS  [protected]
```

Referenced by FileMetaInformation().

10.126.5.3 MetaInformationTS

```
TransferSyntax::NegociatedType gdcm::FileMetaInformation::MetaInformationTS  [protected]
```

Referenced by FileMetaInformation().

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

- gdcmFileMetaInformation.h

10.127 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)

10.127.1 Detailed Description

Class to manipulate file name's.

Note

OS independant representation of a filename (to query path, name and extension from a filename)

10.127.2 Constructor & Destructor Documentation

10.127.2.1 Filename()

```
gdcm::Filename::Filename (
            const char * filename = "" )  [inline]
```

10.127.3 Member Function Documentation

10.127.3.1 EndWith()

```
bool gdcm::Filename::EndWith (
            const char ending[] ) const
```

Does the filename ends with a particular string ?

10.127.3.2 GetExtension()

```
const char* gdcm::Filename::GetExtension ( )
```

return only the extension part of a filename

10.127.3.3 GetFileName()

```
const char* gdcm::Filename::GetFileName ( ) const  [inline]
```

Return the full filename.

10.127.3.4 GetName()

```
const char* gdcm::Filename::GetName ( )
```

return only the name part of a filename

10.127.3.5 GetPath()

```
const char* gdcm::Filename::GetPath ( )
```

Return only the path component of a filename.

Examples:

ClinicalTrialIdentificationWorkflow.cs.

10.127.3.6 IsEmpty()

```
bool gdcm::Filename::IsEmpty ( ) const  [inline]
```

return whether the filename is empty

10.127.3.7 IsIdentical()

```
bool gdcm::Filename::IsIdentical (
            Filename const & fn ) const
```

10.127.3.8 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.

10.127.3.9 operator const char *()

```
gdcm::Filename::operator const char * ( ) const  [inline]
```

Simple operator to allow Filename myfilename("..."); const char * s = myfilename;

10.127.3.10 ToUnixSlashes()

```
const char* gdcm::Filename::ToUnixSlashes ( )
```

Convert backslash (windows style) to UNIX style slash.

10.127.3.11 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:

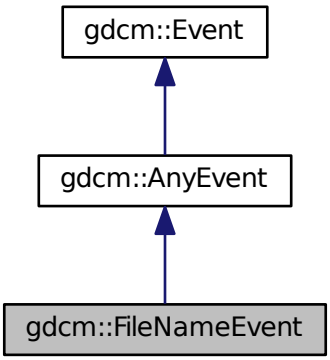
- gdcmFilename.h

10.128 gdcm::FileNameEvent Class Reference

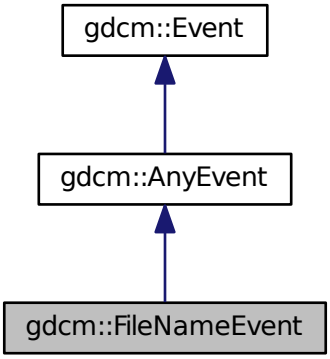
FileNameEvent.

```
#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)
- virtual ~FileNameEvent ()
- virtual bool CheckEvent (const ::gdcm::Event *e) const
- virtual const char * GetEventName () const
- const char * GetFileName () const
- virtual ::gdcm::Event * MakeObject () const
- void SetFileName (const char *f)

10.128.1 Detailed Description

FileNameEvent.

Special type of event triggered during processing of FileSet

See also

AnyEvent

Examples:

SimpleScanner.cxx.

10.128.2 Member Typedef Documentation

10.128.2.1 Self

```
typedef FileNameEvent gdcm::FileNameEvent::Self
```

10.128.2.2 Superclass

```
typedef AnyEvent gdcm::FileNameEvent::Superclass
```

10.128.3 Constructor & Destructor Documentation

10.128.3.1 FileNameEvent() [1/2]

```
gdcm::FileNameEvent::FileNameEvent (
            const char * s = "" )  [inline]
```

10.128.3.2 ~FileNameEvent()

```
virtual gdcm::FileNameEvent::~FileNameEvent ( )  [inline], [virtual]
```

10.128.3.3 FileNameEvent() [2/2]

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

10.128.4 Member Function Documentation

10.128.4.1 CheckEvent()

```
virtual bool gdcm::FileNameEvent::CheckEvent (
            const ::gdcm::Event * e ) const  [inline], [virtual]
```

10.128.4.2 GetEventName()

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

Return the StringName associated with the event.

Implements gdcm::Event.

10.128.4.3 GetFileName()

```
const char* gdcm::FileNameEvent::GetFileName ( ) const  [inline]
```

Examples:

SimpleScanner.cxx.

10.128.4.4 MakeObject()

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

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

Implements gdcm::Event.

10.128.4.5 SetFileName()

```
void gdcm::FileNameEvent::SetFileName (
            const char * f )  [inline]
```

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

- gdcmFileNameEvent.h

10.129 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 ()
- 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.

10.129.1 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.

10.129.2 Member Typedef Documentation

10.129.2.1 FilenamesType

```
typedef std::vector<FilenameType> gdcm::FilenameGenerator::FilenamesType
```

10.129.2.2 FilenameType

```
typedef std::string gdcm::FilenameGenerator::FilenameType
```

10.129.2.3 SizeType

```
typedef FilenamesType::size_type gdcm::FilenameGenerator::SizeType
```

10.129.3 Constructor & Destructor Documentation

10.129.3.1 FilenameGenerator()

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

10.129.3.2 ~FilenameGenerator()

```
gdcm::FilenameGenerator::~FilenameGenerator ( )  [inline]
```

10.129.4 Member Function Documentation

10.129.4.1 Generate()

```
bool gdcm::FilenameGenerator::Generate ( )
```

Generate (return success)

Examples:

ConvertMultiFrameToSingleFrame.cxx, and CreateFakePET.cxx.

10.129.4.2 GetFilename()

```
const char* gdcm::FilenameGenerator::GetFilename (
            SizeType n ) const
```

Get a particular filename (call after Generate)

Examples:

ConvertMultiFrameToSingleFrame.cxx, and CreateFakePET.cxx.

10.129.4.3 GetFilenames()

```
FilenamesType const& gdcm::FilenameGenerator::GetFilenames ( ) const  [inline]
```

Return all filenames.

10.129.4.4 GetNumberOfFilenames()

```
SizeType gdcm::FilenameGenerator::GetNumberOfFilenames ( ) const
```

Examples:

ConvertMultiFrameToSingleFrame.cxx, and CreateFakePET.cxx.

10.129.4.5 GetPattern()

```
const char* gdcm::FilenameGenerator::GetPattern ( ) const  [inline]
```

10.129.4.6 GetPrefix()

```
const char* gdcm::FilenameGenerator::GetPrefix ( ) const  [inline]
```

10.129.4.7 SetNumberOfFilenames()

```
void gdcm::FilenameGenerator::SetNumberOfFilenames (
            SizeType nfiles )
```

Set/Get the number of filenames to generate.

Examples:

ConvertMultiFrameToSingleFrame.cxx, and CreateFakePET.cxx.

10.129.4.8 SetPattern()

```
void gdcm::FilenameGenerator::SetPattern (
            const char * pattern )  [inline]
```

Set/Get pattern.

Examples:

ConvertMultiFrameToSingleFrame.cxx, and CreateFakePET.cxx.

10.129.4.9 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

10.130 gdcm::FileSet Class Reference

```
#include <gdcmFileSet.h>
```

Public Types

- typedef std::vector< FileType > FilesType
- typedef std::string FileType

Public Member Functions

- FileSet ()
- void AddFile (File const &)
- bool AddFile (const char *filename)
- FilesType const & GetFiles () const
- void SetFiles (FilesType const &files)

Friends

- std::ostream & operator<< (std::ostream &_os, const FileSet &d)

10.130.1 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.

10.130.2 Member Typedef Documentation

10.130.2.1 FilesType

```
typedef std::vector<FileType> gdcm::FileSet::FilesType
```

10.130.2.2 FileType

```
typedef std::string gdcm::FileSet::FileType
```

10.130.3 Constructor & Destructor Documentation

10.130.3.1 FileSet()

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

10.130.4 Member Function Documentation

10.130.4.1 AddFile() [1/2]

```
void gdcm::FileSet::AddFile (
            File const &  )  [inline]
```

Deprecated . Does nothing

10.130.4.2 AddFile() [2/2]

```
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.

10.130.4.3 GetFiles()

```
FilesType const& gdcm::FileSet::GetFiles ( ) const  [inline]
```

10.130.4.4 SetFiles()

```
void gdcm::FileSet::SetFiles (
            FilesType const & files )
```

10.130.5 Friends And Related Function Documentation

10.130.5.1 operator<<

```
std::ostream& operator<< (
            std::ostream & _os,
            const FileSet & d )  [friend]
```

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

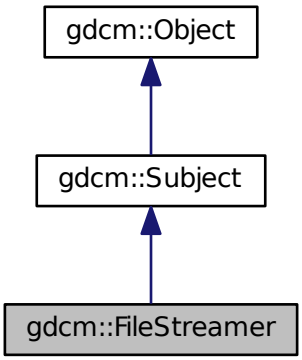
- gdcmFileSet.h

10.131 gdcm::FileStreamer Class Reference

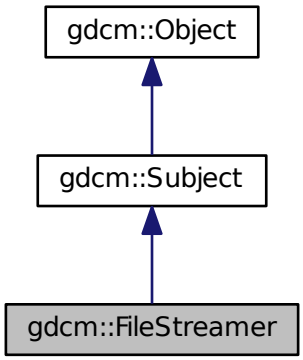
FileStreamer.

```
#include <gdcmFileStreamer.h>
```

Inheritance diagram for gdcm::FileStreamer:



Collaboration diagram for gdcm::FileStreamer:



Public Member Functions

- FileStreamer ()
- ~FileStreamer ()
- 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

10.131.1 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.

10.131.2 Constructor & Destructor Documentation

10.131.2.1 FileStreamer()

```
gdcm::FileStreamer::FileStreamer ( )
```

10.131.2.2 ~FileStreamer()

```
gdcm::FileStreamer::~FileStreamer ( )
```

10.131.3 Member Function Documentation

10.131.3.1 AppendToDataElement()

```
bool gdcm::FileStreamer::AppendToDataElement (
            const Tag & t,
            const char * array,
            size_t len )
```

Append to previously started Tag t.

10.131.3.2 AppendToGroupDataElement()

```
bool gdcm::FileStreamer::AppendToGroupDataElement (
            const PrivateTag & pt,
            const char * array,
            size_t len )
```

Append to previously started private creator.

10.131.3.3 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

10.131.3.4 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.

10.131.3.5 New()

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

for wrapped language: instantiate a reference counted object

10.131.3.6 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

10.131.3.7 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

10.131.3.8 SetOutputFileName()

```
void gdcm::FileStreamer::SetOutputFileName (
            const char * filename_native )
```

Set output filename (target file)

10.131.3.9 SetTemplateFileName()

```
void gdcm::FileStreamer::SetTemplateFileName (
            const char * filename_native )
```

Set input DICOM template filename.

Examples:

FileStreaming.cs.

10.131.3.10 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.

10.131.3.11 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.

10.131.3.12 StopDataElement()

```
bool gdcm::FileStreamer::StopDataElement (
            const Tag & t )
```

Stop appending to tag t. This will compute the proper attribute length.

10.131.3.13 StopGroupDataElement()

```
bool gdcm::FileStreamer::StopGroupDataElement (
            const PrivateTag & pt )
```

Stop appending to private creator.

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

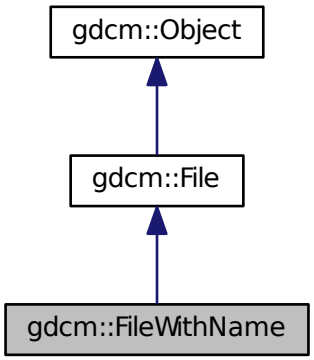
- gdcmFileStreamer.h

10.132 gdcm::FileWithName Class Reference

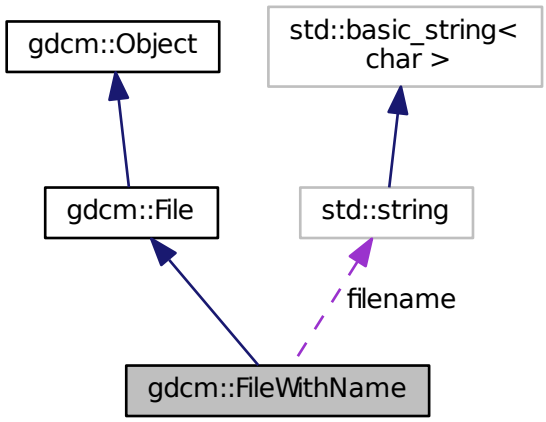
FileWithName.

`#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

10.132.1 Detailed Description

FileWithName.

Backward only class do not use in newer code

10.132.2 Constructor & Destructor Documentation

10.132.2.1 FileWithName()

```
gdcm::FileWithName::FileWithName (
            File & f )  [inline]
```

10.132.3 Member Data Documentation

10.132.3.1 filename

```
std::string gdcm::FileWithName::filename
```

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

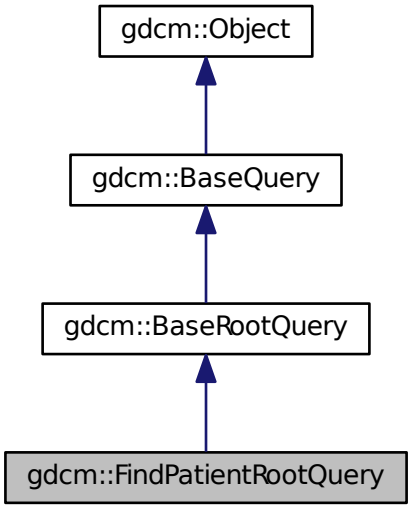
- gdcmSerieHelper.h

10.133 gdcm::FindPatientRootQuery Class Reference

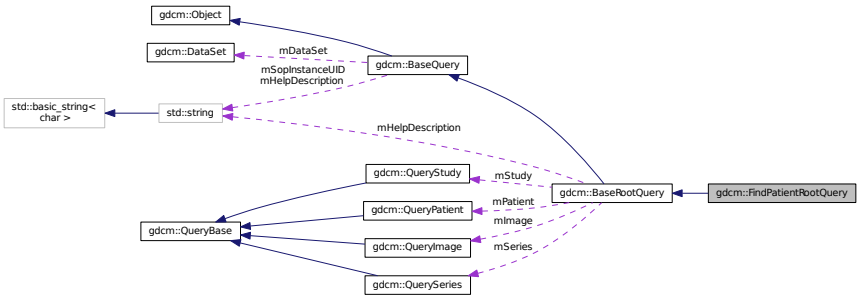
PatientRootQuery.

```
#include <gdcmFindPatientRootQuery.h>
```

Inheritance diagram for gdcm::FindPatientRootQuery:



Collaboration diagram for gdcm::FindPatientRootQuery:



Public Member Functions

- FindPatientRootQuery ()
- UIDs::TSName GetAbstractSyntaxUID () const
- std::vector< Tag > GetTagListByLevel (const EQueryLevel &inQueryLevel)
- void InitializeDataSet (const EQueryLevel &inQueryLevel)
- bool ValidateQuery (bool inStrict=true) const

Friends

- class QueryFactory

Additional Inherited Members

10.133.1 Detailed Description

PatientRootQuery.

contains: the class which will produce a dataset for c-find with patient root

10.133.2 Constructor & Destructor Documentation

10.133.2.1 FindPatientRootQuery()

```
gdcm::FindPatientRootQuery::FindPatientRootQuery ( )
```

10.133.3 Member Function Documentation

10.133.3.1 GetAbstractSyntaxUID()

```
UIDs::TSName gdcm::FindPatientRootQuery::GetAbstractSyntaxUID ( ) const  [virtual]
```

Implements gdcm::BaseQuery.

10.133.3.2 GetTagListByLevel()

```
std::vector<Tag> gdcm::FindPatientRootQuery::GetTagListByLevel (
            const EQueryLevel & inQueryLevel )  [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.

10.133.3.3 InitializeDataSet()

```
void gdcm::FindPatientRootQuery::InitializeDataSet (
            const EQueryLevel & inQueryLevel )  [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.

10.133.3.4 ValidateQuery()

```
bool gdcm::FindPatientRootQuery::ValidateQuery (
            bool inStrict = true ) const  [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.

10.133.4 Friends And Related Function Documentation

10.133.4.1 QueryFactory

```
friend class QueryFactory  [friend]
```

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

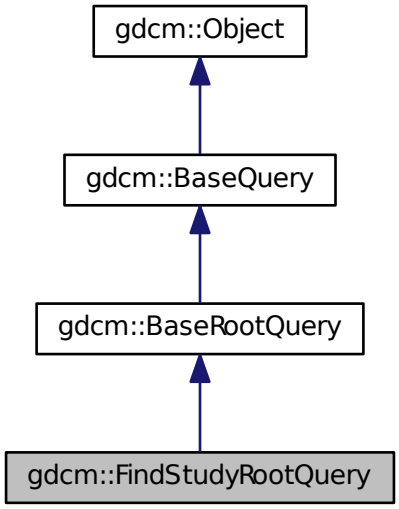
- gdcmFindPatientRootQuery.h

10.134 gdcm::FindStudyRootQuery Class Reference

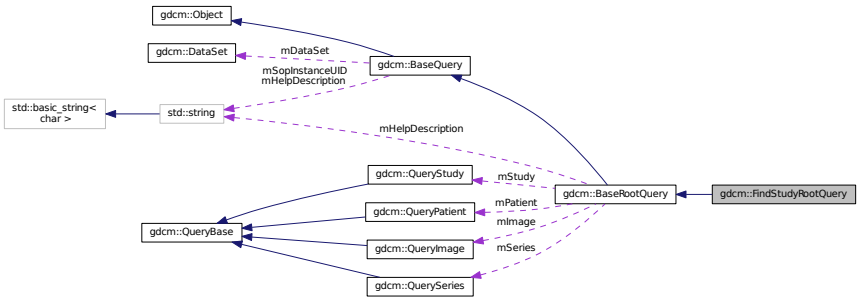
FindStudyRootQuery.

```
#include <gdcmFindStudyRootQuery.h>
```

Inheritance diagram for gdcm::FindStudyRootQuery:



Collaboration diagram for gdcm::FindStudyRootQuery:



Public Member Functions

- FindStudyRootQuery ()
- UIDs::TSName GetAbstractSyntaxUID () const
- std::vector< Tag > GetTagListByLevel (const EQueryLevel &inQueryLevel)
- void InitializeDataSet (const EQueryLevel &inQueryLevel)
- bool ValidateQuery (bool inStrict=true) const

Friends

- class QueryFactory

Additional Inherited Members

10.134.1 Detailed Description

FindStudyRootQuery.

contains: the class which will produce a dataset for C-FIND with study root

10.134.2 Constructor & Destructor Documentation

10.134.2.1 FindStudyRootQuery()

```
gdcm::FindStudyRootQuery::FindStudyRootQuery ( )
```

10.134.3 Member Function Documentation

10.134.3.1 GetAbstractSyntaxUID()

```
UIDs::TSName gdcm::FindStudyRootQuery::GetAbstractSyntaxUID ( ) const  [virtual]
```

Implements gdcm::BaseQuery.

10.134.3.2 GetTagListByLevel()

```
std::vector<Tag> gdcm::FindStudyRootQuery::GetTagListByLevel (
            const EQueryLevel & inQueryLevel )  [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.

10.134.3.3 InitializeDataSet()

```
void gdcm::FindStudyRootQuery::InitializeDataSet (
            const EQueryLevel & inQueryLevel )  [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.

10.134.3.4 ValidateQuery()

```
bool gdcm::FindStudyRootQuery::ValidateQuery (
            bool inStrict = true ) const  [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.

10.134.4 Friends And Related Function Documentation

10.134.4.1 QueryFactory

```
friend class QueryFactory  [friend]
```

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

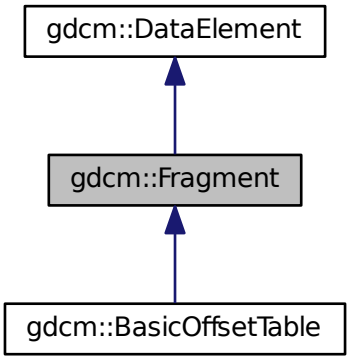
- gdcmFindStudyRootQuery.h

10.135 gdcm::Fragment Class Reference

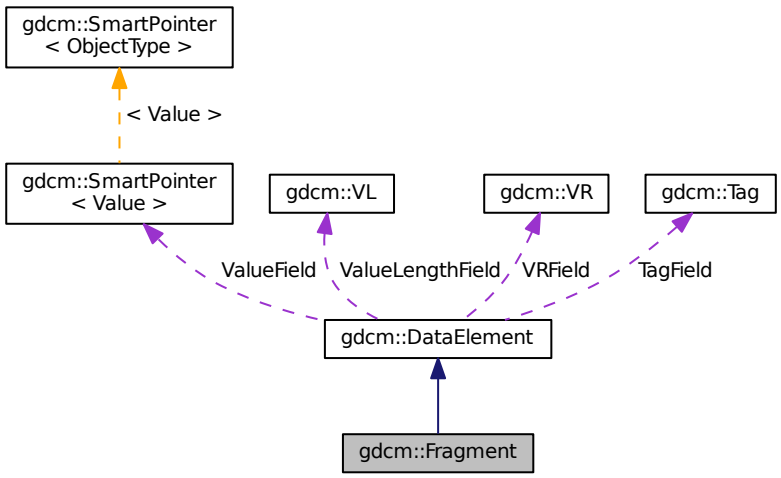
Class to represent a Fragment.

`#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

10.135.1 Detailed Description

Class to represent a Fragment.

Examples:

FixBrokenJ2K.cxx, and FixJAIBugJPEGLS.cxx.

10.135.2 Constructor & Destructor Documentation

10.135.2.1 Fragment()

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

References gdcm::operator<<().

10.135.3 Member Function Documentation

10.135.3.1 ComputeLength()

```
VL gdcm::Fragment::ComputeLength ( ) const
```

10.135.3.2 GetLength()

```
VL gdcm::Fragment::GetLength ( ) const
```

10.135.3.3 Read()

```
template<typename TSwap >
std::istream& gdcm::Fragment::Read (
             std::istream & is )  [inline]
```

Referenced by gdcm::SequenceOfFragments::ReadValue().

10.135.3.4 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().

10.135.3.5 ReadPreValue()

```
template<typename TSwap >
std::istream& gdcm::Fragment::ReadPreValue (
             std::istream & is )  [inline]
```

10.135.3.6 ReadValue()

```
template<typename TSwap >
std::istream& gdcm::Fragment::ReadValue (
             std::istream & is )  [inline]
```

References gdcmWarningMacro, and gdcm::ParseException::SetLastElement().

10.135.3.7 Write()

```
template<typename TSwap >
std::ostream& gdcm::Fragment::Write (
             std::ostream & os ) const  [inline]
```

References gdcm::ByteValue::ComputeLength(), gdcm::ByteValue::GetLength(), gdcm::VL::Write(), and gdcm::Byte↩ Value::Write().

10.135.4 Friends And Related Function Documentation

10.135.4.1 operator<<

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

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

- gdcmFragment.h

10.136 gdcm::Global Class Reference

Global.

```
#include <gdcmGlobal.h>
```

Public Member Functions

- Global ()
- ~Global ()
- bool Append (const char *path)
- Defs const & GetDefs () const
- Dicts const & GetDicts () const
- Dicts & GetDicts ()
- bool LoadResourcesFiles ()
- 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)

10.136.1 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:

GenAllVR.cxx, GenerateStandardSOPClasses.cxx, GenFakeIdentifyFile.cxx, PublicDict.cxx, ReadAndPrint↩ Attributes.cxx, and TraverseModules.cxx.

10.136.2 Constructor & Destructor Documentation

10.136.2.1 Global()

```
gdcm::Global::Global ( )
```

10.136.2.2 ~Global()

```
gdcm::Global::~Global ( )
```

10.136.3 Member Function Documentation

10.136.3.1 Append()

```
bool gdcm::Global::Append (
            const char * path )
```

Append path at the end of the path list

Warning

not thread safe !

10.136.3.2 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.

10.136.3.3 GetDicts() [1/2]

```
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.

10.136.3.4 GetDicts() [2/2]

```
Dicts& gdcm::Global::GetDicts ( )
```

10.136.3.5 GetInstance()

```
static Global& gdcm::Global::GetInstance ( )  [static]
```

return the singleton instance

Examples:

BasicAnonymizer.cs, ClinicalTrialIdentificationWorkflow.cs, GenAllVR.cxx, GenerateStandardSOPClasses.cxx, GenFakeIdentifyFile.cxx, MrProtocol.cxx, PublicDict.cxx, ReadAndPrintAttributes.cxx, and TraverseModules.cxx.

10.136.3.6 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:

GenerateStandardSOPClasses.cxx, and TraverseModules.cxx.

10.136.3.7 Locate()

```
const char* gdcm::Global::Locate (
            const char * resfile ) const  [protected]
```

Locate a resource file.

10.136.3.8 Prepend()

```
bool gdcm::Global::Prepend (
            const char * path )
```

Prepend path at the beginning of the path list

Warning

not thread safe !

10.136.4 Friends And Related Function Documentation

10.136.4.1 operator<<

```
std::ostream& operator<< (
            std::ostream & _os,
            const Global & g )  [friend]
```

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

- gdcmGlobal.h

10.137 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 ()
- 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)

10.137.1 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 ?

10.137.2 Member Typedef Documentation

10.137.2.1 GroupStringVector

```
typedef std::vector<std::string> gdcm::GroupDict::GroupStringVector
```

10.137.3 Constructor & Destructor Documentation

10.137.3.1 GroupDict()

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

10.137.3.2 ~GroupDict()

```
gdcm::GroupDict::~GroupDict ( )  [inline]
```

References gdcm::operator<<().

10.137.4 Member Function Documentation

10.137.4.1 Add()

```
void gdcm::GroupDict::Add (
            std::string const & abbreviation,
            std::string const & name )  [protected]
```

10.137.4.2 GetAbbreviation()

```
std::string const& gdcm::GroupDict::GetAbbreviation (
            uint16_t num ) const
```

Referenced by gdcm::operator<<().

10.137.4.3 GetName()

```
std::string const& gdcm::GroupDict::GetName (
            uint16_t num ) const
```

Referenced by gdcm::operator<<().

10.137.4.4 Insert()

```
void gdcm::GroupDict::Insert (
            uint16_t num,
            std::string const & abbreviation,
            std::string const & name )  [protected]
```

10.137.4.5 Size()

```
size_t gdcm::GroupDict::Size ( ) const  [inline]
```

Referenced by gdcm::operator<<().

10.137.5 Friends And Related Function Documentation

10.137.5.1 operator<<

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

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

- gdcmGroupDict.h

10.138 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 ()

10.138.1 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.

10.138.2 Constructor & Destructor Documentation

10.138.2.1 IconImageFilter()

```
gdcm::IconImageFilter::IconImageFilter ( )
```

10.138.2.2 ~IconImageFilter()

```
gdcm::IconImageFilter::~IconImageFilter ( )
```

10.138.3 Member Function Documentation

10.138.3.1 Extract()

```
bool gdcm::IconImageFilter::Extract ( )
```

Extract all Icon found in File.

Examples:

ExtractIconFromFile.cxx.

10.138.3.2 ExtractIconImages()

```
void gdcm::IconImageFilter::ExtractIconImages ( )  [protected]
```

10.138.3.3 ExtractVeproIconImages()

```
void gdcm::IconImageFilter::ExtractVeproIconImages ( )  [protected]
```

10.138.3.4 GetFile() [1/2]

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

10.138.3.5 GetFile() [2/2]

```
const File& gdcm::IconImageFilter::GetFile ( ) const  [inline]
```

10.138.3.6 GetIconImage()

```
IconImage& gdcm::IconImageFilter::GetIconImage (
            unsigned int i ) const
```

Examples:

ExtractIconFromFile.cxx.

10.138.3.7 GetNumberOfIconImages()

```
unsigned int gdcm::IconImageFilter::GetNumberOfIconImages ( ) const
```

Retrieve extract IconImage (need to call Extract first)

Examples:

ExtractIconFromFile.cxx.

10.138.3.8 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

10.139 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.

10.139.1 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.

10.139.2 Constructor & Destructor Documentation

10.139.2.1 IconImageGenerator()

```
gdcm::IconImageGenerator::IconImageGenerator ( )
```

10.139.2.2 ~IconImageGenerator()

```
gdcm::IconImageGenerator::~IconImageGenerator ( )
```

10.139.3 Member Function Documentation

10.139.3.1 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.

10.139.3.2 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

10.139.3.3 Generate()

```
bool gdcm::IconImageGenerator::Generate ( )
```

Generate Icon.

Examples:

ExtractIconFromFile.cxx.

10.139.3.4 GetIconImage()

```
const IconImage& gdcm::IconImageGenerator::GetIconImage ( ) const  [inline]
```

Retrieve generated Icon.

Examples:

ExtractIconFromFile.cxx.

10.139.3.5 GetPixmap() [1/2]

```
Pixmap& gdcm::IconImageGenerator::GetPixmap ( )  [inline]
```

10.139.3.6 GetPixmap() [2/2]

```
const Pixmap& gdcm::IconImageGenerator::GetPixmap ( ) const  [inline]
```

10.139.3.7 SetOutputDimensions()

```
void gdcm::IconImageGenerator::SetOutputDimensions (
            const unsigned int dims[2] )
```

Set Target dimension of output Icon.

Examples:

ExtractIconFromFile.cxx.

10.139.3.8 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)

10.139.3.9 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.

10.139.3.10 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

10.140 gdcm::ignore_char Struct Reference

`#include <gdcmElement.h>`

Public Member Functions

- ignore_char (char c)

Public Attributes

- char m_char

10.140.1 Constructor & Destructor Documentation

10.140.1.1 ignore_char()

```
gdcm::ignore_char::ignore_char (
            char c )  [inline]
```

10.140.2 Member Data Documentation

10.140.2.1 m_char

```
char gdcm::ignore_char::m_char
```

Referenced by gdcm::operator>>().

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

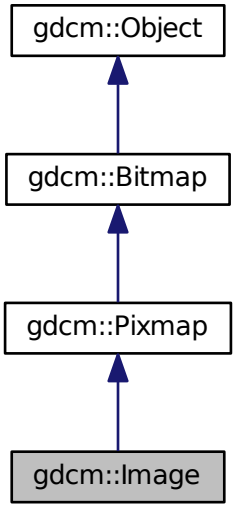
- gdcmElement.h

10.141 gdcm::Image Class Reference

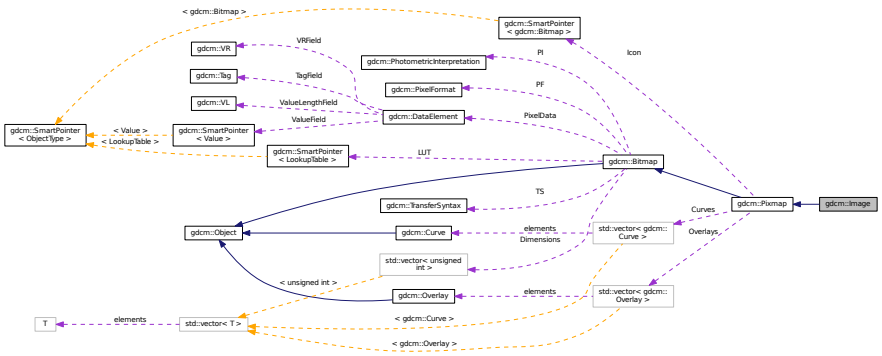
Image.

`#include <gdcmImage.h>`

Inheritance diagram for gdcm::Image:



Collaboration diagram for gdcm::Image:



Public Member Functions

- Image ()
- ~Image ()
- 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

  print
- void SetDirectionCosines (const float *dircos)
- void SetDirectionCosines (const double *dircos)
- void SetDirectionCosines (unsigned int idx, double dircos)
- void SetIntercept (double intercept)

  intercept
- void SetOrigin (const float *ori)
- void SetOrigin (const double *ori)
- void SetOrigin (unsigned int idx, double ori)
- void SetSlope (double slope)

  slope
- void SetSpacing (const double *spacing)
- void SetSpacing (unsigned int idx, double spacing)

Additional Inherited Members

10.141.1 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:

CompressImage.cxx, ConvertToQImage.cxx, CreateARGBImage.cxx, CreateCMYKImage.cxx, csa2img.cxx, ExtractIconFromFile.cxx, FixJAIBugJPEGLS.cxx, GenFakeImage.cxx, GetJPEGSamplePrecision.cxx, GetSub↩SequenceData.cxx, HelloVizWorld.cxx, iU22tomultisc.cxx, PatchFile.cxx, ReadMultiTimesException.cxx, and threadgdcm.cxx.

10.141.2 Constructor & Destructor Documentation

10.141.2.1 Image()

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

10.141.2.2 ~Image()

```
gdcm::Image::~Image ( )  [inline]
```

10.141.3 Member Function Documentation

10.141.3.1 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.

10.141.3.2 GetDirectionCosines() [2/2]

```
double gdcm::Image::GetDirectionCosines (
            unsigned int idx ) const
```

10.141.3.3 GetIntercept()

```
double gdcm::Image::GetIntercept ( ) const  [inline]
```

10.141.3.4 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.

10.141.3.5 GetOrigin() [2/2]

```
double gdcm::Image::GetOrigin (
            unsigned int idx ) const
```

10.141.3.6 GetSlope()

```
double gdcm::Image::GetSlope ( ) const  [inline]
```

10.141.3.7 GetSpacing() [1/2]

```
const double* gdcm::Image::GetSpacing ( ) const
```

Return a 3-tuples specifying the spacing NOTE: 3rd value can be an aribtrary 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

10.141.3.8 GetSpacing() [2/2]

```
double gdcm::Image::GetSpacing (
            unsigned int idx ) const
```

10.141.3.9 Print()

```
void gdcm::Image::Print (
            std::ostream & os ) const  [virtual]
```

print

Reimplemented from gdcm::Bitmap.

Examples:

CompressImage.cxx, and PatchFile.cxx.

10.141.3.10 SetDirectionCosines() [1/3]

```
void gdcm::Image::SetDirectionCosines (
            const float * dircos )
```

10.141.3.11 SetDirectionCosines() [2/3]

```
void gdcm::Image::SetDirectionCosines (
            const double * dircos )
```

10.141.3.12 SetDirectionCosines() [3/3]

```
void gdcm::Image::SetDirectionCosines (
            unsigned int idx,
            double dircos )
```

10.141.3.13 SetIntercept()

```
void gdcm::Image::SetIntercept (
            double intercept )  [inline]
```

intercept

10.141.3.14 SetOrigin() [1/3]

```
void gdcm::Image::SetOrigin (
            const float * ori )
```

10.141.3.15 SetOrigin() [2/3]

```
void gdcm::Image::SetOrigin (
            const double * ori )
```

10.141.3.16 SetOrigin() [3/3]

```
void gdcm::Image::SetOrigin (
            unsigned int idx,
            double ori )
```

10.141.3.17 SetSlope()

```
void gdcm::Image::SetSlope (
            double slope )  [inline]
```

slope

10.141.3.18 SetSpacing() [1/2]

```
void gdcm::Image::SetSpacing (
            const double * spacing )
```

Examples:

csa2img.cxx, and iU22tomultisc.cxx.

10.141.3.19 SetSpacing() [2/2]

```
void gdcm::Image::SetSpacing (
            unsigned int idx,
            double spacing )
```

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

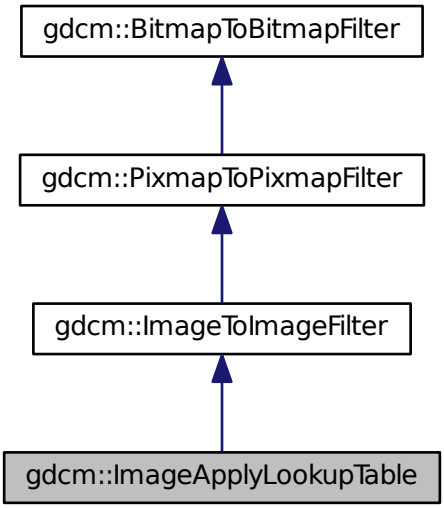
- gdcmImage.h

10.142 gdcm::ImageApplyLookupTable Class Reference

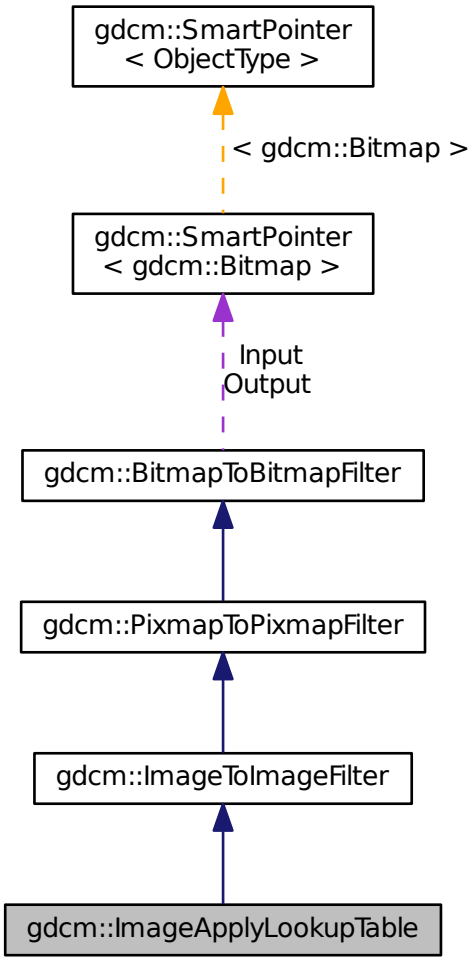
ImageApplyLookupTable class.

```
#include <gdcmImageApplyLookupTable.h>
```

Inheritance diagram for gdcm::ImageApplyLookupTable:



Collaboration diagram for gdcm::ImageApplyLookupTable:



Public Member Functions

- ImageApplyLookupTable ()
- ~ImageApplyLookupTable ()
- bool Apply ()

  Apply.

Additional Inherited Members

10.142.1 Detailed Description

ImageApplyLookupTable class.

It applies the LUT the PixelData (only PALETTE_COLOR images) Output will be a PhotometricInterpretation=RGB image

10.142.2 Constructor & Destructor Documentation

10.142.2.1 ImageApplyLookupTable()

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

10.142.2.2 ~ImageApplyLookupTable()

```
gdcm::ImageApplyLookupTable::~ImageApplyLookupTable ( )  [inline]
```

10.142.3 Member Function Documentation

10.142.3.1 Apply()

```
bool gdcm::ImageApplyLookupTable::Apply ( )
```

Apply.

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

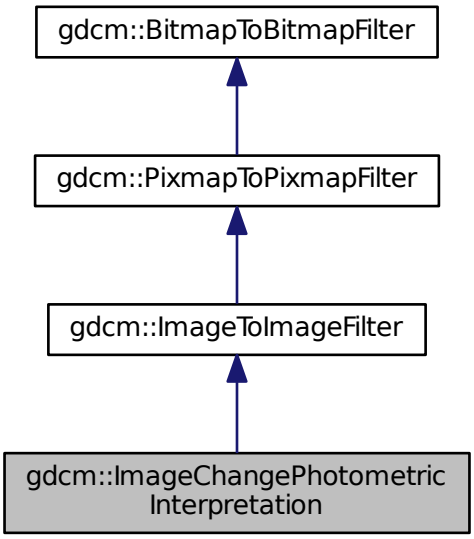
- gdcmImageApplyLookupTable.h

10.143 gdcm::ImageChangePhotometricInterpretation Class Reference

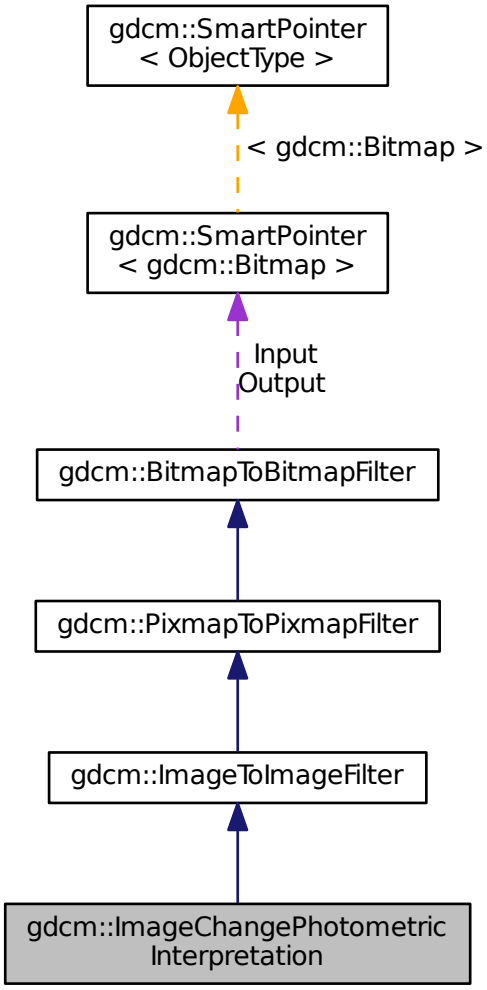
ImageChangePhotometricInterpretation class.

```
#include <gdcmImageChangePhotometricInterpretation.h>
```

Inheritance diagram for gdcm::ImageChangePhotometricInterpretation:



Collaboration diagram for gdcm::ImageChangePhotometricInterpretation:



Public Member Functions

- ImageChangePhotometricInterpretation ()
- ~ImageChangePhotometricInterpretation ()
- 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])

  colorspace converstion (based on CCIR Recommendation 601-2)
- template<typename T >
  static void YBR2RGB (T rgb[3], const T ybr[3])

Protected Member Functions

- bool ChangeMonochrome ()

Additional Inherited Members

10.143.1 Detailed Description

ImageChangePhotometricInterpretation class.

Class to change the Photometric Interpetation of an input DICOM

10.143.2 Constructor & Destructor Documentation

10.143.2.1 ImageChangePhotometricInterpretation()

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

10.143.2.2 ~ImageChangePhotometricInterpretation()

```
gdcm::ImageChangePhotometricInterpretation::~ImageChangePhotometricInterpretation ( )  [inline]
```

10.143.3 Member Function Documentation

10.143.3.1 Change()

```
bool gdcm::ImageChangePhotometricInterpretation::Change ( )
```

Change.

10.143.3.2 ChangeMonochrome()

```
bool gdcm::ImageChangePhotometricInterpretation::ChangeMonochrome ( )  [protected]
```

10.143.3.3 GetPhotometricInterpretation()

```
const PhotometricInterpretation& gdcm::ImageChangePhotometricInterpretation::GetPhotometric↩
Interpretation ( ) const  [inline]
```

10.143.3.4 RGB2YBR()

```
template<typename T >
void gdcm::ImageChangePhotometricInterpretation::RGB2YBR (
            T ybr[3],
            const T rgb[3] )  [static]
```

colorspace converstion (based on CCIR Recommendation 601-2)

10.143.3.5 SetPhotometricInterpretation()

```
void gdcm::ImageChangePhotometricInterpretation::SetPhotometricInterpretation (
            PhotometricInterpretation const & pi )  [inline]
```

Set/Get requested PhotometricInterpretation.

10.143.3.6 YBR2RGB()

```
template<typename T >
void gdcm::ImageChangePhotometricInterpretation::YBR2RGB (
            T rgb[3],
            const T ybr[3] )  [static]
```

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

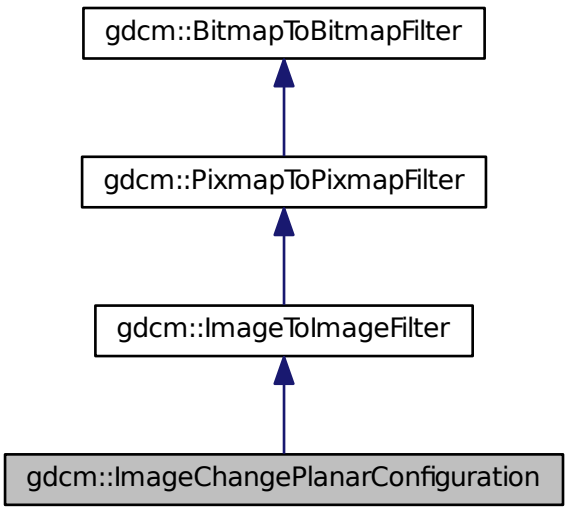
- gdcmImageChangePhotometricInterpretation.h

10.144 gdcm::ImageChangePlanarConfiguration Class Reference

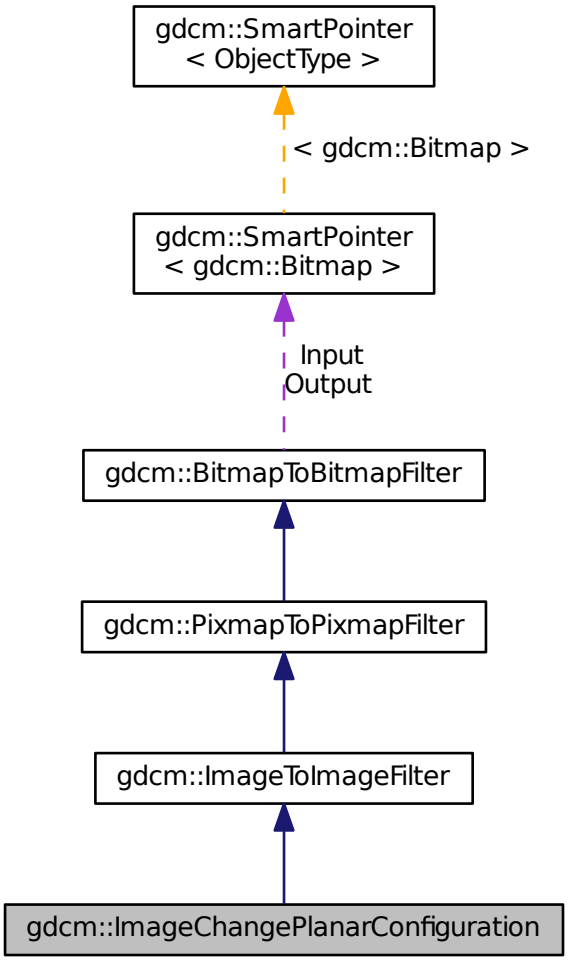
ImageChangePlanarConfiguration class.

```
#include <gdcmImageChangePlanarConfiguration.h>
```

Inheritance diagram for gdcm::ImageChangePlanarConfiguration:



Collaboration diagram for gdcm::ImageChangePlanarConfiguration:



Public Member Functions

- ImageChangePlanarConfiguration ()
- ~ImageChangePlanarConfiguration ()
- 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

10.144.1 Detailed Description

ImageChangePlanarConfiguration class.

Class to change the Planar configuration of an input DICOM By default it will change into the more usual reprensentation: PlanarConfiguration = 0

10.144.2 Constructor & Destructor Documentation

10.144.2.1 ImageChangePlanarConfiguration()

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

10.144.2.2 ~ImageChangePlanarConfiguration()

```
gdcm::ImageChangePlanarConfiguration::~ImageChangePlanarConfiguration ( )  [inline]
```

10.144.3 Member Function Documentation

10.144.3.1 Change()

```
bool gdcm::ImageChangePlanarConfiguration::Change ( )
```

Change.

10.144.3.2 GetPlanarConfiguration()

```
unsigned int gdcm::ImageChangePlanarConfiguration::GetPlanarConfiguration ( ) const  [inline]
```

10.144.3.3 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

10.144.3.4 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

10.144.3.5 SetPlanarConfiguration()

```
void gdcm::ImageChangePlanarConfiguration::SetPlanarConfiguration (
             unsigned int pc )  [inline]
```

Set/Get requested PlanarConfigation.

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

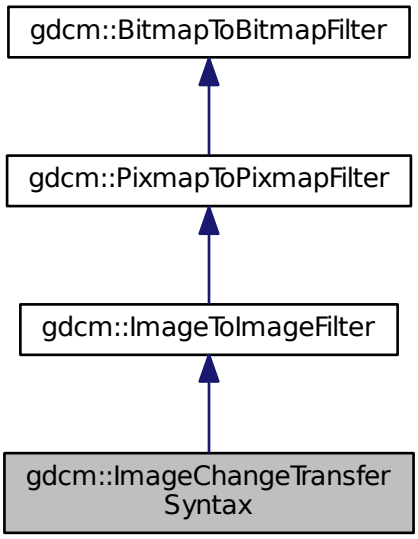
- gdcmImageChangePlanarConfiguration.h

10.145 gdcm::ImageChangeTransferSyntax Class Reference

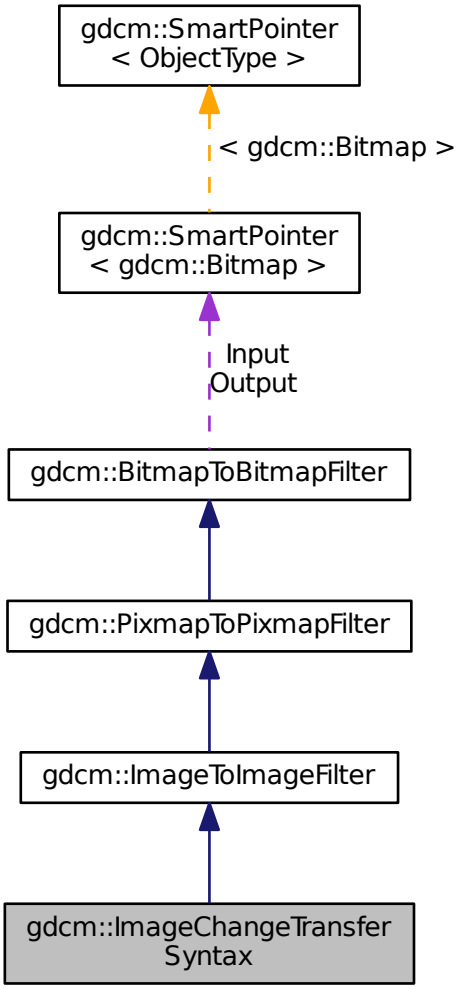
ImageChangeTransferSyntax class.

```
#include <gdcmImageChangeTransferSyntax.h>
```

Inheritance diagram for gdcm::ImageChangeTransferSyntax:



Collaboration diagram for gdcm::ImageChangeTransferSyntax:



Public Member Functions

- ImageChangeTransferSyntax ()
- ~ImageChangeTransferSyntax ()
- 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

10.145.1 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:

CompressImage.cxx.

10.145.2 Constructor & Destructor Documentation

10.145.2.1 ImageChangeTransferSyntax()

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

10.145.2.2 ~ImageChangeTransferSyntax()

```
gdcm::ImageChangeTransferSyntax::~ImageChangeTransferSyntax ( )  [inline]
```

10.145.3 Member Function Documentation

10.145.3.1 Change()

```
bool gdcm::ImageChangeTransferSyntax::Change ( )
```

Change.

Examples:

CompressImage.cxx.

10.145.3.2 GetTransferSyntax()

```
const TransferSyntax& gdcm::ImageChangeTransferSyntax::GetTransferSyntax ( ) const  [inline]
```

Get Transfer Syntax.

10.145.3.3 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

10.145.3.4 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

10.145.3.5 SetTransferSyntax()

```
void gdcm::ImageChangeTransferSyntax::SetTransferSyntax (
            const TransferSyntax & ts )  [inline]
```

Set target Transfer Syntax.

Examples:

CompressImage.cxx.

10.145.3.6 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)

10.145.3.7 TryJPEG2000Codec()

```
bool gdcm::ImageChangeTransferSyntax::TryJPEG2000Codec (
            const DataElement & pixelde,
            Bitmap const & input,
            Bitmap & output )  [protected]
```

10.145.3.8 TryJPEGCodec()

```
bool gdcm::ImageChangeTransferSyntax::TryJPEGCodec (
            const DataElement & pixelde,
            Bitmap const & input,
            Bitmap & output )  [protected]
```

10.145.3.9 TryJPEGLSCodec()

```
bool gdcm::ImageChangeTransferSyntax::TryJPEGLSCodec (
            const DataElement & pixelde,
            Bitmap const & input,
            Bitmap & output )  [protected]
```

10.145.3.10 TryRAWCodec()

```
bool gdcm::ImageChangeTransferSyntax::TryRAWCodec (
            const DataElement & pixelde,
            Bitmap const & input,
            Bitmap & output )  [protected]
```

10.145.3.11 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:

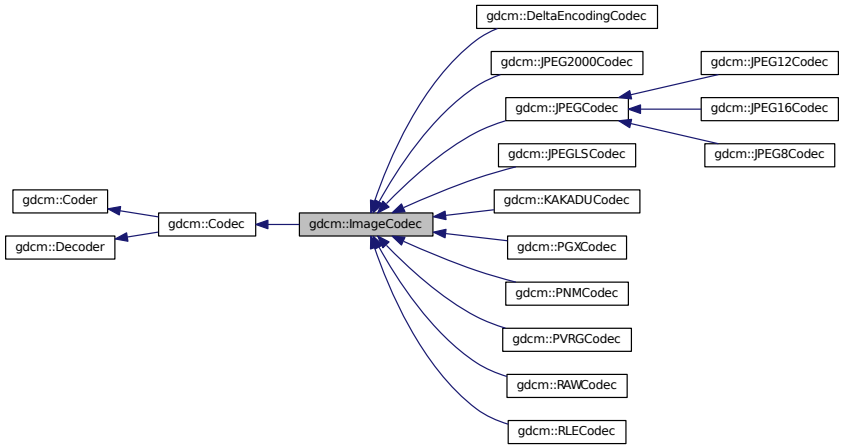
- gdcmImageChangeTransferSyntax.h

10.146 gdcm::ImageCodec Class Reference

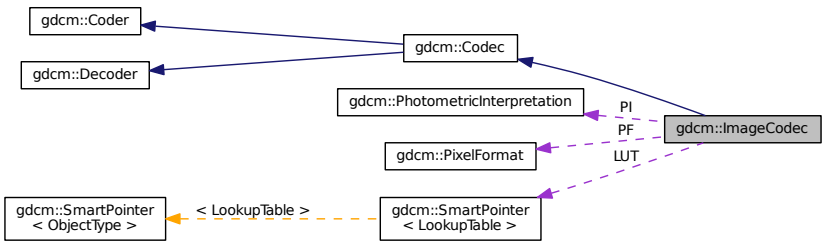
ImageCodec.

`#include <gdcmImageCodec.h>`

Inheritance diagram for gdcm::ImageCodec:



Collaboration diagram for gdcm::ImageCodec:



Public Member Functions

- ImageCodec ()
- ~ImageCodec ()
- bool CanCode (TransferSyntax const &) const

  Return whether this coder support this transfer syntax (can code it)
- bool CanDecode (TransferSyntax const &) const

  Return whether this decoder support this transfer syntax (can decode it)
- virtual ImageCodec * Clone () const =0
- bool Decode (DataElement const &is_, DataElement &os)

  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 unsigned int d[3])
- void SetDimensions (const std::vector< unsigned int > &d)
- 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)
- 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)
- 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

10.146.1 Detailed Description

ImageCodec.

Note

Main codec, this is a central place for all implementation

10.146.2 Member Typedef Documentation

10.146.2.1 LUTPtr

```
typedef SmartPointer<LookupTable> gdcm::ImageCodec::LUTPtr  [protected]
```

10.146.3 Constructor & Destructor Documentation

10.146.3.1 ImageCodec()

```
gdcm::ImageCodec::ImageCodec ( )
```

10.146.3.2 ~ImageCodec()

```
gdcm::ImageCodec::~ImageCodec ( )
```

10.146.4 Member Function Documentation

10.146.4.1 AppendFrameEncode()

```
virtual bool gdcm::ImageCodec::AppendFrameEncode (
            std::ostream & out,
            const char * data,
            size_t datalen )  [protected], [virtual]
```

Reimplemented in gdcm::JPEGCodec, gdcm::JPEGLSCodec, gdcm::JPEG2000Codec, and gdcm::RLECodec.

10.146.4.2 AppendRowEncode()

```
virtual bool gdcm::ImageCodec::AppendRowEncode (
            std::ostream & out,
            const char * data,
            size_t datalen )  [protected], [virtual]
```

Reimplemented in gdcm::JPEGCodec, gdcm::JPEGLSCodec, gdcm::JPEG2000Codec, and gdcm::RLECodec.

10.146.4.3 CanCode()

```
bool gdcm::ImageCodec::CanCode (
            TransferSyntax const &  ) const  [inline], [virtual]
```

Return whether this coder support this transfer syntax (can code it)

Implements gdcm::Coder.

Reimplemented in gdcm::JPEGCodec, gdcm::RLECodec, gdcm::PVRGCodec, gdcm::JPEG2000Codec, gdcm::JPE↩GLSCodec, gdcm::PNMCodec, gdcm::PGXCodec, gdcm::KAKADUCodec, and gdcm::RAWCodec.

10.146.4.4 CanDecode()

```
bool gdcm::ImageCodec::CanDecode (
            TransferSyntax const &  ) const  [inline], [virtual]
```

Return whether this decoder support this transfer syntax (can decode it)

Implements gdcm::Decoder.

Reimplemented in gdcm::JPEGCodec, gdcm::RLECodec, gdcm::PVRGCodec, gdcm::JPEG2000Codec, gdcm::JPE↩GLSCodec, gdcm::PNMCodec, gdcm::RAWCodec, gdcm::PGXCodec, and gdcm::KAKADUCodec.

10.146.4.5 Clone()

```
virtual ImageCodec* gdcm::ImageCodec::Clone ( ) const  [pure virtual]
```

Implemented in gdcm::JPEGCodec, gdcm::RLECodec, gdcm::JPEGLSCodec, gdcm::PVRGCodec, gdcm::JPEG2000Codec, gdcm::PNMCodec, gdcm::RAWCodec, gdcm::KAKADUCodec, and gdcm::PGXCodec.

10.146.4.6 Decode()

```
bool gdcm::ImageCodec::Decode (
            DataElement const & ,
            DataElement &  )  [virtual]
```

Decode.

Reimplemented from gdcm::Decoder.

Reimplemented in gdcm::JPEGCodec, gdcm::RLECodec, gdcm::PVRGCodec, gdcm::JPEGLSCodec, gdcm::JPEG2000Codec, gdcm::KAKADUCodec, and gdcm::RAWCodec.

10.146.4.7 DecodeByStreams()

```
bool gdcm::ImageCodec::DecodeByStreams (
            std::istream & is_,
            std::ostream & os )  [protected], [virtual]
```

Reimplemented from gdcm::Decoder.

Reimplemented in gdcm::JPEGCodec, gdcm::JPEG2000Codec, gdcm::RLECodec, gdcm::RAWCodec, gdcm::JPEG12Codec, gdcm::JPEG16Codec, and gdcm::JPEG8Codec.

10.146.4.8 DoByteSwap()

```
bool gdcm::ImageCodec::DoByteSwap (
            std::istream & is_,
            std::ostream & os )  [protected]
```

10.146.4.9 DoInvertMonochrome()

```
bool gdcm::ImageCodec::DoInvertMonochrome (
            std::istream & is_,
            std::ostream & os )  [protected]
```

10.146.4.10 DoOverlayCleanup()

```
bool gdcm::ImageCodec::DoOverlayCleanup (
            std::istream & is_,
            std::ostream & os )  [protected]
```

10.146.4.11 DoPaddedCompositePixelCode()

```
bool gdcm::ImageCodec::DoPaddedCompositePixelCode (
            std::istream & is_,
            std::ostream & os )  [protected]
```

10.146.4.12 DoPlanarConfiguration()

```
bool gdcm::ImageCodec::DoPlanarConfiguration (
            std::istream & is_,
            std::ostream & os )  [protected]
```

10.146.4.13 DoSimpleCopy()

```
bool gdcm::ImageCodec::DoSimpleCopy (
            std::istream & is_,
            std::ostream & os )  [protected]
```

10.146.4.14 DoYBR()

```
bool gdcm::ImageCodec::DoYBR (
            std::istream & is_,
            std::ostream & os )  [protected]
```

10.146.4.15 GetDimensions()

```
const unsigned int* gdcm::ImageCodec::GetDimensions ( ) const  [inline]
```

References gdcm::terminal::dim.

10.146.4.16 GetHeaderInfo()

```
virtual bool gdcm::ImageCodec::GetHeaderInfo (
            std::istream & is_,
            TransferSyntax & ts )  [virtual]
```

Reimplemented in gdcm::JPEGCodec, gdcm::RLECodec, gdcm::JPEGLSCodec, gdcm::JPEG2000Codec, gdcm::PNMCodec, gdcm::JPEG12Codec, gdcm::JPEG16Codec, gdcm::JPEG8Codec, gdcm::RAWCodec, and gdcm::PGXCodec.

10.146.4.17 GetLossyFlag()

```
bool gdcm::ImageCodec::GetLossyFlag ( ) const
```

10.146.4.18 GetLUT()

```
const LookupTable& gdcm::ImageCodec::GetLUT ( ) const  [inline]
```

10.146.4.19 GetNeedByteSwap()

```
bool gdcm::ImageCodec::GetNeedByteSwap ( ) const  [inline]
```

10.146.4.20 GetNumberOfDimensions()

```
unsigned int gdcm::ImageCodec::GetNumberOfDimensions ( ) const
```

10.146.4.21 GetPhotometricInterpretation()

```
const PhotometricInterpretation& gdcm::ImageCodec::GetPhotometricInterpretation ( ) const
```

10.146.4.22 GetPixelFormat() [1/2]

```
PixelFormat& gdcm::ImageCodec::GetPixelFormat ( )  [inline]
```

Examples:

GetJPEGSamplePrecision.cxx.

10.146.4.23 GetPixelFormat() [2/2]

```
const PixelFormat& gdcm::ImageCodec::GetPixelFormat ( ) const  [inline]
```

10.146.4.24 GetPlanarConfiguration()

```
unsigned int gdcm::ImageCodec::GetPlanarConfiguration ( ) const  [inline]
```

10.146.4.25 IsFrameEncoder()

```
virtual bool gdcm::ImageCodec::IsFrameEncoder ( )  [protected], [virtual]
```

Reimplemented in gdcm::JPEGCodec, gdcm::JPEGLSCodec, gdcm::JPEG2000Codec, and gdcm::RLECodec.

10.146.4.26 IsLossy()

```
bool gdcm::ImageCodec::IsLossy ( ) const
```

10.146.4.27 IsRowEncoder()

```
virtual bool gdcm::ImageCodec::IsRowEncoder ( )  [protected], [virtual]
```

Reimplemented in gdcm::JPEGCodec, gdcm::JPEGLSCodec, gdcm::JPEG2000Codec, and gdcm::RLECodec.

10.146.4.28 IsValid()

```
virtual bool gdcm::ImageCodec::IsValid (
            PhotometricInterpretation const & pi )  [protected], [virtual]
```

Reimplemented in gdcm::JPEGCodec.

10.146.4.29 SetDimensions() [1/2]

```
void gdcm::ImageCodec::SetDimensions (
            const unsigned int d[3] )
```

Examples:

ExtractIconFromFile.cxx.

10.146.4.30 SetDimensions() [2/2]

```
void gdcm::ImageCodec::SetDimensions (
            const std::vector< unsigned int > & d )
```

10.146.4.31 SetLossyFlag()

```
void gdcm::ImageCodec::SetLossyFlag (
            bool l )
```

10.146.4.32 SetLUT()

```
void gdcm::ImageCodec::SetLUT (
            LookupTable const & lut )  [inline]
```

Examples:

ExtractIconFromFile.cxx.

10.146.4.33 SetNeedByteSwap()

```
void gdcm::ImageCodec::SetNeedByteSwap (
            bool b )  [inline]
```

10.146.4.34 SetNeedOverlayCleanup()

```
void gdcm::ImageCodec::SetNeedOverlayCleanup (
            bool b )  [inline]
```

10.146.4.35 SetNumberOfDimensions()

```
void gdcm::ImageCodec::SetNumberOfDimensions (
            unsigned int dim )
```

10.146.4.36 SetPhotometricInterpretation()

```
void gdcm::ImageCodec::SetPhotometricInterpretation (
            PhotometricInterpretation const & pi )
```

Examples:

ExtractIconFromFile.cxx.

10.146.4.37 SetPixelFormat()

```
virtual void gdcm::ImageCodec::SetPixelFormat (
            PixelFormat const & pf )  [inline], [virtual]
```

Reimplemented in gdcm::JPEGCodec.

Examples:

ExtractIconFromFile.cxx.

10.146.4.38 SetPlanarConfiguration()

```
void gdcm::ImageCodec::SetPlanarConfiguration (
            unsigned int pc )  [inline]
```

10.146.4.39 StartEncode()

```
virtual bool gdcm::ImageCodec::StartEncode (
            std::ostream & os )  [protected], [virtual]
```

Reimplemented in gdcm::JPEGCodec, gdcm::JPEGLSCodec, gdcm::JPEG2000Codec, and gdcm::RLECodec.

10.146.4.40 StopEncode()

```
virtual bool gdcm::ImageCodec::StopEncode (
            std::ostream & os )  [protected], [virtual]
```

Reimplemented in gdcm::JPEGCodec, gdcm::JPEGLSCodec, gdcm::JPEG2000Codec, and gdcm::RLECodec.

10.146.5 Friends And Related Function Documentation

10.146.5.1 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

10.146.5.2 ImageChangePhotometricInterpretation

```
friend class ImageChangePhotometricInterpretation  [friend]
```

10.146.6 Member Data Documentation

10.146.6.1 Dimensions

```
unsigned int gdcm::ImageCodec::Dimensions[3]  [protected]
```

10.146.6.2 LossyFlag

```
bool gdcm::ImageCodec::LossyFlag  [protected]
```

10.146.6.3 LUT

```
LUTPtr gdcm::ImageCodec::LUT  [protected]
```

10.146.6.4 NeedByteSwap

```
bool gdcm::ImageCodec::NeedByteSwap  [protected]
```

10.146.6.5 NeedOverlayCleanup

```
bool gdcm::ImageCodec::NeedOverlayCleanup  [protected]
```

10.146.6.6 NumberOfDimensions

```
unsigned int gdcm::ImageCodec::NumberOfDimensions  [protected]
```

10.146.6.7 PF

```
PixelFormat gdcm::ImageCodec::PF  [protected]
```

10.146.6.8 PI

```
PhotometricInterpretation gdcm::ImageCodec::PI  [protected]
```

10.146.6.9 PlanarConfiguration

```
unsigned int gdcm::ImageCodec::PlanarConfiguration  [protected]
```

10.146.6.10 RequestPaddedCompositePixelCode

```
bool gdcm::ImageCodec::RequestPaddedCompositePixelCode  [protected]
```

10.146.6.11 RequestPlanarConfiguration

```
bool gdcm::ImageCodec::RequestPlanarConfiguration  [protected]
```

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

- gdcmImageCodec.h

10.147 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)

10.147.1 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

10.147.2 Constructor & Destructor Documentation

10.147.2.1 ImageConverter()

```
gdcm::ImageConverter::ImageConverter ( )
```

10.147.2.2 ~ImageConverter()

```
gdcm::ImageConverter::~ImageConverter ( )
```

10.147.3 Member Function Documentation

10.147.3.1 Convert()

```
void gdcm::ImageConverter::Convert ( )
```

10.147.3.2 GetOuput()

```
const Image& gdcm::ImageConverter::GetOuput ( ) const
```

10.147.3.3 SetInput()

```
void gdcm::ImageConverter::SetInput (
            Image const & input )
```

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

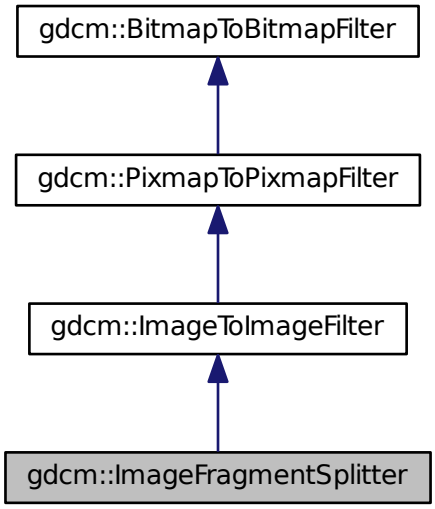
- gdcmImageConverter.h

10.148 gdcm::ImageFragmentSplitter Class Reference

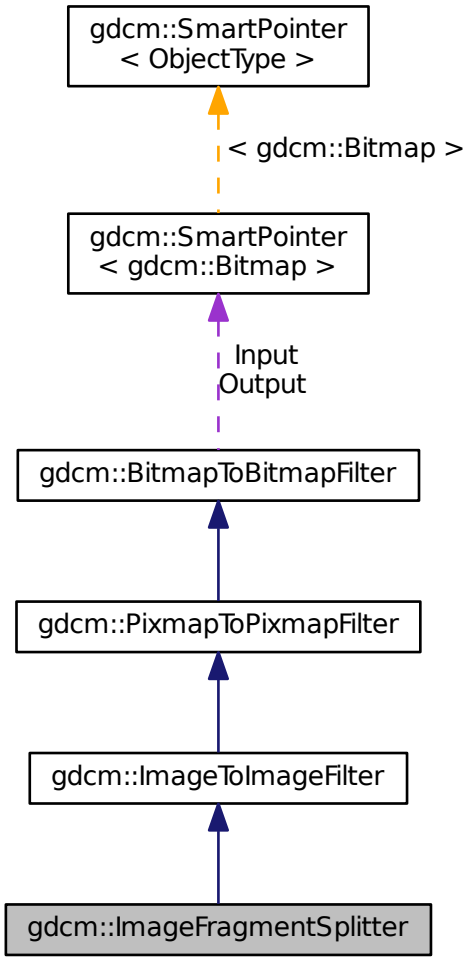
ImageFragmentSplitter class.

```
#include <gdcmImageFragmentSplitter.h>
```

Inheritance diagram for gdcm::ImageFragmentSplitter:



Collaboration diagram for gdcm::ImageFragmentSplitter:



Public Member Functions

- ImageFragmentSplitter ()
- ~ImageFragmentSplitter ()
- 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

10.148.1 Detailed Description

ImageFragmentSplitter class.

For single frame image, DICOM standard allow splitting the frame into multiple fragments

10.148.2 Constructor & Destructor Documentation

10.148.2.1 ImageFragmentSplitter()

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

10.148.2.2 ~ImageFragmentSplitter()

```
gdcm::ImageFragmentSplitter::~ImageFragmentSplitter ( )  [inline]
```

10.148.3 Member Function Documentation

10.148.3.1 GetFragmentSizeMax()

```
unsigned int gdcm::ImageFragmentSplitter::GetFragmentSizeMax ( ) const  [inline]
```

10.148.3.2 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 !

10.148.3.3 SetFragmentSizeMax()

```
void gdcm::ImageFragmentSplitter::SetFragmentSizeMax (
            unsigned int fragsize )
```

FragmentSizeMax needs to be an even number.

10.148.3.4 Split()

```
bool gdcm::ImageFragmentSplitter::Split ( )
```

Split.

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

- gdcmImageFragmentSplitter.h

10.149 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)

10.149.1 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.

10.149.2 Member Function Documentation

10.149.2.1 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.

10.149.2.2 ComputeSpacingFromImagePositionPatient()

```
static bool gdcm::ImageHelper::ComputeSpacingFromImagePositionPatient (
             const std::vector< double > & imageposition,
             std::vector< double > & spacing )  [static]
```

DO NOT USE.

10.149.2.3 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:

Extracting_All_Resolution.cxx, Fake_Image_Using_Stream_Image_Writer.cxx, and StreamImageReaderTest.cxx.

10.149.2.4 GetDirectionCosinesFromDataSet()

```
static bool gdcm::ImageHelper::GetDirectionCosinesFromDataSet (
            DataSet const & ds,
            std::vector< double > & dircos )  [static]
```

10.149.2.5 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

10.149.2.6 GetForcePixelSpacing()

```
static bool gdcm::ImageHelper::GetForcePixelSpacing ( )  [static]
```

10.149.2.7 GetForceRescaleInterceptSlope()

```
static bool gdcm::ImageHelper::GetForceRescaleInterceptSlope ( )  [static]
```

10.149.2.8 GetLUT()

```
static SmartPointer<LookupTable> gdcm::ImageHelper::GetLUT (
            File const & f )  [static]
```

returns the lookup table of an image file

10.149.2.9 GetOriginValue()

```
static std::vector<double> gdcm::ImageHelper::GetOriginValue (
            File const & f )  [static]
```

Set/Get Origin (IPP) from/to a file.

10.149.2.10 GetPhotometricInterpretationValue()

```
static PhotometricInterpretation gdcm::ImageHelper::GetPhotometricInterpretationValue (
            File const & f )  [static]
```

10.149.2.11 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)

10.149.2.12 GetPlanarConfigurationValue()

```
static unsigned int gdcm::ImageHelper::GetPlanarConfigurationValue (
            const File & f )  [static]
```

10.149.2.13 GetPMSRescaleInterceptSlope()

```
static bool gdcm::ImageHelper::GetPMSRescaleInterceptSlope ( )  [static]
```

10.149.2.14 GetPointerFromElement()

```
static const ByteValue* gdcm::ImageHelper::GetPointerFromElement (
            Tag const & tag,
            File const & f )  [static]
```

10.149.2.15 GetRealWorldValueMappingContent()

```
static bool gdcm::ImageHelper::GetRealWorldValueMappingContent (
            File const & f,
            RealWorldValueMappingContent & rwvmc )  [static]
```

10.149.2.16 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

10.149.2.17 GetSpacingTagFromMediaStorage()

```
static Tag gdcm::ImageHelper::GetSpacingTagFromMediaStorage (
            MediaStorage const & ms )  [static], [protected]
```

10.149.2.18 GetSpacingValue()

```
static std::vector<double> gdcm::ImageHelper::GetSpacingValue (
            File const & f )  [static]
```

Set/Get Spacing from/to a File.

10.149.2.19 GetZSpacingTagFromMediaStorage()

```
static Tag gdcm::ImageHelper::GetZSpacingTagFromMediaStorage (
            MediaStorage const & ms )  [static], [protected]
```

10.149.2.20 SetDimensionsValue()

```
static void gdcm::ImageHelper::SetDimensionsValue (
            File & f,
            const Pixmap & img )  [static]
```

10.149.2.21 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.

10.149.2.22 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

10.149.2.23 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.

10.149.2.24 SetOriginValue()

```
static void gdcm::ImageHelper::SetOriginValue (
            DataSet & ds,
            const Image & img )  [static]
```

10.149.2.25 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.

10.149.2.26 SetRescaleInterceptSlopeValue()

```
static void gdcm::ImageHelper::SetRescaleInterceptSlopeValue (
            File & f,
            const Image & img )  [static]
```

10.149.2.27 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:

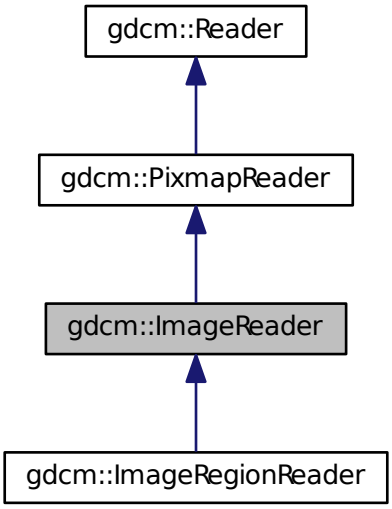
- gdcmImageHelper.h

10.150 gdcm::ImageReader Class Reference

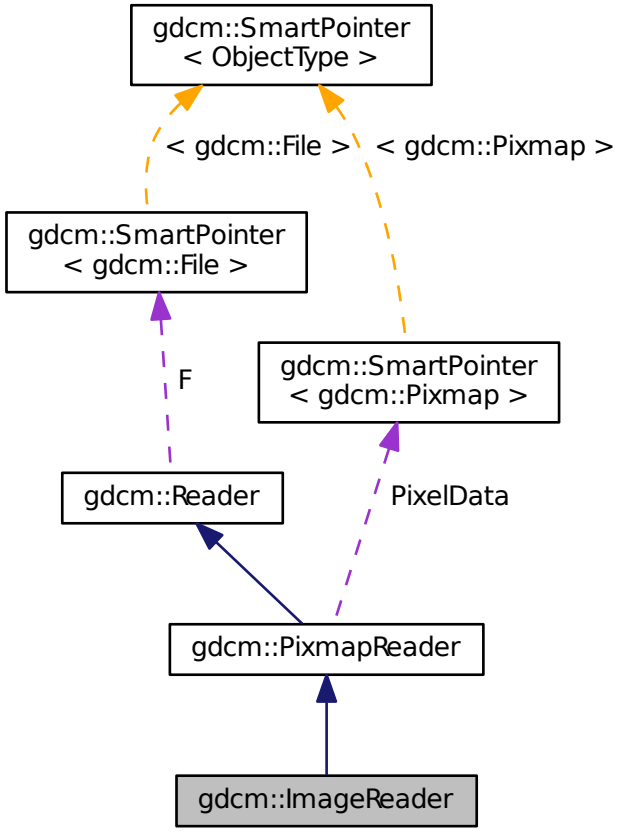
ImageReader.

```
#include <gdcmImageReader.h>
```

Inheritance diagram for gdcm::ImageReader:



Collaboration diagram for gdcm::ImageReader:



Public Member Functions

- ImageReader ()
- virtual ~ImageReader ()
- const Image & GetImage () const

  Return the read image.
- Image & GetImage ()
- virtual bool Read ()

Protected Member Functions

- bool ReadACRNEMAImage ()
- bool ReadImage (MediaStorage const &ms)

Additional Inherited Members

10.150.1 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:

CheckBigEndianBug.cxx, CompressImage.cxx, ConvertToQImage.cxx, ExtractIconFromFile.cxx, FixBroken↩J2K.cxx, FixJAIBugJPEGLS.cxx, GetJPEGSamplePrecision.cxx, HelloVizWorld.cxx, MergeTwoFiles.cxx, Mr↩Protocol.cxx, PatchFile.cxx, ReadMultiTimesException.cxx, and threadgdcm.cxx.

10.150.2 Constructor & Destructor Documentation

10.150.2.1 ImageReader()

```
gdcm::ImageReader::ImageReader ( )
```

10.150.2.2 ~ImageReader()

```
virtual gdcm::ImageReader::~ImageReader ( )  [virtual]
```

10.150.3 Member Function Documentation

10.150.3.1 GetImage() [1/2]

```
const Image& gdcm::ImageReader::GetImage ( ) const
```

Return the read image.

Examples:

CompressImage.cxx, ConvertToQImage.cxx, ExtractIconFromFile.cxx, FixJAIBugJPEGLS.cxx, GetJPEG↩SamplePrecision.cxx, HelloVizWorld.cxx, MergeTwoFiles.cxx, PatchFile.cxx, ReadMultiTimesException.cxx, and threadgdcm.cxx.

10.150.3.2 GetImage() [2/2]

```
Image& gdcm::ImageReader::GetImage ( )
```

10.150.3.3 Read()

```
virtual bool gdcm::ImageReader::Read ( )  [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::PixmapReader.

Reimplemented in gdcm::ImageRegionReader.

Examples:

CheckBigEndianBug.cxx, CompressImage.cxx, ConvertToQImage.cxx, ExtractIconFromFile.cxx, FixBroken↩J2K.cxx, FixJAIBugJPEGLS.cxx, GetJPEGSamplePrecision.cxx, HelloVizWorld.cxx, MergeTwoFiles.cxx, Mr↩Protocol.cxx, PatchFile.cxx, ReadMultiTimesException.cxx, and threadgdcm.cxx.

10.150.3.4 ReadACRNEMAImage()

```
bool gdcm::ImageReader::ReadACRNEMAImage ( )  [protected], [virtual]
```

Reimplemented from gdcm::PixmapReader.

10.150.3.5 ReadImage()

```
bool gdcm::ImageReader::ReadImage (
            MediaStorage const & ms )  [protected], [virtual]
```

Reimplemented from gdcm::PixmapReader.

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

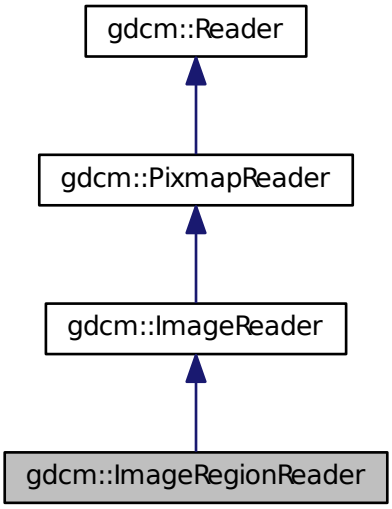
- gdcmImageReader.h

10.151 gdcm::ImageRegionReader Class Reference

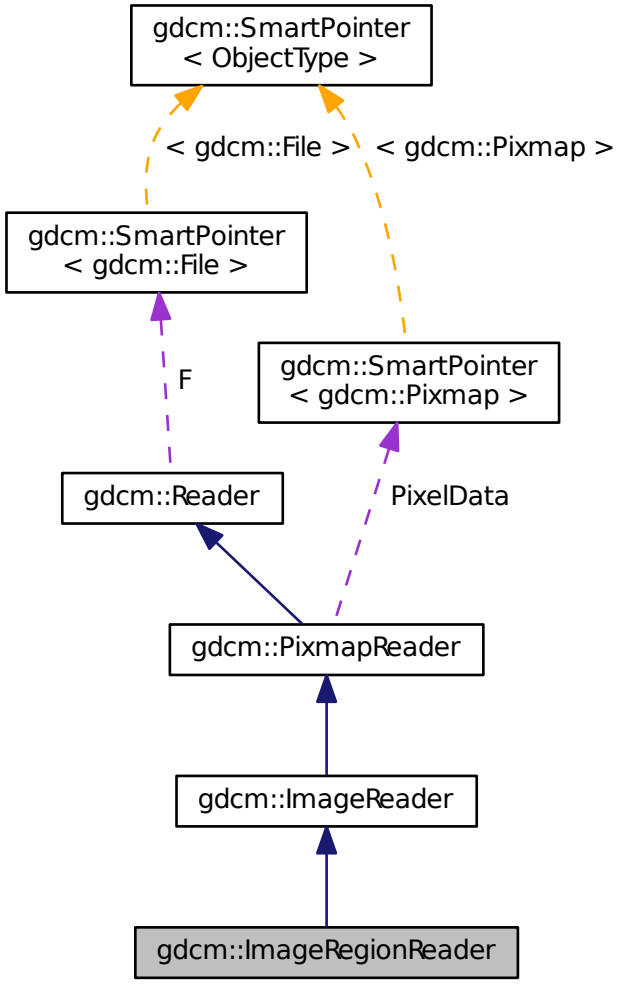
ImageRegionReader.

```
#include <gdcmImageRegionReader.h>
```

Inheritance diagram for gdcm::ImageRegionReader:



Collaboration diagram for gdcm::ImageRegionReader:



Public Member Functions

- ImageRegionReader ()
- ~ImageRegionReader ()
- 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 ()

  To prevent user from calling super class Read() function.

Additional Inherited Members

10.151.1 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

10.151.2 Constructor & Destructor Documentation

10.151.2.1 ImageRegionReader()

```
gdcm::ImageRegionReader::ImageRegionReader ( )
```

10.151.2.2 ~ImageRegionReader()

```
gdcm::ImageRegionReader::~ImageRegionReader ( )
```

10.151.3 Member Function Documentation

10.151.3.1 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

10.151.3.2 GetRegion()

```
Region const& gdcm::ImageRegionReader::GetRegion ( ) const
```

10.151.3.3 Read()

```
bool gdcm::ImageRegionReader::Read ( )  [protected], [virtual]
```

To prevent user from calling super class Read() function.

Reimplemented from gdcm::ImageReader.

10.151.3.4 ReadInformation()

```
bool gdcm::ImageRegionReader::ReadInformation ( )
```

Read meta information (not Pixel Data) from the DICOM file.

Returns

false upon error

10.151.3.5 ReadIntoBuffer()

```
bool gdcm::ImageRegionReader::ReadIntoBuffer (
            char * inreadbuffer,
            size_t buflen )
```

Read into buffer:

Returns

false upon error

10.151.3.6 SetRegion()

```
void gdcm::ImageRegionReader::SetRegion (
            Region const & region )
```

Set/Get Region to be read.

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

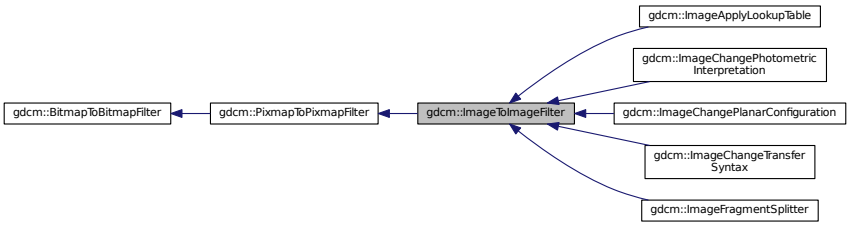
- gdcmImageRegionReader.h

10.152 gdcm::ImageToImageFilter Class Reference

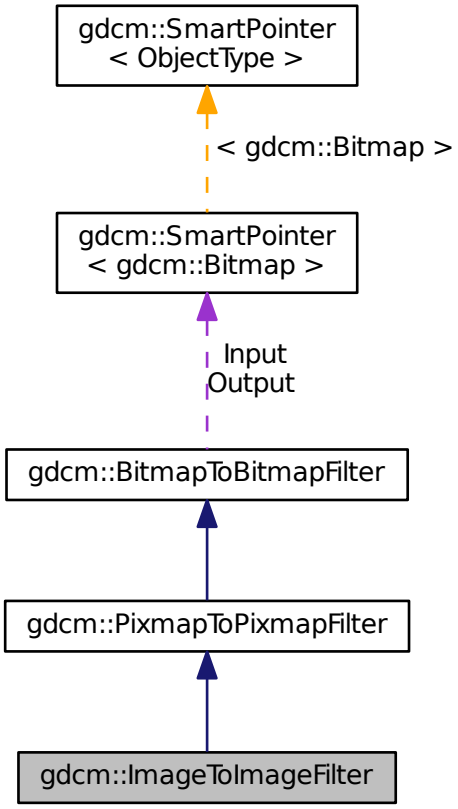
ImageToImageFilter class.

`#include <gdcmImageToImageFilter.h>`

Inheritance diagram for gdcm::ImageToImageFilter:



Collaboration diagram for gdcm::ImageToImageFilter:



Public Member Functions

- ImageToImageFilter ()
- ~ImageToImageFilter ()
- Image & GetInput ()
- const Image & GetOutput () const

  Get Output image.

Additional Inherited Members

10.152.1 Detailed Description

ImageToImageFilter class.

Super class for all filter taking an image and producing an output image

10.152.2 Constructor & Destructor Documentation

10.152.2.1 ImageToImageFilter()

```
gdcm::ImageToImageFilter::ImageToImageFilter ( )
```

10.152.2.2 ~ImageToImageFilter()

```
gdcm::ImageToImageFilter::~ImageToImageFilter ( )  [inline]
```

10.152.3 Member Function Documentation

10.152.3.1 GetInput()

```
Image& gdcm::ImageToImageFilter::GetInput ( )
```

10.152.3.2 GetOutput()

```
const Image& gdcm::ImageToImageFilter::GetOutput ( ) const
```

Get Output image.

Examples:

CompressImage.cxx.

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

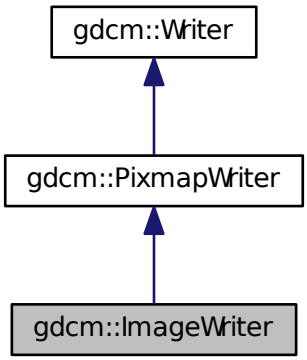
- gdcmImageToImageFilter.h

10.153 gdcm::ImageWriter Class Reference

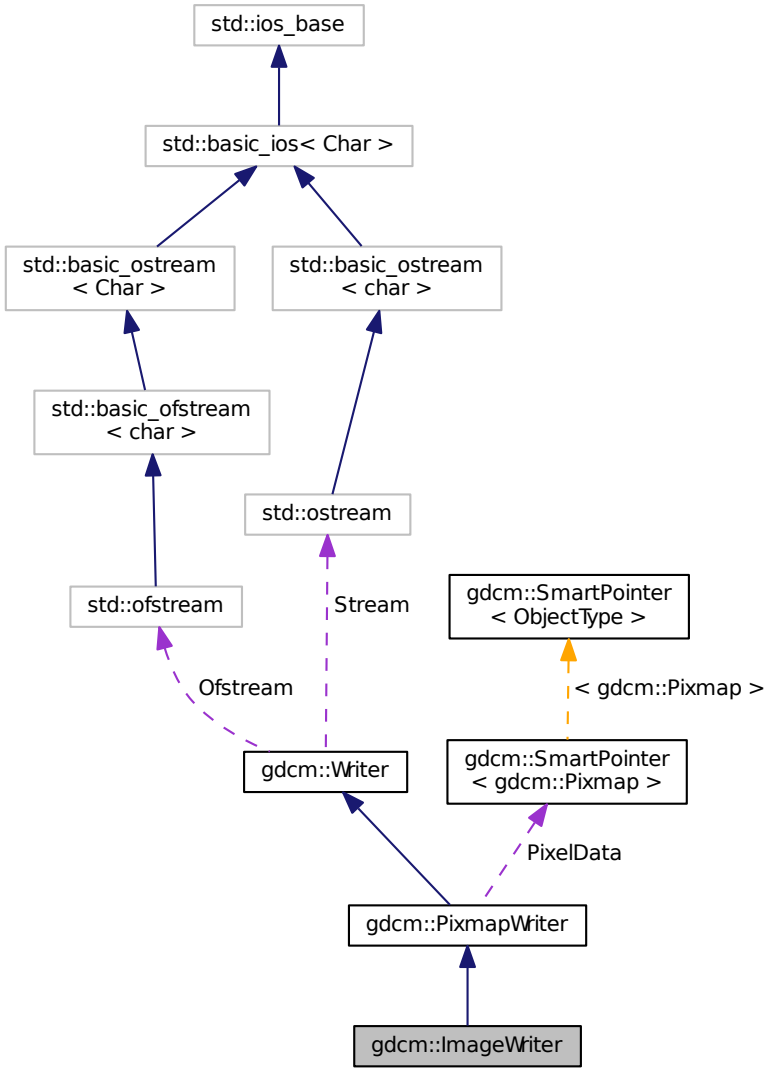
ImageWriter.

```
#include <gdcmImageWriter.h>
```

Inheritance diagram for gdcm::ImageWriter:



Collaboration diagram for gdcm::ImageWriter:



Public Member Functions

- ImageWriter ()
- ~ImageWriter ()
- MediaStorage ComputeTargetMediaStorage ()
- const Image & GetImage () const
- Image & GetImage ()
- bool Write ()

  Write.

Additional Inherited Members

10.153.1 Detailed Description

ImageWriter.

Examples:

CompressImage.cxx, CreateARGBImage.cxx, CreateCMYKImage.cxx, csa2img.cxx, GenFakeImage.cxx, GetSubSequenceData.cxx, HelloVizWorld.cxx, iU22tomultisc.cxx, and MergeTwoFiles.cxx.

10.153.2 Constructor & Destructor Documentation

10.153.2.1 ImageWriter()

```
gdcm::ImageWriter::ImageWriter ( )
```

10.153.2.2 ~ImageWriter()

```
gdcm::ImageWriter::~ImageWriter ( )
```

10.153.3 Member Function Documentation

10.153.3.1 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)

10.153.3.2 GetImage() [1/2]

```
const Image& gdcm::ImageWriter::GetImage ( ) const  [inline], [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.

10.153.3.3 GetImage() [2/2]

```
Image& gdcm::ImageWriter::GetImage ( )  [inline], [virtual]
```

Reimplemented from gdcm::PixmapWriter.

10.153.3.4 Write()

```
bool gdcm::ImageWriter::Write ( )  [virtual]
```

Write.

Reimplemented from gdcm::Writer.

Examples:

CompressImage.cxx, CreateARGBImage.cxx, CreateCMYKImage.cxx, csa2img.cxx, GenFakeImage.cxx, GetSubSequenceData.cxx, HelloVizWorld.cxx, iU22tomultisc.cxx, and MergeTwoFiles.cxx.

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

- gdcmImageWriter.h

10.154 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

10.154.1 Detailed Description

ImplementationClassUIDSub.

PS 3.7 Table D.3-1 IMPLEMENTATION CLASS UID SUB-ITEM FIELDS (A-ASSOCIATE-RQ)

10.154.2 Constructor & Destructor Documentation

10.154.2.1 ImplementationClassUIDSub()

```
gdcm::network::ImplementationClassUIDSub::ImplementationClassUIDSub ( )
```

10.154.3 Member Function Documentation

10.154.3.1 Print()

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

10.154.3.2 Read()

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

10.154.3.3 Size()

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

10.154.3.4 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

10.155 gdcm::network::ImplementationUIDSub Class Reference

ImplementationUIDSub.

```
#include <gdcmImplementationUIDSub.h>
```

Public Member Functions

- ImplementationUIDSub ()
- const std::ostream & Write (std::ostream &os) const

10.155.1 Detailed Description

ImplementationUIDSub.

Table D.3-2 IMPLEMENTATION UID SUB-ITEM FIELDS (A-ASSOCIATE-AC)

10.155.2 Constructor & Destructor Documentation

10.155.2.1 ImplementationUIDSub()

```
gdcm::network::ImplementationUIDSub::ImplementationUIDSub ( )
```

10.155.3 Member Function Documentation

10.155.3.1 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

10.156 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

10.156.1 Detailed Description

ImplementationVersionNameSub.

Table D.3-3 IMPLEMENTATION VERSION NAME SUB-ITEM FIELDS (A-ASSOCIATE-RQ)

10.156.2 Constructor & Destructor Documentation

10.156.2.1 ImplementationVersionNameSub()

```
gdcm::network::ImplementationVersionNameSub::ImplementationVersionNameSub ( )
```

10.156.3 Member Function Documentation

10.156.3.1 Print()

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

10.156.3.2 Read()

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

10.156.3.3 Size()

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

10.156.3.4 Write()

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

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

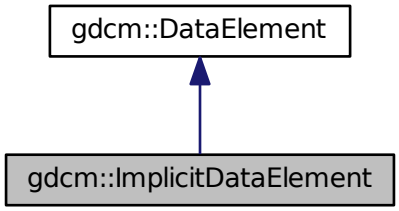
- gdcmImplementationVersionNameSub.h

10.157 gdcm::ImplicitDataElement Class Reference

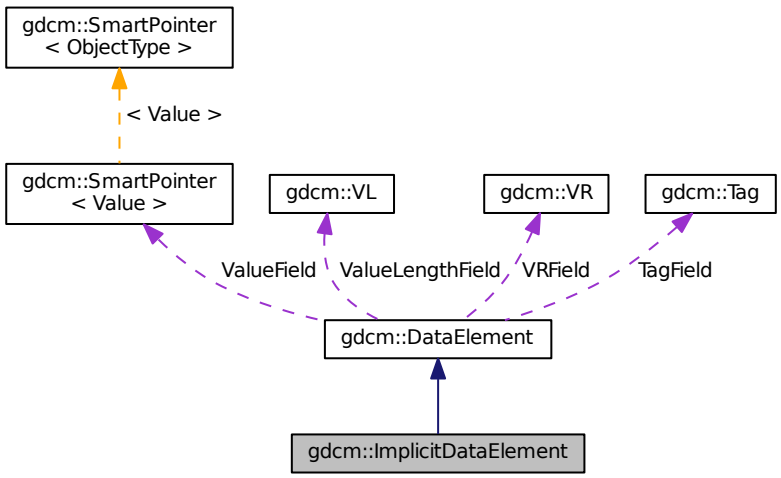
Class to represent an *Implicit VR* Data Element.

`#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

10.157.1 Detailed Description

Class to represent an *Implicit VR* Data Element.

Note

bla

Examples:

ReadExplicitLengthSQIVR.cxx.

10.157.2 Member Function Documentation

10.157.2.1 GetLength()

```
VL gdcm::ImplicitDataElement::GetLength ( ) const
```

10.157.2.2 Read()

```
template<typename TSwap >
std::istream& gdcm::ImplicitDataElement::Read (
             std::istream & is )
```

10.157.2.3 ReadPreValue()

```
template<typename TSwap >
std::istream& gdcm::ImplicitDataElement::ReadPreValue (
             std::istream & is )
```

10.157.2.4 ReadValue()

```
template<typename TSwap >
std::istream& gdcm::ImplicitDataElement::ReadValue (
             std::istream & is,
             bool readvalues = true )
```

10.157.2.5 ReadValueWithLength()

```
template<typename TSwap >
std::istream& gdcm::ImplicitDataElement::ReadValueWithLength (
             std::istream & is,
             VL & length,
             bool readvalues = true )
```

10.157.2.6 ReadWithLength()

```
template<typename TSwap >
std::istream& gdcm::ImplicitDataElement::ReadWithLength (
             std::istream & is,
             VL & length,
             bool readvalues = true )
```

10.157.2.7 Write()

```
template<typename TSwap >
const std::ostream& gdcm::ImplicitDataElement::Write (
             std::ostream & os ) const
```

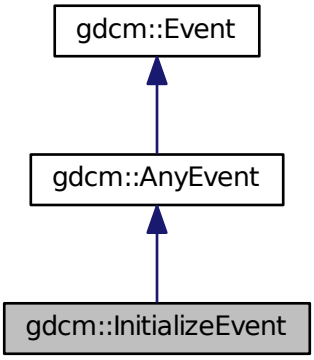
The documentation for this class was generated from the following file:

- gdcmImplicitDataElement.h

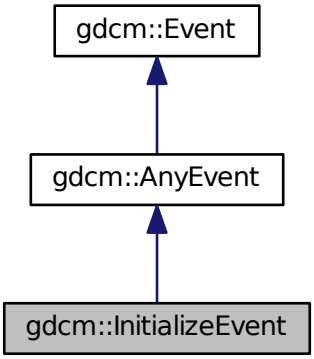
10.158 gdcm::InitializeEvent Class Reference

```
#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

10.159 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 ()
- 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)

10.159.1 Detailed Description

Class for representing a IOD.

Note

bla

See also

Dict

Examples:

TraverseModules.cxx.

10.159.2 Member Typedef Documentation

10.159.2.1 MapIODEntry

`typedef std::vector<IODEntry> gdcm::IOD::MapIODEntry`

10.159.2.2 SizeType

```
typedef MapIODEntry::size_type gdcm::IOD::SizeType
```

10.159.3 Constructor & Destructor Documentation

10.159.3.1 IOD()

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

References gdcm::operator<<().

10.159.4 Member Function Documentation

10.159.4.1 AddIODEntry()

```
void gdcm::IOD::AddIODEntry (
            const IODEntry & iode )  [inline]
```

10.159.4.2 Clear()

```
void gdcm::IOD::Clear ( )  [inline]
```

10.159.4.3 GetIODEntry()

```
const IODEntry& gdcm::IOD::GetIODEntry (
            SizeType idx ) const  [inline]
```

Examples:

TraverseModules.cxx.

10.159.4.4 GetNumberOfIODs()

```
SizeType gdcm::IOD::GetNumberOfIODs ( ) const  [inline]
```

Examples:

TraverseModules.cxx.

10.159.4.5 GetTypeFromTag()

```
Type gdcm::IOD::GetTypeFromTag (
            const Defs & defs,
            const Tag & tag ) const
```

10.159.5 Friends And Related Function Documentation

10.159.5.1 operator<<

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

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

- gdcmIOD.h

10.160 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)

10.160.1 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.

10.160.2 Constructor & Destructor Documentation

10.160.2.1 IODEntry()

```
gdcm::IODEntry::IODEntry (
            const char * name = "",
            const char * ref = "",
            const char * usag = "" )  [inline]
```

References gdcm::operator<<().

10.160.3 Member Function Documentation

10.160.3.1 GetIE()

```
const char* gdcm::IODEntry::GetIE ( ) const  [inline]
```

10.160.3.2 GetName()

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

10.160.3.3 GetRef()

```
const char* gdcm::IODEntry::GetRef ( ) const  [inline]
```

Examples:

TraverseModules.cxx.

10.160.3.4 GetUsage()

```
const char* gdcm::IODEntry::GetUsage ( ) const  [inline]
```

10.160.3.5 GetUsageType()

```
Usage::UsageType gdcm::IODEntry::GetUsageType ( ) const
```

10.160.3.6 SetIE()

```
void gdcm::IODEntry::SetIE (
            const char * ie )  [inline]
```

10.160.3.7 SetName()

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

10.160.3.8 SetRef()

```
void gdcm::IODEntry::SetRef (
            const char * ref )  [inline]
```

10.160.3.9 SetUsage()

```
void gdcm::IODEntry::SetUsage (
            const char * usag )  [inline]
```

10.160.4 Friends And Related Function Documentation

10.160.4.1 operator<<

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

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

- gdcmIODEntry.h

10.161 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 ()
- 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)

10.161.1 Detailed Description

Class for representing a IODs.

Note

bla

See also

IOD

Examples:

TraverseModules.cxx.

10.161.2 Member Typedef Documentation

10.161.2.1 IODMapType

```
typedef std::map<IODName, IOD> gdcm::IODs::IODMapType
```

10.161.2.2 IODMapTypeConstIterator

```
typedef IODMapType::const_iterator gdcm::IODs::IODMapTypeConstIterator
```

10.161.2.3 IODName

```
typedef std::string gdcm::IODs::IODName
```

10.161.3 Constructor & Destructor Documentation

10.161.3.1 IODs()

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

References gdcm::operator<<().

10.161.4 Member Function Documentation

10.161.4.1 AddIOD()

```
void gdcm::IODs::AddIOD (
            const char * name,
            const IOD & module )  [inline]
```

10.161.4.2 Begin()

```
IODMapTypeConstIterator gdcm::IODs::Begin ( ) const  [inline]
```

Examples:

TraverseModules.cxx.

10.161.4.3 Clear()

```
void gdcm::IODs::Clear ( )  [inline]
```

10.161.4.4 End()

```
IODMapTypeConstIterator gdcm::IODs::End ( ) const  [inline]
```

Examples:

TraverseModules.cxx.

10.161.4.5 GetIOD()

```
const IOD& gdcm::IODs::GetIOD (
            const char * name ) const  [inline]
```

10.161.5 Friends And Related Function Documentation

10.161.5.1 operator<<

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

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

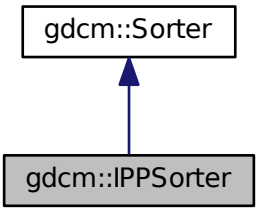
- gdcmIODs.h

10.162 gdcm::IPPSorter Class Reference

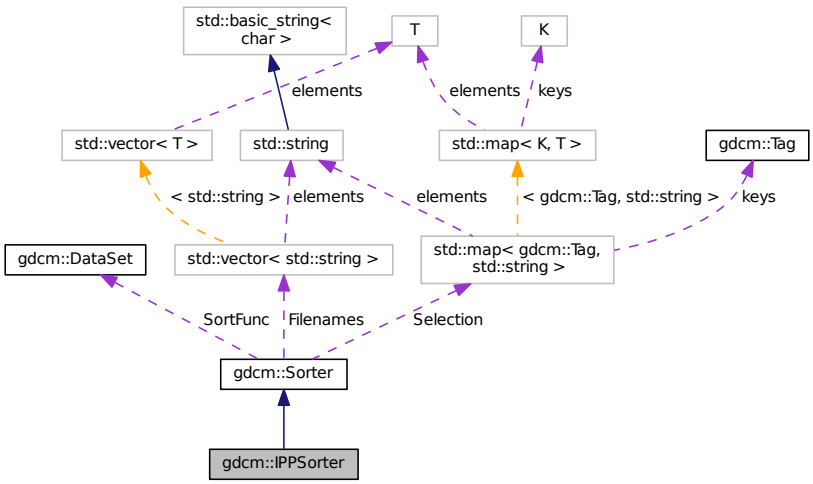
IPPSorter.

`#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)
- virtual bool Sort (std::vector< std::string > const &filenames)

Protected Attributes

- bool ComputeZSpacing
- double DirCosTolerance
- bool DropDuplicatePositions
- double ZSpacing
- double ZTolerance

Additional Inherited Members

10.162.1 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, gdcmorthoplanes.cxx, reslicesphere.cxx, and VolumeSorter.cxx.

10.162.2 Constructor & Destructor Documentation

10.162.2.1 IPPSorter()

```
gdcm::IPPSorter::IPPSorter ( )
```

10.162.3 Member Function Documentation

10.162.3.1 GetDirectionCosinesTolerance()

```
double gdcm::IPPSorter::GetDirectionCosinesTolerance ( ) const  [inline]
```

10.162.3.2 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.

10.162.3.3 GetZSpacingTolerance()

```
double gdcm::IPPSorter::GetZSpacingTolerance ( ) const  [inline]
```

10.162.3.4 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, gdcmorthoplanes.cxx, reslicesphere.cxx, and VolumeSorter.cxx.

10.162.3.5 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).

10.162.3.6 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.

10.162.3.7 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 serie, 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.

10.162.3.8 Sort()

```
virtual bool gdcm::IPPSorter::Sort (
            std::vector< std::string > const & filenames )  [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 achived. 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, gdcmorthoplanes.cxx, reslicesphere.cxx, and VolumeSorter.cxx.

10.162.4 Member Data Documentation

10.162.4.1 ComputeZSpacing

```
bool gdcm::IPPSorter::ComputeZSpacing  [protected]
```

10.162.4.2 DirCosTolerance

```
double gdcm::IPPSorter::DirCosTolerance  [protected]
```

10.162.4.3 DropDuplicatePositions

```
bool gdcm::IPPSorter::DropDuplicatePositions  [protected]
```

10.162.4.4 ZSpacing

```
double gdcm::IPPSorter::ZSpacing  [protected]
```

10.162.4.5 ZTolerance

```
double gdcm::IPPSorter::ZTolerance  [protected]
```

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

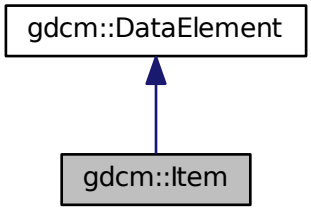
- gdcmIPPSorter.h

10.163 gdcm::Item Class Reference

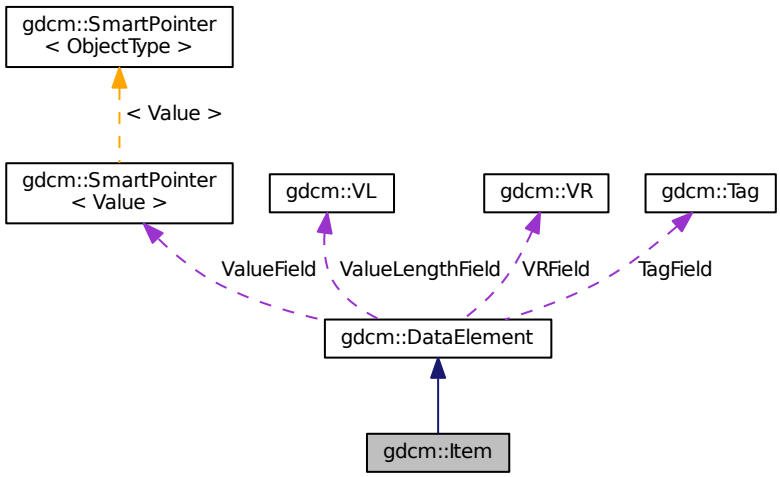
Class to represent an Item.

```
#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
- const DataSet & GetNestedDataSet () const
- DataSet & GetNestedDataSet ()
- 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

10.163.1 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 Standart 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, ExtractEncryptedContent.cxx, Extracting_All_Resolution.cxx, Fake_Image_Using_Stream_Image_Writer.cxx, gdcmrtionplan.cxx, gdcmrtplan.cxx, GenAllVR.cxx, GenFakeIdentifyFile.cxx, GenLongSeqs.cxx, GenSeqs.cxx, GetSequenceUltrasound.cxx, GetSubSequenceData.cxx, and LargeVRDSExplicit.cxx.

10.163.2 Constructor & Destructor Documentation

10.163.2.1 Item() [1/2]

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

References gdcm::operator<<().

10.163.2.2 Item() [2/2]

```
gdcm::Item::Item (
            Item const & val )  [inline]
```

10.163.3 Member Function Documentation

10.163.3.1 Clear()

```
void gdcm::Item::Clear ( )  [inline]
```

Referenced by gdcm::SequenceOfItems::Read().

10.163.3.2 FindDataElement()

```
bool gdcm::Item::FindDataElement (
            const Tag & t ) const  [inline]
```

Examples:

ReadAndDumpDICOMDIR.cxx.

10.163.3.3 GetDataElement()

```
const DataElement& gdcm::Item::GetDataElement (
            const Tag & t ) const  [inline]
```

Examples:

ReadAndDumpDICOMDIR.cxx.

10.163.3.4 GetLength()

```
template<typename TDE >
VL gdcm::Item::GetLength ( ) const
```

10.163.3.5 GetNestedDataSet() [1/2]

```
const DataSet& gdcm::Item::GetNestedDataSet ( ) const  [inline]
```

Examples:

ChangeSequenceUltrasound.cxx, DumpExamCard.cxx, DumpGEMSMovieGroup.cxx, DumpPhilipsECHO.cxx, DumpToshibaDTI.cxx, ExtractEncryptedContent.cxx, Extracting_All_Resolution.cxx, Fake_Image_Using_Stream_Image_Writer.cxx, gdcmrtionplan.cxx, gdcmrtplan.cxx, GenAllVR.cxx, GenFakeIdentifyFile.cxx, GenSeqs.cxx, GetSequenceUltrasound.cxx, GetSubSequenceData.cxx, and LargeVRDSExplicit.cxx.

Referenced by gdcm::SequenceOfItems::Read().

10.163.3.6 GetNestedDataSet() [2/2]

```
DataSet& gdcm::Item::GetNestedDataSet ( )  [inline]
```

10.163.3.7 InsertDataElement()

```
void gdcm::Item::InsertDataElement (
            const DataElement & de )  [inline]
```

10.163.3.8 Read()

```
template<typename TDE , typename TSwap >
std::istream& gdcm::Item::Read (
            std::istream & is )  [inline]
```

References gdcm::DataSet::Clear(), gdcmDebugMacro, gdcmErrorMacro, gdcmWarningMacro, and gdcm::DataSet::IsEmpty().

Referenced by gdcm::SequenceOfItems::Read().

10.163.3.9 SetNestedDataSet()

```
void gdcm::Item::SetNestedDataSet (
            const DataSet & nested )  [inline]
```

10.163.3.10 Write()

```
template<typename TDE , typename TSwap >
const std::ostream& gdcm::Item::Write (
            std::ostream & os ) const  [inline]
```

References gdcmWarningMacro, gdcm::VL::GetLength(), gdcm::VL::Write(), and gdcm::Tag::Write().

10.163.4 Friends And Related Function Documentation

10.163.4.1 operator<<

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

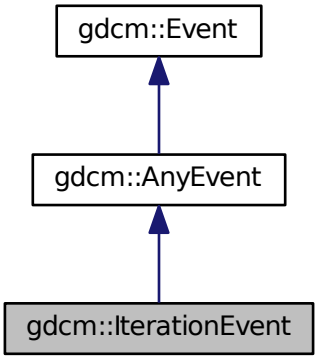
The documentation for this class was generated from the following file:

- gdcmItem.h

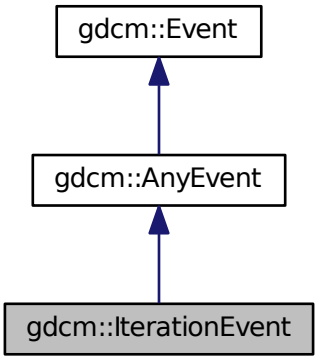
10.164 gdcm::IterationEvent Class Reference

`#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:

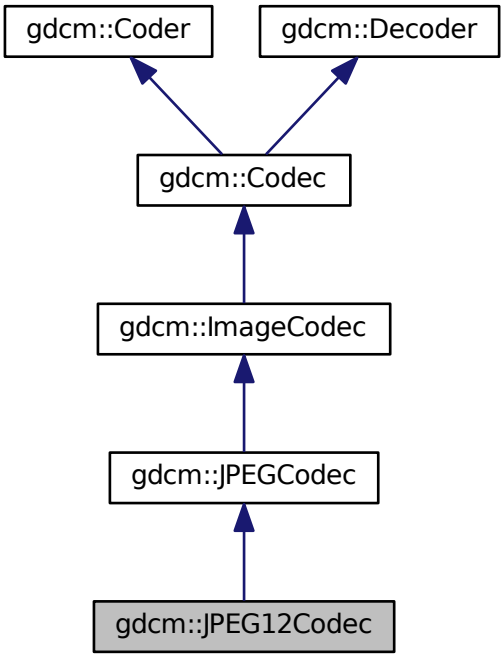
- gdcmEvent.h

10.165 gdcm::JPEG12Codec Class Reference

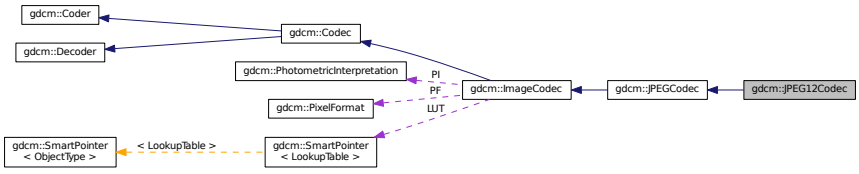
Class to do JPEG 12bits (lossy & lossless)

`#include <gdcmJPEG12Codec.h>`

Inheritance diagram for gdcm::JPEG12Codec:



Collaboration diagram for gdcm::JPEG12Codec:



Public Member Functions

- JPEG12Codec ()
- ∼JPEG12Codec ()
- bool DecodeByStreams (std::istream &is, std::ostream &os)
- bool GetHeaderInfo (std::istream &is, TransferSyntax &ts)
- bool InternalCode (const char ∗input, unsigned long len, std::ostream &os)

Protected Member Functions

- virtual bool EncodeBuffer (std::ostream &os, const char ∗data, size_t datalen)
- bool IsStateSuspension () const

Additional Inherited Members

10.165.1 Detailed Description

Class to do JPEG 12bits (lossy & lossless)

Note

internal class

10.165.2 Constructor & Destructor Documentation

10.165.2.1 JPEG12Codec()

```
gdcm::JPEG12Codec::JPEG12Codec ( )
```

10.165.2.2 ∼JPEG12Codec()

```
gdcm::JPEG12Codec::∼JPEG12Codec ( )
```

10.165.3 Member Function Documentation

10.165.3.1 DecodeByStreams()

```
bool gdcm::JPEG12Codec::DecodeByStreams (
            std::istream & is,
            std::ostream & os ) [virtual]
```

Reimplemented from gdcm::ImageCodec.

10.165.3.2 EncodeBuffer()

```
virtual bool gdcm::JPEG12Codec::EncodeBuffer (
            std::ostream & os,
            const char * data,
            size_t datalen )  [protected], [virtual]
```

Reimplemented from gdcm::JPEGCodec.

10.165.3.3 GetHeaderInfo()

```
bool gdcm::JPEG12Codec::GetHeaderInfo (
            std::istream & is,
            TransferSyntax & ts )  [virtual]
```

Reimplemented from gdcm::JPEGCodec.

10.165.3.4 InternalCode()

```
bool gdcm::JPEG12Codec::InternalCode (
            const char * input,
            unsigned long len,
            std::ostream & os )  [virtual]
```

Reimplemented from gdcm::Coder.

10.165.3.5 IsStateSuspension()

```
bool gdcm::JPEG12Codec::IsStateSuspension ( ) const  [protected], [virtual]
```

Reimplemented from gdcm::JPEGCodec.

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

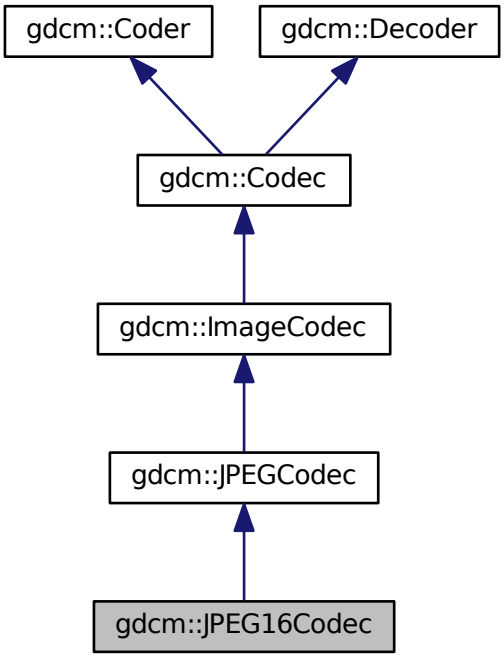
- gdcmJPEG12Codec.h

10.166 gdcm::JPEG16Codec Class Reference

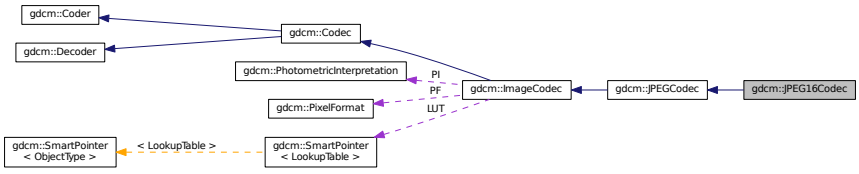
Class to do JPEG 16bits (lossless)

`#include <gdcmJPEG16Codec.h>`

Inheritance diagram for gdcm::JPEG16Codec:



Collaboration diagram for gdcm::JPEG16Codec:



Public Member Functions

- JPEG16Codec ()
- ~JPEG16Codec ()
- bool DecodeByStreams (std::istream &is, std::ostream &os)
- bool GetHeaderInfo (std::istream &is, TransferSyntax &ts)
- bool InternalCode (const char *input, unsigned long len, std::ostream &os)

Protected Member Functions

- virtual bool EncodeBuffer (std::ostream &os, const char *data, size_t datalen)
- bool IsStateSuspension () const

Additional Inherited Members

10.166.1 Detailed Description

Class to do JPEG 16bits (lossless)

Note

internal class

10.166.2 Constructor & Destructor Documentation

10.166.2.1 JPEG16Codec()

```
gdcm::JPEG16Codec::JPEG16Codec ( )
```

10.166.2.2 ~JPEG16Codec()

```
gdcm::JPEG16Codec::~JPEG16Codec ( )
```

10.166.3 Member Function Documentation

10.166.3.1 DecodeByStreams()

```
bool gdcm::JPEG16Codec::DecodeByStreams (
            std::istream & is,
            std::ostream & os )  [virtual]
```

Reimplemented from gdcm::ImageCodec.

10.166.3.2 EncodeBuffer()

```
virtual bool gdcm::JPEG16Codec::EncodeBuffer (
            std::ostream & os,
            const char * data,
            size_t datalen )  [protected], [virtual]
```

Reimplemented from gdcm::JPEGCodec.

10.166.3.3 GetHeaderInfo()

```
bool gdcm::JPEG16Codec::GetHeaderInfo (
            std::istream & is,
            TransferSyntax & ts )  [virtual]
```

Reimplemented from gdcm::JPEGCodec.

10.166.3.4 InternalCode()

```
bool gdcm::JPEG16Codec::InternalCode (
            const char * input,
            unsigned long len,
            std::ostream & os )  [virtual]
```

Reimplemented from gdcm::Coder.

10.166.3.5 IsStateSuspension()

```
bool gdcm::JPEG16Codec::IsStateSuspension ( ) const  [protected], [virtual]
```

Reimplemented from gdcm::JPEGCodec.

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

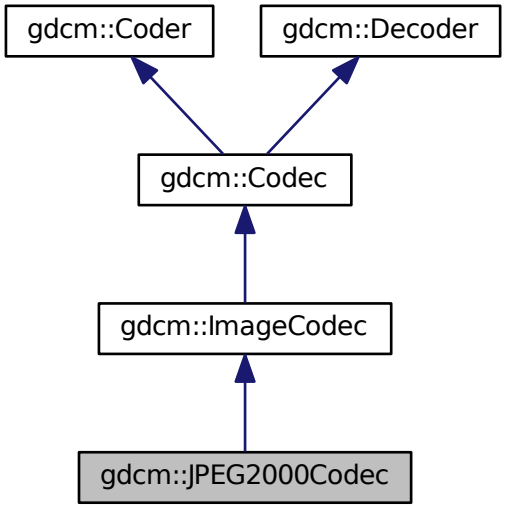
- gdcmJPEG16Codec.h

10.167 gdcm::JPEG2000Codec Class Reference

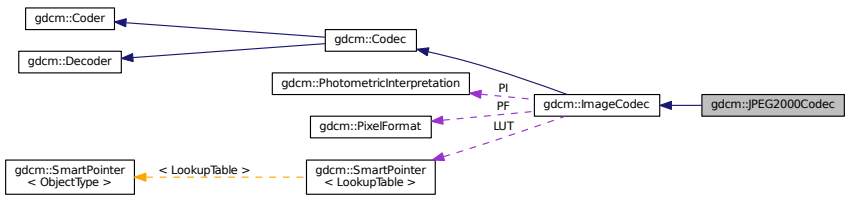
Class to do JPEG 2000.

`#include <gdcmJPEG2000Codec.h>`

Inheritance diagram for gdcm::JPEG2000Codec:



Collaboration diagram for gdcm::JPEG2000Codec:



Public Member Functions

- JPEG2000Codec ()
- ~JPEG2000Codec ()
- bool CanCode (TransferSyntax const &ts) const

Return whether this coder support this transfer syntax (can code it)

- bool CanDecode (TransferSyntax const &ts) const

  Return whether this decoder support this transfer syntax (can decode it)

- virtual ImageCodec * Clone () const
- bool Code (DataElement const &in, DataElement &out)

  Code.

- bool Decode (DataElement const &is, DataElement &os)

  Decode.

- virtual bool GetHeaderInfo (std::istream &is, TransferSyntax &ts)
- double GetQuality (unsigned int idx=0) const
- double GetRate (unsigned int idx=0) const
- void SetNumberOfResolutions (unsigned int nres)
- 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)
- bool AppendRowEncode (std::ostream &out, const char *data, size_t datalen)
- bool DecodeByStreams (std::istream &is, std::ostream &os)
- 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 ()
- bool IsRowEncoder ()
- bool StartEncode (std::ostream &)
- bool StopEncode (std::ostream &)

Friends

- class Bitmap
- class ImageRegionReader

Additional Inherited Members

10.167.1 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

10.167.2 Constructor & Destructor Documentation

10.167.2.1 JPEG2000Codec()

```
gdcm::JPEG2000Codec::JPEG2000Codec ( )
```

10.167.2.2 ~JPEG2000Codec()

```
gdcm::JPEG2000Codec::~JPEG2000Codec ( )
```

10.167.3 Member Function Documentation

10.167.3.1 AppendFrameEncode()

```
bool gdcm::JPEG2000Codec::AppendFrameEncode (
            std::ostream & out,
            const char * data,
            size_t datalen )  [protected], [virtual]
```

Reimplemented from gdcm::ImageCodec.

10.167.3.2 AppendRowEncode()

```
bool gdcm::JPEG2000Codec::AppendRowEncode (
            std::ostream & out,
            const char * data,
            size_t datalen )  [protected], [virtual]
```

Reimplemented from gdcm::ImageCodec.

10.167.3.3 CanCode()

```
bool gdcm::JPEG2000Codec::CanCode (
            TransferSyntax const &  ) const  [virtual]
```

Return whether this coder support this transfer syntax (can code it)

Reimplemented from gdcm::ImageCodec.

10.167.3.4 CanDecode()

```
bool gdcm::JPEG2000Codec::CanDecode (
            TransferSyntax const &  ) const  [virtual]
```

Return whether this decoder support this transfer syntax (can decode it)

Reimplemented from gdcm::ImageCodec.

10.167.3.5 Clone()

```
virtual ImageCodec* gdcm::JPEG2000Codec::Clone ( ) const  [virtual]
```

Implements gdcm::ImageCodec.

10.167.3.6 Code()

```
bool gdcm::JPEG2000Codec::Code (
            DataElement const & in_,
            DataElement & out_ )  [virtual]
```

Code.

Reimplemented from gdcm::Coder.

10.167.3.7 Decode()

```
bool gdcm::JPEG2000Codec::Decode (
            DataElement const & ,
            DataElement &  )  [virtual]
```

Decode.

Reimplemented from gdcm::ImageCodec.

10.167.3.8 DecodeByStreams()

```
bool gdcm::JPEG2000Codec::DecodeByStreams (
            std::istream & is,
            std::ostream & os )  [protected], [virtual]
```

Reimplemented from gdcm::ImageCodec.

10.167.3.9 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]
```

10.167.3.10 GetHeaderInfo()

```
virtual bool gdcm::JPEG2000Codec::GetHeaderInfo (
            std::istream & is,
            TransferSyntax & ts )  [virtual]
```

Reimplemented from gdcm::ImageCodec.

10.167.3.11 GetQuality()

```
double gdcm::JPEG2000Codec::GetQuality (
            unsigned int idx = 0 ) const
```

10.167.3.12 GetRate()

```
double gdcm::JPEG2000Codec::GetRate (
            unsigned int idx = 0 ) const
```

10.167.3.13 IsFrameEncoder()

```
bool gdcm::JPEG2000Codec::IsFrameEncoder ( )  [protected], [virtual]
```

Reimplemented from gdcm::ImageCodec.

10.167.3.14 IsRowEncoder()

```
bool gdcm::JPEG2000Codec::IsRowEncoder ( )  [protected], [virtual]
```

Reimplemented from gdcm::ImageCodec.

10.167.3.15 SetNumberOfResolutions()

```
void gdcm::JPEG2000Codec::SetNumberOfResolutions (
            unsigned int nres )
```

10.167.3.16 SetQuality()

```
void gdcm::JPEG2000Codec::SetQuality (
            unsigned int idx,
            double q )
```

10.167.3.17 SetRate()

```
void gdcm::JPEG2000Codec::SetRate (
            unsigned int idx,
            double rate )
```

10.167.3.18 SetReversible()

```
void gdcm::JPEG2000Codec::SetReversible (
            bool res )
```

10.167.3.19 SetTileSize()

```
void gdcm::JPEG2000Codec::SetTileSize (
            unsigned int tx,
            unsigned int ty )
```

10.167.3.20 StartEncode()

```
bool gdcm::JPEG2000Codec::StartEncode (
            std::ostream & ) [protected], [virtual]
```

Reimplemented from gdcm::ImageCodec.

10.167.3.21 StopEncode()

```
bool gdcm::JPEG2000Codec::StopEncode (
            std::ostream & ) [protected], [virtual]
```

Reimplemented from gdcm::ImageCodec.

10.167.4 Friends And Related Function Documentation

10.167.4.1 Bitmap

```
friend class Bitmap [friend]
```

10.167.4.2 ImageRegionReader

```
friend class ImageRegionReader [friend]
```

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

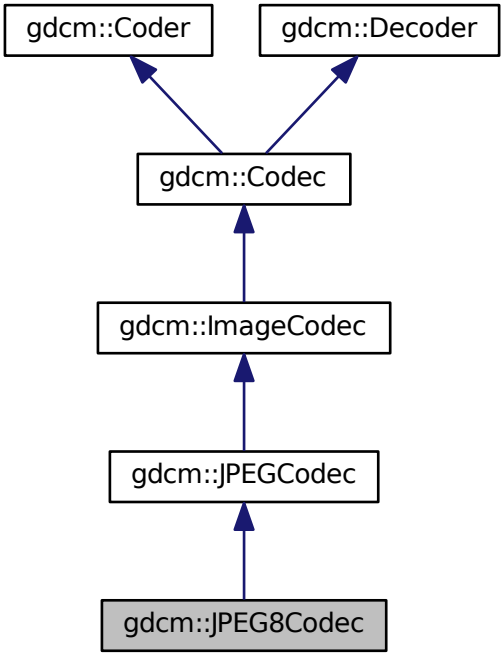
- gdcmJPEG2000Codec.h

10.168 gdcm::JPEG8Codec Class Reference

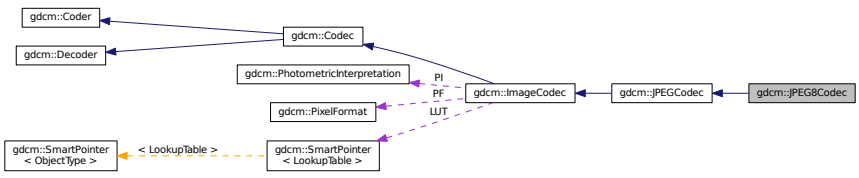
Class to do JPEG 8bits (lossy & lossless)

`#include <gdcmJPEG8Codec.h>`

Inheritance diagram for gdcm::JPEG8Codec:



Collaboration diagram for gdcm::JPEG8Codec:



Public Member Functions

- JPEG8Codec ()
- ∼JPEG8Codec ()
- bool DecodeByStreams (std::istream &is, std::ostream &os)
- bool GetHeaderInfo (std::istream &is, TransferSyntax &ts)
- bool InternalCode (const char ∗input, unsigned long len, std::ostream &os)

Protected Member Functions

- virtual bool EncodeBuffer (std::ostream &os, const char ∗data, size_t datalen)
- bool IsStateSuspension () const

Additional Inherited Members

10.168.1 Detailed Description

Class to do JPEG 8bits (lossy & lossless)

Note

internal class

10.168.2 Constructor & Destructor Documentation

10.168.2.1 JPEG8Codec()

```
gdcm::JPEG8Codec::JPEG8Codec ( )
```

10.168.2.2 ∼JPEG8Codec()

```
gdcm::JPEG8Codec::∼JPEG8Codec ( )
```

10.168.3 Member Function Documentation

10.168.3.1 DecodeByStreams()

```
bool gdcm::JPEG8Codec::DecodeByStreams (
            std::istream & is,
            std::ostream & os ) [virtual]
```

Reimplemented from gdcm::ImageCodec.

10.168.3.2 EncodeBuffer()

```
virtual bool gdcm::JPEG8Codec::EncodeBuffer (
            std::ostream & os,
            const char * data,
            size_t datalen )  [protected], [virtual]
```

Reimplemented from gdcm::JPEGCodec.

10.168.3.3 GetHeaderInfo()

```
bool gdcm::JPEG8Codec::GetHeaderInfo (
            std::istream & is,
            TransferSyntax & ts )  [virtual]
```

Reimplemented from gdcm::JPEGCodec.

10.168.3.4 InternalCode()

```
bool gdcm::JPEG8Codec::InternalCode (
            const char * input,
            unsigned long len,
            std::ostream & os )  [virtual]
```

Reimplemented from gdcm::Coder.

10.168.3.5 IsStateSuspension()

```
bool gdcm::JPEG8Codec::IsStateSuspension ( ) const  [protected], [virtual]
```

Reimplemented from gdcm::JPEGCodec.

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

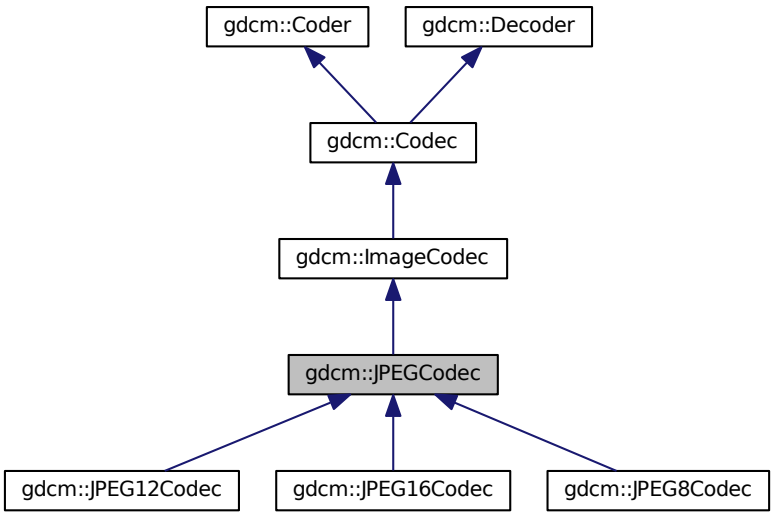
- gdcmJPEG8Codec.h

10.169 gdcm::JPEGCodec Class Reference

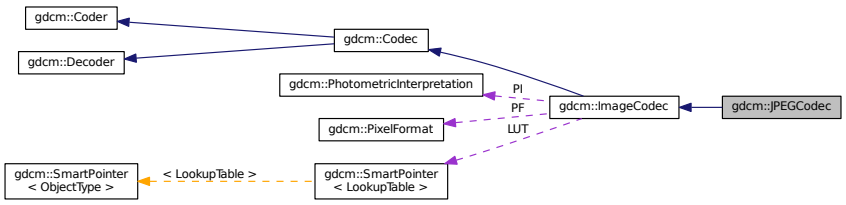
JPEG codec.

`#include <gdcmJPEGCodec.h>`

Inheritance diagram for gdcm::JPEGCodec:



Collaboration diagram for gdcm::JPEGCodec:



Public Member Functions

- JPEGCodec ()
- ~JPEGCodec ()
- bool CanCode (TransferSyntax const &ts) const

  Return whether this coder support this transfer syntax (can code it)

- bool CanDecode (TransferSyntax const &ts) const

  Return whether this decoder support this transfer syntax (can decode it)
- virtual ImageCodec * Clone () const
- bool Code (DataElement const &in, DataElement &out)

  Compress into JPEG.
- void ComputeOffsetTable (bool b)

  Compute the offset table:
- bool Decode (DataElement const &is, DataElement &os)

  Decode.
- virtual bool EncodeBuffer (std::ostream &out, const char *inbuffer, size_t inlen)
- virtual bool GetHeaderInfo (std::istream &is, TransferSyntax &ts)
- bool GetLossless () const
- double GetQuality () const
- void SetLossless (bool l)
- void SetPixelFormat (PixelFormat const &pf)
- void SetQuality (double q)

Protected Member Functions

- bool AppendFrameEncode (std::ostream &out, const char *data, size_t datalen)
- bool AppendRowEncode (std::ostream &out, const char *data, size_t datalen)
- bool DecodeByStreams (std::istream &is, std::ostream &os)
- 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 ()
- bool IsRowEncoder ()
- virtual bool IsStateSuspension () const
- bool IsValid (PhotometricInterpretation const &pi)
- void SetBitSample (int bit)
- bool StartEncode (std::ostream &)
- bool StopEncode (std::ostream &)

Protected Attributes

- int BitSample
- int Quality

Friends

- class ImageRegionReader

Additional Inherited Members

10.169.1 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/6b93af410f8c92

Examples:

GetJPEGSamplePrecision.cxx.

10.169.2 Constructor & Destructor Documentation

10.169.2.1 JPEGCodec()

```
gdcm::JPEGCodec::JPEGCodec ( )
```

10.169.2.2 ~JPEGCodec()

```
gdcm::JPEGCodec::~JPEGCodec ( )
```

10.169.3 Member Function Documentation

10.169.3.1 AppendFrameEncode()

```
bool gdcm::JPEGCodec::AppendFrameEncode (
            std::ostream & out,
            const char * data,
            size_t datalen )  [protected], [virtual]
```

Reimplemented from gdcm::ImageCodec.

10.169.3.2 AppendRowEncode()

```
bool gdcm::JPEGCodec::AppendRowEncode (
            std::ostream & out,
            const char * data,
            size_t datalen ) [protected], [virtual]
```

Reimplemented from gdcm::ImageCodec.

10.169.3.3 CanCode()

```
bool gdcm::JPEGCodec::CanCode (
            TransferSyntax const & ) const [virtual]
```

Return whether this coder support this transfer syntax (can code it)

Reimplemented from gdcm::ImageCodec.

10.169.3.4 CanDecode()

```
bool gdcm::JPEGCodec::CanDecode (
            TransferSyntax const & ) const [virtual]
```

Return whether this decoder support this transfer syntax (can decode it)

Reimplemented from gdcm::ImageCodec.

10.169.3.5 Clone()

```
virtual ImageCodec* gdcm::JPEGCodec::Clone ( ) const [virtual]
```

Implements gdcm::ImageCodec.

10.169.3.6 Code()

```
bool gdcm::JPEGCodec::Code (
            DataElement const & in,
            DataElement & out ) [virtual]
```

Compress into JPEG.

Reimplemented from gdcm::Coder.

10.169.3.7 ComputeOffsetTable()

```
void gdcm::JPEGCodec::ComputeOffsetTable (
            bool b )
```

Compute the offset table:

10.169.3.8 Decode()

```
bool gdcm::JPEGCodec::Decode (
            DataElement const & ,
            DataElement &  )  [virtual]
```

Decode.

Reimplemented from gdcm::ImageCodec.

10.169.3.9 DecodeByStreams()

```
bool gdcm::JPEGCodec::DecodeByStreams (
            std::istream & is,
            std::ostream & os )  [protected], [virtual]
```

Reimplemented from gdcm::ImageCodec.

10.169.3.10 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]
```

10.169.3.11 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.

10.169.3.12 GetHeaderInfo()

```
virtual bool gdcm::JPEGCodec::GetHeaderInfo (
            std::istream & is,
            TransferSyntax & ts )  [virtual]
```

Reimplemented from gdcm::ImageCodec.

Reimplemented in gdcm::JPEG12Codec, gdcm::JPEG16Codec, and gdcm::JPEG8Codec.

Examples:

GetJPEGSamplePrecision.cxx.

10.169.3.13 GetLossless()

```
bool gdcm::JPEGCodec::GetLossless ( ) const
```

10.169.3.14 GetQuality()

```
double gdcm::JPEGCodec::GetQuality ( ) const
```

10.169.3.15 IsFrameEncoder()

```
bool gdcm::JPEGCodec::IsFrameEncoder ( )  [protected], [virtual]
```

Reimplemented from gdcm::ImageCodec.

10.169.3.16 IsRowEncoder()

```
bool gdcm::JPEGCodec::IsRowEncoder ( )  [protected], [virtual]
```

Reimplemented from gdcm::ImageCodec.

10.169.3.17 IsStateSuspension()

```
virtual bool gdcm::JPEGCodec::IsStateSuspension ( ) const  [protected], [virtual]
```

Reimplemented in gdcm::JPEG12Codec, gdcm::JPEG16Codec, and gdcm::JPEG8Codec.

10.169.3.18 IsValid()

```
bool gdcm::JPEGCodec::IsValid (
            PhotometricInterpretation const & pi )  [protected], [virtual]
```

Reimplemented from gdcm::ImageCodec.

10.169.3.19 SetBitSample()

```
void gdcm::JPEGCodec::SetBitSample (
            int bit )  [protected]
```

10.169.3.20 SetLossless()

```
void gdcm::JPEGCodec::SetLossless (
            bool l )
```

10.169.3.21 SetPixelFormat()

```
void gdcm::JPEGCodec::SetPixelFormat (
            PixelFormat const & pf )  [virtual]
```

Reimplemented from gdcm::ImageCodec.

Examples:

GetJPEGSamplePrecision.cxx.

10.169.3.22 SetQuality()

```
void gdcm::JPEGCodec::SetQuality (
            double q )
```

10.169.3.23 StartEncode()

```
bool gdcm::JPEGCodec::StartEncode (
            std::ostream &  )  [protected], [virtual]
```

Reimplemented from gdcm::ImageCodec.

10.169.3.24 StopEncode()

```
bool gdcm::JPEGCodec::StopEncode (
            std::ostream &  )  [protected], [virtual]
```

Reimplemented from gdcm::ImageCodec.

10.169.4 Friends And Related Function Documentation

10.169.4.1 ImageRegionReader

```
friend class ImageRegionReader  [friend]
```

10.169.5 Member Data Documentation

10.169.5.1 BitSample

```
int gdcm::JPEGCodec::BitSample  [protected]
```

10.169.5.2 Quality

```
int gdcm::JPEGCodec::Quality  [protected]
```

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

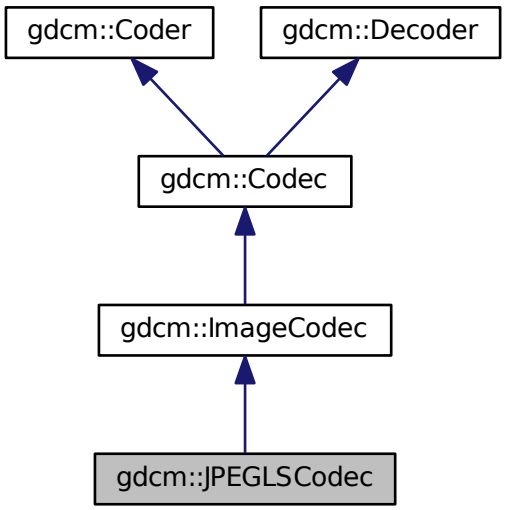
- gdcmJPEGCodec.h

10.170 gdcm::JPEGLSCodec Class Reference

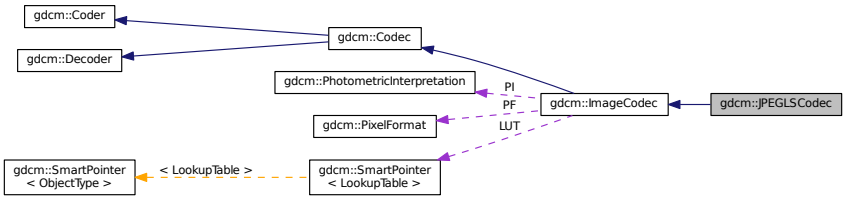
JPEG-LS.

```
#include <gdcmJPEGLSCodec.h>
```

Inheritance diagram for gdcm::JPEGLSCodec:



Collaboration diagram for gdcm::JPEGLSCodec:



Public Member Functions

- JPEGLSCodec ()
- ~JPEGLSCodec ()
- bool CanCode (TransferSyntax const &ts) const

  Return whether this coder support this transfer syntax (can code it)
- bool CanDecode (TransferSyntax const &ts) const

  Return whether this decoder support this transfer syntax (can decode it)
- virtual ImageCodec * Clone () const
- bool Code (DataElement const &in, DataElement &out)

  Code.

- bool Decode (DataElement const &is, DataElement &os)

  Decode.
- 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)
- unsigned long GetBufferLength () const
- bool GetHeaderInfo (std::istream &is, TransferSyntax &ts)
- 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)
- bool AppendRowEncode (std::ostream &out, const char *data, size_t datalen)
- 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 ()
- bool IsRowEncoder ()
- bool StartEncode (std::ostream &)
- bool StopEncode (std::ostream &)

Friends

- class ImageRegionReader

Additional Inherited Members

10.170.1 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`

10.170.2 Constructor & Destructor Documentation

10.170.2.1 JPEGLSCodec()

```
gdcm::JPEGLSCodec::JPEGLSCodec ( )
```

10.170.2.2 ~JPEGLSCodec()

```
gdcm::JPEGLSCodec::~JPEGLSCodec ( )
```

10.170.3 Member Function Documentation

10.170.3.1 AppendFrameEncode()

```
bool gdcm::JPEGLSCodec::AppendFrameEncode (
            std::ostream & out,
            const char * data,
            size_t datalen )  [protected], [virtual]
```

Reimplemented from gdcm::ImageCodec.

10.170.3.2 AppendRowEncode()

```
bool gdcm::JPEGLSCodec::AppendRowEncode (
            std::ostream & out,
            const char * data,
            size_t datalen )  [protected], [virtual]
```

Reimplemented from gdcm::ImageCodec.

10.170.3.3 CanCode()

```
bool gdcm::JPEGLSCodec::CanCode (
            TransferSyntax const &  ) const  [virtual]
```

Return whether this coder support this transfer syntax (can code it)

Reimplemented from gdcm::ImageCodec.

10.170.3.4 CanDecode()

```
bool gdcm::JPEGLSCodec::CanDecode (
            TransferSyntax const &  ) const  [virtual]
```

Return whether this decoder support this transfer syntax (can decode it)

Reimplemented from gdcm::ImageCodec.

10.170.3.5 Clone()

```
virtual ImageCodec* gdcm::JPEGLSCodec::Clone ( ) const  [virtual]
```

Implements gdcm::ImageCodec.

10.170.3.6 Code()

```
bool gdcm::JPEGLSCodec::Code (
            DataElement const & in_,
            DataElement & out_ )  [virtual]
```

Code.

Reimplemented from gdcm::Coder.

10.170.3.7 Decode() [1/2]

```
bool gdcm::JPEGLSCodec::Decode (
            DataElement const & ,
            DataElement &  )  [virtual]
```

Decode.

Reimplemented from gdcm::ImageCodec.

10.170.3.8 Decode() [2/2]

```
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 )
```

10.170.3.9 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]
```

10.170.3.10 GetBufferLength()

```
unsigned long gdcm::JPEGLSCodec::GetBufferLength ( ) const  [inline]
```

10.170.3.11 GetHeaderInfo()

```
bool gdcm::JPEGLSCodec::GetHeaderInfo (
            std::istream & is,
            TransferSyntax & ts )  [virtual]
```

Reimplemented from gdcm::ImageCodec.

10.170.3.12 GetLossless()

```
bool gdcm::JPEGLSCodec::GetLossless ( ) const
```

10.170.3.13 IsFrameEncoder()

```
bool gdcm::JPEGLSCodec::IsFrameEncoder ( )  [protected], [virtual]
```

Reimplemented from gdcm::ImageCodec.

10.170.3.14 IsRowEncoder()

```
bool gdcm::JPEGLSCodec::IsRowEncoder ( )  [protected], [virtual]
```

Reimplemented from gdcm::ImageCodec.

10.170.3.15 SetBufferLength()

```
void gdcm::JPEGLSCodec::SetBufferLength (
            unsigned long l )  [inline]
```

10.170.3.16 SetLossless()

```
void gdcm::JPEGLSCodec::SetLossless (
            bool l )
```

10.170.3.17 SetLossyError()

```
void gdcm::JPEGLSCodec::SetLossyError (
            int error )
```

[0-3] generally

10.170.3.18 StartEncode()

```
bool gdcm::JPEGLSCodec::StartEncode (
            std::ostream &  )  [protected], [virtual]
```

Reimplemented from gdcm::ImageCodec.

10.170.3.19 StopEncode()

```
bool gdcm::JPEGLSCodec::StopEncode (
            std::ostream &  )  [protected], [virtual]
```

Reimplemented from gdcm::ImageCodec.

10.170.4 Friends And Related Function Documentation

10.170.4.1 ImageRegionReader

```
friend class ImageRegionReader  [friend]
```

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

- gdcmJPEGLSCodec.h

10.171 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)

10.171.1 Detailed Description

Examples:

QIDO-RS.cxx.

10.171.2 Constructor & Destructor Documentation

10.171.2.1 JSON()

```
gdcm::JSON::JSON ( )
```

10.171.2.2 ~JSON()

```
gdcm::JSON::~JSON ( )
```

10.171.3 Member Function Documentation

10.171.3.1 Code()

```
bool gdcm::JSON::Code (
            DataSet const & in,
            std::ostream & os )
```

Examples:

QIDO-RS.cxx.

10.171.3.2 Decode()

```
bool gdcm::JSON::Decode (
            std::istream & is,
            DataSet & out )
```

Examples:

QIDO-RS.cxx.

10.171.3.3 GetPrettyPrint()

```
bool gdcm::JSON::GetPrettyPrint ( ) const
```

10.171.3.4 PrettyPrintOff()

```
void gdcm::JSON::PrettyPrintOff ( )
```

10.171.3.5 PrettyPrintOn()

```
void gdcm::JSON::PrettyPrintOn ( )
```

Examples:

QIDO-RS.cxx.

10.171.3.6 SetPrettyPrint()

```
void gdcm::JSON::SetPrettyPrint (
            bool onoff )
```

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

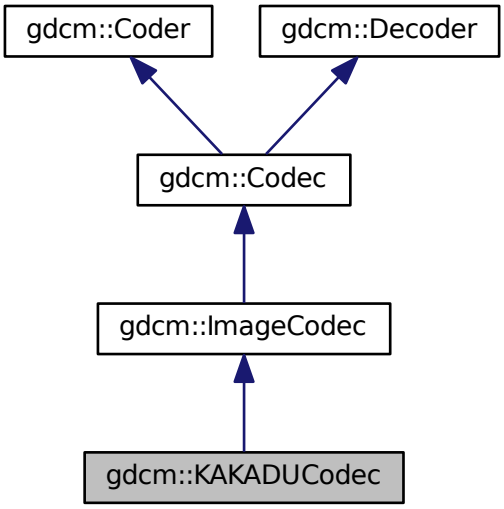
- gdcmJSON.h

10.172 gdcm::KAKADUCodec Class Reference

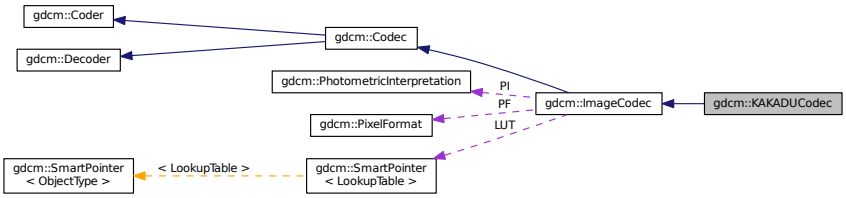
KAKADUCodec.

```
#include <gdcmKAKADUCodec.h>
```

Inheritance diagram for gdcm::KAKADUCodec:



Collaboration diagram for gdcm::KAKADUCodec:



Public Member Functions

- KAKADUCodec ()
- ~KAKADUCodec ()
- bool CanCode (TransferSyntax const &ts) const

  Return whether this coder support this transfer syntax (can code it)
- bool CanDecode (TransferSyntax const &ts) const

  Return whether this decoder support this transfer syntax (can decode it)
- virtual ImageCodec * Clone () const
- bool Code (DataElement const &in, DataElement &out)

  Code.
- bool Decode (DataElement const &is, DataElement &os)

  Decode.

Additional Inherited Members

10.172.1 Detailed Description

KAKADUCodec.

10.172.2 Constructor & Destructor Documentation

10.172.2.1 KAKADUCodec()

```
gdcm::KAKADUCodec::KAKADUCodec ( )
```

10.172.2.2 ~KAKADUCodec()

```
gdcm::KAKADUCodec::~KAKADUCodec ( )
```

10.172.3 Member Function Documentation

10.172.3.1 CanCode()

```
bool gdcm::KAKADUCodec::CanCode (
            TransferSyntax const &  ) const  [virtual]
```

Return whether this coder support this transfer syntax (can code it)

Reimplemented from gdcm::ImageCodec.

10.172.3.2 CanDecode()

```
bool gdcm::KAKADUCodec::CanDecode (
            TransferSyntax const &  ) const  [virtual]
```

Return whether this decoder support this transfer syntax (can decode it)

Reimplemented from gdcm::ImageCodec.

10.172.3.3 Clone()

```
virtual ImageCodec* gdcm::KAKADUCodec::Clone ( ) const  [virtual]
```

Implements gdcm::ImageCodec.

10.172.3.4 Code()

```
bool gdcm::KAKADUCodec::Code (
            DataElement const & in_,
            DataElement & out_ )  [virtual]
```

Code.

Reimplemented from gdcm::Coder.

10.172.3.5 Decode()

```
bool gdcm::KAKADUCodec::Decode (
            DataElement const & ,
            DataElement &  )  [virtual]
```

Decode.

Reimplemented from gdcm::ImageCodec.

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

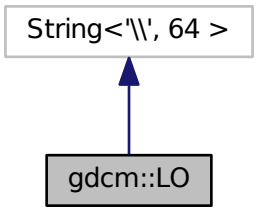
- gdcmKAKADUCodec.h

10.173 gdcm::LO Class Reference

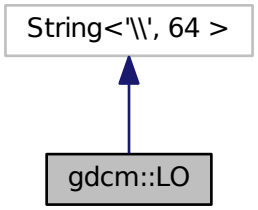
LO.

`#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 value_type *s)
- LO (const value_type *s, size_type n)
- LO (const Superclass &s, size_type pos=0, size_type n=npos)
- bool IsValid () const

10.173.1 Detailed Description

LO.

Note

TODO

10.173.2 Member Typedef Documentation

10.173.2.1 const_iterator

```
typedef Superclass::const_iterator gdcm::LO::const_iterator
```

10.173.2.2 const_reference

```
typedef Superclass::const_reference gdcm::LO::const_reference
```

10.173.2.3 const_reverse_iterator

```
typedef Superclass::const_reverse_iterator gdcm::LO::const_reverse_iterator
```

10.173.2.4 difference_type

```
typedef Superclass::difference_type gdcm::LO::difference_type
```

10.173.2.5 iterator

```
typedef Superclass::iterator gdcm::LO::iterator
```

10.173.2.6 pointer

```
typedef Superclass::pointer gdcm::LO::pointer
```

10.173.2.7 reference

```
typedef Superclass::reference gdcm::LO::reference
```

10.173.2.8 reverse_iterator

```
typedef Superclass::reverse_iterator gdcm::LO::reverse_iterator
```

10.173.2.9 size_type

```
typedef Superclass::size_type gdcm::LO::size_type
```

10.173.2.10 Superclass

```
typedef String<'\\',64> gdcm::LO::Superclass
```

10.173.2.11 value_type

```
typedef Superclass::value_type gdcm::LO::value_type
```

10.173.3 Constructor & Destructor Documentation

10.173.3.1 LO() [1/4]

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

10.173.3.2 LO() [2/4]

```
gdcm::LO::LO (
            const value_type * s )  [inline]
```

10.173.3.3 LO() [3/4]

```
gdcm::LO::LO (
            const value_type * s,
            size_type n )  [inline]
```

10.173.3.4 LO() [4/4]

```
gdcm::LO::LO (
             const Superclass & s,
             size_type pos = 0,
             size_type n = npos )  [inline]
```

10.173.4 Member Function Documentation

10.173.4.1 IsValid()

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

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

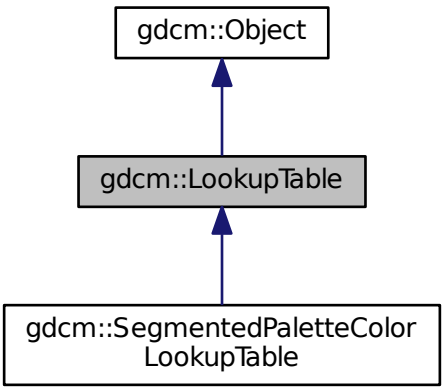
- gdcmLO.h

10.174 gdcm::LookupTable Class Reference

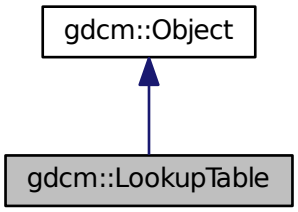
LookupTable class.

```
#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 ()
- void Allocate (unsigned short bitsample=8)

  Allocate the LUT.
- void Clear ()

  Clear the LUT.
- void Decode (std::istream &is, std::ostream &os) const

  Decode the LUT.
- bool Decode (char *outputbuffer, size_t outlen, const char *inputbuffer, size_t inlen) const
- 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:
- void Print (std::ostream &) const
- 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

10.174.1 Detailed Description

LookupTable class.

10.174.2 Member Enumeration Documentation

10.174.2.1 LookupTableType

`enum gdcm::LookupTable::LookupTableType`

Enumerator

RED	
GREEN	
BLUE	
GRAY	
UNKNOWN	

10.174.3 Constructor & Destructor Documentation

10.174.3.1 LookupTable() [1/2]

```
gdcm::LookupTable::LookupTable ( )
```

10.174.3.2 ~LookupTable()

```
gdcm::LookupTable::~LookupTable ( )
```

10.174.3.3 LookupTable() [2/2]

```
gdcm::LookupTable::LookupTable (
            LookupTable const & lut )  [inline]
```

10.174.4 Member Function Documentation

10.174.4.1 Allocate()

```
void gdcm::LookupTable::Allocate (
            unsigned short bitsample = 8 )
```

Allocate the LUT.

10.174.4.2 Clear()

```
void gdcm::LookupTable::Clear ( )
```

Clear the LUT.

10.174.4.3 Decode() [1/2]

```
void gdcm::LookupTable::Decode (
            std::istream & is,
            std::ostream & os ) const
```

Decode the LUT.

10.174.4.4 Decode() [2/2]

```
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

10.174.4.5 GetBitSample()

```
unsigned short gdcm::LookupTable::GetBitSample ( ) const  [inline]
```

return the bit sample

10.174.4.6 GetBufferAsRGBA()

```
bool gdcm::LookupTable::GetBufferAsRGBA (
            unsigned char * rgba ) const
```

return the LUT as RGBA buffer

10.174.4.7 GetLUT()

```
void gdcm::LookupTable::GetLUT (
            LookupTableType type,
            unsigned char * array,
            unsigned int & length ) const
```

Examples:

ExtractImageRegionWithLUT.cs.

10.174.4.8 GetLUTDescriptor()

```
void gdcm::LookupTable::GetLUTDescriptor (
            LookupTableType type,
            unsigned short & length,
            unsigned short & subscript,
            unsigned short & bitsize ) const
```

10.174.4.9 GetLUTLength()

```
unsigned int gdcm::LookupTable::GetLUTLength (
            LookupTableType type ) const
```

10.174.4.10 GetPointer()

```
const unsigned char* gdcm::LookupTable::GetPointer ( ) const
```

return a raw pointer to the LUT

10.174.4.11 InitializeBlueLUT()

```
void gdcm::LookupTable::InitializeBlueLUT (
            unsigned short length,
            unsigned short subscript,
            unsigned short bitsize )
```

10.174.4.12 Initialized()

```
bool gdcm::LookupTable::Initialized ( ) const
```

return whether the LUT has been initialized

10.174.4.13 InitializeGreenLUT()

```
void gdcm::LookupTable::InitializeGreenLUT (
            unsigned short length,
            unsigned short subscript,
            unsigned short bitsize )
```

10.174.4.14 InitializeLUT()

```
void gdcm::LookupTable::InitializeLUT (
            LookupTableType type,
            unsigned short length,
            unsigned short subscript,
            unsigned short bitsize )
```

Generic interface:

10.174.4.15 InitializeRedLUT()

```
void gdcm::LookupTable::InitializeRedLUT (
            unsigned short length,
            unsigned short subscript,
            unsigned short bitsize )
```

RED / GREEN / BLUE specific:

10.174.4.16 Print()

```
void gdcm::LookupTable::Print (
            std::ostream &  ) const  [inline], [virtual]
```

Reimplemented from gdcm::Object.

Reimplemented in gdcm::SegmentedPaletteColorLookupTable.

References gdcm::terminal::blue, gdcm::terminal::green, and gdcm::terminal::red.

10.174.4.17 SetBlueLUT()

```
void gdcm::LookupTable::SetBlueLUT (
            const unsigned char * blue,
            unsigned int length )
```

10.174.4.18 SetGreenLUT()

```
void gdcm::LookupTable::SetGreenLUT (
            const unsigned char * green,
            unsigned int length )
```

10.174.4.19 SetLUT()

```
virtual void gdcm::LookupTable::SetLUT (
            LookupTableType type,
            const unsigned char * array,
            unsigned int length )  [virtual]
```

Reimplemented in gdcm::SegmentedPaletteColorLookupTable.

10.174.4.20 SetRedLUT()

```
void gdcm::LookupTable::SetRedLUT (
            const unsigned char * red,
            unsigned int length )
```

10.174.4.21 WriteBufferAsRGBA()

```
bool gdcm::LookupTable::WriteBufferAsRGBA (
            const unsigned char * rgba )
```

Write the LUT as RGBA.

10.174.5 Member Data Documentation

10.174.5.1 BitSample

```
unsigned short gdcm::LookupTable::BitSample  [protected]
```

10.174.5.2 IncompleteLUT

```
bool gdcm::LookupTable::IncompleteLUT  [protected]
```

10.174.5.3 Internal

```
LookupTableInternal* gdcm::LookupTable::Internal  [protected]
```

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

- gdcmLookupTable.h

10.175 gdcm::Scanner::ltstr Struct Reference

```
#include <gdcmScanner.h>
```

Public Member Functions

- bool operator() (const char *s1, const char *s2) const

10.175.1 Member Function Documentation

10.175.1.1 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

10.176 gdcm::StrictScanner::ltstr Struct Reference

```
#include <gdcmStrictScanner.h>
```

Public Member Functions

- bool operator() (const char *s1, const char *s2) const

10.176.1 Member Function Documentation

10.176.1.1 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

10.177 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 ()
- void AddMacroEntry (const Tag &tag, const MacroEntry &module)

  Will add a ModuleEntry direcly 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)

10.177.1 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

10.177.2 Member Typedef Documentation

10.177.2.1 ArrayIncludeMacrosType

```
typedef std::vector<std::string> gdcm::Macro::ArrayIncludeMacrosType
```

10.177.2.2 MapModuleEntry

```
typedef std::map<Tag, MacroEntry> gdcm::Macro::MapModuleEntry
```

10.177.3 Constructor & Destructor Documentation

10.177.3.1 Macro()

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

References gdcm::operator<<().

10.177.4 Member Function Documentation

10.177.4.1 AddMacroEntry()

```
void gdcm::Macro::AddMacroEntry (
            const Tag & tag,
            const MacroEntry & module )  [inline]
```

Will add a ModuleEntry direcly at root-level. See Macro for nested-included level.

10.177.4.2 Clear()

```
void gdcm::Macro::Clear ( )  [inline]
```

10.177.4.3 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.

10.177.4.4 GetMacroEntry()

```
const MacroEntry& gdcm::Macro::GetMacroEntry (
            const Tag & tag ) const
```

10.177.4.5 GetName()

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

10.177.4.6 SetName()

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

10.177.4.7 Verify()

```
bool gdcm::Macro::Verify (
            const DataSet & ds,
            Usage const & usage ) const
```

10.177.5 Friends And Related Function Documentation

10.177.5.1 operator<<

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

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

- gdcmMacro.h

10.178 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 ()
- 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)

10.178.1 Detailed Description

Class for representing a Modules.

Note

bla

See also

Module

Examples:

TraverseModules.cxx.

10.178.2 Member Typedef Documentation

10.178.2.1 ModuleMapType

`typedef std::map<std::string, Macro> gdcm::Macros::ModuleMapType`

10.178.3 Constructor & Destructor Documentation

10.178.3.1 Macros()

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

References gdcm::operator<<().

10.178.4 Member Function Documentation

10.178.4.1 AddMacro()

```
void gdcm::Macros::AddMacro (
            const char * ref,
            const Macro & module )  [inline]
```

10.178.4.2 Clear()

```
void gdcm::Macros::Clear ( )  [inline]
```

10.178.4.3 GetMacro()

```
const Macro& gdcm::Macros::GetMacro (
            const char * name ) const  [inline]
```

10.178.4.4 IsEmpty()

```
bool gdcm::Macros::IsEmpty ( ) const  [inline]
```

10.178.5 Friends And Related Function Documentation

10.178.5.1 operator<<

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

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

- gdcmMacros.h

10.179 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

10.179.1 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)

10.179.2 Constructor & Destructor Documentation

10.179.2.1 MaximumLengthSub()

```
gdcm::network::MaximumLengthSub::MaximumLengthSub ( )
```

10.179.3 Member Function Documentation

10.179.3.1 GetMaximumLength()

```
uint32_t gdcm::network::MaximumLengthSub::GetMaximumLength ( ) const  [inline]
```

References Print(), and SetMaximumLength().

10.179.3.2 Print()

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

Referenced by GetMaximumLength().

10.179.3.3 Read()

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

10.179.3.4 SetMaximumLength()

```
void gdcm::network::MaximumLengthSub::SetMaximumLength (
            uint32_t maximumlength )
```

Referenced by GetMaximumLength().

10.179.3.5 Size()

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

10.179.3.6 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

10.180 gdcm::MD5 Class Reference

Class for MD5.

```
#include <gdcmMD5.h>
```

Public Member Functions

- MD5 ()
- ~MD5 ()

Static Public Member Functions

- static bool Compute (const char *buffer, unsigned long buf_len, char digest_str[33])
- static bool ComputeFile (const char *filename, char digest_str[33])

10.180.1 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

10.180.2 Constructor & Destructor Documentation

10.180.2.1 MD5()

```
gdcm::MD5::MD5 ( )
```

10.180.2.2 ~MD5()

```
gdcm::MD5::~MD5 ( )
```

10.180.3 Member Function Documentation

10.180.3.1 Compute()

```
static bool gdcm::MD5::Compute (
            const char * buffer,
            unsigned long buf_len,
            char digest_str[33] )  [static]
```

10.180.3.2 ComputeFile()

```
static bool gdcm::MD5::ComputeFile (
            const char * filename,
            char digest_str[33] )  [static]
```

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

- gdcmMD5.h

10.181 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,
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)

10.181.1 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.cxx, gdcmrtionplan.cxx, gdcmrtplan.cxx, GenAllVR.cxx, GenerateStandardSOPClasses.cxx, GenFakeIdentifyFile.cxx, GetSubSequenceData.cxx, iU22tomultisc.cxx, ReadAndDumpDICOMDIR.cxx, StreamImageReaderTest.cxx, and TestReader.cxx.

10.181.2 Member Enumeration Documentation

10.181.2.1 MSType

`enum gdcm::MediaStorage::MSType`

Enumerator

MediaStorageDirectoryStorage	
ComputedRadiographyImageStorage	
DigitalXRayImageStorageForPresentation	
DigitalXRayImageStorageForProcessing	
DigitalMammographyImageStorageForPresentation	
DigitalMammographyImageStorageForProcessing	
DigitalIntraoralXrayImageStorageForPresentation	
DigitalIntraoralXRayImageStorageForProcessing	
CTImageStorage	
EnhancedCTImageStorage	
UltrasoundImageStorageRetired	
UltrasoundImageStorage	
UltrasoundMultiFrameImageStorageRetired	
UltrasoundMultiFrameImageStorage	
MRImageStorage	
EnhancedMRImageStorage	
MRSpectroscopyStorage	
NuclearMedicineImageStorageRetired	

Enumerator

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	

Enumerator

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	
MS_END	

Examples:

GenerateStandardSOPClasses.cxx.

10.181.2.2 ObjectType

`enum gdcm::MediaStorage::ObjectType`

Enumerator

NoObject	
Video	
Waveform	
Audio	

Enumerator

PDF	
URI	
Segmentation	
ObjectEnd	

10.181.3 Constructor & Destructor Documentation

10.181.3.1 MediaStorage()

```
gdcm::MediaStorage::MediaStorage (
            MSType type = MS_END )  [inline]
```

10.181.4 Member Function Documentation

10.181.4.1 GetModality()

```
const char* gdcm::MediaStorage::GetModality ( ) const
```

10.181.4.2 GetModalityDimension()

```
unsigned int gdcm::MediaStorage::GetModalityDimension ( ) const
```

10.181.4.3 GetMSString()

```
static const char* gdcm::MediaStorage::GetMSString (
            MSType ts )  [static]
```

Return the Media String associated. Will return NULL for MS_END.

Examples:

GenerateStandardSOPClasses.cxx.

Referenced by gdcm::operator<<().

10.181.4.4 GetMSType()

```
static MSType gdcm::MediaStorage::GetMSType (
            const char * str )  [static]
```

Examples:

MetaImageMD5Activiz.cs, and TestReader.cxx.

10.181.4.5 GetNumberOfModality()

```
static unsigned int gdcm::MediaStorage::GetNumberOfModality ( )  [static]
```

10.181.4.6 GetNumberOfMSString()

```
static unsigned int gdcm::MediaStorage::GetNumberOfMSString ( )  [static]
```

10.181.4.7 GetNumberOfMSType()

```
static unsigned int gdcm::MediaStorage::GetNumberOfMSType ( )  [static]
```

10.181.4.8 GetString()

```
const char* gdcm::MediaStorage::GetString ( ) const
```

Return the Media String of the object.

Examples:

CreateJPIPDataSet.cxx, EncapsulateFileInRawData.cxx, Extracting_All_Resolution.cxx, Fake_Image_Using_Stream_Image_Writer.cxx, GenAllVR.cxx, GenFakeIdentifyFile.cxx, GetSubSequenceData.cxx, iU22tomultisc.cxx, and StreamImageReaderTest.cxx.

10.181.4.9 GuessFromModality()

```
void gdcm::MediaStorage::GuessFromModality (
            const char * modality,
            unsigned int dimension = 2 )
```

10.181.4.10 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.

10.181.4.11 IsUndefined()

```
bool gdcm::MediaStorage::IsUndefined ( ) const  [inline]
```

Examples:

TestReader.cxx.

10.181.4.12 operator MSType()

```
gdcm::MediaStorage::operator MSType ( ) const  [inline]
```

References gdcm::operator<<().

10.181.4.13 SetFromDataSet()

```
bool gdcm::MediaStorage::SetFromDataSet (
            DataSet const & ds )
```

Advanced user only (functions should be protected level...) Those function are lower level than SetFromFile

10.181.4.14 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:

gdcmrtionplan.cxx, gdcmrtplan.cxx, ReadAndDumpDICOMDIR.cxx, and TestReader.cxx.

10.181.4.15 SetFromHeader()

```
bool gdcm::MediaStorage::SetFromHeader (
            FileMetaInformation const & fmi )
```

10.181.4.16 SetFromModality()

```
bool gdcm::MediaStorage::SetFromModality (
            DataSet const & ds )
```

10.181.4.17 SetFromSourceImageSequence()

```
void gdcm::MediaStorage::SetFromSourceImageSequence (
            DataSet const & ds )  [protected]
```

10.181.5 Friends And Related Function Documentation

10.181.5.1 operator<<

```
std::ostream& operator<< (
            std::ostream & os,
            const MediaStorage & ms )  [friend]
```

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

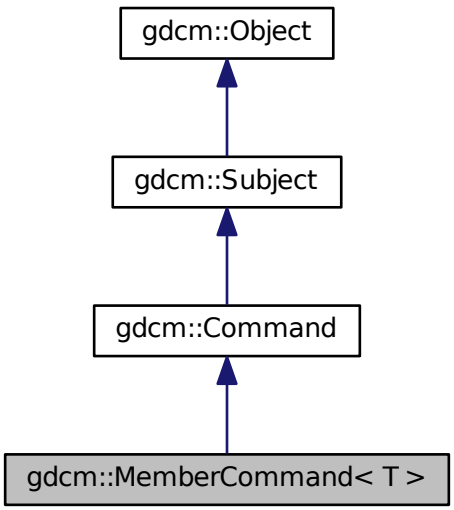
- gdcmMediaStorage.h

10.182 gdcm::MemberCommand< T > Class Template Reference

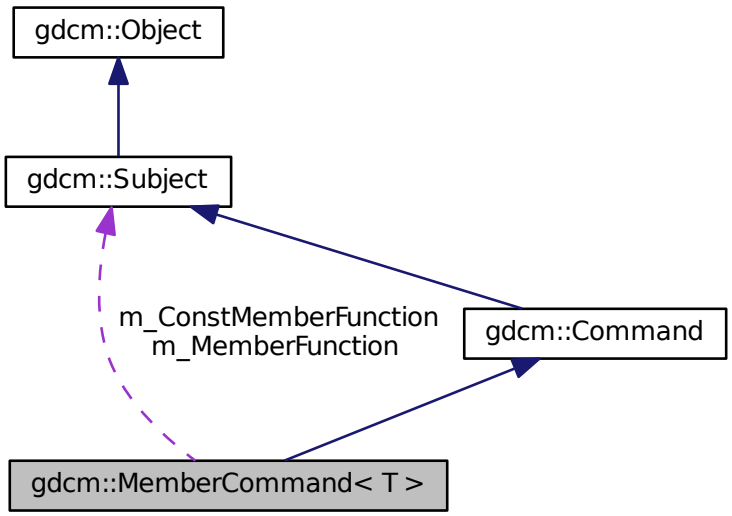
Command subclass that calls a pointer to a member function.

`#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

- virtual void Execute (Subject *caller, const Event &event)
- virtual void Execute (const Subject *caller, const Event &event)
- void SetCallbackFunction (T *object, TMemberFunctionPointer memberFunction)
- void SetCallbackFunction (T *object, TConstMemberFunctionPointer memberFunction)

Static Public Member Functions

- static SmartPointer< MemberCommand > New ()

Protected Member Functions

- MemberCommand ()
- virtual ~MemberCommand ()

Protected Attributes

- TConstMemberFunctionPointer m_ConstMemberFunction
- TMemberFunctionPointer m_MemberFunction
- T * m_This

10.182.1 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.

10.182.2 Member Typedef Documentation

10.182.2.1 Self

```
template<class T >
typedef MemberCommand gdcm::MemberCommand< T >::Self
```

Standard class typedefs.

10.182.2.2 TConstMemberFunctionPointer

```
template<class T >
typedef void(T::* gdcm::MemberCommand< T >::TConstMemberFunctionPointer) (const Subject *, const
Event &)
```

10.182.2.3 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

10.182.3 Constructor & Destructor Documentation

10.182.3.1 MemberCommand()

```
template<class T >
gdcm::MemberCommand< T >::MemberCommand ( )  [inline], [protected]
```

10.182.3.2 ~MemberCommand()

```
template<class T >
virtual gdcm::MemberCommand< T >::~MemberCommand ( )  [inline], [protected], [virtual]
```

10.182.4 Member Function Documentation

10.182.4.1 Execute() [1/2]

```
template<class T >
virtual void gdcm::MemberCommand< T >::Execute (
             Subject * caller,
             const Event & event )  [inline], [virtual]
```

Invoke the member function.

Implements gdcm::Command.

10.182.4.2 Execute() [2/2]

```
template<class T >
virtual void gdcm::MemberCommand< T >::Execute (
             const Subject * caller,
             const Event & event )  [inline], [virtual]
```

Invoke the member function with a const object.

Implements gdcm::Command.

10.182.4.3 New()

```
template<class T >
static SmartPointer<MemberCommand> gdcm::MemberCommand< T >::New ( )  [inline], [static]
```

Method for creation through the object factory.

10.182.4.4 SetCallbackFunction() [1/2]

```
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.

10.182.4.5 SetCallbackFunction() [2/2]

```
template<class T >
void gdcm::MemberCommand< T >::SetCallbackFunction (
            T * object,
            TConstMemberFunctionPointer memberFunction )  [inline]
```

10.182.5 Member Data Documentation

10.182.5.1 m_ConstMemberFunction

```
template<class T >
TConstMemberFunctionPointer gdcm::MemberCommand< T >::m_ConstMemberFunction  [protected]
```

10.182.5.2 m_MemberFunction

```
template<class T >
TMemberFunctionPointer gdcm::MemberCommand< T >::m_MemberFunction  [protected]
```

10.182.5.3 m_This

```
template<class T >
T* gdcm::MemberCommand< T >::m_This  [protected]
```

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

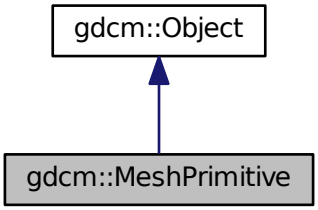
- gdcmCommand.h

10.183 gdcm::MeshPrimitive Class Reference

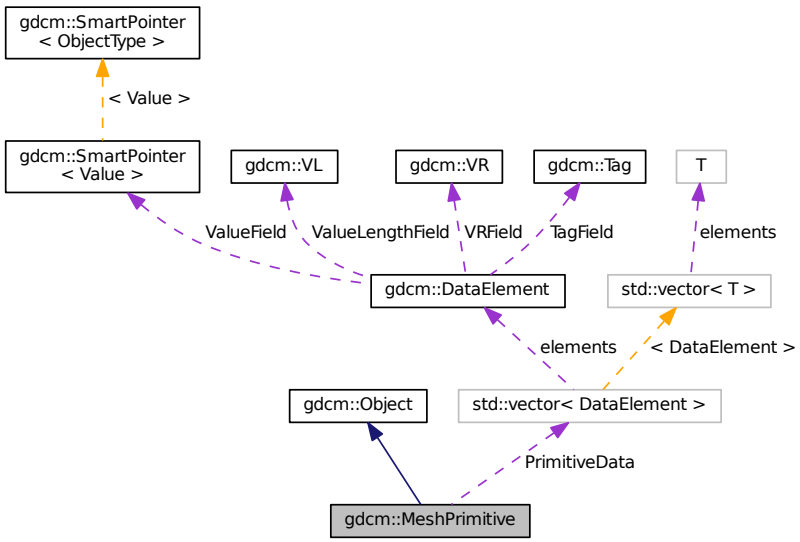
This class defines surface mesh primitives.

`#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 ()
- virtual ~MeshPrimitive ()
- void AddPrimitiveData (DataElement const &de)
- unsigned int GetNumberOfPrimitivesData () const
- 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 ()
- MPType GetPrimitiveType () const
- void SetPrimitiveData (DataElement const &de)
- void SetPrimitiveData (const unsigned int idx, 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

10.183.1 Detailed Description

This class defines surface mesh primitives.

It is designed from surface mesh primitives macro.

See also

PS 3.3 C.27.4

10.183.2 Member Typedef Documentation

10.183.2.1 PrimitivesData

```
typedef std::vector< DataElement > gdcm::MeshPrimitive::PrimitivesData
```

10.183.3 Member Enumeration Documentation

10.183.3.1 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	

10.183.4 Constructor & Destructor Documentation

10.183.4.1 MeshPrimitive()

```
gdcm::MeshPrimitive::MeshPrimitive ( )
```

10.183.4.2 ~MeshPrimitive()

```
virtual gdcm::MeshPrimitive::~MeshPrimitive ( )  [virtual]
```

10.183.5 Member Function Documentation

10.183.5.1 AddPrimitiveData()

```
void gdcm::MeshPrimitive::AddPrimitiveData (
            DataElement const & de )
```

10.183.5.2 GetMPType()

```
static MPType gdcm::MeshPrimitive::GetMPType (
            const char * type )  [static]
```

10.183.5.3 GetMPTypeString()

```
static const char* gdcm::MeshPrimitive::GetMPTypeString (
            const MPType type )  [static]
```

10.183.5.4 GetNumberOfPrimitivesData()

```
unsigned int gdcm::MeshPrimitive::GetNumberOfPrimitivesData ( ) const
```

10.183.5.5 GetPrimitiveData() [1/4]

```
const DataElement& gdcm::MeshPrimitive::GetPrimitiveData ( ) const
```

10.183.5.6 GetPrimitiveData() [2/4]

```
DataElement& gdcm::MeshPrimitive::GetPrimitiveData ( )
```

10.183.5.7 GetPrimitiveData() [3/4]

```
const DataElement& gdcm::MeshPrimitive::GetPrimitiveData (
            const unsigned int idx ) const
```

10.183.5.8 GetPrimitiveData() [4/4]

```
DataElement& gdcm::MeshPrimitive::GetPrimitiveData (
            const unsigned int idx )
```

10.183.5.9 GetPrimitivesData() [1/2]

```
const PrimitivesData& gdcm::MeshPrimitive::GetPrimitivesData ( ) const
```

10.183.5.10 GetPrimitivesData() [2/2]

```
PrimitivesData& gdcm::MeshPrimitive::GetPrimitivesData ( )
```

10.183.5.11 GetPrimitiveType()

```
MPType gdcm::MeshPrimitive::GetPrimitiveType ( ) const
```

10.183.5.12 SetPrimitiveData() [1/2]

```
void gdcm::MeshPrimitive::SetPrimitiveData (
            DataElement const & de )
```

10.183.5.13 SetPrimitiveData() [2/2]

```
void gdcm::MeshPrimitive::SetPrimitiveData (
            const unsigned int idx,
            DataElement const & de )
```

10.183.5.14 SetPrimitivesData()

```
void gdcm::MeshPrimitive::SetPrimitivesData (
            PrimitivesData const & DEs )
```

10.183.5.15 SetPrimitiveType()

```
void gdcm::MeshPrimitive::SetPrimitiveType (
            const MPType type )
```

10.183.6 Member Data Documentation

10.183.6.1 PrimitiveData

```
PrimitivesData gdcm::MeshPrimitive::PrimitiveData  [protected]
```

10.183.6.2 PrimitiveType

```
MPType gdcm::MeshPrimitive::PrimitiveType  [protected]
```

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

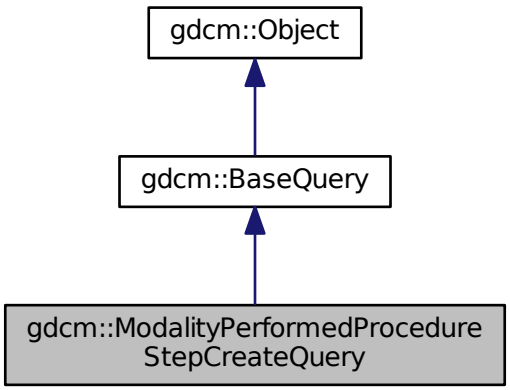
- gdcmMeshPrimitive.h

10.184 gdcm::ModalityPerformedProcedureStepCreateQuery Class Reference

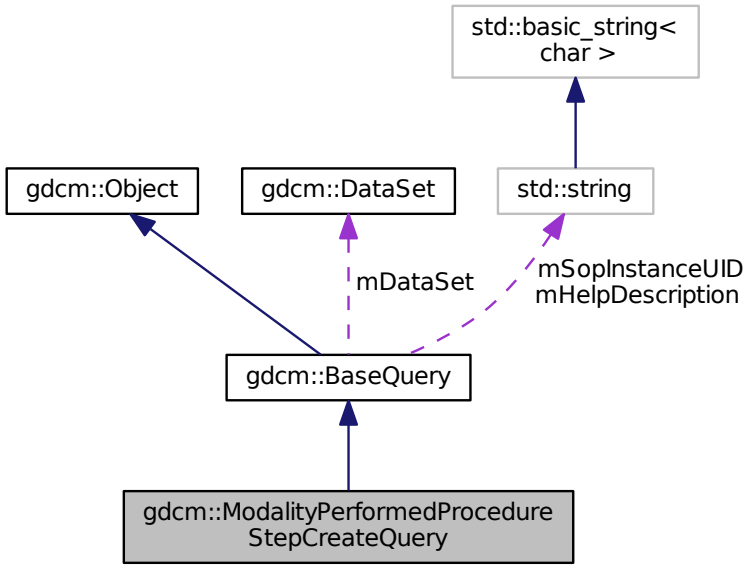
ModalityPerformedProcedureStepCreateQuery.

`#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
- gdcm::DataSet GetRequiredDataSet () const
- bool ValidateQuery (bool inStrict=true) const

Friends

- class QueryFactory

Additional Inherited Members

10.184.1 Detailed Description

ModalityPerformedProcedureStepCreateQuery.

contains: the class which will produce a dataset for n-create for Modality Performed Procedure Step sop class

10.184.2 Constructor & Destructor Documentation

10.184.2.1 ModalityPerformedProcedureStepCreateQuery()

```
gdcm::ModalityPerformedProcedureStepCreateQuery::ModalityPerformedProcedureStepCreateQuery (
            const std::string & iSopInstanceUID )
```

10.184.3 Member Function Documentation

10.184.3.1 GetAbstractSyntaxUID()

```
UIDs::TSName gdcm::ModalityPerformedProcedureStepCreateQuery::GetAbstractSyntaxUID ( ) const  [virtual]
```

Implements gdcm::BaseQuery.

10.184.3.2 GetRequiredDataSet()

```
gdcm::DataSet gdcm::ModalityPerformedProcedureStepCreateQuery::GetRequiredDataSet ( ) const
```

10.184.3.3 ValidateQuery()

```
bool gdcm::ModalityPerformedProcedureStepCreateQuery::ValidateQuery (
            bool inStrict = true ) const  [virtual]
```

Implements gdcm::BaseQuery.

10.184.4 Friends And Related Function Documentation

10.184.4.1 QueryFactory

```
friend class QueryFactory  [friend]
```

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

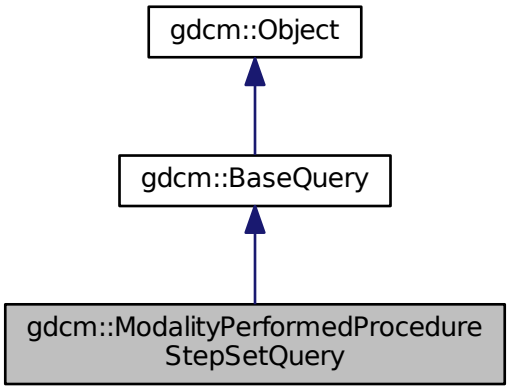
- gdcmModalityPerformedProcedureStepCreateQuery.h

10.185 gdcm::ModalityPerformedProcedureStepSetQuery Class Reference

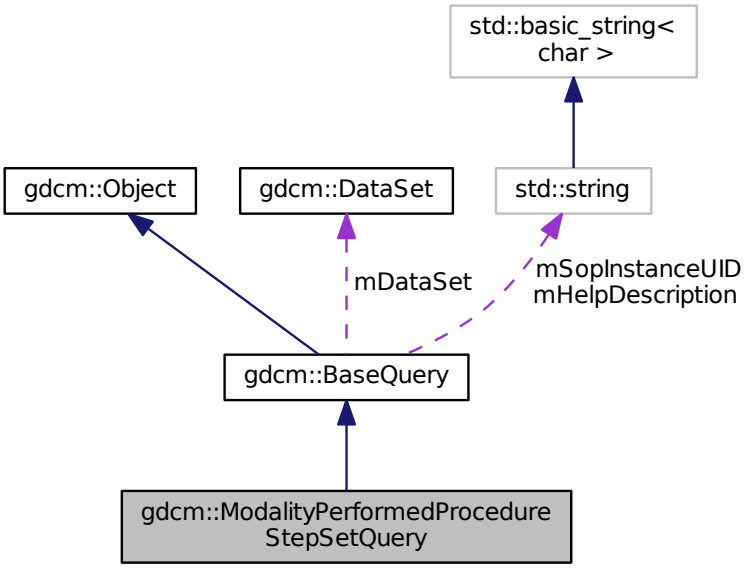
ModalityPerformedProcedureStepSetQuery.

`#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
- gdcm::DataSet GetRequiredDataSet () const
- bool ValidateQuery (bool inStrict=true) const

Friends

- class QueryFactory

Additional Inherited Members

10.185.1 Detailed Description

ModalityPerformedProcedureStepSetQuery.

contains: the class which will produce a dataset for n-set for Modality Performed Procedure Step sop class

10.185.2 Constructor & Destructor Documentation

10.185.2.1 ModalityPerformedProcedureStepSetQuery()

```
gdcm::ModalityPerformedProcedureStepSetQuery::ModalityPerformedProcedureStepSetQuery (
            const std::string & iSopInstanceUID )
```

10.185.3 Member Function Documentation

10.185.3.1 GetAbstractSyntaxUID()

```
UIDs::TSName gdcm::ModalityPerformedProcedureStepSetQuery::GetAbstractSyntaxUID ( ) const  [virtual]
```

Implements gdcm::BaseQuery.

10.185.3.2 GetRequiredDataSet()

```
gdcm::DataSet gdcm::ModalityPerformedProcedureStepSetQuery::GetRequiredDataSet ( ) const
```

10.185.3.3 ValidateQuery()

```
bool gdcm::ModalityPerformedProcedureStepSetQuery::ValidateQuery (
            bool inStrict = true ) const  [virtual]
```

Implements gdcm::BaseQuery.

10.185.4 Friends And Related Function Documentation

10.185.4.1 QueryFactory

```
friend class QueryFactory  [friend]
```

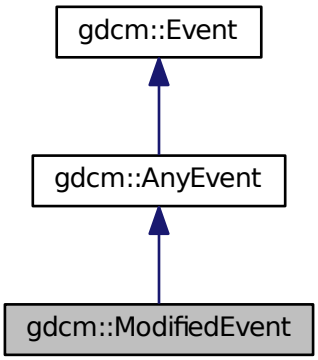
The documentation for this class was generated from the following file:

- gdcmModalityPerformedProcedureStepSetQuery.h

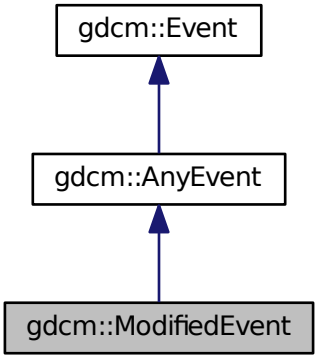
10.186 gdcm::ModifiedEvent Class Reference

```
#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

10.187 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 ()
- void AddMacro (const char *include)
- void AddModuleEntry (const Tag &tag, const ModuleEntry &module)

  Will add a ModuleEntry direcly 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)

10.187.1 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.

10.187.2 Member Typedef Documentation

10.187.2.1 ArrayIncludeMacrosType

```
typedef std::vector<std::string> gdcm::Module::ArrayIncludeMacrosType
```

10.187.2.2 MapModuleEntry

```
typedef std::map<Tag, ModuleEntry> gdcm::Module::MapModuleEntry
```

10.187.3 Constructor & Destructor Documentation

10.187.3.1 Module()

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

References gdcm::operator<<().

10.187.4 Member Function Documentation

10.187.4.1 AddMacro()

```
void gdcm::Module::AddMacro (
            const char * include )  [inline]
```

10.187.4.2 AddModuleEntry()

```
void gdcm::Module::AddModuleEntry (
            const Tag & tag,
            const ModuleEntry & module )  [inline]
```

Will add a ModuleEntry direcly at root-level. See Macro for nested-included level.

10.187.4.3 Clear()

```
void gdcm::Module::Clear ( )  [inline]
```

10.187.4.4 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.

10.187.4.5 GetModuleEntryInMacros()

```
const ModuleEntry& gdcm::Module::GetModuleEntryInMacros (
            Macros const & macros,
            const Tag & tag ) const
```

Examples:

TraverseModules.cxx.

10.187.4.6 GetName()

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

10.187.4.7 SetName()

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

10.187.4.8 Verify()

```
bool gdcm::Module::Verify (
             const DataSet & ds,
             Usage const & usage ) const
```

10.187.5 Friends And Related Function Documentation

10.187.5.1 operator<<

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

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

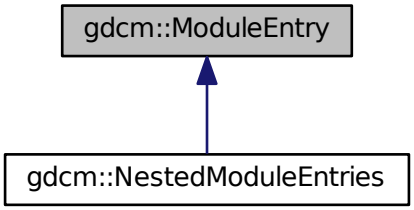
- gdcmModule.h

10.188 gdcm::ModuleEntry Class Reference

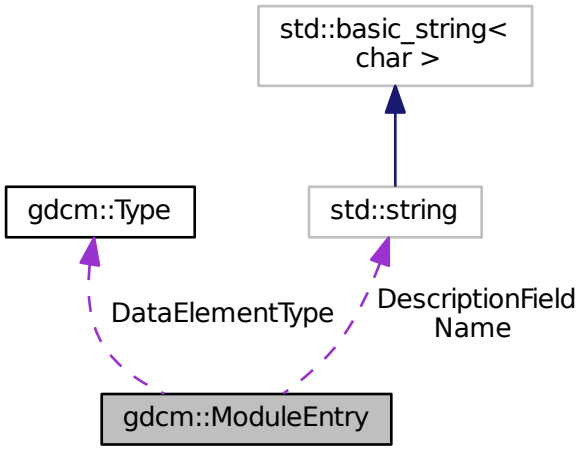
Class for representing a ModuleEntry.

```
#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 ()
- 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)

10.188.1 Detailed Description

Class for representing a ModuleEntry.

Note

bla

See also

DictEntry

Examples:

TraverseModules.cxx.

10.188.2 Member Typedef Documentation

10.188.2.1 Description

```
typedef std::string gdcm::ModuleEntry::Description
```

10.188.3 Constructor & Destructor Documentation

10.188.3.1 ModuleEntry()

```
gdcm::ModuleEntry::ModuleEntry (
            const char * name = "",
            const char * type = "3",
            const char * description = "" )  [inline]
```

References gdcm::Type::GetTypeType().

10.188.3.2 ~ModuleEntry()

```
virtual gdcm::ModuleEntry::~ModuleEntry ( )  [inline], [virtual]
```

References gdcm::operator<<().

10.188.4 Member Function Documentation

10.188.4.1 GetDescription()

```
const Description& gdcm::ModuleEntry::GetDescription ( ) const  [inline]
```

10.188.4.2 GetName()

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

10.188.4.3 GetType()

```
const Type& gdcm::ModuleEntry::GetType ( ) const  [inline]
```

Examples:

TraverseModules.cxx.

10.188.4.4 SetDescription()

```
void gdcm::ModuleEntry::SetDescription (
            const char * d )  [inline]
```

10.188.4.5 SetName()

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

10.188.4.6 SetType()

```
void gdcm::ModuleEntry::SetType (
            const Type & type )  [inline]
```

10.188.5 Friends And Related Function Documentation

10.188.5.1 operator<<

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

10.188.6 Member Data Documentation

10.188.6.1 DataElementType

```
Type gdcm::ModuleEntry::DataElementType  [protected]
```

Referenced by gdcm::operator<<().

10.188.6.2 DescriptionField

```
Description gdcm::ModuleEntry::DescriptionField  [protected]
```

Referenced by gdcm::operator<<().

10.188.6.3 Name

```
std::string gdcm::ModuleEntry::Name  [protected]
```

Referenced by gdcm::operator<<().

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

- gdcmModuleEntry.h

10.189 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 ()
- 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)

10.189.1 Detailed Description

Class for representing a Modules.

Note

bla

See also

Module

Examples:

TraverseModules.cxx.

10.189.2 Member Typedef Documentation

10.189.2.1 ModuleMapType

```
typedef std::map<std::string, Module> gdcm::Modules::ModuleMapType
```

10.189.3 Constructor & Destructor Documentation

10.189.3.1 Modules()

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

References gdcm::operator<<().

10.189.4 Member Function Documentation

10.189.4.1 AddModule()

```
void gdcm::Modules::AddModule (
            const char * ref,
            const Module & module )  [inline]
```

10.189.4.2 Clear()

```
void gdcm::Modules::Clear ( )  [inline]
```

10.189.4.3 GetModule()

```
const Module& gdcm::Modules::GetModule (
            const char * name ) const  [inline]
```

Examples:

TraverseModules.cxx.

10.189.4.4 IsEmpty()

```
bool gdcm::Modules::IsEmpty ( ) const  [inline]
```

10.189.5 Friends And Related Function Documentation

10.189.5.1 operator<<

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

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

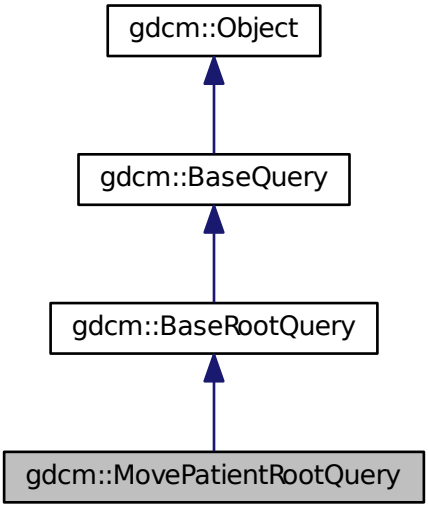
- gdcmModules.h

10.190 gdcm::MovePatientRootQuery Class Reference

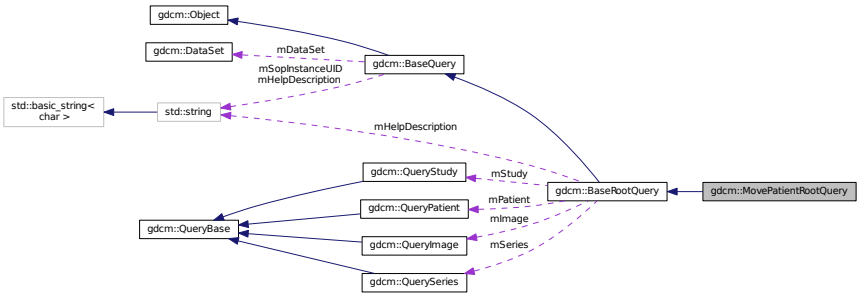
MovePatientRootQuery.

```
#include <gdcmMovePatientRootQuery.h>
```

Inheritance diagram for gdcm::MovePatientRootQuery:



Collaboration diagram for gdcm::MovePatientRootQuery:



Public Member Functions

- MovePatientRootQuery ()
- UIDs::TSName GetAbstractSyntaxUID () const
- std::vector< Tag > GetTagListByLevel (const EQueryLevel &inQueryLevel)
- void InitializeDataSet (const EQueryLevel &inQueryLevel)
- bool ValidateQuery (bool inStrict=true) const

Friends

- class QueryFactory

Additional Inherited Members

10.190.1 Detailed Description

MovePatientRootQuery.

contains: the class which will produce a dataset for c-move with patient root

10.190.2 Constructor & Destructor Documentation

10.190.2.1 MovePatientRootQuery()

```
gdcm::MovePatientRootQuery::MovePatientRootQuery ( )
```

10.190.3 Member Function Documentation

10.190.3.1 GetAbstractSyntaxUID()

```
UIDs::TSName gdcm::MovePatientRootQuery::GetAbstractSyntaxUID ( ) const  [virtual]
```

Implements gdcm::BaseQuery.

10.190.3.2 GetTagListByLevel()

```
std::vector<Tag> gdcm::MovePatientRootQuery::GetTagListByLevel (
            const EQueryLevel & inQueryLevel )  [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.

10.190.3.3 InitializeDataSet()

```
void gdcm::MovePatientRootQuery::InitializeDataSet (
            const EQueryLevel & inQueryLevel )  [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.

10.190.3.4 ValidateQuery()

```
bool gdcm::MovePatientRootQuery::ValidateQuery (
            bool inStrict = true ) const  [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.

10.190.4 Friends And Related Function Documentation

10.190.4.1 QueryFactory

```
friend class QueryFactory  [friend]
```

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

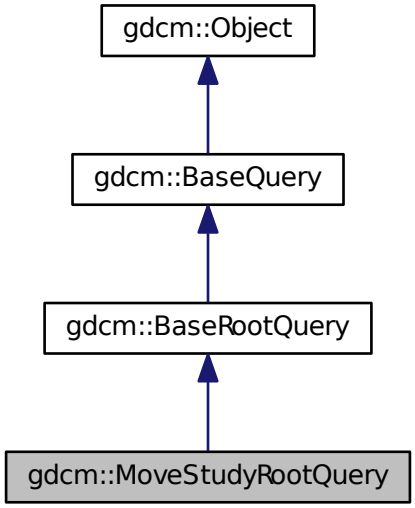
- gdcmMovePatientRootQuery.h

10.191 gdcm::MoveStudyRootQuery Class Reference

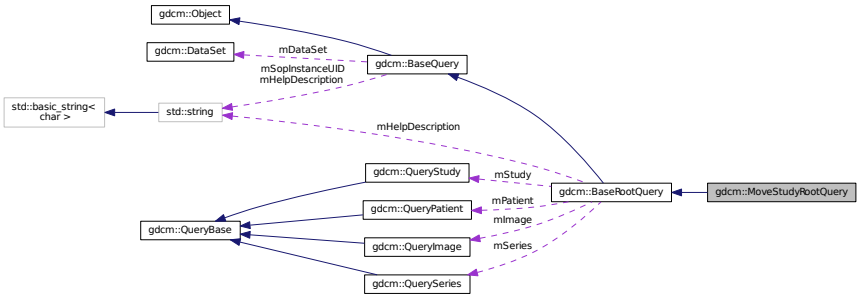
MoveStudyRootQuery.

```
#include <gdcmMoveStudyRootQuery.h>
```

Inheritance diagram for gdcm::MoveStudyRootQuery:



Collaboration diagram for gdcm::MoveStudyRootQuery:



Public Member Functions

- MoveStudyRootQuery ()
- UIDs::TSName GetAbstractSyntaxUID () const
- std::vector< Tag > GetTagListByLevel (const EQueryLevel &inQueryLevel)
- void InitializeDataSet (const EQueryLevel &inQueryLevel)
- bool ValidateQuery (bool inStrict=true) const

Friends

- class QueryFactory

Additional Inherited Members

10.191.1 Detailed Description

MoveStudyRootQuery.

contains: the class which will produce a dataset for C-MOVE with study root

10.191.2 Constructor & Destructor Documentation

10.191.2.1 MoveStudyRootQuery()

```
gdcm::MoveStudyRootQuery::MoveStudyRootQuery ( )
```

10.191.3 Member Function Documentation

10.191.3.1 GetAbstractSyntaxUID()

```
UIDs::TSName gdcm::MoveStudyRootQuery::GetAbstractSyntaxUID ( ) const  [virtual]
```

Implements gdcm::BaseQuery.

10.191.3.2 GetTagListByLevel()

```
std::vector<Tag> gdcm::MoveStudyRootQuery::GetTagListByLevel (
            const EQueryLevel & inQueryLevel )  [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.

10.191.3.3 InitializeDataSet()

```
void gdcm::MoveStudyRootQuery::InitializeDataSet (
            const EQueryLevel & inQueryLevel )  [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.

10.191.3.4 ValidateQuery()

```
bool gdcm::MoveStudyRootQuery::ValidateQuery (
            bool inStrict = true ) const  [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.

10.191.4 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:

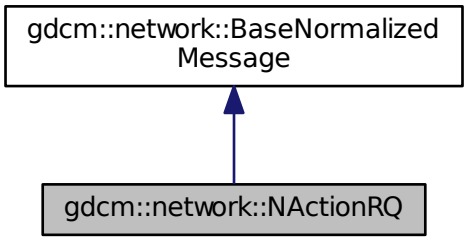
- gdcmMoveStudyRootQuery.h

10.192 gdcm::network::NActionRQ Class Reference

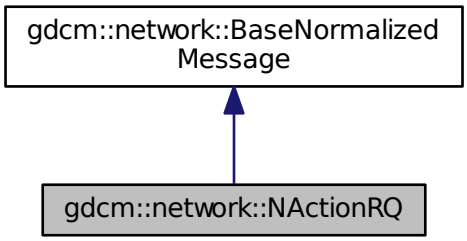
NActionRQ.

```
#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)

10.192.1 Detailed Description

NActionRQ.

this file defines the messages for the NAction action

10.192.2 Member Function Documentation

10.192.2.1 ConstructPDV()

```
std::vector<PresentationDataValue> gdcm::network::NActionRQ::ConstructPDV (
            const ULConnection & inConnection,
            const BaseQuery * inQuery ) [virtual]
```

Implements gdcm::network::BaseNormalizedMessage.

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

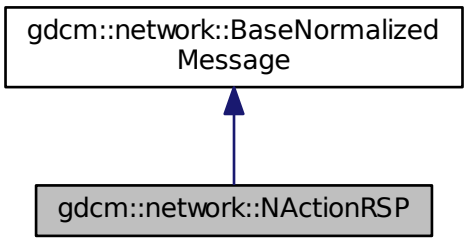
- gdcmNActionMessages.h

10.193 gdcm::network::NActionRSP Class Reference

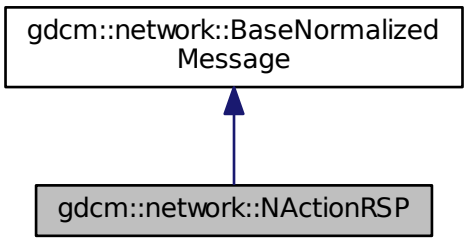
NActionRSP this file defines the messages for the NAction action.

```
#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)

10.193.1 Detailed Description

NActionRSP this file defines the messages for the NAction action.

10.193.2 Member Function Documentation

10.193.2.1 ConstructPDVByDataSet()

```
std::vector<PresentationDataValue> gdcm::network::NActionRSP::ConstructPDVByDataSet (
            const DataSet * inDataSet )
```

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

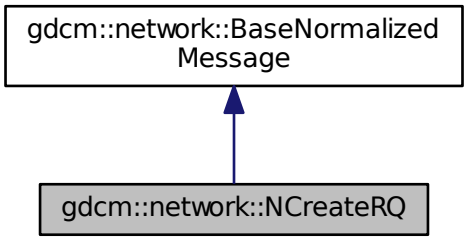
- gdcmNActionMessages.h

10.194 gdcm::network::NCreateRQ Class Reference

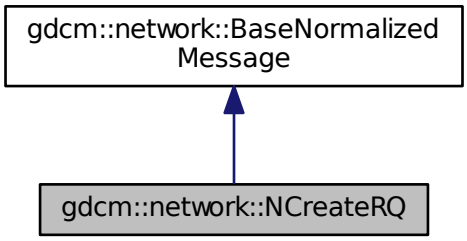
NCreateRQ.

```
#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)

10.194.1 Detailed Description

NCreateRQ.

this file defines the messages for the ncreate action

10.194.2 Member Function Documentation

10.194.2.1 ConstructPDV()

```
std::vector<PresentationDataValue> gdcm::network::NCreateRQ::ConstructPDV (
            const ULConnection & inConnection,
            const BaseQuery * inQuery )  [virtual]
```

Implements gdcm::network::BaseNormalizedMessage.

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

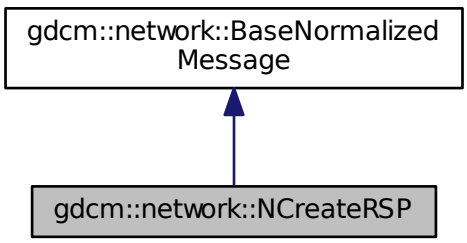
- gdcmNCreateMessages.h

10.195 gdcm::network::NCreateRSP Class Reference

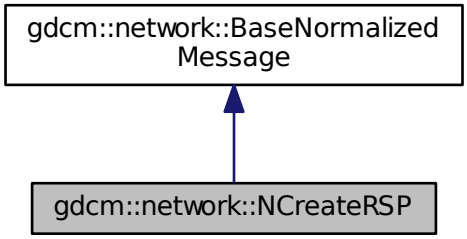
NCreateRSP this file defines the messages for the ncreate action.

```
#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)

10.195.1 Detailed Description

NCreateRSP this file defines the messages for the ncreate action.

10.195.2 Member Function Documentation

10.195.2.1 ConstructPDVByDataSet()

```
std::vector<PresentationDataValue> gdcm::network::NCreateRSP::ConstructPDVByDataSet (
            const DataSet * inDataSet )
```

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

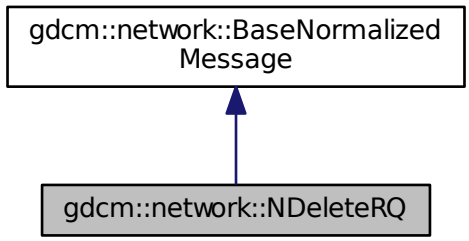
- gdcmNCreateMessages.h

10.196 gdcm::network::NDeleteRQ Class Reference

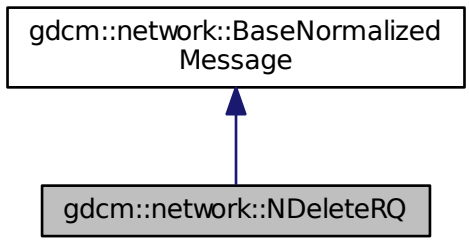
NDeleteRQ.

`#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)

10.196.1 Detailed Description

NDeleteRQ.

this file defines the messages for the ndelete action

10.196.2 Member Function Documentation

10.196.2.1 ConstructPDV()

```
std::vector<PresentationDataValue> gdcm::network::NDeleteRQ::ConstructPDV (
            const ULConnection & inConnection,
            const BaseQuery * inQuery )  [virtual]
```

Implements gdcm::network::BaseNormalizedMessage.

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

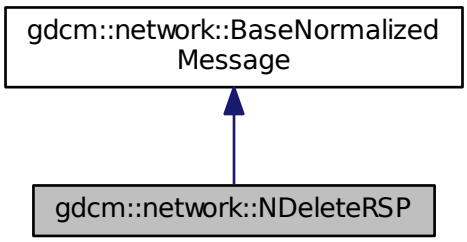
- gdcmNDeleteMessages.h

10.197 gdcm::network::NDeleteRSP Class Reference

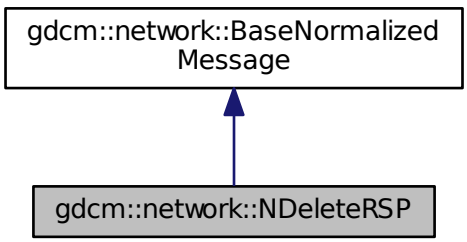
NDeleteRSP this file defines the messages for the ndelete action.

`#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)

10.197.1 Detailed Description

NDeleteRSP this file defines the messages for the ndelete action.

10.197.2 Member Function Documentation

10.197.2.1 ConstructPDVByDataSet()

```
std::vector<PresentationDataValue> gdcm::network::NDeleteRSP::ConstructPDVByDataSet (
            const DataSet * inDataSet )
```

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

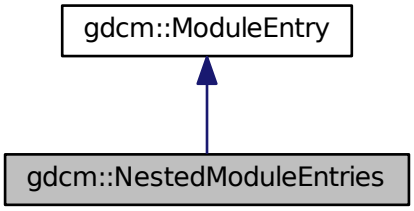
- gdcmNDeleteMessages.h

10.198 gdcm::NestedModuleEntries Class Reference

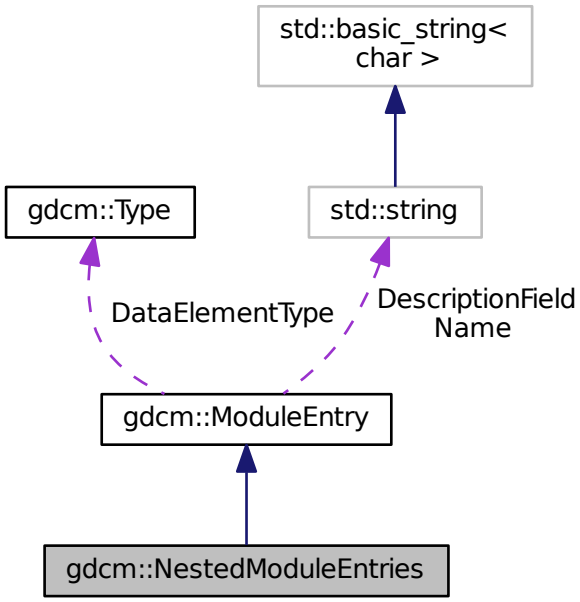
Class for representing a NestedModuleEntries.

```
#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)
- const ModuleEntry & GetModuleEntry (SizeType idx) const
- ModuleEntry & GetModuleEntry (SizeType idx)
- SizeType GetNumberOfModuleEntries ()

Friends

- std::ostream & operator<< (std::ostream &_os, const NestedModuleEntries &_val)

Additional Inherited Members

10.198.1 Detailed Description

Class for representing a NestedModuleEntries.

Note

bla

See also

ModuleEntry

10.198.2 Member Typedef Documentation

10.198.2.1 SizeType

```
typedef std::vector<ModuleEntry>::size_type gdcm::NestedModuleEntries::SizeType
```

10.198.3 Constructor & Destructor Documentation

10.198.3.1 NestedModuleEntries()

```
gdcm::NestedModuleEntries::NestedModuleEntries (
            const char * name = "",
            const char * type = "3",
            const char * description = "" )  [inline]
```

References gdcm::operator<<().

10.198.4 Member Function Documentation

10.198.4.1 AddModuleEntry()

```
void gdcm::NestedModuleEntries::AddModuleEntry (
            const ModuleEntry & me )  [inline]
```

10.198.4.2 GetModuleEntry() [1/2]

```
const ModuleEntry& gdcm::NestedModuleEntries::GetModuleEntry (
            SizeType idx ) const  [inline]
```

10.198.4.3 GetModuleEntry() [2/2]

```
ModuleEntry& gdcm::NestedModuleEntries::GetModuleEntry (
            SizeType idx )  [inline]
```

10.198.4.4 GetNumberOfModuleEntries()

```
SizeType gdcm::NestedModuleEntries::GetNumberOfModuleEntries ( )  [inline]
```

10.198.5 Friends And Related Function Documentation

10.198.5.1 operator<<

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

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

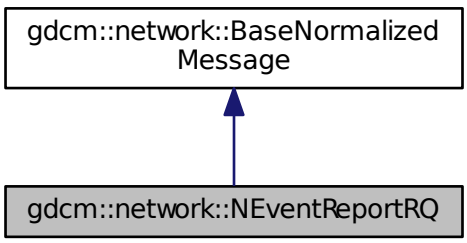
- gdcmNestedModuleEntries.h

10.199 gdcm::network::NEventReportRQ Class Reference

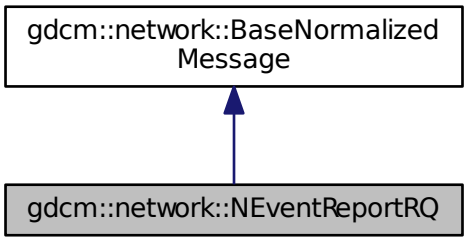
NEventReportRQ.

```
#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)

10.199.1 Detailed Description

NEventReportRQ.

this file defines the messages for the neventreport action

10.199.2 Member Function Documentation

10.199.2.1 ConstructPDV()

```
std::vector<PresentationDataValue> gdcm::network::NEventReportRQ::ConstructPDV (
            const ULConnection & inConnection,
            const BaseQuery * inQuery )  [virtual]
```

Implements gdcm::network::BaseNormalizedMessage.

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

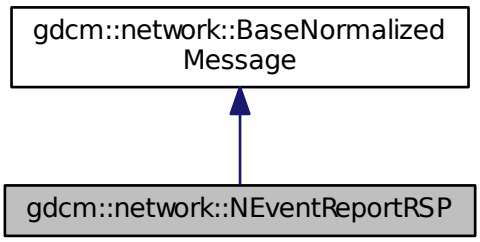
- gdcmNEventReportMessages.h

10.200 gdcm::network::NEventReportRSP Class Reference

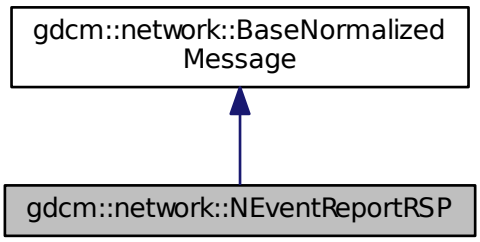
NEventReportRSP this file defines the messages for the neventreport action.

`#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)

10.200.1 Detailed Description

NEventReportRSP this file defines the messages for the neventreport action.

10.200.2 Member Function Documentation

10.200.2.1 ConstructPDVByDataSet()

```
std::vector<PresentationDataValue> gdcm::network::NEventReportRSP::ConstructPDVByDataSet (
            const DataSet * inDataSet )
```

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

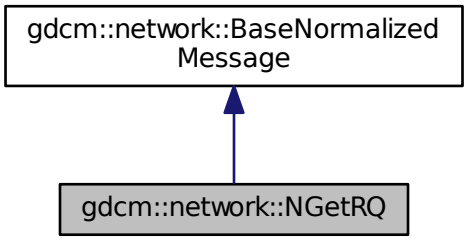
- gdcmNEventReportMessages.h

10.201 gdcm::network::NGetRQ Class Reference

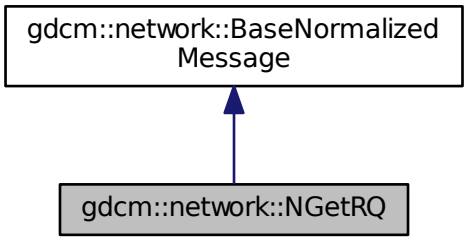
NGetRQ.

```
#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)

10.201.1 Detailed Description

NGetRQ.

this file defines the messages for the nget action

10.201.2 Member Function Documentation

10.201.2.1 ConstructPDV()

```
std::vector<PresentationDataValue> gdcm::network::NGetRQ::ConstructPDV (
            const ULConnection & inConnection,
            const BaseQuery * inQuery )  [virtual]
```

Implements gdcm::network::BaseNormalizedMessage.

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

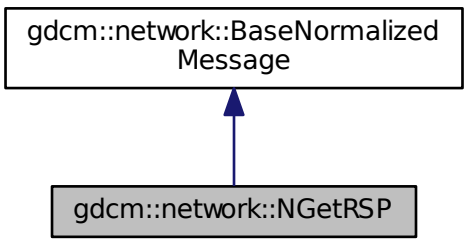
- gdcmNGetMessages.h

10.202 gdcm::network::NGetRSP Class Reference

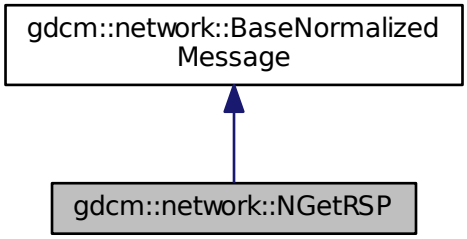
NGetRSP this file defines the messages for the nget action.

```
#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)

10.202.1 Detailed Description

NGetRSP this file defines the messages for the nget action.

10.202.2 Member Function Documentation

10.202.2.1 ConstructPDVByDataSet()

```
std::vector<PresentationDataValue> gdcm::network::NGetRSP::ConstructPDVByDataSet (
            const DataSet * inDataSet )
```

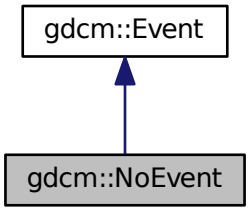
The documentation for this class was generated from the following file:

- gdcmNGetMessages.h

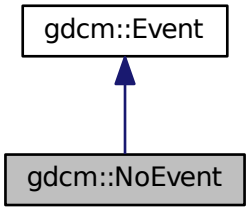
10.203 gdcm::NoEvent Class Reference

`#include <gdcmEvent.h>`

Inheritance diagram for gdcm::NoEvent:



Collaboration diagram for gdcm::NoEvent:



Additional Inherited Members

10.203.1 Detailed Description

Define some common GDCM events

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

- gdcmEvent.h

10.204 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)

10.204.1 Member Function Documentation

10.204.1.1 ConstructNAction()

```
static std::vector<PresentationDataValue> gdcm::network::NormalizedMessageFactory::ConstructN↩
Action (
            const ULConnection & inConnection,
            const BaseQuery * inQuery )  [static]
```

10.204.1.2 ConstructNCreate()

```
static std::vector<PresentationDataValue> gdcm::network::NormalizedMessageFactory::ConstructN↩
Create (
            const ULConnection & inConnection,
            const BaseQuery * inQuery )  [static]
```

10.204.1.3 ConstructNDelete()

```
static std::vector<PresentationDataValue> gdcm::network::NormalizedMessageFactory::ConstructN↩
Delete (
            const ULConnection & inConnection,
            const BaseQuery * inQuery )  [static]
```

10.204.1.4 ConstructNEventReport()

```
static std::vector<PresentationDataValue> gdcm::network::NormalizedMessageFactory::ConstructN↩
EventReport (
            const ULConnection & inConnection,
            const BaseQuery * inQuery )  [static]
```

10.204.1.5 ConstructNGet()

```
static std::vector<PresentationDataValue> gdcm::network::NormalizedMessageFactory::ConstructNGet
(
            const ULConnection & inConnection,
            const BaseQuery * inQuery )  [static]
```

10.204.1.6 ConstructNSet()

```
static std::vector<PresentationDataValue> gdcm::network::NormalizedMessageFactory::ConstructNSet
(
            const ULConnection & inConnection,
            const BaseQuery * inQuery )  [static]
```

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

- gdcmNormalizedMessageFactory.h

10.205 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)

10.205.1 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

10.205.2 Member Function Documentation

10.205.2.1 ConstructQuery()

```
static BaseQuery* gdcm::NormalizedNetworkFunctions::ConstructQuery (
             const std::string & sopInstanceUID,
             const DataSet & queryds,
             ENQueryType queryType = eCreateMMPS )  [static]
```

10.205.2.2 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]
```

10.205.2.3 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]
```

10.205.2.4 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]
```

10.205.2.5 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]
```

10.205.2.6 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]
```

10.205.2.7 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:

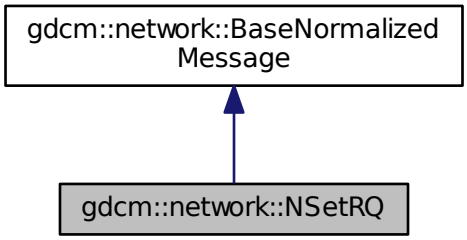
- gdcmNormalizedNetworkFunctions.h

10.206 gdcm::network::NSetRQ Class Reference

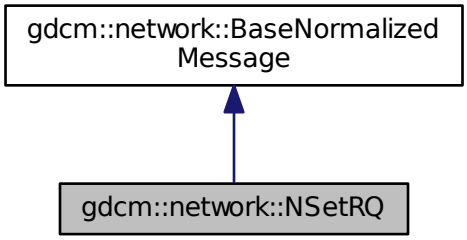
NSetRQ.

`#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)

10.206.1 Detailed Description

NSetRQ.

this file defines the messages for the nset action

10.206.2 Member Function Documentation

10.206.2.1 ConstructPDV()

```
std::vector<PresentationDataValue> gdcm::network::NSetRQ::ConstructPDV (
            const ULConnection & inConnection,
            const BaseQuery * inQuery )  [virtual]
```

Implements gdcm::network::BaseNormalizedMessage.

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

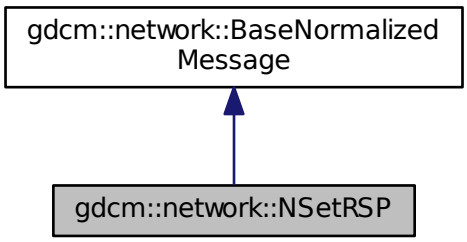
- gdcmNSetMessages.h

10.207 gdcm::network::NSetRSP Class Reference

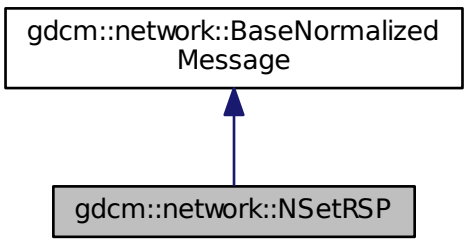
NSetRSP this file defines the messages for the nset action.

`#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)

10.207.1 Detailed Description

NSetRSP this file defines the messages for the nset action.

10.207.2 Member Function Documentation

10.207.2.1 ConstructPDVByDataSet()

```
std::vector<PresentationDataValue> gdcm::network::NSetRSP::ConstructPDVByDataSet (
            const DataSet * inDataSet )
```

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

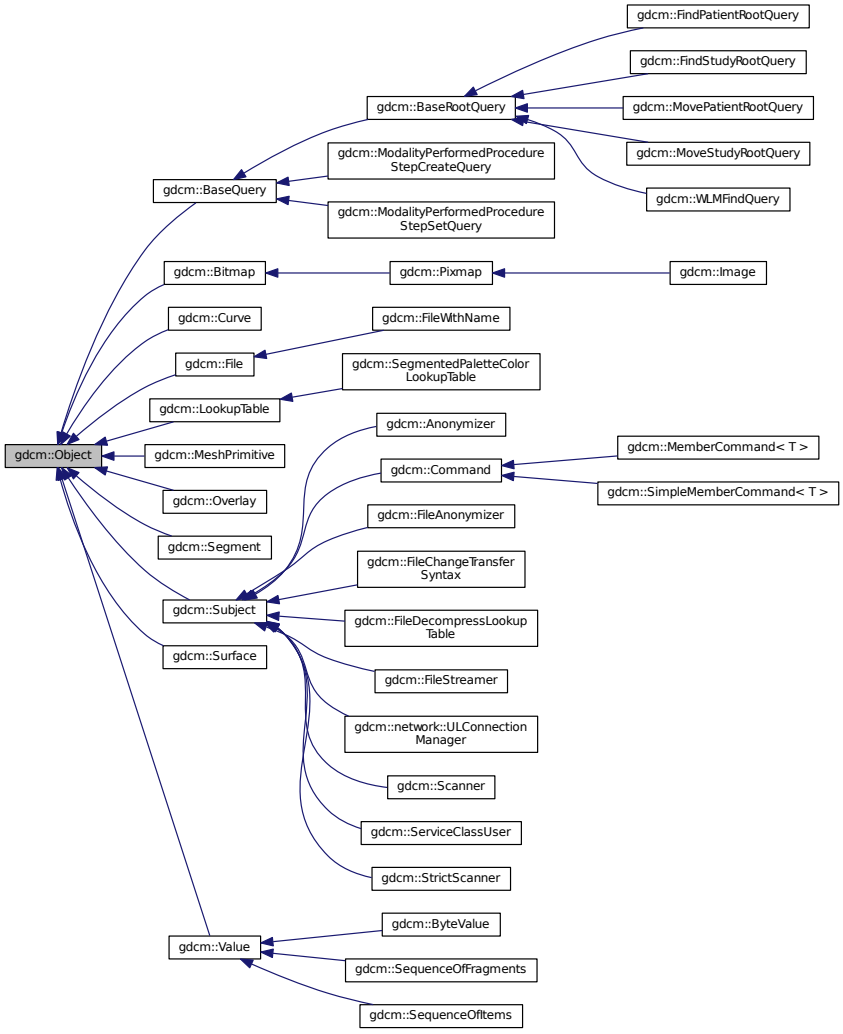
- gdcmNSetMessages.h

10.208 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

10.208.1 Detailed Description

Object.

Note

main superclass for object that want to use SmartPointer invasive ref counting system

See also

SmartPointer

10.208.2 Constructor & Destructor Documentation

10.208.2.1 Object() [1/2]

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

10.208.2.2 ~Object()

```
virtual gdcm::Object::~Object ( )  [inline], [virtual]
```

10.208.2.3 Object() [2/2]

```
gdcm::Object::Object (
            const Object &  )  [inline]
```

Special requirement for copy/cstor, assignment operator.

10.208.3 Member Function Documentation

10.208.3.1 operator=()

```
void gdcm::Object::operator= (
            const Object &  )  [inline]
```

10.208.3.2 Print()

```
virtual void gdcm::Object::Print (
            std::ostream &  ) const  [inline], [virtual]
```

Reimplemented in gdcm::SequenceOfFragments, gdcm::SequenceOfItems, gdcm::ByteValue, gdcm::Scanner, gdcm::StrictScanner, gdcm::Image, gdcm::BaseQuery, gdcm::Curve, gdcm::Overlay, gdcm::Bitmap, gdcm::LookupTable, gdcm::Pixmap, and gdcm::SegmentedPaletteColorLookupTable.

Examples:

ReadAndDumpDICOMDIR.cxx.

Referenced by gdcm::operator<<().

10.208.3.3 Register()

```
void gdcm::Object::Register ( )  [inline], [protected]
```

10.208.3.4 UnRegister()

```
void gdcm::Object::UnRegister ( )  [inline], [protected]
```

10.208.4 Friends And Related Function Documentation

10.208.4.1 operator<<

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

10.208.4.2 SmartPointer

```
template<class ObjectType >
friend class SmartPointer  [friend]
```

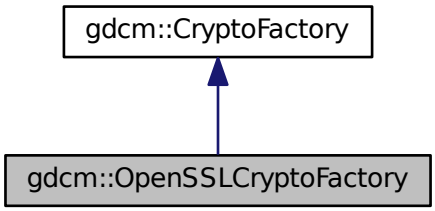
The documentation for this class was generated from the following file:

- gdcmObject.h

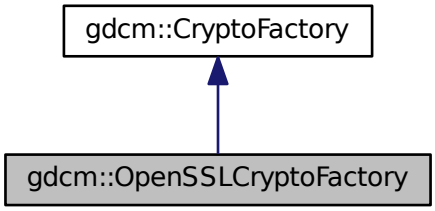
10.209 gdcm::OpenSSLCryptoFactory Class Reference

`#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

10.209.1 Constructor & Destructor Documentation

10.209.1.1 OpenSSLCryptoFactory()

```
gdcm::OpenSSLCryptoFactory::OpenSSLCryptoFactory (
            CryptoLib id )  [inline]
```

References gdcmDebugMacro.

10.209.2 Member Function Documentation

10.209.2.1 CreateCMSProvider()

```
CryptographicMessageSyntax* gdcm::OpenSSLCryptoFactory::CreateCMSProvider ( )  [inline], [virtual]
```

Implements gdcm::CryptoFactory.

10.209.2.2 InitOpenSSL()

```
void gdcm::OpenSSLCryptoFactory::InitOpenSSL ( )  [protected]
```

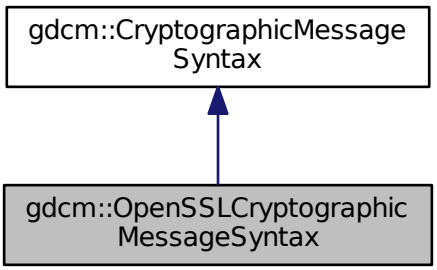
The documentation for this class was generated from the following file:

- gdcmOpenSSLCryptoFactory.h

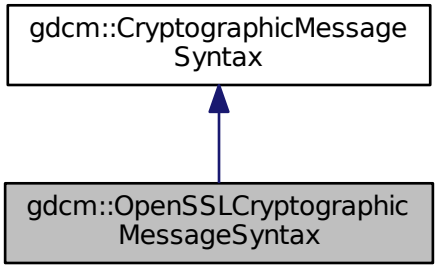
10.210 gdcm::OpenSSLCryptographicMessageSyntax Class Reference

```
#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

10.210.1 Constructor & Destructor Documentation

10.210.1.1 OpenSSLCryptographicMessageSyntax()

```
gdcm::OpenSSLCryptographicMessageSyntax::OpenSSLCryptographicMessageSyntax ( )
```

10.210.1.2 ~OpenSSLCryptographicMessageSyntax()

```
gdcm::OpenSSLCryptographicMessageSyntax::~OpenSSLCryptographicMessageSyntax ( )
```

10.210.2 Member Function Documentation

10.210.2.1 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.

10.210.2.2 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.

10.210.2.3 GetCipherType()

```
CipherTypes gdcm::OpenSSLCryptographicMessageSyntax::GetCipherType ( ) const  [virtual]
```

Implements gdcm::CryptographicMessageSyntax.

10.210.2.4 ParseCertificateFile()

```
bool gdcm::OpenSSLCryptographicMessageSyntax::ParseCertificateFile (
            const char * filename )  [virtual]
```

Implements gdcm::CryptographicMessageSyntax.

10.210.2.5 ParseKeyFile()

```
bool gdcm::OpenSSLCryptographicMessageSyntax::ParseKeyFile (
            const char * filename )  [virtual]
```

Implements gdcm::CryptographicMessageSyntax.

10.210.2.6 SetCipherType()

```
void gdcm::OpenSSLCryptographicMessageSyntax::SetCipherType (
            CipherTypes type )  [virtual]
```

Set Cipher Type. Default is: AES256_CIPHER

Implements gdcm::CryptographicMessageSyntax.

10.210.2.7 SetPassword()

```
bool gdcm::OpenSSLCryptographicMessageSyntax::SetPassword (
            const char * pass,
            size_t passLen )  [virtual]
```

Implements gdcm::CryptographicMessageSyntax.

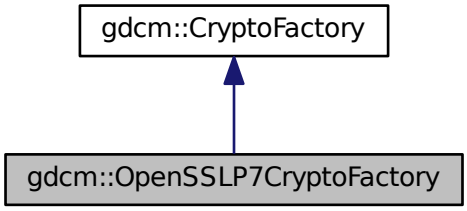
The documentation for this class was generated from the following file:

- gdcmOpenSSLCryptographicMessageSyntax.h

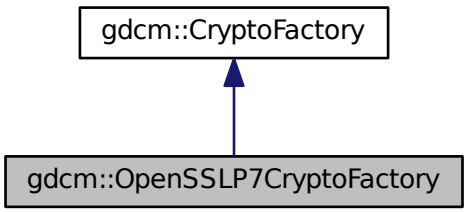
10.211 gdcm::OpenSSLP7CryptoFactory Class Reference

```
#include <gdcmOpenSSLP7CryptoFactory.h>
```

Inheritance diagram for gdcm::OpenSSLP7CryptoFactory:



Collaboration diagram for gdcm::OpenSSLP7CryptoFactory:



Public Member Functions

- OpenSSLP7CryptoFactory (CryptoLib id)
- CryptographicMessageSyntax * CreateCMSProvider ()

Additional Inherited Members

10.211.1 Constructor & Destructor Documentation

10.211.1.1 OpenSSLP7CryptoFactory()

```
gdcm::OpenSSLP7CryptoFactory::OpenSSLP7CryptoFactory (
            CryptoLib id ) [inline]
```

References gdcmDebugMacro.

10.211.2 Member Function Documentation

10.211.2.1 CreateCMSProvider()

```
CryptographicMessageSyntax* gdcm::OpenSSLP7CryptoFactory::CreateCMSProvider ( ) [inline], [virtual]
```

Implements gdcm::CryptoFactory.

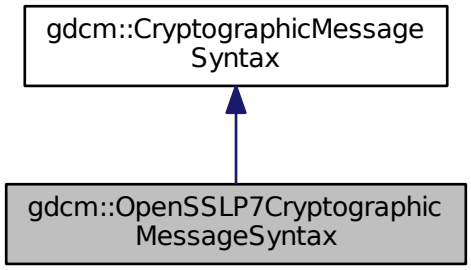
The documentation for this class was generated from the following file:

- gdcmOpenSSLP7CryptoFactory.h

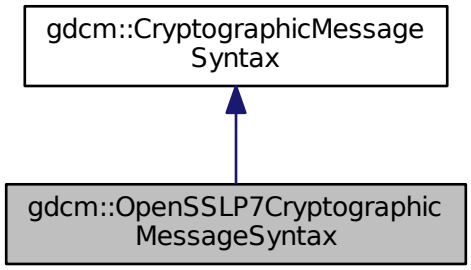
10.212 gdcm::OpenSSLP7CryptographicMessageSyntax Class Reference

`#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

10.212.1 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`

10.212.2 Constructor & Destructor Documentation

10.212.2.1 OpenSSLP7CryptographicMessageSyntax()

```
gdcm::OpenSSLP7CryptographicMessageSyntax::OpenSSLP7CryptographicMessageSyntax ( )
```

10.212.2.2 ~OpenSSLP7CryptographicMessageSyntax()

```
gdcm::OpenSSLP7CryptographicMessageSyntax::~OpenSSLP7CryptographicMessageSyntax ( )
```

10.212.3 Member Function Documentation

10.212.3.1 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.

10.212.3.2 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.

10.212.3.3 GetCipherType()

```
CipherTypes gdcm::OpenSSLP7CryptographicMessageSyntax::GetCipherType ( ) const  [virtual]
```

Implements gdcm::CryptographicMessageSyntax.

10.212.3.4 ParseCertificateFile()

```
bool gdcm::OpenSSLP7CryptographicMessageSyntax::ParseCertificateFile (
            const char * filename )  [virtual]
```

Implements gdcm::CryptographicMessageSyntax.

10.212.3.5 ParseKeyFile()

```
bool gdcm::OpenSSLP7CryptographicMessageSyntax::ParseKeyFile (
            const char * filename )  [virtual]
```

Implements gdcm::CryptographicMessageSyntax.

10.212.3.6 SetCipherType()

```
void gdcm::OpenSSLP7CryptographicMessageSyntax::SetCipherType (
            CipherTypes type )  [virtual]
```

Set Cipher Type. Default is: AES256_CIPHER

Implements gdcm::CryptographicMessageSyntax.

10.212.3.7 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

10.213 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)

10.213.1 Detailed Description

class to handle Orientation

10.213.2 Member Enumeration Documentation

10.213.2.1 OrientationType

```
enum gdcm::Orientation::OrientationType
```

Enumerator

UNKNOWN	
AXIAL	
CORONAL	
SAGITTAL	
OBLIQUE	

10.213.3 Constructor & Destructor Documentation

10.213.3.1 Orientation()

```
gdcm::Orientation::Orientation ( )
```

10.213.3.2 ~Orientation()

```
gdcm::Orientation::~Orientation ( )
```

10.213.4 Member Function Documentation

10.213.4.1 GetLabel()

```
static const char* gdcm::Orientation::GetLabel (
            OrientationType type )  [static]
```

Return the label of an Orientation.

Examples:

FixOrientation.cxx.

10.213.4.2 GetMajorAxisFromPatientRelativeDirectionCosine()

```
static char gdcm::Orientation::GetMajorAxisFromPatientRelativeDirectionCosine (
            double x,
            double y,
            double z )  [static], [protected]
```

10.213.4.3 GetObliquityThresholdCosineValue()

```
static double gdcm::Orientation::GetObliquityThresholdCosineValue ( )  [static]
```

10.213.4.4 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.

10.213.4.5 Print()

```
void gdcm::Orientation::Print (
            std::ostream &  ) const
```

Print.

Referenced by gdcm::operator<<().

10.213.4.6 SetObliquityThresholdCosineValue()

```
static void gdcm::Orientation::SetObliquityThresholdCosineValue (
            double val )  [static]
```

ObliquityThresholdCosineValue stuff.

10.213.5 Friends And Related Function Documentation

10.213.5.1 operator<<

```
std::ostream& operator<< (
            std::ostream & _os,
            const Orientation & o )  [friend]
```

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

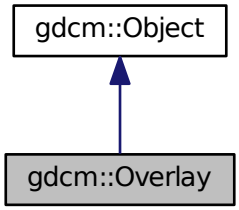
- gdcmOrientation.h

10.214 gdcm::Overlay Class Reference

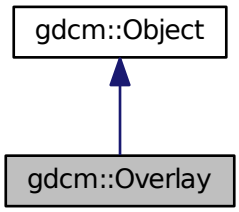
Overlay class.

```
#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 ()
- 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

  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

10.214.1 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 differenciate them (other than the group tag).

Example:

10.214.2 Member Enumeration Documentation

10.214.2.1 OverlayType

```
enum gdcm::Overlay::OverlayType
```

Enumerator

Invalid	
Graphics	
ROI	

10.214.3 Constructor & Destructor Documentation

10.214.3.1 Overlay() [1/2]

```
gdcm::Overlay::Overlay ( )
```

10.214.3.2 ~Overlay()

```
gdcm::Overlay::~Overlay ( )
```

10.214.3.3 Overlay() [2/2]

```
gdcm::Overlay::Overlay (
            Overlay const & ov )
```

10.214.4 Member Function Documentation

10.214.4.1 Decompress()

```
void gdcm::Overlay::Decompress (
            std::ostream & os ) const
```

Decode the internal OverlayData (packed bits) into unpacked representation.

10.214.4.2 GetBitPosition()

```
unsigned short gdcm::Overlay::GetBitPosition ( ) const
```

return bit position

10.214.4.3 GetBitsAllocated()

```
unsigned short gdcm::Overlay::GetBitsAllocated ( ) const
```

return bits allocated

10.214.4.4 GetColumns()

```
unsigned short gdcm::Overlay::GetColumns ( ) const
```

get columns

10.214.4.5 GetDescription()

```
const char* gdcm::Overlay::GetDescription ( ) const
```

get description

10.214.4.6 GetGroup()

```
unsigned short gdcm::Overlay::GetGroup ( ) const
```

Get Group number.

10.214.4.7 GetOrigin()

```
const signed short* gdcm::Overlay::GetOrigin ( ) const
```

get origin

10.214.4.8 GetOverlayData()

```
const ByteValue& gdcm::Overlay::GetOverlayData ( ) const
```

Return the Overlay Data as ByteValue: Not thread safe

10.214.4.9 GetOverlayTypeAsString()

```
static const char* gdcm::Overlay::GetOverlayTypeAsString (
            OverlayType ot )  [static]
```

10.214.4.10 GetOverlayTypeFromString()

```
static OverlayType gdcm::Overlay::GetOverlayTypeFromString (
            const char *  )  [static]
```

10.214.4.11 GetRows()

```
unsigned short gdcm::Overlay::GetRows ( ) const
```

get rows

10.214.4.12 GetType()

```
const char* gdcm::Overlay::GetType ( ) const
```

get type

10.214.4.13 GetTypeAsEnum()

```
OverlayType gdcm::Overlay::GetTypeAsEnum ( ) const
```

10.214.4.14 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()

10.214.4.15 GetUnpackBufferLength()

```
size_t gdcm::Overlay::GetUnpackBufferLength ( ) const
```

Retrieve the size of the buffer needed to hold the Overlay as specified by Col & Row parameters

10.214.4.16 GrabOverlayFromPixelData()

```
bool gdcm::Overlay::GrabOverlayFromPixelData (
            DataSet const & ds )
```

10.214.4.17 IsEmpty()

```
bool gdcm::Overlay::IsEmpty ( ) const
```

Return whether or not the Overlay is empty:

10.214.4.18 IsInPixelData() [1/2]

```
bool gdcm::Overlay::IsInPixelData ( ) const
```

return if the Overlay is stored in the pixel data or not

10.214.4.19 IsInPixelData() [2/2]

```
void gdcm::Overlay::IsInPixelData (
            bool b )
```

Set whether or no the OverlayData is in the Pixel Data:

10.214.4.20 IsZero()

```
bool gdcm::Overlay::IsZero ( ) const
```

return true if all bits are set to 0

10.214.4.21 operator=()

```
Overlay& gdcm::Overlay::operator= (
            Overlay const & ov )
```

10.214.4.22 Print()

```
void gdcm::Overlay::Print (
            std::ostream &  ) const  [virtual]
```

Print.

Reimplemented from gdcm::Object.

10.214.4.23 SetBitPosition()

```
void gdcm::Overlay::SetBitPosition (
            unsigned short bitposition )
```

set bit position

10.214.4.24 SetBitsAllocated()

```
void gdcm::Overlay::SetBitsAllocated (
            unsigned short bitsallocated )
```

set bits allocated

10.214.4.25 SetColumns()

```
void gdcm::Overlay::SetColumns (
            unsigned short columns )
```

set columns

10.214.4.26 SetDescription()

```
void gdcm::Overlay::SetDescription (
            const char * description )
```

set description

10.214.4.27 SetFrameOrigin()

```
void gdcm::Overlay::SetFrameOrigin (
            unsigned short frameorigin )
```

set frame origin

10.214.4.28 SetGroup()

```
void gdcm::Overlay::SetGroup (
            unsigned short group )
```

Set Group number.

10.214.4.29 SetNumberOfFrames()

```
void gdcm::Overlay::SetNumberOfFrames (
            unsigned int numberofframes )
```

set number of frames

10.214.4.30 SetOrigin()

```
void gdcm::Overlay::SetOrigin (
            const signed short origin[2] )
```

set origin

10.214.4.31 SetOverlay()

```
void gdcm::Overlay::SetOverlay (
            const char * array,
            size_t length )
```

set overlay from byte array + length

10.214.4.32 SetRows()

```
void gdcm::Overlay::SetRows (
            unsigned short rows )
```

set rows

10.214.4.33 SetType()

```
void gdcm::Overlay::SetType (
            const char * type )
```

set type

10.214.4.34 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:

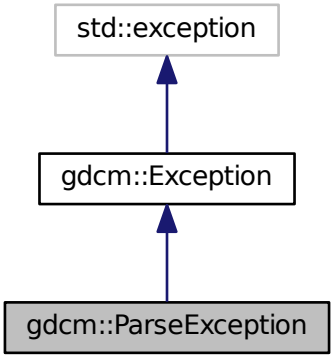
- gdcmOverlay.h

10.215 gdcm::ParseException Class Reference

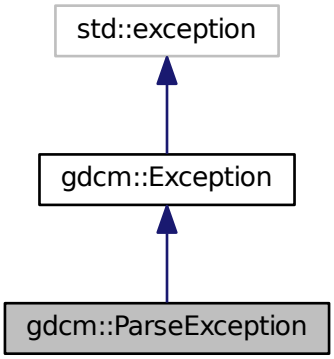
ParseException Standard exception handling object.

```
#include <gdcmParseException.h>
```

Inheritance diagram for gdcm::ParseException:



Collaboration diagram for gdcm::ParseException:



Public Member Functions

- ParseException ()
- virtual ~ParseException () throw ()
- const DataElement & GetLastElement () const
- ParseException & operator= (const ParseException &orig)
- void SetLastElement (DataElement &de)

10.215.1 Detailed Description

ParseException Standard exception handling object.

10.215.2 Constructor & Destructor Documentation

10.215.2.1 ParseException()

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

10.215.2.2 ~ParseException()

```
virtual gdcm::ParseException::~ParseException ( ) throw )   [inline], [virtual]
```

10.215.3 Member Function Documentation

10.215.3.1 GetLastElement()

```
const DataElement& gdcm::ParseException::GetLastElement ( ) const  [inline]
```

10.215.3.2 operator=()

```
ParseException& gdcm::ParseException::operator= (
            const ParseException & orig )  [inline]
```

Assignment operator.

10.215.3.3 SetLastElement()

```
void gdcm::ParseException::SetLastElement (
            DataElement & de )  [inline]
```

Equivalence operator.

Referenced by gdcm::Fragment::ReadBacktrack(), and gdcm::Fragment::ReadValue().

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

- gdcmParseException.h

10.216 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 ()

10.216.1 Detailed Description

Parser ala XML_Parser from expat (SAX)

Detailled description here

Note

Simple API for DICOM

10.216.2 Member Typedef Documentation

10.216.2.1 EndElementHandler

```
typedef void(* gdcm::Parser::EndElementHandler) (void *userData, const Tag &name)
```

10.216.2.2 StartElementHandler

```
typedef void(* gdcm::Parser::StartElementHandler) (void *userData, const Tag &tag, const char
*atts[])
```

10.216.3 Member Enumeration Documentation

10.216.3.1 ErrorType

```
enum gdcm::Parser::ErrorType
```

Enumerator

NoError	
NoMemoryError	
SyntaxError	
NoElementsError	
TagMismatchError	
DuplicateAttributeError	
JunkAfterDocElementError	
UndefinedEntityError	
UnexpectedStateError	

10.216.4 Constructor & Destructor Documentation

10.216.4.1 Parser()

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

10.216.4.2 ~Parser()

```
gdcm::Parser::~Parser ( )  [inline]
```

10.216.5 Member Function Documentation

10.216.5.1 GetBuffer()

```
char* gdcm::Parser::GetBuffer (
            int len )  [protected]
```

10.216.5.2 GetCurrentByteIndex()

```
unsigned long gdcm::Parser::GetCurrentByteIndex ( ) const
```

10.216.5.3 GetErrorCode()

```
ErrorType gdcm::Parser::GetErrorCode ( ) const
```

10.216.5.4 GetErrorString()

```
static const char* gdcm::Parser::GetErrorString (
            ErrorType const & err )  [static]
```

10.216.5.5 GetUserData()

```
void* gdcm::Parser::GetUserData ( ) const
```

10.216.5.6 Parse()

```
bool gdcm::Parser::Parse (
            const char * s,
            int len,
            bool isFinal )
```

10.216.5.7 ParseBuffer()

```
bool gdcm::Parser::ParseBuffer (
            int len,
            bool isFinal )  [protected]
```

10.216.5.8 Process()

```
ErrorType gdcm::Parser::Process ( )  [protected]
```

10.216.5.9 SetElementHandler()

```
void gdcm::Parser::SetElementHandler (
            StartElementHandler start,
            EndElementHandler end )
```

10.216.5.10 SetUserData()

```
void gdcm::Parser::SetUserData (
            void * userData )
```

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

- gdcmParser.h

10.217 gdcm::Patient Class Reference

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

```
#include <gdcmPatient.h>
```

Public Member Functions

- Patient ()

10.217.1 Detailed Description

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

10.217.2 Constructor & Destructor Documentation

10.217.2.1 Patient()

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

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

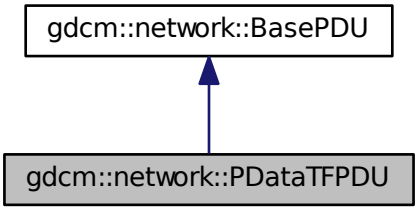
- gdcmPatient.h

10.218 gdcm::network::PDataTFPDU Class Reference

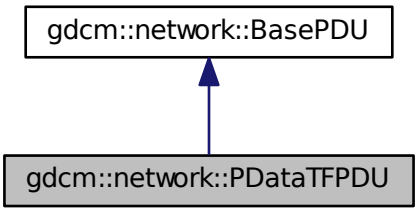
PDataTFPDU.

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

Protected Member Functions

- std::istream & ReadInto (std::istream &is, std::ostream &os)

10.218.1 Detailed Description

PDataTFPDU.

Table 9-22 P-DATA-TF PDU FIELDS

10.218.2 Member Typedef Documentation

10.218.2.1 SizeType

```
typedef std::vector<PresentationDataValue>::size_type gdcm::network::PDataTFPDU::SizeType
```

10.218.3 Constructor & Destructor Documentation

10.218.3.1 PDataTFPDU()

```
gdcm::network::PDataTFPDU::PDataTFPDU ( )
```

10.218.4 Member Function Documentation

10.218.4.1 AddPresentationDataValue()

```
void gdcm::network::PDataTFPDU::AddPresentationDataValue (
            PresentationDataValue const & pdv )  [inline]
```

10.218.4.2 GetNumberOfPresentationDataValues()

```
SizeType gdcm::network::PDataTFPDU::GetNumberOfPresentationDataValues ( ) const  [inline]
```

10.218.4.3 GetPresentationDataValue()

```
PresentationDataValue const& gdcm::network::PDataTFPDU::GetPresentationDataValue (
            SizeType i ) const  [inline]
```

10.218.4.4 IsLastFragment()

```
bool gdcm::network::PDataTFPDU::IsLastFragment ( ) const  [virtual]
```

Implements gdcm::network::BasePDU.

10.218.4.5 Print()

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

Implements gdcm::network::BasePDU.

10.218.4.6 Read()

```
std::istream& gdcm::network::PDataTFPDU::Read (
            std::istream & is )  [virtual]
```

Implements gdcm::network::BasePDU.

10.218.4.7 ReadInto()

```
std::istream& gdcm::network::PDataTFPDU::ReadInto (
            std::istream & is,
            std::ostream & os )  [protected]
```

10.218.4.8 Size()

```
size_t gdcm::network::PDataTFPDU::Size ( ) const  [virtual]
```

Implements gdcm::network::BasePDU.

10.218.4.9 Write()

```
const std::ostream& gdcm::network::PDataTFPDU::Write (
            std::ostream & os ) const  [virtual]
```

Implements gdcm::network::BasePDU.

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

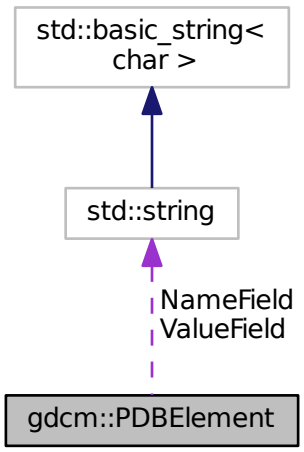
- gdcmPDataTFPDU.h

10.219 gdcm::PDBElement Class Reference

Class to represent a PDB Element.

`#include <gdcmPDBElement.h>`

Collaboration diagram for gdcm::PDBElement:



Public Member Functions

- PDBElement ()
- 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)

10.219.1 Detailed Description

Class to represent a PDB Element.

See also

PDBHeader

10.219.2 Constructor & Destructor Documentation

10.219.2.1 PDBElement()

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

References gdcm::operator<<().

10.219.3 Member Function Documentation

10.219.3.1 GetName()

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

Set/Get Name.

10.219.3.2 GetValue()

```
const char* gdcm::PDBElement::GetValue ( ) const  [inline]
```

Set/Get Value.

10.219.3.3 operator==()

```
bool gdcm::PDBElement::operator== (
            const PDBElement & de ) const  [inline]
```

References NameField, and ValueField.

10.219.3.4 SetName()

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

10.219.3.5 SetValue()

```
void gdcm::PDBElement::SetValue (
            const char * value )  [inline]
```

10.219.4 Friends And Related Function Documentation

10.219.4.1 operator<<

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

10.219.5 Member Data Documentation

10.219.5.1 NameField

```
std::string gdcm::PDBElement::NameField  [protected]
```

Referenced by gdcm::operator<<(), and operator==().

10.219.5.2 ValueField

```
std::string gdcm::PDBElement::ValueField  [protected]
```

Referenced by gdcm::operator<<(), and operator==().

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

- gdcmPDBElement.h

10.220 gdcm::PDBHeader Class Reference

Class for PDBHeader.

```
#include <gdcmPDBHeader.h>
```

Public Member Functions

- PDBHeader ()
- ~PDBHeader ()
- 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)

10.220.1 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.

See also

CSAHeader

10.220.2 Constructor & Destructor Documentation

10.220.2.1 PDBHeader()

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

10.220.2.2 ~PDBHeader()

```
gdcm::PDBHeader::~PDBHeader ( )  [inline]
```

10.220.3 Member Function Documentation

10.220.3.1 FindPDBElementByName()

```
bool gdcm::PDBHeader::FindPDBElementByName (
            const char * name )
```

Return true if the PDB element matching name is found or not.

10.220.3.2 GetPDBEEnd()

```
const PDBElement& gdcm::PDBHeader::GetPDBEEnd ( ) const  [protected]
```

10.220.3.3 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

10.220.3.4 GetPDBInfoTag()

```
static const PrivateTag& gdcm::PDBHeader::GetPDBInfoTag ( )  [static]
```

Return the Private Tag where the PDB header is stored within a DICOM DataSet.

10.220.3.5 LoadFromDataElement()

```
bool gdcm::PDBHeader::LoadFromDataElement (
            DataElement const & de )
```

Load the PDB Header from a DataElement of a DataSet.

10.220.3.6 Print()

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

Print.

Referenced by gdcm::operator<<().

10.220.4 Friends And Related Function Documentation

10.220.4.1 operator<<

```
std::ostream& operator<< (
            std::ostream & _os,
            const PDBHeader & d )  [friend]
```

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

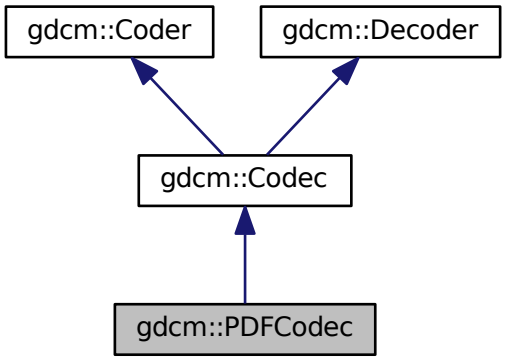
- gdcmPDBHeader.h

10.221 gdcm::PDFCodec Class Reference

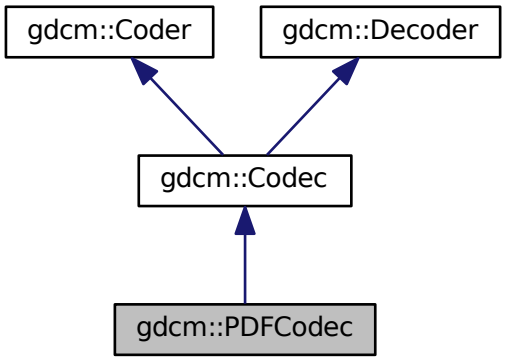
PDFCodec class.

```
#include <gdcmPDFCodec.h>
```

Inheritance diagram for gdcm::PDFCodec:



Collaboration diagram for gdcm::PDFCodec:



Public Member Functions

- PDFCodec ()
- ~PDFCodec ()
- bool CanCode (TransferSyntax const &) const

  Return whether this coder support this transfer syntax (can code it)
- bool CanDecode (TransferSyntax const &) const

  Return whether this decoder support this transfer syntax (can decode it)
- bool Decode (DataElement const &is, DataElement &os)

  Decode.

Additional Inherited Members

10.221.1 Detailed Description

PDFCodec class.

10.221.2 Constructor & Destructor Documentation

10.221.2.1 PDFCodec()

```
gdcm::PDFCodec::PDFCodec ( )
```

10.221.2.2 ~PDFCodec()

```
gdcm::PDFCodec::~PDFCodec ( )
```

10.221.3 Member Function Documentation

10.221.3.1 CanCode()

```
bool gdcm::PDFCodec::CanCode (
            TransferSyntax const &  ) const  [inline], [virtual]
```

Return whether this coder support this transfer syntax (can code it)

Implements gdcm::Coder.

10.221.3.2 CanDecode()

```
bool gdcm::PDFCodec::CanDecode (
            TransferSyntax const &  ) const  [inline], [virtual]
```

Return whether this decoder support this transfer syntax (can decode it)

Implements gdcm::Decoder.

10.221.3.3 Decode()

```
bool gdcm::PDFCodec::Decode (
             DataElement const & ,
             DataElement &  )  [virtual]
```

Decode.

Reimplemented from gdcm::Decoder.

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

- gdcmPDFCodec.h

10.222 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 *inQuery)
- static std::vector< BasePDU * > CreateNSetPDU (const ULConnection &inConnection, const BaseQuery *inQuery)
- static EEventID DetermineEventByPDU (const BasePDU *inPDU)
- static std::vector< PresentationDataValue > GetPDVs (const std::vector< BasePDU *> &inDataPDUs)

10.222.1 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.

10.222.2 Member Function Documentation

10.222.2.1 ConstructAbortPDU()

```
static BasePDU* gdcm::network::PDUFactory::ConstructAbortPDU ( )  [static]
```

10.222.2.2 ConstructPDU()

```
static BasePDU* gdcm::network::PDUFactory::ConstructPDU (
            uint8_t itemtype )  [static]
```

10.222.2.3 ConstructReleasePDU()

```
static BasePDU* gdcm::network::PDUFactory::ConstructReleasePDU ( )  [static]
```

10.222.2.4 CreateCEchoPDU()

```
static std::vector<BasePDU*> gdcm::network::PDUFactory::CreateCEchoPDU (
            const ULConnection & inConnection )  [static]
```

10.222.2.5 CreateCFindPDU()

```
static std::vector<BasePDU*> gdcm::network::PDUFactory::CreateCFindPDU (
            const ULConnection & inConnection,
            const BaseRootQuery * inRootQuery )  [static]
```

10.222.2.6 CreateCMovePDU()

```
static std::vector<BasePDU*> gdcm::network::PDUFactory::CreateCMovePDU (
            const ULConnection & inConnection,
            const BaseRootQuery * inRootQuery )  [static]
```

10.222.2.7 CreateCStoreRQPDU()

```
static std::vector<BasePDU*> gdcm::network::PDUFactory::CreateCStoreRQPDU (
            const ULConnection & inConnection,
            const File & file,
            bool writeDataSet = true )  [static]
```

10.222.2.8 CreateCStoreRSPPDU()

```
static std::vector<BasePDU*> gdcm::network::PDUFactory::CreateCStoreRSPPDU (
            const DataSet * inDataSet,
            const BasePDU * inPC )  [static]
```

10.222.2.9 CreateNActionPDU()

```
static std::vector<BasePDU*> gdcm::network::PDUFactory::CreateNActionPDU (
            const ULConnection & inConnection,
            const BaseQuery * inQuery )  [static]
```

10.222.2.10 CreateNCreatePDU()

```
static std::vector<BasePDU*> gdcm::network::PDUFactory::CreateNCreatePDU (
            const ULConnection & inConnection,
            const BaseQuery * inQuery )  [static]
```

10.222.2.11 CreateNDeletePDU()

```
static std::vector<BasePDU*> gdcm::network::PDUFactory::CreateNDeletePDU (
            const ULConnection & inConnection,
            const BaseQuery * inQuery )  [static]
```

10.222.2.12 CreateNEventReportPDU()

```
static std::vector<BasePDU*> gdcm::network::PDUFactory::CreateNEventReportPDU (
            const ULConnection & inConnection,
            const BaseQuery * inQuery )  [static]
```

10.222.2.13 CreateNGetPDU()

```
static std::vector<BasePDU*> gdcm::network::PDUFactory::CreateNGetPDU (
            const ULConnection & inConnection,
            const BaseQuery * inQuery )  [static]
```

10.222.2.14 CreateNSetPDU()

```
static std::vector<BasePDU*> gdcm::network::PDUFactory::CreateNSetPDU (
            const ULConnection & inConnection,
            const BaseQuery * inQuery )  [static]
```

10.222.2.15 DetermineEventByPDU()

```
static EEventID gdcm::network::PDUFactory::DetermineEventByPDU (
            const BasePDU * inPDU )  [static]
```

10.222.2.16 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

10.223 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 = '^'

10.223.1 Detailed Description

PersonName class.

10.223.2 Member Function Documentation

10.223.2.1 GetMaxLength()

```
unsigned int gdcm::PersonName::GetMaxLength ( ) const  [inline]
```

10.223.2.2 GetNumberOfComponents()

```
unsigned int gdcm::PersonName::GetNumberOfComponents ( ) const  [inline]
```

10.223.2.3 Print()

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

10.223.2.4 SetBlob()

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

10.223.2.5 SetComponents() [1/2]

```
void gdcm::PersonName::SetComponents (
            const char * comp1 = "",
            const char * comp2 = "",
            const char * comp3 = "",
            const char * comp4 = "",
            const char * comp5 = "" )  [inline]
```

10.223.2.6 SetComponents() [2/2]

```
void gdcm::PersonName::SetComponents (
            const char * components[] )  [inline]
```

10.223.3 Member Data Documentation

10.223.3.1 Component

```
char gdcm::PersonName::Component[MaxNumberOfComponents][MaxLength+1]
```

10.223.3.2 MaxLength

```
const unsigned int gdcm::PersonName::MaxLength = 64  [static]
```

10.223.3.3 MaxNumberOfComponents

```
const unsigned int gdcm::PersonName::MaxNumberOfComponents = 5  [static]
```

10.223.3.4 Padding

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

10.223.3.5 Separator

```
const char gdcm::PersonName::Separator = '^'  [static]
```

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

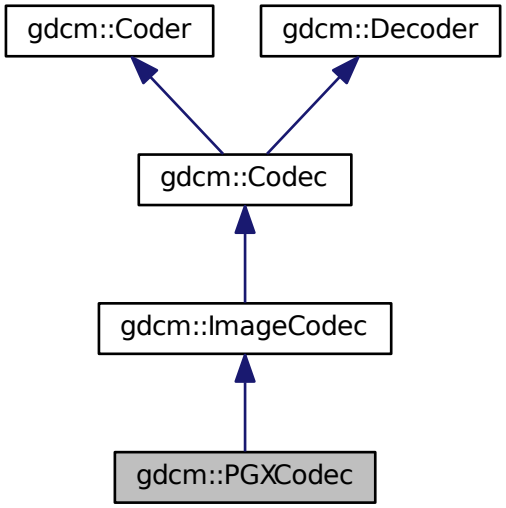
- gdcmPersonName.h

10.224 gdcm::PGXCodec Class Reference

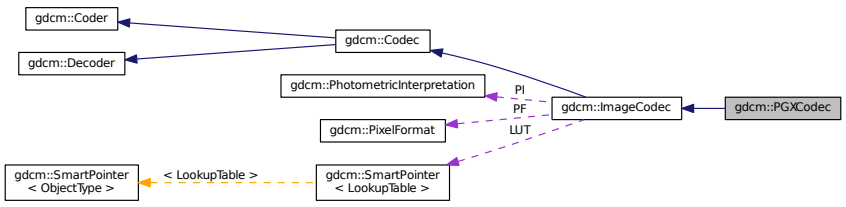
Class to do PGX.

`#include <gdcmPGXCodec.h>`

Inheritance diagram for gdcm::PGXCodec:



Collaboration diagram for gdcm::PGXCodec:



Public Member Functions

- PGXCodec ()
- ~PGXCodec ()
- bool CanCode (TransferSyntax const &ts) const

Return whether this coder support this transfer syntax (can code it)

- bool CanDecode (TransferSyntax const &ts) const

  Return whether this decoder support this transfer syntax (can decode it)
- virtual ImageCodec * Clone () const
- bool GetHeaderInfo (std::istream &is, TransferSyntax &ts)
- bool Read (const char *filename, DataElement &out) const
- bool Write (const char *filename, const DataElement &out) const

Additional Inherited Members

10.224.1 Detailed Description

Class to do PGX.

See PGX as used in JPEG 2000 implementation and reference images

10.224.2 Constructor & Destructor Documentation

10.224.2.1 PGXCodec()

```
gdcm::PGXCodec::PGXCodec ( )
```

10.224.2.2 ~PGXCodec()

```
gdcm::PGXCodec::~PGXCodec ( )
```

10.224.3 Member Function Documentation

10.224.3.1 CanCode()

```
bool gdcm::PGXCodec::CanCode (
            TransferSyntax const &  ) const  [virtual]
```

Return whether this coder support this transfer syntax (can code it)

Reimplemented from gdcm::ImageCodec.

10.224.3.2 CanDecode()

```
bool gdcm::PGXCodec::CanDecode (
            TransferSyntax const &  ) const  [virtual]
```

Return whether this decoder support this transfer syntax (can decode it)

Reimplemented from gdcm::ImageCodec.

10.224.3.3 Clone()

```
virtual ImageCodec* gdcm::PGXCodec::Clone ( ) const  [virtual]
```

Implements gdcm::ImageCodec.

10.224.3.4 GetHeaderInfo()

```
bool gdcm::PGXCodec::GetHeaderInfo (
             std::istream & is,
             TransferSyntax & ts )  [virtual]
```

Reimplemented from gdcm::ImageCodec.

10.224.3.5 Read()

```
bool gdcm::PGXCodec::Read (
             const char * filename,
             DataElement & out ) const
```

10.224.3.6 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

10.225 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)

10.225.1 Detailed Description

Class to represent an PhotometricInterpretation.

Examples:

CreateARGBImage.cxx, CreateCMYKImage.cxx, csa2img.cxx, HelloVizWorld.cxx, and iU22tomultisc.cxx.

10.225.2 Member Enumeration Documentation

10.225.2.1 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	

10.225.3 Constructor & Destructor Documentation

10.225.3.1 PhotometricInterpretation()

```
gdcm::PhotometricInterpretation::PhotometricInterpretation (
            PIType pi = UNKNOWN )  [inline]
```

References gdcm::operator<<().

10.225.4 Member Function Documentation

10.225.4.1 GetPIString()

```
static const char* gdcm::PhotometricInterpretation::GetPIString (
            PIType pi )  [static]
```

Referenced by gdcm::operator<<().

10.225.4.2 GetPIType()

```
static PIType gdcm::PhotometricInterpretation::GetPIType (
            const char * pi )  [static]
```

10.225.4.3 GetSamplesPerPixel()

```
unsigned short gdcm::PhotometricInterpretation::GetSamplesPerPixel ( ) const
```

return the value for Sample Per Pixel associated with a particular Photometric Interpretation

10.225.4.4 GetString()

```
const char* gdcm::PhotometricInterpretation::GetString ( ) const
```

10.225.4.5 GetType()

```
PIType gdcm::PhotometricInterpretation::GetType ( ) const  [inline]
```

10.225.4.6 IsLossless()

```
bool gdcm::PhotometricInterpretation::IsLossless ( ) const
```

10.225.4.7 IsLossy()

```
bool gdcm::PhotometricInterpretation::IsLossy ( ) const
```

10.225.4.8 IsRetired()

```
static bool gdcm::PhotometricInterpretation::IsRetired (
            PIType pi )  [static]
```

10.225.4.9 IsSameColorSpace()

```
bool gdcm::PhotometricInterpretation::IsSameColorSpace (
            PhotometricInterpretation const & pi ) const
```

10.225.4.10 operator PIType()

```
gdcm::PhotometricInterpretation::operator PIType ( ) const  [inline]
```

10.225.5 Friends And Related Function Documentation

10.225.5.1 operator<<

```
std::ostream& operator<< (
            std::ostream & os,
            const PhotometricInterpretation & pi )  [friend]
```

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

- gdcmPhotometricInterpretation.h

10.226 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 (unsigned short samplesperpixel=1, unsigned short bitsallocated=8, unsigned short bitsstored=8, unsigned short highbit=7, unsigned short pixelrepresentation=0)
- PixelFormat (ScalarType st)
- 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!= (ScalarType st) const
- bool operator!= (const PixelFormat &pf) const
- bool operator== (ScalarType st) const
- bool operator== (const PixelFormat &pf) 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)

10.226.1 Detailed Description

PixelFormat.

By default the Pixel Type will be instanciated 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, csa2img.cxx, FixJAIBugJPEGLS.cxx, GetJPEGSample↩Precision.cxx, iU22tomultisc.cxx, and threadgdcm.cxx.

10.226.2 Member Enumeration Documentation

10.226.2.1 ScalarType

`enum gdcm::PixelFormat::ScalarType`

Enumerator

UINT8	
INT8	
UINT12	
INT12	
UINT16	
INT16	
UINT32	
INT32	
UINT64	
INT64	
FLOAT16	
FLOAT32	
FLOAT64	
SINGLEBIT	
UNKNOWN	

10.226.3 Constructor & Destructor Documentation

10.226.3.1 PixelFormat() [1/2]

```
gdcm::PixelFormat::PixelFormat (
            unsigned short samplesperpixel = 1,
            unsigned short bitsallocated = 8,
            unsigned short bitsstored = 8,
            unsigned short highbit = 7,
            unsigned short pixelrepresentation = 0 )  [inline], [explicit]
```

10.226.3.2 PixelFormat() [2/2]

```
gdcm::PixelFormat::PixelFormat (
            ScalarType st )
```

10.226.4 Member Function Documentation

10.226.4.1 GetBitsAllocated()

```
unsigned short gdcm::PixelFormat::GetBitsAllocated ( ) const  [inline]
```

BitsAllocated see Tag (0028,0100) US Bits Allocated.

Examples:

GetJPEGSamplePrecision.cxx.

10.226.4.2 GetBitsStored()

```
unsigned short gdcm::PixelFormat::GetBitsStored ( ) const  [inline]
```

BitsStored see Tag (0028,0101) US Bits Stored.

Examples:

GetJPEGSamplePrecision.cxx.

10.226.4.3 GetHighBit()

```
unsigned short gdcm::PixelFormat::GetHighBit ( ) const  [inline]
```

HighBit see Tag (0028,0102) US High Bit.

10.226.4.4 GetMax()

```
int64_t gdcm::PixelFormat::GetMax ( ) const
```

return the max possible of the pixel

10.226.4.5 GetMin()

```
int64_t gdcm::PixelFormat::GetMin ( ) const
```

return the min possible of the pixel

10.226.4.6 GetPixelRepresentation()

```
unsigned short gdcm::PixelFormat::GetPixelRepresentation ( ) const  [inline]
```

PixelRepresentation: 0 or 1, see Tag (0028,0103) US Pixel Representation.

10.226.4.7 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:

threadgdcm.cxx.

10.226.4.8 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.

10.226.4.9 GetScalarType()

```
ScalarType gdcm::PixelFormat::GetScalarType ( ) const
```

ScalarType does not take into account the sample per pixel.

10.226.4.10 GetScalarTypeAsString()

```
const char* gdcm::PixelFormat::GetScalarTypeAsString ( ) const
```

10.226.4.11 IsCompatible()

```
bool gdcm::PixelFormat::IsCompatible (
            const TransferSyntax & ts ) const
```

10.226.4.12 IsValid()

```
bool gdcm::PixelFormat::IsValid ( ) const
```

return IsValid

10.226.4.13 operator ScalarType()

```
gdcm::PixelFormat::operator ScalarType ( ) const  [inline]
```

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

```
bool gdcm::PixelFormat::operator!= (
            ScalarType st ) const  [inline]
```

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

```
bool gdcm::PixelFormat::operator!= (
            const PixelFormat & pf ) const  [inline]
```

10.226.4.16 operator==() [1/2]

```
bool gdcm::PixelFormat::operator== (
            ScalarType st ) const  [inline]
```

10.226.4.17 operator==() [2/2]

```
bool gdcm::PixelFormat::operator== (
            const PixelFormat & pf ) const  [inline]
```

10.226.4.18 Print()

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

Print.

Referenced by gdcm::operator<<().

10.226.4.19 SetBitsAllocated()

```
void gdcm::PixelFormat::SetBitsAllocated (
            unsigned short ba )  [inline]
```

10.226.4.20 SetBitsStored()

```
void gdcm::PixelFormat::SetBitsStored (
            unsigned short bs )  [inline]
```

10.226.4.21 SetHighBit()

```
void gdcm::PixelFormat::SetHighBit (
            unsigned short hb )  [inline]
```

10.226.4.22 SetPixelRepresentation()

```
void gdcm::PixelFormat::SetPixelRepresentation (
            unsigned short pr )  [inline]
```

10.226.4.23 SetSamplesPerPixel()

```
void gdcm::PixelFormat::SetSamplesPerPixel (
            unsigned short spp )  [inline]
```

Examples:

CreateARGBImage.cxx, CreateCMYKImage.cxx, and GenFakeImage.cxx.

References gdcmAssertMacro.

10.226.4.24 SetScalarType()

```
void gdcm::PixelFormat::SetScalarType (
            ScalarType st )
```

Set PixelFormat based only on the ScalarType

Warning

: You need to call SetScalarType *before* SetSamplesPerPixel

10.226.4.25 Validate()

```
bool gdcm::PixelFormat::Validate ( )  [protected]
```

When image with 24/24/23 was read, need to validate.

Referenced by gdcm::Bitmap::SetPixelFormat().

10.226.5 Friends And Related Function Documentation

10.226.5.1 Bitmap

```
friend class Bitmap  [friend]
```

10.226.5.2 operator<<

```
std::ostream& operator<< (
            std::ostream & _os,
            const PixelFormat & pf )  [friend]
```

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

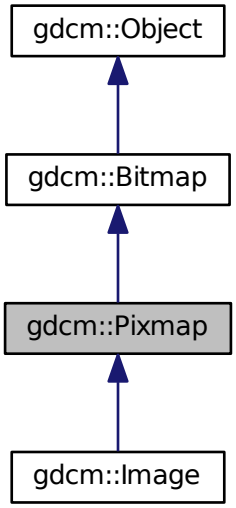
- gdcmPixelFormat.h

10.227 gdcm::Pixmap Class Reference

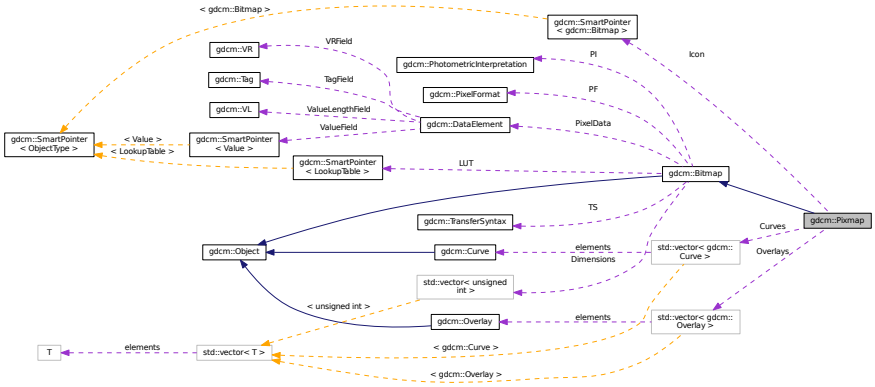
Pixmap class.

```
#include <gdcmPixmap.h>
```

Inheritance diagram for gdcm::Pixmap:



Collaboration diagram for gdcm::Pixmap:



Public Member Functions

- Pixmap ()
- ~Pixmap ()
- bool AreOverlaysInPixelData () const

  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
- const IconImage & GetIconImage () const

  Set/Get Icon Image.
- IconImage & GetIconImage ()
- 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
- void RemoveOverlay (size_t i)
- void SetIconImage (IconImage const &ii)
- void SetNumberOfCurves (size_t n)
- void SetNumberOfOverlays (size_t n)

Protected Attributes

- std::vector< Curve > Curves
- SmartPointer< IconImage > Icon
- std::vector< Overlay > Overlays

Additional Inherited Members

10.227.1 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

10.227.2 Constructor & Destructor Documentation

10.227.2.1 Pixmap()

```
gdcm::Pixmap::Pixmap ( )
```

10.227.2.2 ~Pixmap()

```
gdcm::Pixmap::~Pixmap ( )
```

10.227.3 Member Function Documentation

10.227.3.1 AreOverlaysInPixelData()

```
bool gdcm::Pixmap::AreOverlaysInPixelData ( ) const  [virtual]
```

returns if Overlays are stored in the unused bit of the pixel data:

Reimplemented from gdcm::Bitmap.

10.227.3.2 GetCurve() [1/2]

```
Curve& gdcm::Pixmap::GetCurve (
            size_t i = 0 )  [inline]
```

Curve: group 50xx.

10.227.3.3 GetCurve() [2/2]

```
const Curve& gdcm::Pixmap::GetCurve (
            size_t i = 0 ) const  [inline]
```

10.227.3.4 GetIconImage() [1/2]

```
const IconImage& gdcm::Pixmap::GetIconImage ( ) const  [inline]
```

Set/Get Icon Image.

10.227.3.5 GetIconImage() [2/2]

```
IconImage& gdcm::Pixmap::GetIconImage ( )  [inline]
```

10.227.3.6 GetNumberOfCurves()

```
size_t gdcm::Pixmap::GetNumberOfCurves ( ) const  [inline]
```

10.227.3.7 GetNumberOfOverlays()

```
size_t gdcm::Pixmap::GetNumberOfOverlays ( ) const  [inline]
```

10.227.3.8 GetOverlay() [1/2]

```
Overlay& gdcm::Pixmap::GetOverlay (
            size_t i = 0 )  [inline]
```

Overlay: group 60xx.

10.227.3.9 GetOverlay() [2/2]

```
const Overlay& gdcm::Pixmap::GetOverlay (
            size_t i = 0 ) const  [inline]
```

10.227.3.10 Print()

```
void gdcm::Pixmap::Print (
            std::ostream &  ) const  [virtual]
```

Reimplemented from gdcm::Bitmap.

10.227.3.11 RemoveOverlay()

```
void gdcm::Pixmap::RemoveOverlay (
            size_t i )  [inline]
```

10.227.3.12 SetIconImage()

```
void gdcm::Pixmap::SetIconImage (
            IconImage const & ii )  [inline]
```

10.227.3.13 SetNumberOfCurves()

```
void gdcm::Pixmap::SetNumberOfCurves (
            size_t n )  [inline]
```

10.227.3.14 SetNumberOfOverlays()

```
void gdcm::Pixmap::SetNumberOfOverlays (
            size_t n )  [inline]
```

10.227.4 Member Data Documentation

10.227.4.1 Curves

```
std::vector<Curve> gdcm::Pixmap::Curves  [protected]
```

10.227.4.2 Icon

```
SmartPointer<IconImage> gdcm::Pixmap::Icon  [protected]
```

10.227.4.3 Overlays

```
std::vector<Overlay> gdcm::Pixmap::Overlays  [protected]
```

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

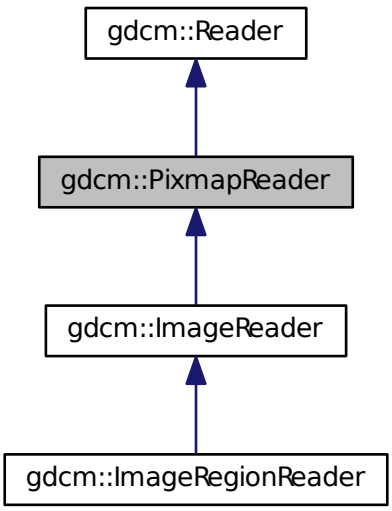
- gdcmPixmap.h

10.228 gdcm::PixmapReader Class Reference

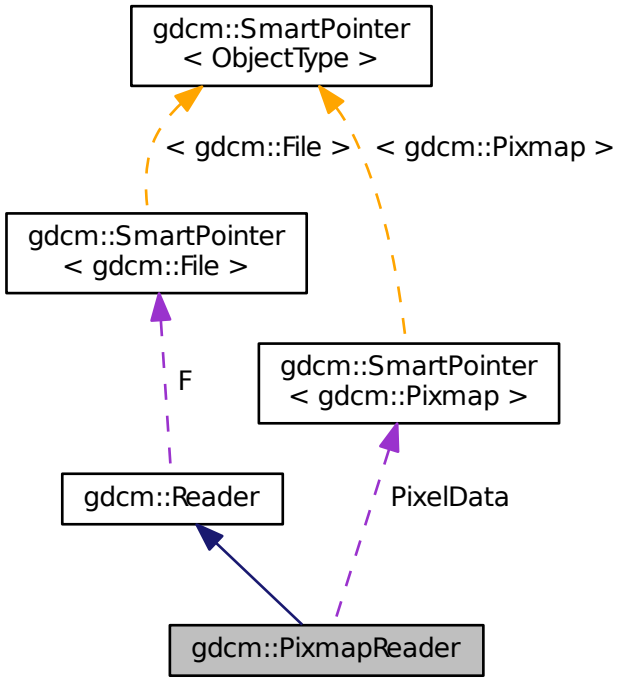
PixmapReader.

```
#include <gdcmPixmapReader.h>
```

Inheritance diagram for gdcm::PixmapReader:



Collaboration diagram for gdcm::PixmapReader:



Public Member Functions

- PixmapReader ()
- virtual ~PixmapReader ()
- const Pixmap & GetPixmap () const

  Return the read image (need to call Read() first)
- Pixmap & GetPixmap ()
- virtual bool Read ()

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

10.228.1 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

10.228.2 Constructor & Destructor Documentation

10.228.2.1 PixmapReader()

```
gdcm::PixmapReader::PixmapReader ( )
```

10.228.2.2 ~PixmapReader()

```
virtual gdcm::PixmapReader::~PixmapReader ( )  [virtual]
```

10.228.3 Member Function Documentation

10.228.3.1 GetPixmap() [1/2]

```
const Pixmap& gdcm::PixmapReader::GetPixmap ( ) const
```

Return the read image (need to call Read() first)

10.228.3.2 GetPixmap() [2/2]

```
Pixmap& gdcm::PixmapReader::GetPixmap ( )
```

10.228.3.3 Read()

```
virtual bool gdcm::PixmapReader::Read ( )  [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.

Reimplemented in gdcm::ImageRegionReader, and gdcm::ImageReader.

10.228.3.4 ReadACRNEMAImage()

```
virtual bool gdcm::PixmapReader::ReadACRNEMAImage ( )  [protected], [virtual]
```

Reimplemented in gdcm::ImageReader.

10.228.3.5 ReadImage()

```
virtual bool gdcm::PixmapReader::ReadImage (
            MediaStorage const & ms )  [protected], [virtual]
```

Reimplemented in gdcm::ImageReader.

10.228.3.6 ReadImageInternal()

```
bool gdcm::PixmapReader::ReadImageInternal (
            MediaStorage const & ms,
            bool handlepixeldata = true )  [protected]
```

10.228.4 Member Data Documentation

10.228.4.1 PixelData

```
SmartPointer<Pixmap> gdcm::PixmapReader::PixelData  [protected]
```

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

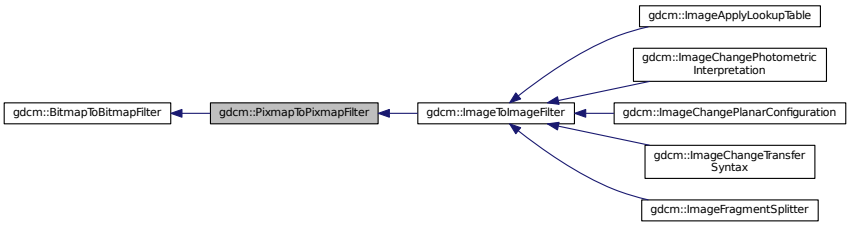
- gdcmPixmapReader.h

10.229 gdcm::PixmapToPixmapFilter Class Reference

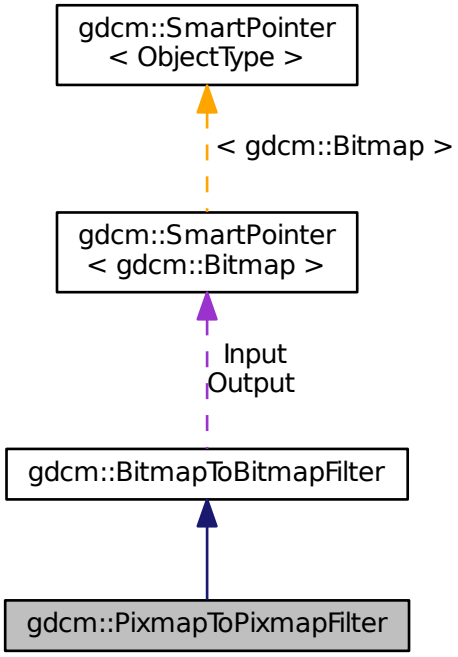
PixmapToPixmapFilter class.

```
#include <gdcmPixmapToPixmapFilter.h>
```

Inheritance diagram for gdcm::PixmapToPixmapFilter:



Collaboration diagram for gdcm::PixmapToPixmapFilter:



Public Member Functions

- PixmapToPixmapFilter ()
- ~PixmapToPixmapFilter ()
- Pixmap & GetInput ()
- const Pixmap & GetOutput () const

  Get Output image.
- const Pixmap & GetOutputAsPixmap () const

Additional Inherited Members

10.229.1 Detailed Description

PixmapToPixmapFilter class.

Super class for all filter taking an image and producing an output image

10.229.2 Constructor & Destructor Documentation

10.229.2.1 PixmapToPixmapFilter()

```
gdcm::PixmapToPixmapFilter::PixmapToPixmapFilter ( )
```

10.229.2.2 ~PixmapToPixmapFilter()

```
gdcm::PixmapToPixmapFilter::~PixmapToPixmapFilter ( )  [inline]
```

10.229.3 Member Function Documentation

10.229.3.1 GetInput()

```
Pixmap& gdcm::PixmapToPixmapFilter::GetInput ( )
```

10.229.3.2 GetOutput()

```
const Pixmap& gdcm::PixmapToPixmapFilter::GetOutput ( ) const
```

Get Output image.

10.229.3.3 GetOutputAsPixmap()

```
const Pixmap& gdcm::PixmapToPixmapFilter::GetOutputAsPixmap ( ) const
```

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

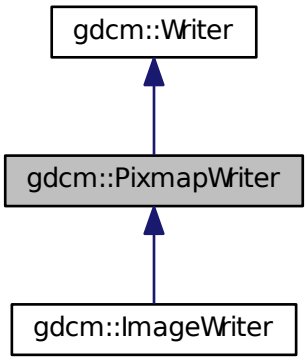
- gdcmPixmapToPixmapFilter.h

10.230 gdcm::PixmapWriter Class Reference

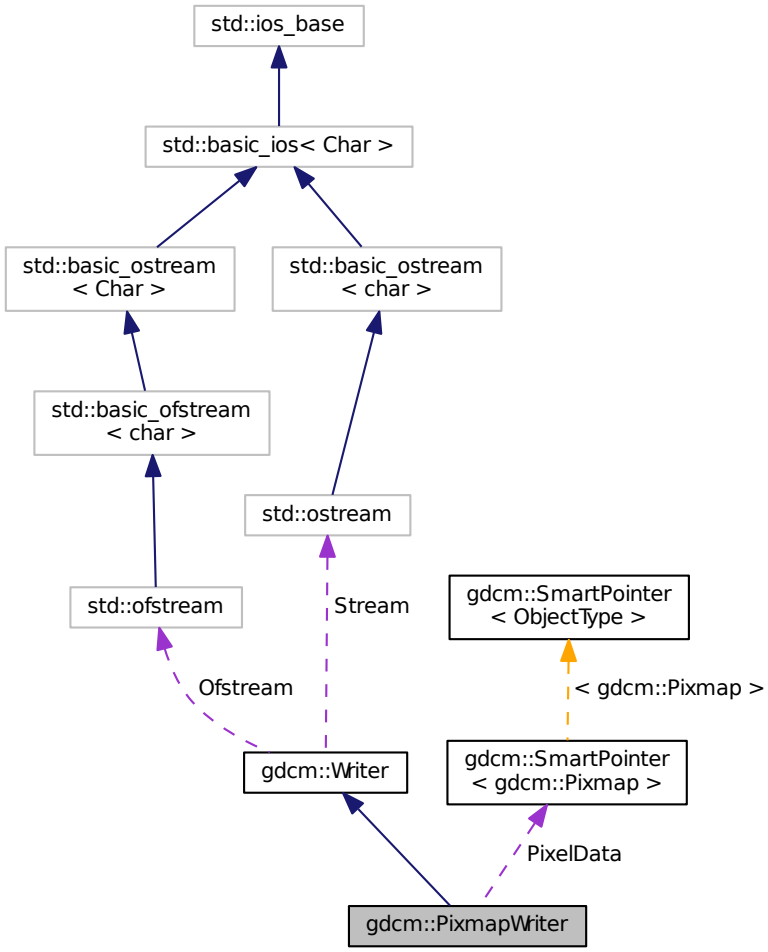
PixmapWriter.

```
#include <gdcmPixmapWriter.h>
```

Inheritance diagram for gdcm::PixmapWriter:



Collaboration diagram for gdcm::PixmapWriter:



Public Member Functions

- PixmapWriter ()
- ~PixmapWriter ()
- virtual const Pixmap & GetImage () const
- virtual Pixmap & GetImage ()
- const Pixmap & GetPixmap () const
- Pixmap & GetPixmap ()
- virtual void SetImage (Pixmap const &img)
- void SetPixmap (Pixmap const &img)
- bool Write ()

  Write.

Protected Member Functions

- void DoIconImage (DataSet &ds, Pixmap const &image)
- bool PrepareWrite ()
- bool PrepareWrite (MediaStorage const &refms)

Protected Attributes

- SmartPointer< Pixmap > PixelData

10.230.1 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 implicitely lossless) then some attribute are definitely needed to mark this dataset as Lossy (typically 0028,2114)

10.230.2 Constructor & Destructor Documentation

10.230.2.1 PixmapWriter()

```
gdcm::PixmapWriter::PixmapWriter ( )
```

10.230.2.2 ~PixmapWriter()

```
gdcm::PixmapWriter::~PixmapWriter ( )
```

10.230.3 Member Function Documentation

10.230.3.1 DoIconImage()

```
void gdcm::PixmapWriter::DoIconImage (
            DataSet & ds,
            Pixmap const & image )  [protected]
```

10.230.3.2 GetImage() [1/2]

```
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.

10.230.3.3 GetImage() [2/2]

```
virtual Pixmap& gdcm::PixmapWriter::GetImage ( )  [inline], [virtual]
```

Reimplemented in gdcm::ImageWriter.

10.230.3.4 GetPixmap() [1/2]

```
const Pixmap& gdcm::PixmapWriter::GetPixmap ( ) const  [inline]
```

10.230.3.5 GetPixmap() [2/2]

```
Pixmap& gdcm::PixmapWriter::GetPixmap ( )  [inline]
```

10.230.3.6 PrepareWrite() [1/2]

```
bool gdcm::PixmapWriter::PrepareWrite ( )  [protected]
```

10.230.3.7 PrepareWrite() [2/2]

```
bool gdcm::PixmapWriter::PrepareWrite (
            MediaStorage const & refms )  [protected]
```

10.230.3.8 SetImage()

```
virtual void gdcm::PixmapWriter::SetImage (
            Pixmap const & img )  [virtual]
```

Examples:

CompressImage.cxx, GenFakeImage.cxx, GetSubSequenceData.cxx, HelloVizWorld.cxx, and MergeTwoFiles.cxx.

10.230.3.9 SetPixmap()

```
void gdcm::PixmapWriter::SetPixmap (
            Pixmap const & img )
```

10.230.3.10 Write()

```
bool gdcm::PixmapWriter::Write ( )  [virtual]
```

Write.

Reimplemented from gdcm::Writer.

10.230.4 Member Data Documentation

10.230.4.1 PixelData

```
SmartPointer<Pixmap> gdcm::PixmapWriter::PixelData  [protected]
```

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

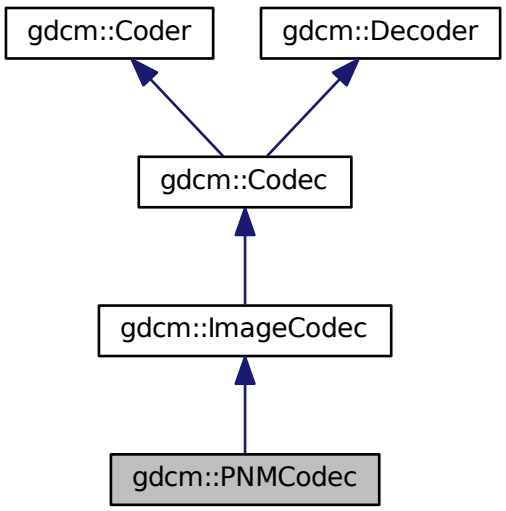
- gdcmPixmapWriter.h

10.231 gdcm::PNMCodec Class Reference

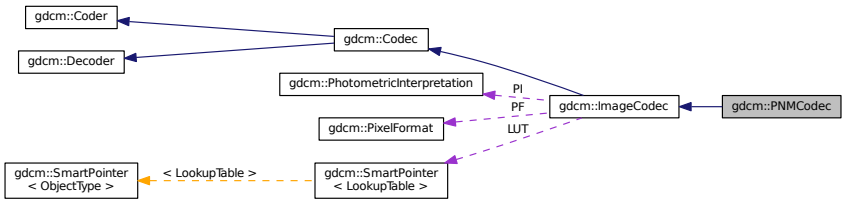
Class to do PNM.

```
#include <gdcmPNMCodec.h>
```

Inheritance diagram for gdcm::PNMCodec:



Collaboration diagram for gdcm::PNMCodec:



Public Member Functions

- PNMCodec ()
- ~PNMCodec ()
- bool CanCode (TransferSyntax const &ts) const

  Return whether this coder support this transfer syntax (can code it)
- bool CanDecode (TransferSyntax const &ts) const

  Return whether this decoder support this transfer syntax (can decode it)
- virtual ImageCodec * Clone () const
- unsigned long GetBufferLength () const
- bool GetHeaderInfo (std::istream &is, TransferSyntax &ts)
- 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

10.231.1 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.

10.231.2 Constructor & Destructor Documentation

10.231.2.1 PNMCodec()

```
gdcm::PNMCodec::PNMCodec ( )
```

10.231.2.2 ~PNMCodec()

```
gdcm::PNMCodec::~PNMCodec ( )
```

10.231.3 Member Function Documentation

10.231.3.1 CanCode()

```
bool gdcm::PNMCodec::CanCode (
            TransferSyntax const &  ) const  [virtual]
```

Return whether this coder support this transfer syntax (can code it)

Reimplemented from gdcm::ImageCodec.

10.231.3.2 CanDecode()

```
bool gdcm::PNMCodec::CanDecode (
            TransferSyntax const &  ) const  [virtual]
```

Return whether this decoder support this transfer syntax (can decode it)

Reimplemented from gdcm::ImageCodec.

10.231.3.3 Clone()

```
virtual ImageCodec* gdcm::PNMCodec::Clone ( ) const  [virtual]
```

Implements gdcm::ImageCodec.

10.231.3.4 GetBufferLength()

```
unsigned long gdcm::PNMCodec::GetBufferLength ( ) const  [inline]
```

10.231.3.5 GetHeaderInfo()

```
bool gdcm::PNMCodec::GetHeaderInfo (
            std::istream & is,
            TransferSyntax & ts )  [virtual]
```

Reimplemented from gdcm::ImageCodec.

10.231.3.6 Read()

```
bool gdcm::PNMCodec::Read (
            const char * filename,
            DataElement & out ) const
```

10.231.3.7 SetBufferLength()

```
void gdcm::PNMCodec::SetBufferLength (
            unsigned long l )  [inline]
```

10.231.3.8 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

10.232 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)

10.232.1 Detailed Description

DICOM Preamble (Part 10)

10.232.2 Constructor & Destructor Documentation

10.232.2.1 Preamble() [1/2]

```
gdcm::Preamble::Preamble ( )
```

10.232.2.2 ~Preamble()

```
gdcm::Preamble::~Preamble ( )
```

10.232.2.3 Preamble() [2/2]

```
gdcm::Preamble::Preamble (
            Preamble const &  )  [inline]
```

10.232.3 Member Function Documentation

10.232.3.1 Clear()

```
void gdcm::Preamble::Clear ( )
```

Clear.

10.232.3.2 Create()

```
void gdcm::Preamble::Create ( )
```

10.232.3.3 GetInternal()

```
const char* gdcm::Preamble::GetInternal ( ) const  [inline]
```

Get internal pointer to preamble.

10.232.3.4 GetLength()

```
VL gdcm::Preamble::GetLength ( ) const  [inline]
```

Return size of Preamble.

10.232.3.5 IsEmpty()

```
bool gdcm::Preamble::IsEmpty ( ) const  [inline]
```

Check if Preamble is empty.

10.232.3.6 IsValid()

```
bool gdcm::Preamble::IsValid ( ) const  [inline], [protected]
```

10.232.3.7 operator=()

```
Preamble& gdcm::Preamble::operator= (
            Preamble const &  )  [inline]
```

10.232.3.8 Print()

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

Print Preamble.

10.232.3.9 Read()

```
std::istream& gdcm::Preamble::Read (
            std::istream & is )
```

Read Preamble.

10.232.3.10 Remove()

```
void gdcm::Preamble::Remove ( )
```

10.232.3.11 Valid()

```
void gdcm::Preamble::Valid ( )
```

Set Preamble to the default one.

10.232.3.12 Write()

```
std::ostream const& gdcm::Preamble::Write (
            std::ostream & os ) const
```

Write Preamble.

10.232.4 Friends And Related Function Documentation

10.232.4.1 operator<<

```
std::ostream& operator<< (
            std::ostream & _os,
            const Preamble & _val )  [friend]
```

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

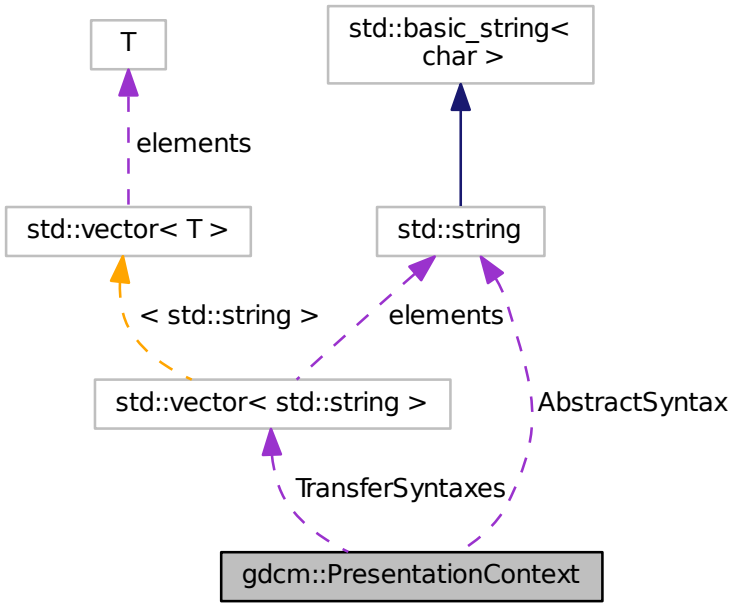
- gdcmPreamble.h

10.233 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::ImplicitVRLittleEndianDefault↩ TransferSyntaxforDICOM)
- 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

10.233.1 Detailed Description

PresentationContext.

See also

PresentationContextAC PresentationContextRQ

10.233.2 Member Typedef Documentation

10.233.2.1 SizeType

```
typedef TransferSyntaxArrayType::size_type gdcm::PresentationContext::SizeType
```

10.233.2.2 TransferSyntaxArrayType

```
typedef std::vector<std::string> gdcm::PresentationContext::TransferSyntaxArrayType
```

10.233.3 Constructor & Destructor Documentation

10.233.3.1 PresentationContext() [1/2]

```
gdcm::PresentationContext::PresentationContext ( )
```

10.233.3.2 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).

10.233.4 Member Function Documentation

10.233.4.1 AddTransferSyntax()

```
void gdcm::PresentationContext::AddTransferSyntax (
            const char * tsstr )
```

10.233.4.2 GetAbstractSyntax()

```
const char* gdcm::PresentationContext::GetAbstractSyntax ( ) const  [inline]
```

10.233.4.3 GetNumberOfTransferSyntaxes()

```
SizeType gdcm::PresentationContext::GetNumberOfTransferSyntaxes ( ) const  [inline]
```

10.233.4.4 GetPresentationContextID()

```
uint8_t gdcm::PresentationContext::GetPresentationContextID ( ) const
```

10.233.4.5 GetTransferSyntax()

```
const char* gdcm::PresentationContext::GetTransferSyntax (
            SizeType i ) const  [inline]
```

10.233.4.6 operator==()

```
bool gdcm::PresentationContext::operator== (
            const PresentationContext & pc ) const  [inline]
```

References AbstractSyntax, and TransferSyntaxes.

10.233.4.7 Print()

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

10.233.4.8 SetAbstractSyntax()

```
void gdcm::PresentationContext::SetAbstractSyntax (
            const char * absyn )  [inline]
```

10.233.4.9 SetPresentationContextID()

```
void gdcm::PresentationContext::SetPresentationContextID (
            uint8_t id )
```

10.233.5 Member Data Documentation

10.233.5.1 AbstractSyntax

```
std::string gdcm::PresentationContext::AbstractSyntax  [protected]
```

Referenced by operator==().

10.233.5.2 ID

```
uint8_t gdcm::PresentationContext::ID  [protected]
```

10.233.5.3 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

10.234 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

10.234.1 Detailed Description

PresentationContextAC.

Table 9-18 PRESENTATION CONTEXT ITEM FIELDS

See also

PresentationContext

10.234.2 Constructor & Destructor Documentation

10.234.2.1 PresentationContextAC()

```
gdcm::network::PresentationContextAC::PresentationContextAC ( )
```

10.234.3 Member Function Documentation

10.234.3.1 GetPresentationContextID()

```
uint8_t gdcm::network::PresentationContextAC::GetPresentationContextID ( ) const  [inline]
```

10.234.3.2 GetReason()

```
uint8_t gdcm::network::PresentationContextAC::GetReason ( ) const  [inline]
```

10.234.3.3 GetTransferSyntax()

```
TransferSyntaxSub const& gdcm::network::PresentationContextAC::GetTransferSyntax ( ) const  [inline]
```

10.234.3.4 Print()

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

10.234.3.5 Read()

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

10.234.3.6 SetPresentationContextID()

```
void gdcm::network::PresentationContextAC::SetPresentationContextID (
            uint8_t id )
```

10.234.3.7 SetReason()

```
void gdcm::network::PresentationContextAC::SetReason (
            uint8_t r )  [inline]
```

10.234.3.8 SetTransferSyntax()

```
void gdcm::network::PresentationContextAC::SetTransferSyntax (
            TransferSyntaxSub const & ts )
```

10.234.3.9 Size()

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

10.234.3.10 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

10.235 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

10.235.1 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 GenerateFrom↩ UID() 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.

10.235.2 Member Typedef Documentation

10.235.2.1 PresentationContextArrayType

```
typedef std::vector<PresentationContext> gdcm::PresentationContextGenerator::PresentationContext↩
ArrayType
```

10.235.2.2 SizeType

```
typedef PresentationContextArrayType::size_type gdcm::PresentationContextGenerator::SizeType
```

10.235.3 Constructor & Destructor Documentation

10.235.3.1 PresentationContextGenerator()

```
gdcm::PresentationContextGenerator::PresentationContextGenerator ( )
```

10.235.4 Member Function Documentation

10.235.4.1 AddFromFile()

```
bool gdcm::PresentationContextGenerator::AddFromFile (
            const File & file )
```

Add a single PresentationContext from a single File. Call multiple times when dealing with multiple files.

10.235.4.2 AddPresentationContext()

```
bool gdcm::PresentationContextGenerator::AddPresentationContext (
            const char * absyn,
            const char * ts )  [protected]
```

10.235.4.3 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-ST↩ ORE operations

Examples:

CStoreQtProgress.cxx.

10.235.4.4 GenerateFromUID()

```
bool gdcm::PresentationContextGenerator::GenerateFromUID (
            UIDs::TSName asname )
```

Generate the PresentationContext array from a UID (eg. VerificationSOPClass)

10.235.4.5 GetDefaultTransferSyntax()

```
const char* gdcm::PresentationContextGenerator::GetDefaultTransferSyntax ( ) const  [protected]
```

10.235.4.6 GetPresentationContexts()

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

Examples:

CStoreQtProgress.cxx.

10.235.4.7 SetDefaultTransferSyntax()

```
void gdcm::PresentationContextGenerator::SetDefaultTransferSyntax (
            const TransferSyntax & ts )
```

Not implemented for now. GDCM internally uses Implicit Little Endian.

10.235.4.8 SetMergeModeToAbstractSyntax()

```
void gdcm::PresentationContextGenerator::SetMergeModeToAbstractSyntax ( )
```

10.235.4.9 SetMergeModeToTransferSyntax()

```
void gdcm::PresentationContextGenerator::SetMergeModeToTransferSyntax ( )
```

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

- gdcmPresentationContextGenerator.h

10.236 gdcm::network::PresentationContextRQ Class Reference

PresentationContextRQ.

`#include <gdcmPresentationContextRQ.h>`

Public Types

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

Public Member Functions

- PresentationContextRQ ()
- PresentationContextRQ (UIDs::TSName asname, UIDs::TSName tsname=UIDs::ImplicitVRLittleEndianDefault↩ TransferSyntaxforDICOM)
- PresentationContextRQ (const PresentationContext &pc)
- void AddTransferSyntax (TransferSyntaxSub const &ts)
- AbstractSyntax const & GetAbstractSyntax () const
- AbstractSyntax & GetAbstractSyntax ()
- SizeType GetNumberOfTransferSyntaxes () const
- uint8_t GetPresentationContextID () const
- TransferSyntaxSub const & GetTransferSyntax (SizeType i) const
- TransferSyntaxSub & GetTransferSyntax (SizeType i)
- 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

10.236.1 Detailed Description

PresentationContextRQ.

Table 9-13 PRESENTATION CONTEXT ITEM FIELDS

See also

PresentationContextAC

10.236.2 Member Typedef Documentation

10.236.2.1 SizeType

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

10.236.3 Constructor & Destructor Documentation

10.236.3.1 PresentationContextRQ() [1/3]

```
gdcm::network::PresentationContextRQ::PresentationContextRQ ( )
```

10.236.3.2 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).

10.236.3.3 PresentationContextRQ() [3/3]

```
gdcm::network::PresentationContextRQ::PresentationContextRQ (
            const PresentationContext & pc )
```

10.236.4 Member Function Documentation

10.236.4.1 AddTransferSyntax()

```
void gdcm::network::PresentationContextRQ::AddTransferSyntax (
            TransferSyntaxSub const & ts )
```

10.236.4.2 GetAbstractSyntax() [1/2]

```
AbstractSyntax const& gdcm::network::PresentationContextRQ::GetAbstractSyntax ( ) const  [inline]
```

10.236.4.3 GetAbstractSyntax() [2/2]

```
AbstractSyntax& gdcm::network::PresentationContextRQ::GetAbstractSyntax ( )  [inline]
```

10.236.4.4 GetNumberOfTransferSyntaxes()

```
SizeType gdcm::network::PresentationContextRQ::GetNumberOfTransferSyntaxes ( ) const  [inline]
```

10.236.4.5 GetPresentationContextID()

```
uint8_t gdcm::network::PresentationContextRQ::GetPresentationContextID ( ) const
```

10.236.4.6 GetTransferSyntax() [1/2]

```
TransferSyntaxSub const& gdcm::network::PresentationContextRQ::GetTransferSyntax (
            SizeType i ) const  [inline]
```

10.236.4.7 GetTransferSyntax() [2/2]

```
TransferSyntaxSub& gdcm::network::PresentationContextRQ::GetTransferSyntax (
            SizeType i )  [inline]
```

10.236.4.8 GetTransferSyntaxes()

```
std::vector<TransferSyntaxSub> const& gdcm::network::PresentationContextRQ::GetTransferSyntaxes (
) const  [inline]
```

10.236.4.9 operator==()

```
bool gdcm::network::PresentationContextRQ::operator== (
            const PresentationContextRQ & pc ) const  [inline]
```

10.236.4.10 Print()

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

10.236.4.11 Read()

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

10.236.4.12 SetAbstractSyntax()

```
void gdcm::network::PresentationContextRQ::SetAbstractSyntax (
             AbstractSyntax const & absyn )
```

10.236.4.13 SetPresentationContextID()

```
void gdcm::network::PresentationContextRQ::SetPresentationContextID (
             uint8_t id )
```

10.236.4.14 Size()

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

10.236.4.15 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

10.237 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)

10.237.1 Detailed Description

PresentationDataValue.

Table 9-23 PRESENTATION-DATA-VALUE ITEM FIELDS

10.237.2 Constructor & Destructor Documentation

10.237.2.1 PresentationDataValue()

```
gdcm::network::PresentationDataValue::PresentationDataValue ( )
```

10.237.3 Member Function Documentation

10.237.3.1 ConcatenatePDVBlobs()

```
static DataSet gdcm::network::PresentationDataValue::ConcatenatePDVBlobs (
            const std::vector< PresentationDataValue > & inPDVs )  [static]
```

Warning

DataSet will be read as Implicit Little Endian TS

10.237.3.2 ConcatenatePDVBlobsAsExplicit()

```
static DataSet gdcm::network::PresentationDataValue::ConcatenatePDVBlobsAsExplicit (
            const std::vector< PresentationDataValue > & inPDVs )  [static]
```

10.237.3.3 GetBlob()

```
const std::string& gdcm::network::PresentationDataValue::GetBlob ( ) const
```

10.237.3.4 GetIsCommand()

```
bool gdcm::network::PresentationDataValue::GetIsCommand ( ) const
```

10.237.3.5 GetIsLastFragment()

```
bool gdcm::network::PresentationDataValue::GetIsLastFragment ( ) const
```

10.237.3.6 GetMessageHeader()

```
uint8_t gdcm::network::PresentationDataValue::GetMessageHeader ( ) const  [inline]
```

10.237.3.7 GetPresentationContextID()

```
uint8_t gdcm::network::PresentationDataValue::GetPresentationContextID ( ) const  [inline]
```

10.237.3.8 Print()

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

10.237.3.9 Read()

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

10.237.3.10 ReadInto()

```
std::istream& gdcm::network::PresentationDataValue::ReadInto (
            std::istream & is,
            std::ostream & os )
```

10.237.3.11 SetBlob()

```
void gdcm::network::PresentationDataValue::SetBlob (
            const std::string & partialblob )
```

10.237.3.12 SetCommand()

```
void gdcm::network::PresentationDataValue::SetCommand (
            bool inCommand )
```

10.237.3.13 SetDataSet()

```
void gdcm::network::PresentationDataValue::SetDataSet (
            const DataSet & ds )
```

Set DataSet. Write DataSet in implicit.

Warning

size of dataset should be below maxpdusize

10.237.3.14 SetLastFragment()

```
void gdcm::network::PresentationDataValue::SetLastFragment (
            bool inLast )
```

10.237.3.15 SetMessageHeader()

```
void gdcm::network::PresentationDataValue::SetMessageHeader (
            uint8_t messageheader )  [inline]
```

10.237.3.16 SetPresentationContextID()

```
void gdcm::network::PresentationDataValue::SetPresentationContextID (
            uint8_t id )  [inline]
```

10.237.3.17 Size()

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

10.237.3.18 Write()

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

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

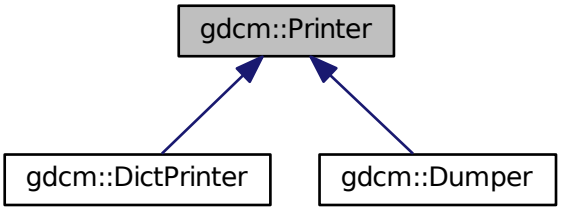
- gdcmPresentationDataValue.h

10.238 gdcm::Printer Class Reference

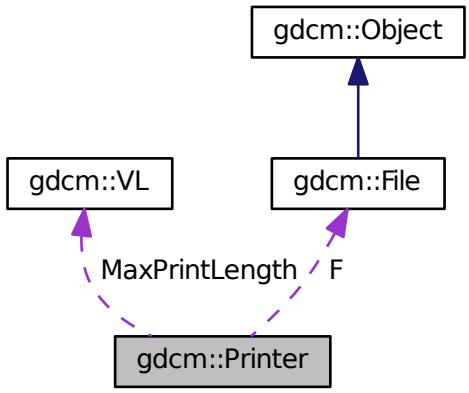
Printer class.

`#include <gdcmPrinter.h>`

Inheritance diagram for gdcm::Printer:



Collaboration diagram for gdcm::Printer:



Public Types

- enum PrintStyles {
  VERBOSE_STYLE = 0,
  CONDENSED_STYLE,
  XML }

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

10.238.1 Detailed Description

Printer class.

Examples:

DumpToshibaDTI.cxx.

10.238.2 Member Enumeration Documentation

10.238.2.1 PrintStyles

```
enum gdcm::Printer::PrintStyles
```

Enumerator

VERBOSE_STYLE	
CONDENSED_STYLE	
XML	

10.238.3 Constructor & Destructor Documentation

10.238.3.1 Printer()

```
gdcm::Printer::Printer ( )
```

10.238.3.2 ~Printer()

```
gdcm::Printer::~Printer ( )
```

10.238.4 Member Function Documentation

10.238.4.1 GetPrintStyle()

```
PrintStyles gdcm::Printer::GetPrintStyle ( ) const  [inline]
```

Get PrintStyle value.

10.238.4.2 Print()

```
void gdcm::Printer::Print (
            std::ostream & os )
```

Print.

Examples:

DumpToshibaDTI.cxx.

10.238.4.3 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]
```

10.238.4.4 PrintDataSet()

```
void gdcm::Printer::PrintDataSet (
            const DataSet & ds,
            std::ostream & os,
            const std::string & s = "" )
```

Print an individual dataset.

10.238.4.5 PrintSQ()

```
void gdcm::Printer::PrintSQ (
            const SequenceOfItems * sqi,
            std::ostream & os,
            std::string const & indent )  [protected]
```

10.238.4.6 SetColor()

```
void gdcm::Printer::SetColor (
            bool c )
```

Set color mode or not.

Examples:

DumpToshibaDTI.cxx.

10.238.4.7 SetFile()

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

Set file.

Examples:

DumpToshibaDTI.cxx.

10.238.4.8 SetStyle()

```
void gdcm::Printer::SetStyle (
            PrintStyles ps )  [inline]
```

Set PrintStyle value.

10.238.5 Member Data Documentation

10.238.5.1 F

```
const File* gdcm::Printer::F  [protected]
```

10.238.5.2 MaxPrintLength

```
VL gdcm::Printer::MaxPrintLength  [protected]
```

10.238.5.3 PrintStyle

```
PrintStyles gdcm::Printer::PrintStyle  [protected]
```

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

- gdcmPrinter.h

10.239 gdcm::PrivateDict Class Reference

Private Dict.

```
#include <gdcmDict.h>
```

Public Member Functions

- PrivateDict ()
- ~PrivateDict ()
- 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)

10.239.1 Detailed Description

Private Dict.

10.239.2 Constructor & Destructor Documentation

10.239.2.1 PrivateDict()

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

10.239.2.2 ~PrivateDict()

```
gdcm::PrivateDict::~PrivateDict ( )  [inline]
```

10.239.3 Member Function Documentation

10.239.3.1 AddDictEntry()

```
void gdcm::PrivateDict::AddDictEntry (
            const PrivateTag & tag,
            const DictEntry & de )  [inline]
```

References gdcm::DictEntry::GetVM(), gdcm::DictEntry::GetVR(), gdcm::DictEntry::SetVR(), and gdcm::VR::UN.

10.239.3.2 FindDictEntry()

```
bool gdcm::PrivateDict::FindDictEntry (
            const PrivateTag & tag ) const  [inline]
```

10.239.3.3 GetDictEntry()

```
const DictEntry& gdcm::PrivateDict::GetDictEntry (
            const PrivateTag & tag ) const  [inline]
```

10.239.3.4 IsEmpty()

```
bool gdcm::PrivateDict::IsEmpty ( ) const  [inline]
```

10.239.3.5 LoadDefault()

```
void gdcm::PrivateDict::LoadDefault ( )  [protected]
```

10.239.3.6 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().

10.239.3.7 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.

10.239.4 Friends And Related Function Documentation

10.239.4.1 Dicts

```
friend class Dicts  [friend]
```

10.239.4.2 operator<<

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

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

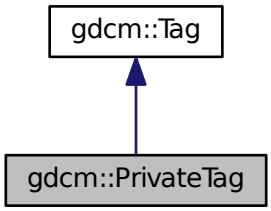
- gdcmDict.h

10.240 gdcm::PrivateTag Class Reference

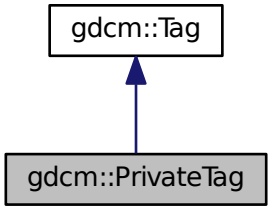
Class to represent a Private DICOM Data Element (Attribute) Tag (Group, Element, Owner)

```
#include <gdcmPrivateTag.h>
```

Inheritance diagram for gdcm::PrivateTag:



Collaboration diagram for gdcm::PrivateTag:



Public Member Functions

- PrivateTag (uint16_t group=0, uint16_t element=0, const char ∗owner="")
- PrivateTag (Tag const &t, const char ∗owner="")
- DataElement GetAsDataElement () const
- const char ∗ GetOwner () const
- bool operator< (const PrivateTag &_val) const
- bool ReadFromCommaSeparatedString (const char ∗str)
- void SetOwner (const char ∗owner)

Friends

- std::ostream & operator<< (std::ostream &_os, const PrivateTag &_val)

10.240.1 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, csa2img.cxx, DumpADAC.cxx, DumpExamCard.cxx, DumpGEMSMovieGroup.cxx, DumpImageHeaderInfo.cxx, DumpPhilipsECHO.cxx, DumpToshibaDTI.cxx, ELSCINT1WaveToText.cxx, GetSubSequenceData.cxx, iU22tomultisc.cxx, MrProtocol.cxx, pmsct_rgb1.cxx, PublicDict.cxx, ReadGEMSSDO.cxx, and rle2img.cxx.

10.240.2 Constructor & Destructor Documentation

10.240.2.1 PrivateTag() [1/2]

```
gdcm::PrivateTag::PrivateTag (
            uint16_t group = 0,
            uint16_t element = 0,
            const char * owner = "" )  [inline]
```

10.240.2.2 PrivateTag() [2/2]

```
gdcm::PrivateTag::PrivateTag (
            Tag const & t,
            const char * owner = "" )  [inline]
```

References gdcm::Tag::GetElement().

10.240.3 Member Function Documentation

10.240.3.1 GetAsDataElement()

```
DataElement gdcm::PrivateTag::GetAsDataElement ( ) const
```

10.240.3.2 GetOwner()

```
const char* gdcm::PrivateTag::GetOwner ( ) const  [inline]
```

Examples:

PublicDict.cxx.

Referenced by gdcm::PrivateDict::PrintXML().

10.240.3.3 operator<()

```
bool gdcm::PrivateTag::operator< (
            const PrivateTag & _val ) const
```

10.240.3.4 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")

10.240.3.5 SetOwner()

```
void gdcm::PrivateTag::SetOwner (
            const char * owner )  [inline]
```

10.240.4 Friends And Related Function Documentation

10.240.4.1 operator<<

```
std::ostream& operator<< (
            std::ostream & _os,
            const PrivateTag & _val )  [friend]
```

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

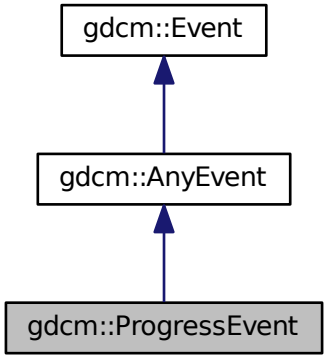
- gdcmPrivateTag.h

10.241 gdcm::ProgressEvent Class Reference

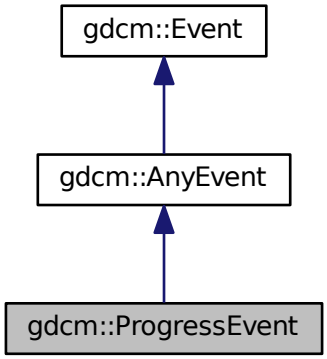
ProgressEvent.

```
#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 (double p=0)
- ProgressEvent (const Self &s)
- virtual ~ProgressEvent ()
- virtual bool CheckEvent (const ::gdcm::Event *e) const
- virtual const char * GetEventName () const
- double GetProgress () const
- virtual ::gdcm::Event * MakeObject () const
- void SetProgress (double p)

10.241.1 Detailed Description

ProgressEvent.

Special type of event triggered during

See also

AnyEvent

10.241.2 Member Typedef Documentation

10.241.2.1 Self

```
typedef ProgressEvent gdcm::ProgressEvent::Self
```

10.241.2.2 Superclass

```
typedef AnyEvent gdcm::ProgressEvent::Superclass
```

10.241.3 Constructor & Destructor Documentation

10.241.3.1 ProgressEvent() [1/2]

```
gdcm::ProgressEvent::ProgressEvent (
            double p = 0 )  [inline]
```

10.241.3.2 ~ProgressEvent()

```
virtual gdcm::ProgressEvent::~ProgressEvent ( )  [inline], [virtual]
```

10.241.3.3 ProgressEvent() [2/2]

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

10.241.4 Member Function Documentation

10.241.4.1 CheckEvent()

```
virtual bool gdcm::ProgressEvent::CheckEvent (
            const ::gdcm::Event * e ) const  [inline], [virtual]
```

10.241.4.2 GetEventName()

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

Return the StringName associated with the event.

Implements gdcm::Event.

10.241.4.3 GetProgress()

```
double gdcm::ProgressEvent::GetProgress ( ) const  [inline]
```

10.241.4.4 MakeObject()

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

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

Implements gdcm::Event.

10.241.4.5 SetProgress()

```
void gdcm::ProgressEvent::SetProgress (
            double p )  [inline]
```

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

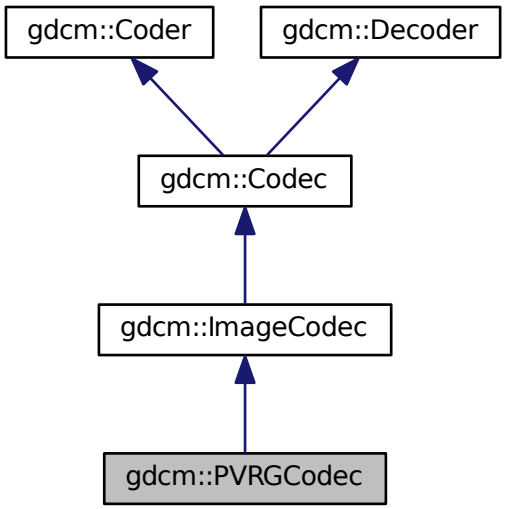
- gdcmProgressEvent.h

10.242 gdcm::PVRGCodec Class Reference

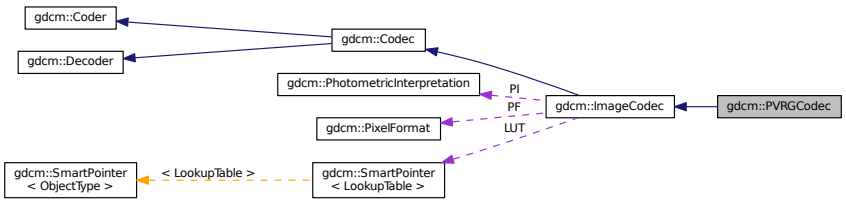
PVRGCodec.

`#include <gdcmPVRGCodec.h>`

Inheritance diagram for gdcm::PVRGCodec:



Collaboration diagram for gdcm::PVRGCodec:



Public Member Functions

- PVRGCodec ()
- ~PVRGCodec ()
- bool CanCode (TransferSyntax const &ts) const

Return whether this coder support this transfer syntax (can code it)

- bool CanDecode (TransferSyntax const &ts) const

  Return whether this decoder support this transfer syntax (can decode it)
- virtual ImageCodec * Clone () const
- bool Code (DataElement const &in, DataElement &out)

  Code.
- bool Decode (DataElement const &is, DataElement &os)

  Decode.
- void SetLossyFlag (bool l)

Additional Inherited Members

10.242.1 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...

10.242.2 Constructor & Destructor Documentation

10.242.2.1 PVRGCodec()

```
gdcm::PVRGCodec::PVRGCodec ( )
```

10.242.2.2 ~PVRGCodec()

```
gdcm::PVRGCodec::~PVRGCodec ( )
```

10.242.3 Member Function Documentation

10.242.3.1 CanCode()

```
bool gdcm::PVRGCodec::CanCode (
            TransferSyntax const &  ) const  [virtual]
```

Return whether this coder support this transfer syntax (can code it)

Reimplemented from gdcm::ImageCodec.

10.242.3.2 CanDecode()

```
bool gdcm::PVRGCodec::CanDecode (
            TransferSyntax const &  ) const  [virtual]
```

Return whether this decoder support this transfer syntax (can decode it)

Reimplemented from gdcm::ImageCodec.

10.242.3.3 Clone()

```
virtual ImageCodec* gdcm::PVRGCodec::Clone ( ) const  [virtual]
```

Implements gdcm::ImageCodec.

10.242.3.4 Code()

```
bool gdcm::PVRGCodec::Code (
            DataElement const & in_,
            DataElement & out_ )  [virtual]
```

Code.

Reimplemented from gdcm::Coder.

10.242.3.5 Decode()

```
bool gdcm::PVRGCodec::Decode (
            DataElement const & ,
            DataElement &  )  [virtual]
```

Decode.

Reimplemented from gdcm::ImageCodec.

10.242.3.6 SetLossyFlag()

```
void gdcm::PVRGCodec::SetLossyFlag (
            bool l )
```

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

- gdcmPVRGCodec.h

10.243 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)

10.243.1 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.

10.243.2 Constructor & Destructor Documentation

10.243.2.1 PythonFilter()

```
gdcm::PythonFilter::PythonFilter ( )
```

10.243.2.2 ∼PythonFilter()

```
gdcm::PythonFilter::∼PythonFilter ( )
```

10.243.3 Member Function Documentation

10.243.3.1 GetFile() [1/2]

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

10.243.3.2 GetFile() [2/2]

```
const File& gdcm::PythonFilter::GetFile ( ) const  [inline]
```

10.243.3.3 SetDicts()

```
void gdcm::PythonFilter::SetDicts (
            const Dicts & dicts )
```

10.243.3.4 SetFile()

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

10.243.3.5 ToPyObject()

```
PyObject* gdcm::PythonFilter::ToPyObject (
            const Tag & t ) const
```

10.243.3.6 UseDictAlways()

```
void gdcm::PythonFilter::UseDictAlways (
            bool  )  [inline]
```

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

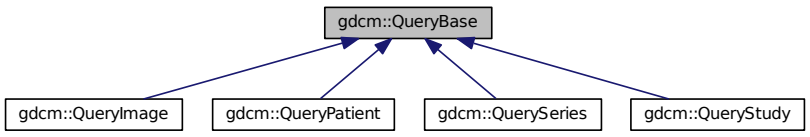
- gdcmPythonFilter.h

10.244 gdcm::QueryBase Class Reference

QueryBase.

```
#include <gdcmQueryBase.h>
```

Inheritance diagram for gdcm::QueryBase:



Public Member Functions

- virtual ~QueryBase ()
- 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

10.244.1 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.

10.244.2 Constructor & Destructor Documentation

10.244.2.1 ~QueryBase()

```
virtual gdcm::QueryBase::~QueryBase ( )  [inline], [virtual]
```

10.244.3 Member Function Documentation

10.244.3.1 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

10.244.3.2 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

10.244.3.3 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.

10.244.3.4 GetName()

```
virtual const char* gdcm::QueryBase::GetName ( ) const  [pure virtual]
```

Implemented in gdcm::QueryImage, gdcm::QueryPatient, gdcm::QuerySeries, and gdcm::QueryStudy.

10.244.3.5 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.

10.244.3.6 GetQueryLevel()

```
virtual DataElement gdcm::QueryBase::GetQueryLevel ( ) const  [pure virtual]
```

Implemented in gdcm::QueryImage, gdcm::QueryPatient, gdcm::QuerySeries, and gdcm::QueryStudy.

10.244.3.7 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.

10.244.3.8 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

10.245 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)

10.245.1 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

10.245.2 Member Function Documentation

10.245.2.1 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()).

10.245.2.2 ListCharSets()

```
static void gdcm::QueryFactory::ListCharSets (
            std::ostream & os )  [static]
```

List all possible CharSet.

10.245.2.3 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

10.245.2.4 ProduceQuery() [1/2]

```
static BaseQuery* gdcm::QueryFactory::ProduceQuery (
            const std::string & sopInstanceUID,
            ENQueryType inQueryType )  [static]
```

10.245.2.5 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:

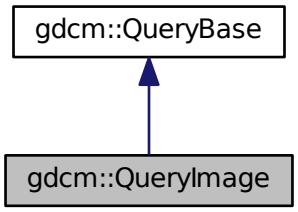
- gdcmQueryFactory.h

10.246 gdcm::QueryImage Class Reference

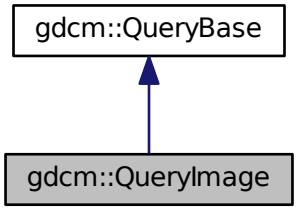
QueryImage.

`#include <gdcmQueryImage.h>`

Inheritance diagram for gdcm::QueryImage:



Collaboration diagram for gdcm::QueryImage:



Public Member Functions

- std::vector< Tag > GetHierachicalSearchTags (const ERootType &inRootType) const
  Return all Unique Key for a particular Query Root type (from the same level and above).
- const char * GetName () const
- std::vector< Tag > GetOptionalTags (const ERootType &inRootType) const
- DataElement GetQueryLevel () const
- std::vector< Tag > GetRequiredTags (const ERootType &inRootType) const
- std::vector< Tag > GetUniqueTags (const ERootType &inRootType) const

10.246.1 Detailed Description

QueryImage.

contains: class to construct an image-based query for C-FIND and C-MOVE

10.246.2 Member Function Documentation

10.246.2.1 GetHierachicalSearchTags()

```
std::vector<Tag> gdcm::QueryImage::GetHierachicalSearchTags (
            const ERootType & inRootType ) const  [virtual]
```

Return all Unique Key for a particular Query Root type (from the same level and above).

Implements gdcm::QueryBase.

10.246.2.2 GetName()

```
const char* gdcm::QueryImage::GetName ( ) const  [virtual]
```

Implements gdcm::QueryBase.

10.246.2.3 GetOptionalTags()

```
std::vector<Tag> gdcm::QueryImage::GetOptionalTags (
            const ERootType & inRootType ) const  [virtual]
```

Implements gdcm::QueryBase.

10.246.2.4 GetQueryLevel()

```
DataElement gdcm::QueryImage::GetQueryLevel ( ) const  [virtual]
```

Implements gdcm::QueryBase.

10.246.2.5 GetRequiredTags()

```
std::vector<Tag> gdcm::QueryImage::GetRequiredTags (
            const ERootType & inRootType ) const  [virtual]
```

Implements gdcm::QueryBase.

10.246.2.6 GetUniqueTags()

```
std::vector<Tag> gdcm::QueryImage::GetUniqueTags (
            const ERootType & inRootType ) const  [virtual]
```

Implements gdcm::QueryBase.

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

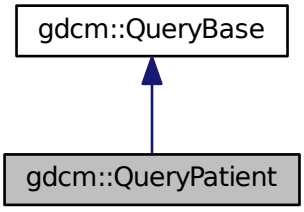
- gdcmQueryImage.h

10.247 gdcm::QueryPatient Class Reference

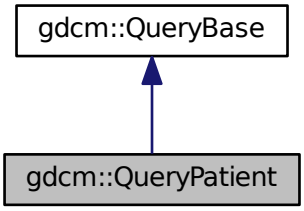
QueryPatient.

```
#include <gdcmQueryPatient.h>
```

Inheritance diagram for gdcm::QueryPatient:



Collaboration diagram for gdcm::QueryPatient:



Public Member Functions

- std::vector< Tag > GetHierachicalSearchTags (const ERootType &inRootType) const

  Return all Unique Key for a particular Query Root type (from the same level and above).
- const char * GetName () const
- std::vector< Tag > GetOptionalTags (const ERootType &inRootType) const
- DataElement GetQueryLevel () const
- std::vector< Tag > GetRequiredTags (const ERootType &inRootType) const
- std::vector< Tag > GetUniqueTags (const ERootType &inRootType) const

10.247.1 Detailed Description

QueryPatient.

contains: class to construct a patient-based query for c-find and c-move

10.247.2 Member Function Documentation

10.247.2.1 GetHierachicalSearchTags()

```
std::vector<Tag> gdcm::QueryPatient::GetHierachicalSearchTags (
            const ERootType & inRootType ) const  [virtual]
```

Return all Unique Key for a particular Query Root type (from the same level and above).

Implements gdcm::QueryBase.

10.247.2.2 GetName()

```
const char* gdcm::QueryPatient::GetName ( ) const  [virtual]
```

Implements gdcm::QueryBase.

10.247.2.3 GetOptionalTags()

```
std::vector<Tag> gdcm::QueryPatient::GetOptionalTags (
            const ERootType & inRootType ) const  [virtual]
```

Implements gdcm::QueryBase.

10.247.2.4 GetQueryLevel()

```
DataElement gdcm::QueryPatient::GetQueryLevel ( ) const  [virtual]
```

Implements gdcm::QueryBase.

10.247.2.5 GetRequiredTags()

```
std::vector<Tag> gdcm::QueryPatient::GetRequiredTags (
            const ERootType & inRootType ) const  [virtual]
```

Implements gdcm::QueryBase.

10.247.2.6 GetUniqueTags()

```
std::vector<Tag> gdcm::QueryPatient::GetUniqueTags (
            const ERootType & inRootType ) const  [virtual]
```

Implements gdcm::QueryBase.

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

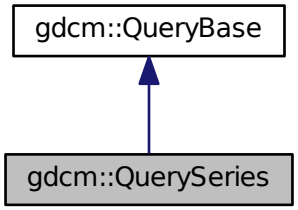
- gdcmQueryPatient.h

10.248 gdcm::QuerySeries Class Reference

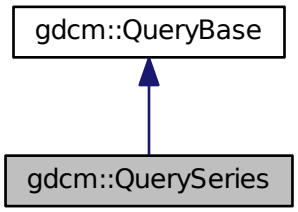
QuerySeries.

```
#include <gdcmQuerySeries.h>
```

Inheritance diagram for gdcm::QuerySeries:



Collaboration diagram for gdcm::QuerySeries:



Public Member Functions

- std::vector< Tag > GetHierachicalSearchTags (const ERootType &inRootType) const
  Return all Unique Key for a particular Query Root type (from the same level and above).
- const char * GetName () const
- std::vector< Tag > GetOptionalTags (const ERootType &inRootType) const
- DataElement GetQueryLevel () const
- std::vector< Tag > GetRequiredTags (const ERootType &inRootType) const
- std::vector< Tag > GetUniqueTags (const ERootType &inRootType) const

10.248.1 Detailed Description

QuerySeries.

contains: class to construct a series-based query for c-find and c-move

10.248.2 Member Function Documentation

10.248.2.1 GetHierachicalSearchTags()

```
std::vector<Tag> gdcm::QuerySeries::GetHierachicalSearchTags (
            const ERootType & inRootType ) const  [virtual]
```

Return all Unique Key for a particular Query Root type (from the same level and above).

Implements gdcm::QueryBase.

10.248.2.2 GetName()

```
const char* gdcm::QuerySeries::GetName ( ) const  [virtual]
```

Implements gdcm::QueryBase.

10.248.2.3 GetOptionalTags()

```
std::vector<Tag> gdcm::QuerySeries::GetOptionalTags (
            const ERootType & inRootType ) const  [virtual]
```

Implements gdcm::QueryBase.

10.248.2.4 GetQueryLevel()

```
DataElement gdcm::QuerySeries::GetQueryLevel ( ) const  [virtual]
```

Implements gdcm::QueryBase.

10.248.2.5 GetRequiredTags()

```
std::vector<Tag> gdcm::QuerySeries::GetRequiredTags (
            const ERootType & inRootType ) const  [virtual]
```

Implements gdcm::QueryBase.

10.248.2.6 GetUniqueTags()

```
std::vector<Tag> gdcm::QuerySeries::GetUniqueTags (
            const ERootType & inRootType ) const  [virtual]
```

Implements gdcm::QueryBase.

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

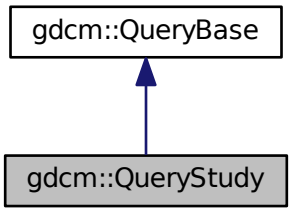
- gdcmQuerySeries.h

10.249 gdcm::QueryStudy Class Reference

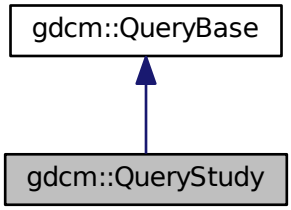
QueryStudy.h.

```
#include <gdcmQueryStudy.h>
```

Inheritance diagram for gdcm::QueryStudy:



Collaboration diagram for gdcm::QueryStudy:



Public Member Functions

- std::vector< Tag > GetHierachicalSearchTags (const ERootType &inRootType) const
  Return all Unique Key for a particular Query Root type (from the same level and above).
- const char * GetName () const
- std::vector< Tag > GetOptionalTags (const ERootType &inRootType) const
- DataElement GetQueryLevel () const
- std::vector< Tag > GetRequiredTags (const ERootType &inRootType) const
- std::vector< Tag > GetUniqueTags (const ERootType &inRootType) const

10.249.1 Detailed Description

QueryStudy.h.

contains: class to construct a study-based query for C-FIND and C-MOVE

10.249.2 Member Function Documentation

10.249.2.1 GetHierachicalSearchTags()

```
std::vector<Tag> gdcm::QueryStudy::GetHierachicalSearchTags (
            const ERootType & inRootType ) const  [virtual]
```

Return all Unique Key for a particular Query Root type (from the same level and above).

Implements gdcm::QueryBase.

10.249.2.2 GetName()

```
const char* gdcm::QueryStudy::GetName ( ) const  [virtual]
```

Implements gdcm::QueryBase.

10.249.2.3 GetOptionalTags()

```
std::vector<Tag> gdcm::QueryStudy::GetOptionalTags (
            const ERootType & inRootType ) const  [virtual]
```

Implements gdcm::QueryBase.

10.249.2.4 GetQueryLevel()

```
DataElement gdcm::QueryStudy::GetQueryLevel ( ) const  [virtual]
```

Implements gdcm::QueryBase.

10.249.2.5 GetRequiredTags()

```
std::vector<Tag> gdcm::QueryStudy::GetRequiredTags (
            const ERootType & inRootType ) const  [virtual]
```

Implements gdcm::QueryBase.

10.249.2.6 GetUniqueTags()

```
std::vector<Tag> gdcm::QueryStudy::GetUniqueTags (
            const ERootType & inRootType ) const  [virtual]
```

Implements gdcm::QueryBase.

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

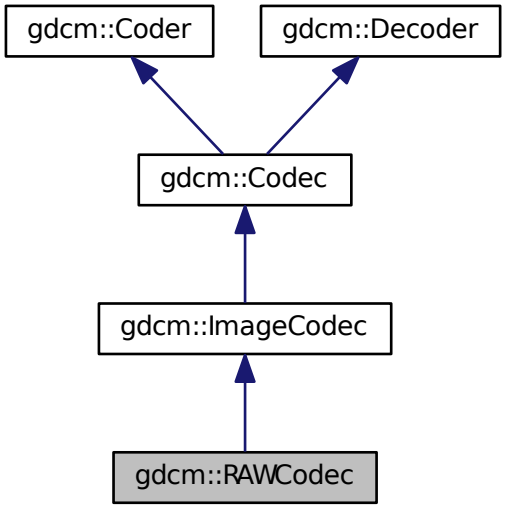
- gdcmQueryStudy.h

10.250 gdcm::RAWCodec Class Reference

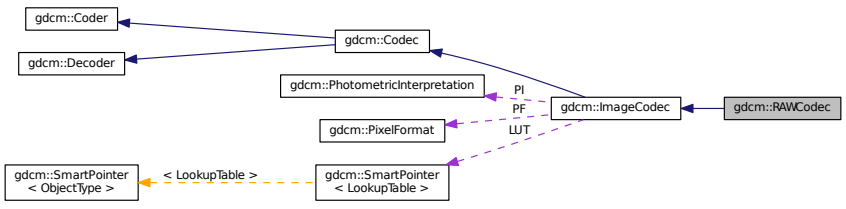
RAWCodec class.

`#include <gdcmRAWCodec.h>`

Inheritance diagram for gdcm::RAWCodec:



Collaboration diagram for gdcm::RAWCodec:



Public Member Functions

- RAWCodec ()
- ~RAWCodec ()
- bool CanCode (TransferSyntax const &ts) const

Return whether this coder support this transfer syntax (can code it)

- bool CanDecode (TransferSyntax const &ts) const

  Return whether this decoder support this transfer syntax (can decode it)
- virtual ImageCodec * Clone () const
- bool Code (DataElement const &in, DataElement &out)

  Code.
- bool Decode (DataElement const &is, DataElement &os)

  Decode.
- bool DecodeBytes (const char *inBytes, size_t inBufferLength, char *outBytes, size_t inOutBufferLength)
- bool GetHeaderInfo (std::istream &is, TransferSyntax &ts)

Protected Member Functions

- bool DecodeByStreams (std::istream &is, std::ostream &os)

Additional Inherited Members

10.250.1 Detailed Description

RAWCodec class.

10.250.2 Constructor & Destructor Documentation

10.250.2.1 RAWCodec()

```
gdcm::RAWCodec::RAWCodec ( )
```

10.250.2.2 ~RAWCodec()

```
gdcm::RAWCodec::~RAWCodec ( )
```

10.250.3 Member Function Documentation

10.250.3.1 CanCode()

```
bool gdcm::RAWCodec::CanCode (
            TransferSyntax const &  ) const  [virtual]
```

Return whether this coder support this transfer syntax (can code it)

Reimplemented from gdcm::ImageCodec.

10.250.3.2 CanDecode()

```
bool gdcm::RAWCodec::CanDecode (
            TransferSyntax const &  ) const  [virtual]
```

Return whether this decoder support this transfer syntax (can decode it)

Reimplemented from gdcm::ImageCodec.

10.250.3.3 Clone()

```
virtual ImageCodec* gdcm::RAWCodec::Clone ( ) const  [virtual]
```

Implements gdcm::ImageCodec.

10.250.3.4 Code()

```
bool gdcm::RAWCodec::Code (
            DataElement const & in_,
            DataElement & out_ )  [virtual]
```

Code.

Reimplemented from gdcm::Coder.

10.250.3.5 Decode()

```
bool gdcm::RAWCodec::Decode (
            DataElement const & ,
            DataElement &  )  [virtual]
```

Decode.

Reimplemented from gdcm::ImageCodec.

10.250.3.6 DecodeByStreams()

```
bool gdcm::RAWCodec::DecodeByStreams (
            std::istream & is,
            std::ostream & os )  [protected], [virtual]
```

Reimplemented from gdcm::ImageCodec.

10.250.3.7 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.

10.250.3.8 GetHeaderInfo()

```
bool gdcm::RAWCodec::GetHeaderInfo (
            std::istream & is,
            TransferSyntax & ts )  [virtual]
```

Reimplemented from gdcm::ImageCodec.

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

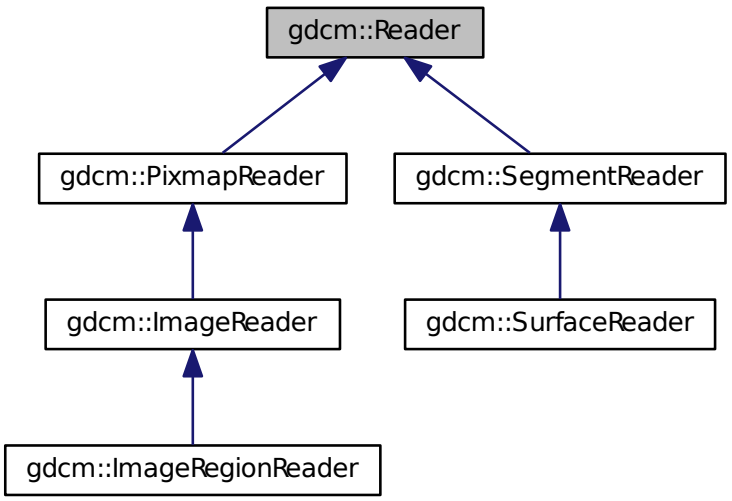
- gdcmRAWCodec.h

10.251 gdcm::Reader Class Reference

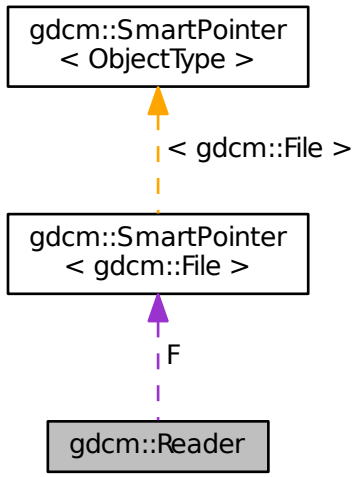
Reader ala DOM (Document Object Model)

`#include <gdcmReader.h>`

Inheritance diagram for gdcm::Reader:



Collaboration diagram for gdcm::Reader:



Public Member Functions

- Reader ()
- virtual ~Reader ()
- bool CanRead () const
- const File & GetFile () const

  Set/Get File.
- File & GetFile ()

  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

10.251.1 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).

Detailled 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:

ChangePrivateTags.cxx, ChangeSequenceUltrasound.cxx, ClinicalTrialAnnotate.cxx, CreateFakeRTDOSE.cxx, csa2img.cxx, DiffFile.cxx, DumpADAC.cxx, DumpExamCard.cxx, DumpGEMSMovieGroup.cxx, DumpImageHeaderInfo.cxx, DumpPhilipsECHO.cxx, DumpToshibaDTI.cxx, DuplicatePCDE.cxx, ELSCINT1WaveToText.cxx, ExtractEncryptedContent.cxx, FixBrokenJ2K.cxx, FixOrientation.cxx, gdcmrtionplan.cxx, gdcmrtplan.cxx, GenLongSeqs.cxx, GenSeqs.cxx, GetSequenceUltrasound.cxx, GetSubSequenceData.cxx, HelloWorld.cxx, iU22tomultisc.cxx, LargeVRDSExplicit.cxx, MakeTemplate.cxx, PatchFile.cxx, pmsct_rgb1.cxx, QIDO-RS.cxx, ReadAndDumpDICOMDIR.cxx, ReadAndPrintAttributes.cxx, ReadExplicitLengthSQIVR.cxx, ReadGEMSSDO.cxx, ReadUTF8QtDir.cxx, rle2img.cxx, and TestReader.cxx.

10.251.2 Constructor & Destructor Documentation

10.251.2.1 Reader()

```
gdcm::Reader::Reader ( )
```

10.251.2.2 ~Reader()

```
virtual gdcm::Reader::~Reader ( )  [virtual]
```

10.251.3 Member Function Documentation

10.251.3.1 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.

10.251.3.2 GetFile() [1/2]

```
const File& gdcm::Reader::GetFile ( ) const  [inline]
```

Set/Get File.

Examples:

ChangePrivateTags.cxx, ChangeSequenceUltrasound.cxx, ClinicalTrialAnnotate.cxx, CompressImage.cxx, CreateFakeRTDOSE.cxx, csa2img.cxx, DiffFile.cxx, DumpADAC.cxx, DumpExamCard.cxx, DumpGEMSMovieGroup.cxx, DumpImageHeaderInfo.cxx, DumpPhilipsECHO.cxx, DumpToshibaDTI.cxx, DuplicatePCDE.cxx, ELSCINT1WaveToText.cxx, ExtractEncryptedContent.cxx, ExtractIconFromFile.cxx, FixBrokenJ2K.cxx, FixJAIBugJPEGLS.cxx, FixOrientation.cxx, gdcmrtionplan.cxx, gdcmrtplan.cxx, GenLongSeqs.cxx, GenSeqs.cxx, GetJPEGSamplePrecision.cxx, GetSequenceUltrasound.cxx, GetSubSequenceData.cxx, HelloWorld.cxx, iU22tomultisc.cxx, LargeVRDSExplicit.cxx, MakeTemplate.cxx, MergeTwoFiles.cxx, MrProtocol.cxx, PatchFile.cxx, pmsct_rgb1.cxx, QIDO-RS.cxx, ReadAndDumpDICOMDIR.cxx, ReadAndPrintAttributes.cxx, ReadExplicitLengthSQIVR.cxx, ReadGEMSSDO.cxx, rle2img.cxx, and TestReader.cxx.

10.251.3.3 GetFile() [2/2]

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

Set/Get File.

10.251.3.4 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++

10.251.3.5 GetStreamPtr()

```
std::istream* gdcm::Reader::GetStreamPtr ( ) const  [inline], [protected]
```

10.251.3.6 Read()

```
virtual bool gdcm::Reader::Read ( )  [virtual]
```

Main function to read a file.

Reimplemented in gdcm::ImageRegionReader, gdcm::PixmapReader, gdcm::ImageReader, gdcm::SegmentReader, and gdcm::SurfaceReader.

Examples:

ChangePrivateTags.cxx, ChangeSequenceUltrasound.cxx, ClinicalTrialAnnotate.cxx, CreateFakeRTDOSE.cxx, csa2img.cxx, DiffFile.cxx, DumpADAC.cxx, DumpExamCard.cxx, DumpGEMSMovieGroup.cxx, DumpImageHeaderInfo.cxx, DumpPhilipsECHO.cxx, DumpToshibaDTI.cxx, DuplicatePCDE.cxx, ELSCINT1WaveToText.cxx, ExtractEncryptedContent.cxx, FixBrokenJ2K.cxx, FixOrientation.cxx, gdcmrtionplan.cxx, gdcmrtplan.cxx, GenLongSeqs.cxx, GenSeqs.cxx, GetSequenceUltrasound.cxx, GetSubSequenceData.cxx, HelloWorld.cxx, iU22tomultisc.cxx, LargeVRDSExplicit.cxx, MakeTemplate.cxx, PatchFile.cxx, pmsct_rgb1.cxx, QIDO-RS.cxx, ReadAndDumpDICOMDIR.cxx, ReadAndPrintAttributes.cxx, ReadExplicitLengthSQIVR.cxx, ReadGEMSSDO.cxx, rle2img.cxx, and TestReader.cxx.

10.251.3.7 ReadDataSet()

```
bool gdcm::Reader::ReadDataSet ( )  [protected]
```

10.251.3.8 ReadMetaInformation()

```
bool gdcm::Reader::ReadMetaInformation ( )  [protected]
```

10.251.3.9 ReadPreamble()

```
bool gdcm::Reader::ReadPreamble ( )  [protected]
```

10.251.3.10 ReadSelectedPrivateTags()

```
bool gdcm::Reader::ReadSelectedPrivateTags (
            std::set< PrivateTag > const & ptags,
            bool readvalues = true )
```

Will only read the specified selected private tags.

10.251.3.11 ReadSelectedTags()

```
bool gdcm::Reader::ReadSelectedTags (
            std::set< Tag > const & tags,
            bool readvalues = true )
```

Will only read the specified selected tags.

10.251.3.12 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	

10.251.3.13 SetFile()

```
void gdcm::Reader::SetFile (
            File & file )  [inline]
```

Set/Get File.

10.251.3.14 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:

BasicImageAnonymizer.cs, ChangePrivateTags.cxx, ChangeSequenceUltrasound.cxx, CheckBigEndianBug.cxx, ClinicalTrialAnnotate.cxx, CompressImage.cxx, ConvertToQImage.cxx, CreateFakeRTDOSE.cxx, csa2img.cxx, DiffFile.cxx, DumpADAC.cxx, DumpCSA.cs, DumpExamCard.cxx, DumpGEMSMovieGroup.cxx, DumpImageHeaderInfo.cxx, DumpPhilipsECHO.cxx, DumpToshibaDTI.cxx, DuplicatePCDE.cxx, ELSCINT1WaveToText.cxx, ExtractEncryptedContent.cxx, ExtractIconFromFile.cxx, ExtractImageRegion.cs, ExtractImageRegionWithLUT.cs, FixBrokenJ2K.cxx, FixJAIBugJPEGLS.cxx, FixOrientation.cxx, gdcmrtionplan.cxx, gdcmrtplan.cxx, GenLongSeqs.cxx, GenSeqs.cxx, GetJPEGSamplePrecision.cxx, GetSequenceUltrasound.cxx, GetSubSequenceData.cxx, HelloVizWorld.cxx, HelloWorld.cxx, iU22tomultisc.cxx, LargeVRDSExplicit.cxx, MakeTemplate.cxx, MergeTwoFiles.cxx, MrProtocol.cxx, NewSequence.cs, PatchFile.cxx, pmsct_rgb1.cxx, QIDO-RS.cxx, ReadAndDumpDICOMDIR.cxx, ReadAndPrintAttributes.cxx, ReadExplicitLengthSQIVR.cxx, ReadGEMSSDO.cxx, ReadMultiTimesException.cxx, ReadUTF8QtDir.cxx, rle2img.cxx, SimplePrintPatientName.cs, TestReader.cxx, and threadgdcm.cxx.

10.251.3.15 SetStream()

```
void gdcm::Reader::SetStream (
            std::istream & input_stream )  [inline]
```

Set the open-ed stream directly.

Examples:

DumpToshibaDTI.cxx, and ReadUTF8QtDir.cxx.

10.251.4 Friends And Related Function Documentation

10.251.4.1 StreamImageReader

```
friend class StreamImageReader  [friend]
```

10.251.5 Member Data Documentation

10.251.5.1 F

```
SmartPointer<File> gdcm::Reader::F  [protected]
```

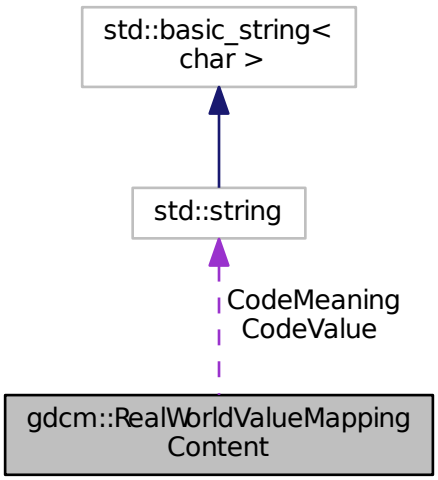
The documentation for this class was generated from the following file:

- gdcmReader.h

10.252 gdcm::RealWorldValueMappingContent Struct Reference

`#include <gdcmImageHelper.h>`

Collaboration diagram for gdcm::RealWorldValueMappingContent:



Public Attributes

- std::string CodeMeaning
- std::string CodeValue
- double RealWorldValueIntercept
- double RealWorldValueSlope

10.252.1 Member Data Documentation

10.252.1.1 CodeMeaning

`std::string gdcm::RealWorldValueMappingContent::CodeMeaning`

10.252.1.2 CodeValue

`std::string gdcm::RealWorldValueMappingContent::CodeValue`

10.252.1.3 RealWorldValueIntercept

```
double gdcm::RealWorldValueMappingContent::RealWorldValueIntercept
```

10.252.1.4 RealWorldValueSlope

```
double gdcm::RealWorldValueMappingContent::RealWorldValueSlope
```

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

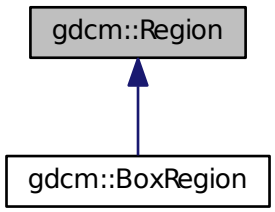
- gdcmImageHelper.h

10.253 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.

10.253.1 Detailed Description

Class for manipulation region.

10.253.2 Constructor & Destructor Documentation

10.253.2.1 Region()

```
gdcm::Region::Region ( )
```

10.253.2.2 ~Region()

```
virtual gdcm::Region::~Region ( )  [virtual]
```

10.253.3 Member Function Documentation

10.253.3.1 Area()

```
virtual size_t gdcm::Region::Area ( ) const  [pure virtual]
```

compute the area

Implemented in gdcm::BoxRegion.

10.253.3.2 Clone()

```
virtual Region* gdcm::Region::Clone ( ) const  [pure virtual]
```

Implemented in gdcm::BoxRegion.

10.253.3.3 ComputeBoundingBox()

```
virtual BoxRegion gdcm::Region::ComputeBoundingBox ( )  [pure virtual]
```

Return the Axis-Aligned minimum bounding box for all regions.

Implemented in gdcm::BoxRegion.

10.253.3.4 Empty()

```
virtual bool gdcm::Region::Empty ( ) const  [pure virtual]
```

return whether this domain is empty:

Implemented in gdcm::BoxRegion.

10.253.3.5 IsValid()

```
virtual bool gdcm::Region::IsValid ( ) const  [pure virtual]
```

return whether this is valid domain

Implemented in gdcm::BoxRegion.

10.253.3.6 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

10.254 gdcm::Rescaler Class Reference

Rescale class.

```
#include <gdcmRescaler.h>
```

Public Member Functions

- Rescaler ()
- ∼Rescaler ()
- 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)

10.254.1 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 ouput, 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

10.254.2 Constructor & Destructor Documentation

10.254.2.1 Rescaler()

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

10.254.2.2 ~Rescaler()

```
gdcm::Rescaler::~Rescaler ( )  [inline]
```

10.254.3 Member Function Documentation

10.254.3.1 ComputeInterceptSlopePixelType()

```
PixelFormat::ScalarType gdcm::Rescaler::ComputeInterceptSlopePixelType ( )
```

Compute the Pixel Format of the output data Used for direct transformation

10.254.3.2 ComputePixelTypeFromMinMax()

```
PixelFormat gdcm::Rescaler::ComputePixelTypeFromMinMax ( )
```

Compute the Pixel Format of the output data Used for inverse transformation

10.254.3.3 GetIntercept()

```
double gdcm::Rescaler::GetIntercept ( ) const  [inline]
```

10.254.3.4 GetSlope()

```
double gdcm::Rescaler::GetSlope ( ) const  [inline]
```

10.254.3.5 InverseRescale()

```
bool gdcm::Rescaler::InverseRescale (
            char * out,
            const char * in,
            size_t n )
```

Inverse transform.

10.254.3.6 InverseRescaleFunctionIntoBestFit()

```
template<typename TIn >
void gdcm::Rescaler::InverseRescaleFunctionIntoBestFit (
            char * out,
            const TIn * in,
            size_t n )  [protected]
```

10.254.3.7 Rescale()

```
bool gdcm::Rescaler::Rescale (
            char * out,
            const char * in,
            size_t n )
```

Direct transform.

10.254.3.8 RescaleFunctionIntoBestFit()

```
template<typename TIn >
void gdcm::Rescaler::RescaleFunctionIntoBestFit (
            char * out,
            const TIn * in,
            size_t n )  [protected]
```

10.254.3.9 SetIntercept()

```
void gdcm::Rescaler::SetIntercept (
            double i )  [inline]
```

Set Intercept: used for both direct&inverse transformation.

10.254.3.10 SetMinMaxForPixelType()

```
void gdcm::Rescaler::SetMinMaxForPixelType (
            double min,
            double max )  [inline]
```

Set target interval for output data. A best match will be computed (if possible) Used for inverse transformation

10.254.3.11 SetPixelFormat()

```
void gdcm::Rescaler::SetPixelFormat (
            PixelFormat const & pf )  [inline]
```

Set Pixel Format of input data.

10.254.3.12 SetSlope()

```
void gdcm::Rescaler::SetSlope (
            double s )  [inline]
```

Set Slope: user for both direct&inverse transformation.

10.254.3.13 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

10.254.3.14 SetUseTargetPixelType()

```
void gdcm::Rescaler::SetUseTargetPixelType (
            bool b )
```

Override default behavior of Rescale.

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

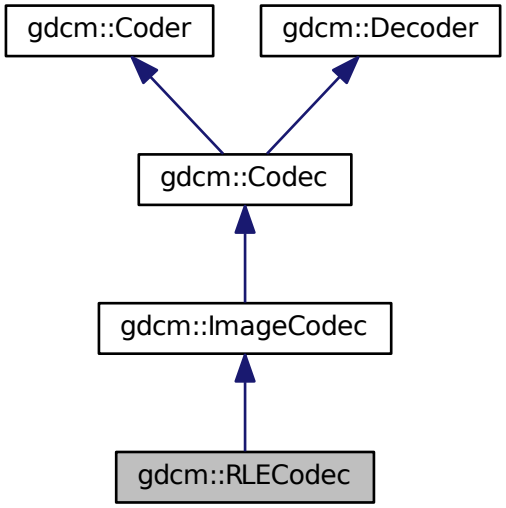
- gdcmRescaler.h

10.255 gdcm::RLECodec Class Reference

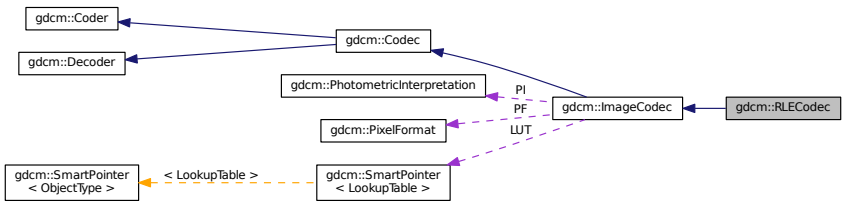
Class to do RLE.

`#include <gdcmRLECodec.h>`

Inheritance diagram for gdcm::RLECodec:



Collaboration diagram for gdcm::RLECodec:



Public Member Functions

- RLECodec ()
- ~RLECodec ()
- bool CanCode (TransferSyntax const &ts) const

Return whether this coder support this transfer syntax (can code it)

- bool CanDecode (TransferSyntax const &ts) const

  Return whether this decoder support this transfer syntax (can decode it)

- virtual ImageCodec * Clone () const
- bool Code (DataElement const &in, DataElement &out)

  Code.

- bool Decode (DataElement const &is, DataElement &os)

  Decode.

- unsigned long GetBufferLength () const
- bool GetHeaderInfo (std::istream &is, TransferSyntax &ts)
- void SetBufferLength (unsigned long l)
- void SetLength (unsigned long l)

Protected Member Functions

- bool AppendFrameEncode (std::ostream &out, const char *data, size_t datalen)
- bool AppendRowEncode (std::ostream &out, const char *data, size_t datalen)
- bool DecodeByStreams (std::istream &is, std::ostream &os)
- 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 ()
- bool IsRowEncoder ()
- bool StartEncode (std::ostream &)
- bool StopEncode (std::ostream &)

Friends

- class ImageRegionReader

Additional Inherited Members

10.255.1 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).

10.255.2 Constructor & Destructor Documentation

10.255.2.1 RLECodec()

```
gdcm::RLECodec::RLECodec ( )
```

10.255.2.2 ~RLECodec()

```
gdcm::RLECodec::~RLECodec ( )
```

10.255.3 Member Function Documentation

10.255.3.1 AppendFrameEncode()

```
bool gdcm::RLECodec::AppendFrameEncode (
            std::ostream & out,
            const char * data,
            size_t datalen )  [protected], [virtual]
```

Reimplemented from gdcm::ImageCodec.

10.255.3.2 AppendRowEncode()

```
bool gdcm::RLECodec::AppendRowEncode (
            std::ostream & out,
            const char * data,
            size_t datalen )  [protected], [virtual]
```

Reimplemented from gdcm::ImageCodec.

10.255.3.3 CanCode()

```
bool gdcm::RLECodec::CanCode (
            TransferSyntax const &  ) const  [virtual]
```

Return whether this coder support this transfer syntax (can code it)

Reimplemented from gdcm::ImageCodec.

10.255.3.4 CanDecode()

```
bool gdcm::RLECodec::CanDecode (
            TransferSyntax const &  ) const  [virtual]
```

Return whether this decoder support this transfer syntax (can decode it)

Reimplemented from gdcm::ImageCodec.

10.255.3.5 Clone()

```
virtual ImageCodec* gdcm::RLECodec::Clone ( ) const  [virtual]
```

Implements gdcm::ImageCodec.

10.255.3.6 Code()

```
bool gdcm::RLECodec::Code (
            DataElement const & in_,
            DataElement & out_ )  [virtual]
```

Code.

Reimplemented from gdcm::Coder.

10.255.3.7 Decode()

```
bool gdcm::RLECodec::Decode (
            DataElement const & ,
            DataElement &  )  [virtual]
```

Decode.

Reimplemented from gdcm::ImageCodec.

10.255.3.8 DecodeByStreams()

```
bool gdcm::RLECodec::DecodeByStreams (
            std::istream & is,
            std::ostream & os )  [protected], [virtual]
```

Reimplemented from gdcm::ImageCodec.

10.255.3.9 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]
```

10.255.3.10 GetBufferLength()

```
unsigned long gdcm::RLECodec::GetBufferLength ( ) const  [inline]
```

10.255.3.11 GetHeaderInfo()

```
bool gdcm::RLECodec::GetHeaderInfo (
            std::istream & is,
            TransferSyntax & ts )  [virtual]
```

Reimplemented from gdcm::ImageCodec.

10.255.3.12 IsFrameEncoder()

```
bool gdcm::RLECodec::IsFrameEncoder ( )  [protected], [virtual]
```

Reimplemented from gdcm::ImageCodec.

10.255.3.13 IsRowEncoder()

```
bool gdcm::RLECodec::IsRowEncoder ( )  [protected], [virtual]
```

Reimplemented from gdcm::ImageCodec.

10.255.3.14 SetBufferLength()

```
void gdcm::RLECodec::SetBufferLength (
            unsigned long l )  [inline]
```

10.255.3.15 SetLength()

```
void gdcm::RLECodec::SetLength (
            unsigned long l )  [inline]
```

10.255.3.16 StartEncode()

```
bool gdcm::RLECodec::StartEncode (
            std::ostream &  )  [protected], [virtual]
```

Reimplemented from gdcm::ImageCodec.

10.255.3.17 StopEncode()

```
bool gdcm::RLECodec::StopEncode (
            std::ostream &  )  [protected], [virtual]
```

Reimplemented from gdcm::ImageCodec.

10.255.4 Friends And Related Function Documentation

10.255.4.1 ImageRegionReader

```
friend class ImageRegionReader  [friend]
```

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

- gdcmRLECodec.h

10.256 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

10.256.1 Detailed Description

RoleSelectionSub.

PS 3.7 Table D.3-9 SCP/SCU ROLE SELECTION SUB-ITEM FIELDS (A-ASSOCIATE-RQ)

10.256.2 Constructor & Destructor Documentation

10.256.2.1 RoleSelectionSub()

```
gdcm::network::RoleSelectionSub::RoleSelectionSub ( )
```

10.256.3 Member Function Documentation

10.256.3.1 Print()

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

10.256.3.2 Read()

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

10.256.3.3 SetTuple()

```
void gdcm::network::RoleSelectionSub::SetTuple (
            const char * uid,
            uint8_t scurole,
            uint8_t scprole )
```

10.256.3.4 Size()

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

10.256.3.5 Write()

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

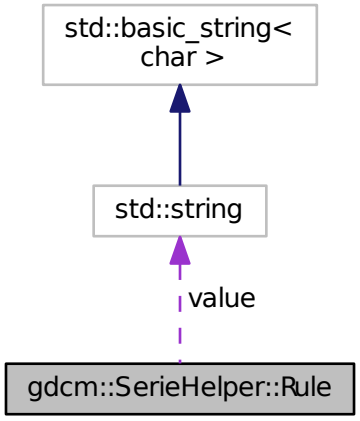
The documentation for this class was generated from the following file:

- gdcmRoleSelectionSub.h

10.257 gdcm::SerieHelper::Rule Struct Reference

`#include <gdcmSerieHelper.h>`

Collaboration diagram for gdcm::SerieHelper::Rule:



Public Attributes

- uint16_t elem
- uint16_t group
- int op
- std::string value

10.257.1 Member Data Documentation

10.257.1.1 elem

`uint16_t gdcm::SerieHelper::Rule::elem`

10.257.1.2 group

`uint16_t gdcm::SerieHelper::Rule::group`

10.257.1.3 op

`int gdcm::SerieHelper::Rule::op`

10.257.1.4 value

```
std::string gdcm::SerieHelper::Rule::value
```

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

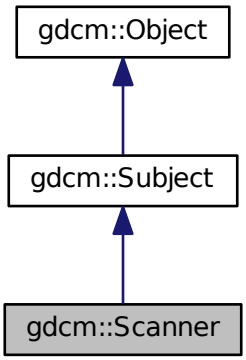
- gdcmSerieHelper.h

10.258 gdcm::Scanner Class Reference

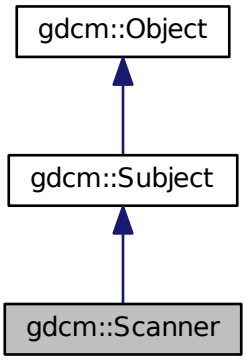
Scanner.

```
#include <gdcmScanner.h>
```

Inheritance diagram for gdcm::Scanner:



Collaboration diagram for gdcm::Scanner:



Classes

- struct Itstr

Public Types

- typedef MappingType::const_iterator ConstIterator
- typedef std::map< const char *, TagToValue, Itstr > MappingType
- typedef std::map< Tag, const char * > TagToValue
- typedef TagToValue::value_type TagToValueValueType
- typedef std::set< std::string > ValuesType

Public Member Functions

- Scanner ()
- ~Scanner ()
- 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

  Print result.
- bool Scan (Directory::FilenamesType const &filenames)

  Start the scan !

Static Public Member Functions

- static SmartPointer< Scanner > New ()

  for wrapped language: instanciate a reference counted object

Protected Member Functions

- void ProcessPublicTag (StringFilter &sf, const char *filename)

Friends

- std::ostream & operator<< (std::ostream &_os, const Scanner &s)

10.258.1 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 mimimum 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.

10.258.2 Member Typedef Documentation

10.258.2.1 ConstIterator

```
typedef MappingType::const_iterator gdcm::Scanner::ConstIterator
```

10.258.2.2 MappingType

```
typedef std::map<const char *,TagToValue, ltstr> gdcm::Scanner::MappingType
```

10.258.2.3 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)

10.258.2.4 TagToValueValueType

```
typedef TagToValue::value_type gdcm::Scanner::TagToValueValueType
```

10.258.2.5 ValuesType

```
typedef std::set< std::string > gdcm::Scanner::ValuesType
```

10.258.3 Constructor & Destructor Documentation

10.258.3.1 Scanner()

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

10.258.3.2 ~Scanner()

```
gdcm::Scanner::~Scanner ( )
```

10.258.4 Member Function Documentation

10.258.4.1 AddPrivateTag()

```
void gdcm::Scanner::AddPrivateTag (
            PrivateTag const & t )
```

10.258.4.2 AddSkipTag()

```
void gdcm::Scanner::AddSkipTag (
            Tag const & t )
```

Add a tag that will need to be skipped. Those are root level skip tags.

10.258.4.3 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.

10.258.4.4 Begin()

```
ConstIterator gdcm::Scanner::Begin ( ) const  [inline]
```

10.258.4.5 ClearSkipTags()

```
void gdcm::Scanner::ClearSkipTags ( )
```

10.258.4.6 ClearTags()

```
void gdcm::Scanner::ClearTags ( )
```

10.258.4.7 End()

```
ConstIterator gdcm::Scanner::End ( ) const  [inline]
```

10.258.4.8 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'

10.258.4.9 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'

10.258.4.10 GetFilenames()

```
Directory::FilenamesType const& gdcm::Scanner::GetFilenames ( ) const  [inline]
```

10.258.4.11 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.

10.258.4.12 GetMapping()

```
TagToValue const& gdcm::Scanner::GetMapping (
            const char * filename ) const
```

Get the std::map mapping filenames to value for file 'filename'.

Examples:

DumpToSQLITE3.cxx.

10.258.4.13 GetMappingFromTagToValue()

```
TagToValue const& gdcm::Scanner::GetMappingFromTagToValue (
            Tag const & t,
            const char * value ) const
```

See GetFilenameFromTagToValue(). This is simply GetFilenameFromTagToValue followed.

10.258.4.14 GetMappings()

```
MappingType const& gdcm::Scanner::GetMappings ( ) const  [inline]
```

Mappings are the mapping from a particular tag to the map, mapping filename to value:

10.258.4.15 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)

10.258.4.16 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 !

10.258.4.17 GetValues() [1/2]

```
ValuesType const& gdcm::Scanner::GetValues ( ) const  [inline]
```

Get all the values found (in lexicographic order)

Examples:

SortImage.cxx, and VolumeSorter.cxx.

10.258.4.18 GetValues() [2/2]

```
ValuesType gdcm::Scanner::GetValues (
            Tag const & t ) const
```

Get all the values found (in lexicographic order) associated with Tag 't'.

10.258.4.19 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.

10.258.4.20 New()

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

for wrapped language: instanciate a reference counted object

10.258.4.21 Print()

```
void gdcm::Scanner::Print (
            std::ostream & os ) const  [virtual]
```

Print result.

Reimplemented from gdcm::Object.

Referenced by gdcm::operator<<().

10.258.4.22 ProcessPublicTag()

```
void gdcm::Scanner::ProcessPublicTag (
            StringFilter & sf,
            const char * filename )  [protected]
```

10.258.4.23 Scan()

```
bool gdcm::Scanner::Scan (
            Directory::FilenamesType const & filenames )
```

Start the scan !

Examples:

DiscriminateVolume.cxx, DumpToSQLITE3.cxx, SortImage.cxx, and VolumeSorter.cxx.

10.258.5 Friends And Related Function Documentation

10.258.5.1 operator<<

```
std::ostream& operator<< (
            std::ostream & _os,
            const Scanner & s )  [friend]
```

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

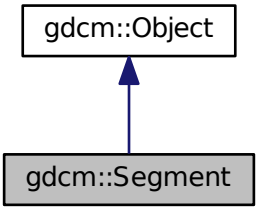
- gdcmScanner.h

10.259 gdcm::Segment Class Reference

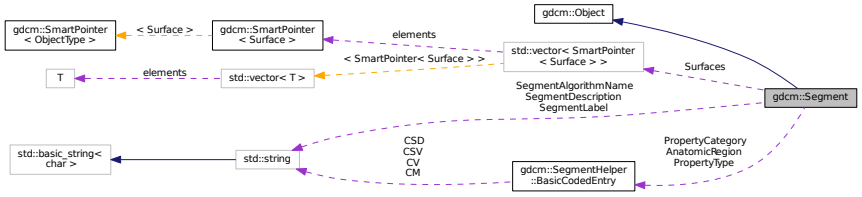
This class defines a segment.

`#include <gdcmSegment.h>`

Inheritance diagram for gdcm::Segment:



Collaboration diagram for gdcm::Segment:



Public Types

- enum ALGOType {
  MANUAL = 0,
  AUTOMATIC,
  ALGOType_END }
- typedef std::vector< SmartPointer< Surface > > SurfaceVector

Public Member Functions

- Segment ()
- virtual ~Segment ()
- void AddSurface (SmartPointer< Surface > surface)
- SegmentHelper::BasicCodedEntry const & GetAnatomicRegion () const
- SegmentHelper::BasicCodedEntry & GetAnatomicRegion ()
- SegmentHelper::BasicCodedEntry const & GetPropertyCategory () const
- SegmentHelper::BasicCodedEntry & GetPropertyCategory ()
- SegmentHelper::BasicCodedEntry const & GetPropertyType () const
- SegmentHelper::BasicCodedEntry & GetPropertyType ()
- 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 const & GetSurfaces () const
- SurfaceVector & GetSurfaces ()
- void SetAnatomicRegion (SegmentHelper::BasicCodedEntry const &BSE)
- void SetPropertyCategory (SegmentHelper::BasicCodedEntry const &BSE)
- void SetPropertyType (SegmentHelper::BasicCodedEntry const &BSE)
- 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
- SegmentHelper::BasicCodedEntry PropertyCategory
- SegmentHelper::BasicCodedEntry PropertyType
- std::string SegmentAlgorithmName
- ALGOType SegmentAlgorithmType
- std::string SegmentDescription
- std::string SegmentLabel
- unsigned short SegmentNumber
- unsigned long SurfaceCount
- SurfaceVector Surfaces

Additional Inherited Members

10.259.1 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

10.259.2 Member Typedef Documentation

10.259.2.1 SurfaceVector

```
typedef std::vector< SmartPointer< Surface > > gdcm::Segment::SurfaceVector
```

10.259.3 Member Enumeration Documentation

10.259.3.1 ALGOType

```
enum gdcm::Segment::ALGOType
```

Enumerator

MANUAL	
AUTOMATIC	
ALGOType_END	

10.259.4 Constructor & Destructor Documentation

10.259.4.1 Segment()

```
gdcm::Segment::Segment ( )
```

10.259.4.2 ~Segment()

```
virtual gdcm::Segment::~Segment ( )  [virtual]
```

10.259.5 Member Function Documentation

10.259.5.1 AddSurface()

```
void gdcm::Segment::AddSurface (
            SmartPointer< Surface > surface )
```

10.259.5.2 GetALGOType()

```
static ALGOType gdcm::Segment::GetALGOType (
            const char * type )  [static]
```

10.259.5.3 GetALGOTypeString()

```
static const char* gdcm::Segment::GetALGOTypeString (
            ALGOType type )  [static]
```

10.259.5.4 GetAnatomicRegion() [1/2]

```
SegmentHelper::BasicCodedEntry const& gdcm::Segment::GetAnatomicRegion ( ) const
```

10.259.5.5 GetAnatomicRegion() [2/2]

```
SegmentHelper::BasicCodedEntry& gdcm::Segment::GetAnatomicRegion ( )
```

10.259.5.6 GetPropertyCategory() [1/2]

```
SegmentHelper::BasicCodedEntry const& gdcm::Segment::GetPropertyCategory ( ) const
```

10.259.5.7 GetPropertyCategory() [2/2]

```
SegmentHelper::BasicCodedEntry& gdcm::Segment::GetPropertyCategory ( )
```

10.259.5.8 GetPropertyType() [1/2]

```
SegmentHelper::BasicCodedEntry const& gdcm::Segment::GetPropertyType ( ) const
```

10.259.5.9 GetPropertyType() [2/2]

```
SegmentHelper::BasicCodedEntry& gdcm::Segment::GetPropertyType ( )
```

10.259.5.10 GetSegmentAlgorithmName()

```
const char* gdcm::Segment::GetSegmentAlgorithmName ( ) const
```

10.259.5.11 GetSegmentAlgorithmType()

```
ALGOType gdcm::Segment::GetSegmentAlgorithmType ( ) const
```

10.259.5.12 GetSegmentDescription()

```
const char* gdcm::Segment::GetSegmentDescription ( ) const
```

10.259.5.13 GetSegmentLabel()

```
const char* gdcm::Segment::GetSegmentLabel ( ) const
```

10.259.5.14 GetSegmentNumber()

```
unsigned short gdcm::Segment::GetSegmentNumber ( ) const
```

10.259.5.15 GetSurface()

```
SmartPointer< Surface > gdcm::Segment::GetSurface (
            const unsigned int idx = 0 ) const
```

10.259.5.16 GetSurfaceCount()

```
unsigned long gdcm::Segment::GetSurfaceCount ( )
```

10.259.5.17 GetSurfaces() [1/2]

```
SurfaceVector const& gdcm::Segment::GetSurfaces ( ) const
```

10.259.5.18 GetSurfaces() [2/2]

```
SurfaceVector& gdcm::Segment::GetSurfaces ( )
```

10.259.5.19 SetAnatomicRegion()

```
void gdcm::Segment::SetAnatomicRegion (
            SegmentHelper::BasicCodedEntry const & BSE )
```

10.259.5.20 SetPropertyCategory()

```
void gdcm::Segment::SetPropertyCategory (
            SegmentHelper::BasicCodedEntry const & BSE )
```

10.259.5.21 SetPropertyType()

```
void gdcm::Segment::SetPropertyType (
            SegmentHelper::BasicCodedEntry const & BSE )
```

10.259.5.22 SetSegmentAlgorithmName()

```
void gdcm::Segment::SetSegmentAlgorithmName (
            const char * name )
```

10.259.5.23 SetSegmentAlgorithmType() [1/2]

```
void gdcm::Segment::SetSegmentAlgorithmType (
            ALGOType type )
```

10.259.5.24 SetSegmentAlgorithmType() [2/2]

```
void gdcm::Segment::SetSegmentAlgorithmType (
            const char * typeStr )
```

10.259.5.25 SetSegmentDescription()

```
void gdcm::Segment::SetSegmentDescription (
            const char * description )
```

10.259.5.26 SetSegmentLabel()

```
void gdcm::Segment::SetSegmentLabel (
            const char * label )
```

10.259.5.27 SetSegmentNumber()

```
void gdcm::Segment::SetSegmentNumber (
            const unsigned short num )
```

10.259.5.28 SetSurfaceCount()

```
void gdcm::Segment::SetSurfaceCount (
            const unsigned long nb )
```

10.259.6 Member Data Documentation

10.259.6.1 AnatomicRegion

```
SegmentHelper::BasicCodedEntry gdcm::Segment::AnatomicRegion  [protected]
```

10.259.6.2 PropertyCategory

```
SegmentHelper::BasicCodedEntry gdcm::Segment::PropertyCategory  [protected]
```

10.259.6.3 PropertyType

```
SegmentHelper::BasicCodedEntry gdcm::Segment::PropertyType  [protected]
```

10.259.6.4 SegmentAlgorithmName

```
std::string gdcm::Segment::SegmentAlgorithmName  [protected]
```

10.259.6.5 SegmentAlgorithmType

```
ALGOType gdcm::Segment::SegmentAlgorithmType  [protected]
```

10.259.6.6 SegmentDescription

```
std::string gdcm::Segment::SegmentDescription  [protected]
```

10.259.6.7 SegmentLabel

```
std::string gdcm::Segment::SegmentLabel  [protected]
```

10.259.6.8 SegmentNumber

```
unsigned short gdcm::Segment::SegmentNumber  [protected]
```

10.259.6.9 SurfaceCount

```
unsigned long gdcm::Segment::SurfaceCount  [protected]
```

10.259.6.10 Surfaces

```
SurfaceVector gdcm::Segment::Surfaces  [protected]
```

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

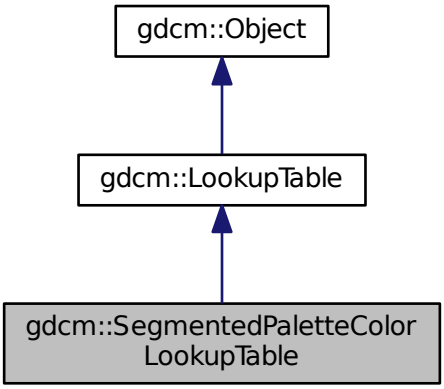
- gdcmSegment.h

10.260 gdcm::SegmentedPaletteColorLookupTable Class Reference

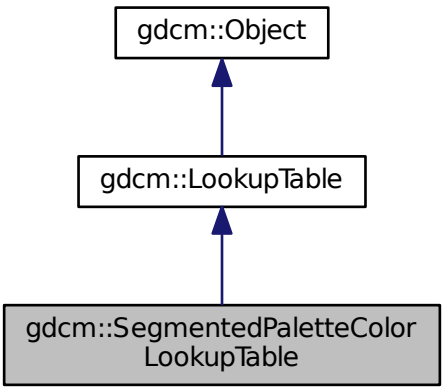
SegmentedPaletteColorLookupTable class.

```
#include <gdcmSegmentedPaletteColorLookupTable.h>
```

Inheritance diagram for gdcm::SegmentedPaletteColorLookupTable:



Collaboration diagram for gdcm::SegmentedPaletteColorLookupTable:



Public Member Functions

- SegmentedPaletteColorLookupTable ()
- ~SegmentedPaletteColorLookupTable ()
- void Print (std::ostream &) const
- void SetLUT (LookupTableType type, const unsigned char *array, unsigned int length)

  Initialize a SegmentedPaletteColorLookupTable.

Additional Inherited Members

10.260.1 Detailed Description

SegmentedPaletteColorLookupTable class.

10.260.2 Constructor & Destructor Documentation

10.260.2.1 SegmentedPaletteColorLookupTable()

```
gdcm::SegmentedPaletteColorLookupTable::SegmentedPaletteColorLookupTable ( )
```

10.260.2.2 ~SegmentedPaletteColorLookupTable()

```
gdcm::SegmentedPaletteColorLookupTable::~SegmentedPaletteColorLookupTable ( )
```

10.260.3 Member Function Documentation

10.260.3.1 Print()

```
void gdcm::SegmentedPaletteColorLookupTable::Print (
            std::ostream &  ) const  [inline], [virtual]
```

Reimplemented from gdcm::LookupTable.

10.260.3.2 SetLUT()

```
void gdcm::SegmentedPaletteColorLookupTable::SetLUT (
            LookupTableType type,
            const unsigned char * array,
            unsigned int length )  [virtual]
```

Initialize a SegmentedPaletteColorLookupTable.

Reimplemented from gdcm::LookupTable.

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

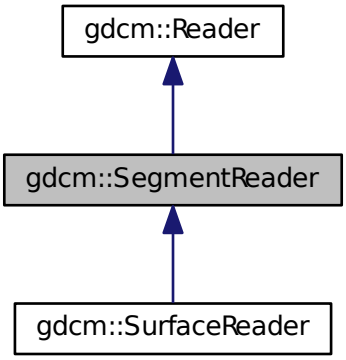
- gdcmSegmentedPaletteColorLookupTable.h

10.261 gdcm::SegmentReader Class Reference

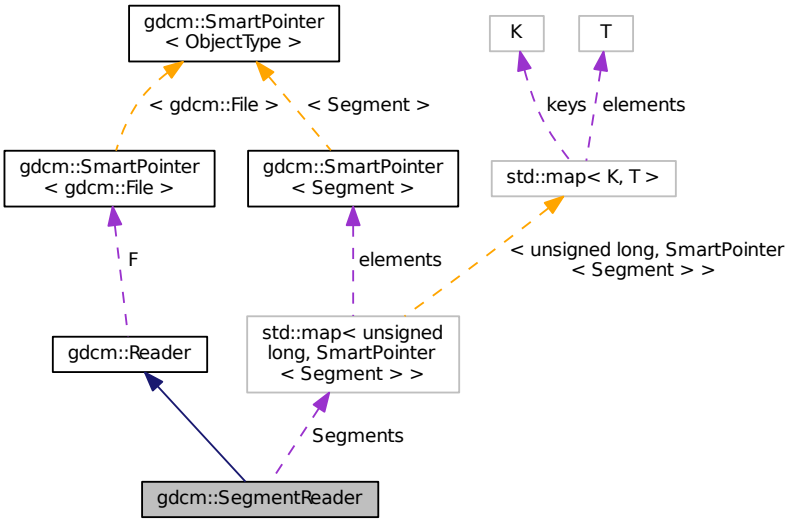
This class defines a segment reader.

`#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 ()
- virtual ~SegmentReader ()
- const SegmentVector GetSegments () const
- SegmentVector GetSegments ()
- virtual bool Read ()

  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

10.261.1 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

10.261.2 Member Typedef Documentation

10.261.2.1 SegmentMap

```
typedef std::map< unsigned long, SmartPointer< Segment > > gdcm::SegmentReader::SegmentMap  [protected]
```

10.261.2.2 SegmentVector

```
typedef std::vector< SmartPointer< Segment > > gdcm::SegmentReader::SegmentVector
```

10.261.3 Constructor & Destructor Documentation

10.261.3.1 SegmentReader()

```
gdcm::SegmentReader::SegmentReader ( )
```

10.261.3.2 ~SegmentReader()

```
virtual gdcm::SegmentReader::~SegmentReader ( )  [virtual]
```

10.261.4 Member Function Documentation

10.261.4.1 GetSegments() [1/2]

```
const SegmentVector gdcm::SegmentReader::GetSegments ( ) const
```

10.261.4.2 GetSegments() [2/2]

```
SegmentVector gdcm::SegmentReader::GetSegments ( )
```

10.261.4.3 Read()

```
virtual bool gdcm::SegmentReader::Read ( )  [virtual]
```

Read.

Reimplemented from gdcm::Reader.

Reimplemented in gdcm::SurfaceReader.

10.261.4.4 ReadSegment()

```
bool gdcm::SegmentReader::ReadSegment (
            const Item & segmentItem,
            const unsigned int idx )  [protected]
```

10.261.4.5 ReadSegments()

```
bool gdcm::SegmentReader::ReadSegments ( )  [protected]
```

10.261.5 Member Data Documentation

10.261.5.1 Segments

```
SegmentMap gdcm::SegmentReader::Segments  [protected]
```

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

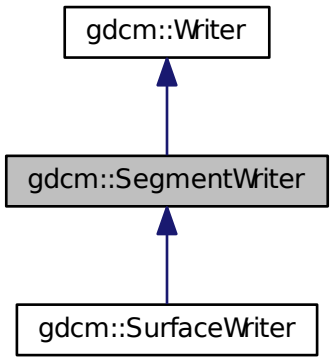
- gdcmSegmentReader.h

10.262 gdcm::SegmentWriter Class Reference

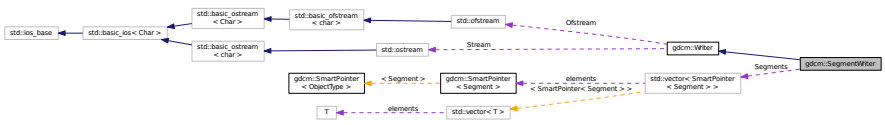
This class defines a segment writer.

```
#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 ()
- virtual ~SegmentWriter ()
- void AddSegment (SmartPointer< Segment > segment)
- unsigned int GetNumberOfSegments () const
- SmartPointer< Segment > GetSegment (const unsigned int idx=0) const
- const SegmentVector & GetSegments () const
- SegmentVector & GetSegments ()
- void SetNumberOfSegments (const unsigned int size)
- void SetSegments (SegmentVector &segments)
- bool Write ()

  Write.

Protected Member Functions

- bool PrepareWrite ()

Protected Attributes

- SegmentVector Segments

10.262.1 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

10.262.2 Member Typedef Documentation

10.262.2.1 SegmentVector

```
typedef std::vector< SmartPointer< Segment > > gdcm::SegmentWriter::SegmentVector
```

10.262.3 Constructor & Destructor Documentation

10.262.3.1 SegmentWriter()

```
gdcm::SegmentWriter::SegmentWriter ( )
```

10.262.3.2 ~SegmentWriter()

```
virtual gdcm::SegmentWriter::~SegmentWriter ( )  [virtual]
```

10.262.4 Member Function Documentation

10.262.4.1 AddSegment()

```
void gdcm::SegmentWriter::AddSegment (
            SmartPointer< Segment > segment )
```

10.262.4.2 GetNumberOfSegments()

```
unsigned int gdcm::SegmentWriter::GetNumberOfSegments ( ) const
```

10.262.4.3 GetSegment()

```
SmartPointer< Segment > gdcm::SegmentWriter::GetSegment (
            const unsigned int idx = 0 ) const
```

10.262.4.4 GetSegments() [1/2]

```
const SegmentVector& gdcm::SegmentWriter::GetSegments ( ) const
```

10.262.4.5 GetSegments() [2/2]

```
SegmentVector& gdcm::SegmentWriter::GetSegments ( )
```

10.262.4.6 PrepareWrite()

```
bool gdcm::SegmentWriter::PrepareWrite ( )  [protected]
```

10.262.4.7 SetNumberOfSegments()

```
void gdcm::SegmentWriter::SetNumberOfSegments (
            const unsigned int size )
```

10.262.4.8 SetSegments()

```
void gdcm::SegmentWriter::SetSegments (
            SegmentVector & segments )
```

10.262.4.9 Write()

```
bool gdcm::SegmentWriter::Write ( )  [virtual]
```

Write.

Reimplemented from gdcm::Writer.

Reimplemented in gdcm::SurfaceWriter.

10.262.5 Member Data Documentation

10.262.5.1 Segments

```
SegmentVector gdcm::SegmentWriter::Segments  [protected]
```

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

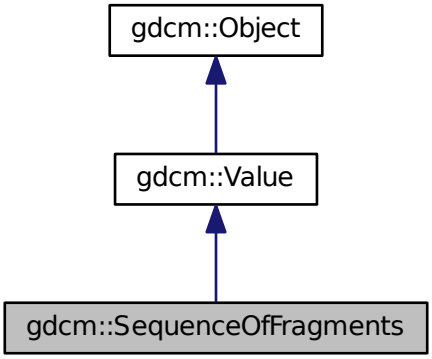
- gdcmSegmentWriter.h

10.263 gdcm::SequenceOfFragments Class Reference

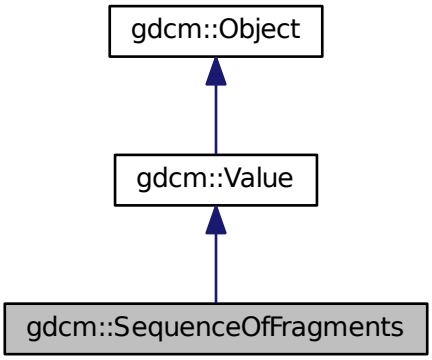
Class to represent a Sequence Of Fragments.

```
#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 ()

  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

  Returns the SQ length, as read from disk.
- SizeType GetNumberOfFragments () const
- const BasicOffsetTable & GetTable () const
- BasicOffsetTable & GetTable ()
- bool operator== (const Value &val) const
- void Print (std::ostream &os) const
- 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)

  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

10.263.1 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:

FixBrokenJ2K.cxx, FixJAIBugJPEGLS.cxx, and GetJPEGSamplePrecision.cxx.

10.263.2 Member Typedef Documentation

10.263.2.1 ConstIterator

```
typedef FragmentVector::const_iterator gdcm::SequenceOfFragments::ConstIterator
```

10.263.2.2 FragmentVector

```
typedef std::vector<Fragment> gdcm::SequenceOfFragments::FragmentVector
```

10.263.2.3 Iterator

```
typedef FragmentVector::iterator gdcm::SequenceOfFragments::Iterator
```

10.263.2.4 SizeType

```
typedef FragmentVector::size_type gdcm::SequenceOfFragments::SizeType
```

10.263.3 Constructor & Destructor Documentation

10.263.3.1 SequenceOfFragments()

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

constructor (UndefinedLength by default)

10.263.4 Member Function Documentation

10.263.4.1 AddFragment()

```
void gdcm::SequenceOfFragments::AddFragment (
            Fragment const & item )
```

Appends a Fragment to the already added ones.

Examples:

FixBrokenJ2K.cxx.

10.263.4.2 Begin() [1/2]

```
Iterator gdcm::SequenceOfFragments::Begin ( )  [inline]
```

10.263.4.3 Begin() [2/2]

```
ConstIterator gdcm::SequenceOfFragments::Begin ( ) const  [inline]
```

10.263.4.4 Clear()

```
void gdcm::SequenceOfFragments::Clear ( )  [virtual]
```

Clear.

Implements gdcm::Value.

10.263.4.5 ComputeByteLength()

```
unsigned long gdcm::SequenceOfFragments::ComputeByteLength ( ) const
```

10.263.4.6 ComputeLength()

```
VL gdcm::SequenceOfFragments::ComputeLength ( ) const
```

10.263.4.7 End() [1/2]

```
Iterator gdcm::SequenceOfFragments::End ( )  [inline]
```

10.263.4.8 End() [2/2]

```
ConstIterator gdcm::SequenceOfFragments::End ( ) const  [inline]
```

10.263.4.9 GetBuffer()

```
bool gdcm::SequenceOfFragments::GetBuffer (
            char * buffer,
            unsigned long length ) const
```

10.263.4.10 GetFragBuffer()

```
bool gdcm::SequenceOfFragments::GetFragBuffer (
            unsigned int fragNb,
            char * buffer,
            unsigned long & length ) const
```

10.263.4.11 GetFragment()

```
const Fragment& gdcm::SequenceOfFragments::GetFragment (
            SizeType num ) const
```

Examples:

FixBrokenJ2K.cxx, and FixJAIBugJPEGLS.cxx.

10.263.4.12 GetLength()

```
VL gdcm::SequenceOfFragments::GetLength ( ) const  [inline], [virtual]
```

Returns the SQ length, as read from disk.

Implements gdcm::Value.

10.263.4.13 GetNumberOfFragments()

```
SizeType gdcm::SequenceOfFragments::GetNumberOfFragments ( ) const
```

Examples:

FixJAIBugJPEGLS.cxx.

10.263.4.14 GetTable() [1/2]

```
const BasicOffsetTable& gdcm::SequenceOfFragments::GetTable ( ) const  [inline]
```

10.263.4.15 GetTable() [2/2]

```
BasicOffsetTable& gdcm::SequenceOfFragments::GetTable ( )  [inline]
```

10.263.4.16 New()

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

10.263.4.17 operator==()

```
bool gdcm::SequenceOfFragments::operator== (
            const Value & val ) const  [inline], [virtual]
```

Implements gdcm::Value.

10.263.4.18 Print()

```
void gdcm::SequenceOfFragments::Print (
            std::ostream & os ) const  [inline], [virtual]
```

Reimplemented from gdcm::Object.

10.263.4.19 Read()

```
template<typename TSwap >
std::istream& gdcm::SequenceOfFragments::Read (
            std::istream & is,
            bool readvalues = true )  [inline]
```

10.263.4.20 ReadPreValue()

```
template<typename TSwap >
std::istream& gdcm::SequenceOfFragments::ReadPreValue (
            std::istream & is )  [inline]
```

References gdcmDebugMacro.

10.263.4.21 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().

10.263.4.22 SetLength()

```
void gdcm::SequenceOfFragments::SetLength (
            VL length )  [inline], [virtual]
```

Sets the actual SQ length.

Implements gdcm::Value.

10.263.4.23 Write()

```
template<typename TSwap >
std::ostream const& gdcm::SequenceOfFragments::Write (
            std::ostream & os ) const  [inline]
```

References gdcm::VL::Write().

10.263.4.24 WriteBuffer()

```
bool gdcm::SequenceOfFragments::WriteBuffer (
            std::ostream & os ) const
```

Examples:

GetJPEGSamplePrecision.cxx.

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

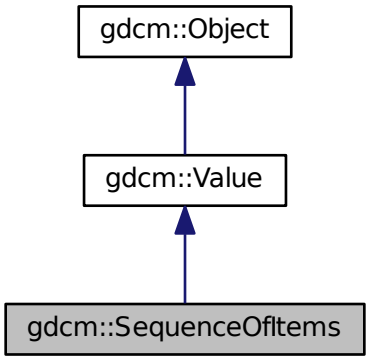
- gdcmSequenceOfFragments.h

10.264 gdcm::SequenceOfItems Class Reference

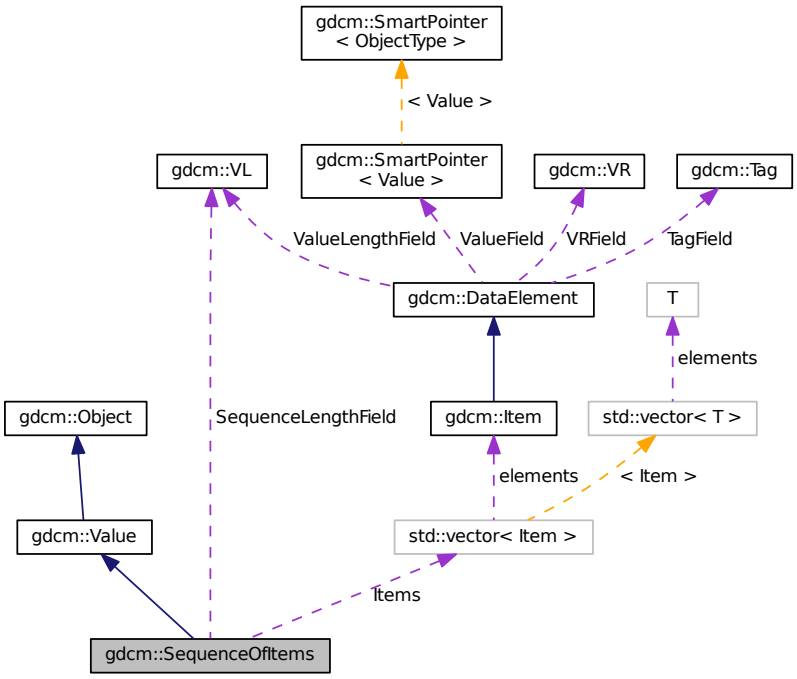
Class to represent a Sequence Of Items.

```
#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 ()

  remove all items within the sequence
- template<typename TDE >
  VL ComputeLength () const
- Iterator End ()
- ConstIterator End () const
- bool FindDataElement (const Tag &t) const
- const Item & GetItem (SizeType position) const
- Item & GetItem (SizeType position)
- VL GetLength () const

  Returns the SQ length, as read from disk.
- SizeType GetNumberOfItems () const
- bool IsUndefinedLength () const

  return if Value Length if of undefined length
- SequenceOfItems & operator= (const SequenceOfItems &val)
- bool operator== (const Value &val) const
- void Print (std::ostream &os) const
- template<typename TDE , typename TSwap >
  std::istream & Read (std::istream &is, bool readvalues=true)
- bool RemoveItemByIndex (const SizeType index)
- void SetLength (VL length)

  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

10.264.1 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.

10.264.2 Member Typedef Documentation

10.264.2.1 ConstIterator

```
typedef ItemVector::const_iterator gdcm::SequenceOfItems::ConstIterator
```

10.264.2.2 ItemVector

```
typedef std::vector< Item > gdcm::SequenceOfItems::ItemVector
```

10.264.2.3 Iterator

```
typedef ItemVector::iterator gdcm::SequenceOfItems::Iterator
```

10.264.2.4 SizeType

```
typedef ItemVector::size_type gdcm::SequenceOfItems::SizeType
```

10.264.3 Constructor & Destructor Documentation

10.264.3.1 SequenceOfItems()

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

constructor (UndefinedLength by default)

10.264.4 Member Function Documentation

10.264.4.1 AddItem()

```
void gdcm::SequenceOfItems::AddItem (
            Item const & item )
```

Appends an Item to the already added ones.

Examples:

Extracting_All_Resolution.cxx, Fake_Image_Using_Stream_Image_Writer.cxx, GenAllVR.cxx, GenLongSeqs.cxx, and GenSeqs.cxx.

10.264.4.2 AddNewUndefinedLengthItem()

```
Item& gdcm::SequenceOfItems::AddNewUndefinedLengthItem ( )
```

Appends an Item to the already added ones.

10.264.4.3 Begin() [1/2]

```
Iterator gdcm::SequenceOfItems::Begin ( )  [inline]
```

10.264.4.4 Begin() [2/2]

```
ConstIterator gdcm::SequenceOfItems::Begin ( ) const  [inline]
```

10.264.4.5 Clear()

```
void gdcm::SequenceOfItems::Clear ( )  [virtual]
```

remove all items within the sequence

Implements gdcm::Value.

10.264.4.6 ComputeLength()

```
template<typename TDE >
VL gdcm::SequenceOfItems::ComputeLength ( ) const
```

10.264.4.7 End() [1/2]

```
Iterator gdcm::SequenceOfItems::End ( )  [inline]
```

10.264.4.8 End() [2/2]

```
ConstIterator gdcm::SequenceOfItems::End ( ) const  [inline]
```

10.264.4.9 FindDataElement()

```
bool gdcm::SequenceOfItems::FindDataElement (
            const Tag & t ) const
```

10.264.4.10 GetItem() [1/2]

```
const Item& gdcm::SequenceOfItems::GetItem (
            SizeType position ) const
```

Examples:

ChangeSequenceUltrasound.cxx, DumpExamCard.cxx, DumpGEMSMovieGroup.cxx, DumpPhilipsECHO.cxx, DumpToshibaDTI.cxx, ExtractEncryptedContent.cxx, gdcmrtionplan.cxx, gdcmrtplan.cxx, GetSequenceUltrasound.cxx, GetSubSequenceData.cxx, LargeVRDSExplicit.cxx, and ReadAndDumpDICOMDIR.cxx.

10.264.4.11 GetItem() [2/2]

```
Item& gdcm::SequenceOfItems::GetItem (
            SizeType position )
```

10.264.4.12 GetLength()

```
VL gdcm::SequenceOfItems::GetLength ( ) const  [inline], [virtual]
```

Returns the SQ length, as read from disk.

Implements gdcm::Value.

10.264.4.13 GetNumberOfItems()

```
SizeType gdcm::SequenceOfItems::GetNumberOfItems ( ) const  [inline]
```

Examples:

ChangeSequenceUltrasound.cxx, DumpExamCard.cxx, DumpGEMSMovieGroup.cxx, DumpPhilipsECHO.cxx, DumpToshibaDTI.cxx, ExtractEncryptedContent.cxx, gdcmrtionplan.cxx, gdcmrtplan.cxx, GetSequenceUltrasound.cxx, GetSubSequenceData.cxx, and LargeVRDSExplicit.cxx.

10.264.4.14 IsUndefinedLength()

```
bool gdcm::SequenceOfItems::IsUndefinedLength ( ) const  [inline]
```

return if Value Length if of undefined length

10.264.4.15 New()

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

Examples:

NewSequence.cs.

10.264.4.16 operator=()

```
SequenceOfItems& gdcm::SequenceOfItems::operator= (
            const SequenceOfItems & val )  [inline]
```

References Items, and SequenceLengthField.

10.264.4.17 operator==()

```
bool gdcm::SequenceOfItems::operator== (
            const Value & val ) const  [inline], [virtual]
```

Implements gdcm::Value.

References Items, and SequenceLengthField.

10.264.4.18 Print()

```
void gdcm::SequenceOfItems::Print (
            std::ostream & os ) const  [inline], [virtual]
```

Reimplemented from gdcm::Object.

10.264.4.19 Read()

```
template<typename TDE , typename TSwap >
std::istream& gdcm::SequenceOfItems::Read (
            std::istream & is,
            bool readvalues = true )  [inline]
```

Examples:

ReadExplicitLengthSQIVR.cxx.

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().

10.264.4.20 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

10.264.4.21 SetLength()

```
void gdcm::SequenceOfItems::SetLength (
            VL length )  [inline], [virtual]
```

Sets the actual SQ length.

Implements gdcm::Value.

Examples:

ReadExplicitLengthSQIVR.cxx.

10.264.4.22 SetLengthToUndefined()

```
void gdcm::SequenceOfItems::SetLengthToUndefined ( )
```

Properly set the Sequence of Item to be undefined length.

Examples:

Extracting_All_Resolution.cxx, Fake_Image_Using_Stream_Image_Writer.cxx, GenAllVR.cxx, GenLongSeqs.cxx, and GenSeqs.cxx.

10.264.4.23 SetNumberOfItems()

```
void gdcm::SequenceOfItems::SetNumberOfItems (
            SizeType n )  [inline]
```

10.264.4.24 Write()

```
template<typename TDE , typename TSwap >
std::ostream const& gdcm::SequenceOfItems::Write (
            std::ostream & os ) const  [inline]
```

References gdcm::VL::Write(), and gdcm::Tag::Write().

10.264.5 Member Data Documentation

10.264.5.1 Items

```
ItemVector gdcm::SequenceOfItems::Items
```

Vector of Sequence Items.

Referenced by operator=(), and operator==().

10.264.5.2 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:

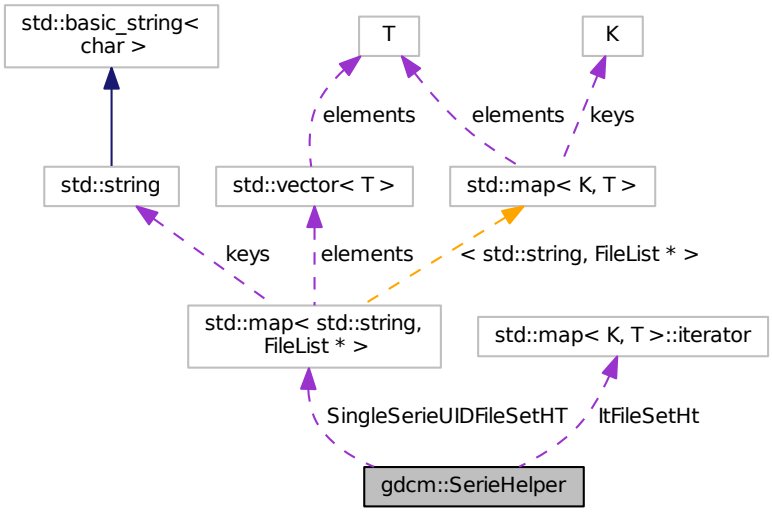
- gdcmSequenceOfItems.h

10.265 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 disapear soon, you've been warned.

`#include <gdcmSerieHelper.h>`

Collaboration diagram for gdcm::SerieHelper:



Classes

- struct Rule

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

- 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 ImagePositionPatientOrdering (FileList *fileSet)
- bool UserOrdering (FileList *fileSet)

Protected Attributes

- SingleSerieUIDFileSetmap::iterator ItFileSetHt
- SingleSerieUIDFileSetmap SingleSerieUIDFileSetHT

10.265.1 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 disapear soon, you've been warned.

Instead see ImageHelper or IPPSorter

10.265.2 Member Typedef Documentation

10.265.2.1 SerieRestrictions

```
typedef std::vector<Rule> gdcm::SerieHelper::SerieRestrictions  [protected]
```

10.265.2.2 SingleSerieUIDFileSetmap

```
typedef std::map<std::string, FileList *> gdcm::SerieHelper::SingleSerieUIDFileSetmap  [protected]
```

10.265.3 Constructor & Destructor Documentation

10.265.3.1 SerieHelper()

```
gdcm::SerieHelper::SerieHelper ( )
```

10.265.3.2 ~SerieHelper()

```
gdcm::SerieHelper::~SerieHelper ( )
```

10.265.4 Member Function Documentation

10.265.4.1 AddFile()

```
bool gdcm::SerieHelper::AddFile (
            FileWithName & header )  [protected]
```

10.265.4.2 AddFileName()

```
void gdcm::SerieHelper::AddFileName (
            std::string const & filename )  [protected]
```

10.265.4.3 AddRestriction() [1/3]

```
void gdcm::SerieHelper::AddRestriction (
            const std::string & tag )
```

10.265.4.4 AddRestriction() [2/3]

```
void gdcm::SerieHelper::AddRestriction (
            uint16_t group,
            uint16_t elem,
            std::string const & value,
            int op )
```

10.265.4.5 AddRestriction() [3/3]

```
void gdcm::SerieHelper::AddRestriction (
            const Tag & tag )  [protected]
```

10.265.4.6 Clear()

```
void gdcm::SerieHelper::Clear ( )
```

10.265.4.7 CreateDefaultUniqueSeriesIdentifier()

```
void gdcm::SerieHelper::CreateDefaultUniqueSeriesIdentifier ( )
```

10.265.4.8 CreateUniqueSeriesIdentifier()

```
std::string gdcm::SerieHelper::CreateUniqueSeriesIdentifier (
            File * inFile )
```

10.265.4.9 FileNameOrdering()

```
bool gdcm::SerieHelper::FileNameOrdering (
            FileList * fileList )  [protected]
```

10.265.4.10 GetFirstSingleSerieUIDFileSet()

```
FileList* gdcm::SerieHelper::GetFirstSingleSerieUIDFileSet ( )
```

10.265.4.11 GetNextSingleSerieUIDFileSet()

```
FileList* gdcm::SerieHelper::GetNextSingleSerieUIDFileSet ( )
```

10.265.4.12 ImagePositionPatientOrdering()

```
bool gdcm::SerieHelper::ImagePositionPatientOrdering (
            FileList * fileSet )  [protected]
```

10.265.4.13 OrderFileList()

```
void gdcm::SerieHelper::OrderFileList (
            FileList * fileSet )
```

10.265.4.14 SetDirectory()

```
void gdcm::SerieHelper::SetDirectory (
            std::string const & dir,
            bool recursive = false )
```

10.265.4.15 SetLoadMode()

```
void gdcm::SerieHelper::SetLoadMode (
            int  )  [inline]
```

10.265.4.16 SetUseSeriesDetails()

```
void gdcm::SerieHelper::SetUseSeriesDetails (
            bool useSeriesDetails )
```

10.265.4.17 UserOrdering()

```
bool gdcm::SerieHelper::UserOrdering (
            FileList * fileSet )  [protected]
```

10.265.5 Member Data Documentation

10.265.5.1 ItFileSetHt

```
SingleSerieUIDFileSetmap::iterator gdcm::SerieHelper::ItFileSetHt  [protected]
```

10.265.5.2 SingleSerieUIDFileSetHT

```
SingleSerieUIDFileSetmap gdcm::SerieHelper::SingleSerieUIDFileSetHT  [protected]
```

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

- gdcmSerieHelper.h

10.266 gdcm::Series Class Reference

Series.

```
#include <gdcmSeries.h>
```

Public Member Functions

- Series ()

10.266.1 Detailed Description

Series.

10.266.2 Constructor & Destructor Documentation

10.266.2.1 Series()

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

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

- gdcmSeries.h

10.267 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

10.267.1 Detailed Description

PS 3.4 Table B.3-1 SERVICE-CLASS-APPLICATION-INFORMATION (A-ASSOCIATE-RQ)

10.267.2 Constructor & Destructor Documentation

10.267.2.1 ServiceClassApplicationInformation()

```
gdcm::network::ServiceClassApplicationInformation::ServiceClassApplicationInformation ( )
```

10.267.3 Member Function Documentation

10.267.3.1 Print()

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

10.267.3.2 Read()

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

10.267.3.3 SetTuple()

```
void gdcm::network::ServiceClassApplicationInformation::SetTuple (
             uint8_t levelofsupport,
             uint8_t levelofdigitalsig,
             uint8_t elementcoercion )
```

10.267.3.4 Size()

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

10.267.3.5 Write()

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

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

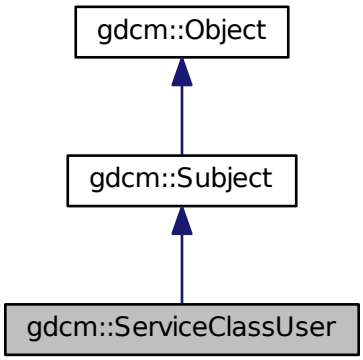
- gdcmServiceClassApplicationInformation.h

10.268 gdcm::ServiceClassUser Class Reference

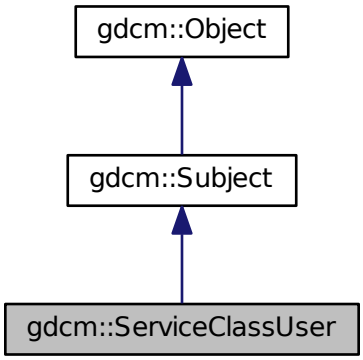
ServiceClassUser.

`#include <gdcmServiceClassUser.h>`

Inheritance diagram for gdcm::ServiceClassUser:



Collaboration diagram for gdcm::ServiceClassUser:



Public Member Functions

- ServiceClassUser ()
- ~ServiceClassUser ()
- 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.
- 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 (File const &file)
- bool SendStore (DataSet const &ds)

  Execute a C-STORE on a DataSet, the transfer syntax used will be Implicit.
- 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: instanciate a reference counted object

Additional Inherited Members

10.268.1 Detailed Description

ServiceClassUser.

Examples:

CStoreQtProgress.cxx.

10.268.2 Constructor & Destructor Documentation

10.268.2.1 ServiceClassUser()

```
gdcm::ServiceClassUser::ServiceClassUser ( )
```

Construct a SCU with default:

- hostname = localhost
- port = 104

10.268.2.2 ~ServiceClassUser()

```
gdcm::ServiceClassUser::~ServiceClassUser ( )
```

10.268.3 Member Function Documentation

10.268.3.1 GetAETitle()

```
const char* gdcm::ServiceClassUser::GetAETitle ( ) const
```

10.268.3.2 GetCalledAETitle()

```
const char* gdcm::ServiceClassUser::GetCalledAETitle ( ) const
```

10.268.3.3 GetTimeout()

```
double gdcm::ServiceClassUser::GetTimeout ( ) const
```

10.268.3.4 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.

10.268.3.5 IsPresentationContextAccepted()

```
bool gdcm::ServiceClassUser::IsPresentationContextAccepted (
            const PresentationContext & pc ) const
```

Return if the passed in presentation was accepted during association negotiation.

10.268.3.6 New()

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

for wrapped language: instanciate a reference counted object

10.268.3.7 SendEcho()

```
bool gdcm::ServiceClassUser::SendEcho ( )
```

C-ECHO.

10.268.3.8 SendFind()

```
bool gdcm::ServiceClassUser::SendFind (
            const BaseRootQuery * query,
            std::vector< DataSet > & retDatasets )
```

C-FIND a query, return result are in retDatasets.

10.268.3.9 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.

10.268.3.10 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.

10.268.3.11 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.

10.268.3.12 SendStore() [1/3]

```
bool gdcm::ServiceClassUser::SendStore (
            const char * filename )
```

Execute a C-STORE on file on disk, named filename.

Examples:

CStoreQtProgress.cxx.

10.268.3.13 SendStore() [2/3]

```
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.

10.268.3.14 SendStore() [3/3]

```
bool gdcm::ServiceClassUser::SendStore (
            DataSet const & ds )
```

Execute a C-STORE on a DataSet, the transfer syntax used will be Implicit.

10.268.3.15 SetAETitle()

```
void gdcm::ServiceClassUser::SetAETitle (
            const char * aetitle )
```

set calling ae title

10.268.3.16 SetCalledAETitle()

```
void gdcm::ServiceClassUser::SetCalledAETitle (
            const char * aetitle )
```

set called ae title

Examples:

CStoreQtProgress.cxx.

10.268.3.17 SetHostname()

```
void gdcm::ServiceClassUser::SetHostname (
            const char * hostname )
```

Set the name of the called hostname (hostname or IP address)

Examples:

CStoreQtProgress.cxx.

10.268.3.18 SetPort()

```
void gdcm::ServiceClassUser::SetPort (
            uint16_t port )
```

Set port of remote host (called application)

Examples:

CStoreQtProgress.cxx.

10.268.3.19 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)

10.268.3.20 SetPresentationContexts()

```
void gdcm::ServiceClassUser::SetPresentationContexts (
            std::vector< PresentationContext > const & pcs )
```

Set the Presentation Context used for the Association.

Examples:

CStoreQtProgress.cxx.

10.268.3.21 SetTimeout()

```
void gdcm::ServiceClassUser::SetTimeout (
            double t )
```

set/get Timeout

Examples:

CStoreQtProgress.cxx.

10.268.3.22 StartAssociation()

```
bool gdcm::ServiceClassUser::StartAssociation ( )
```

Start the association. Need to call SetPresentationContexts before.

Examples:

CStoreQtProgress.cxx.

10.268.3.23 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

10.269 gdcm::SHA1 Class Reference

Class for SHA1.

```
#include <gdcmSHA1.h>
```

Public Member Functions

- SHA1 ()
- ~SHA1 ()

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])

10.269.1 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

10.269.2 Constructor & Destructor Documentation

10.269.2.1 SHA1()

```
gdcm::SHA1::SHA1 ( )
```

10.269.2.2 ~SHA1()

```
gdcm::SHA1::~SHA1 ( )
```

10.269.3 Member Function Documentation

10.269.3.1 Compute()

```
static bool gdcm::SHA1::Compute (
             const char * buffer,
             unsigned long buf_len,
             char digest_str[20 *2+1] )  [static]
```

10.269.3.2 ComputeFile()

```
static bool gdcm::SHA1::ComputeFile (
             const char * filename,
             char digest_str[20 *2+1] )  [static]
```

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

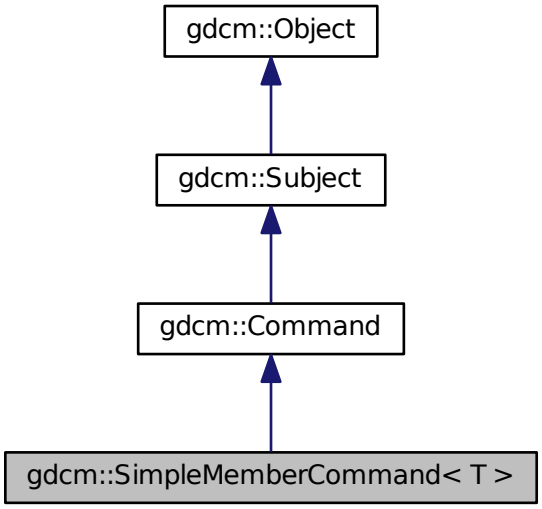
- gdcmSHA1.h

10.270 gdcm::SimpleMemberCommand< T > Class Template Reference

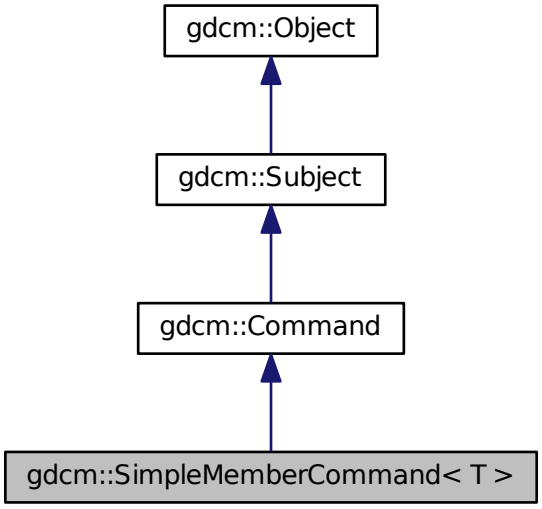
Command subclass that calls a pointer to a member function.

```
#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

- virtual void Execute (Subject *, const Event &)
- virtual void Execute (const Subject *, const Event &)
- void SetCallbackFunction (T *object, TMemberFunctionPointer memberFunction)

Static Public Member Functions

- static SmartPointer< SimpleMemberCommand > New ()

Protected Member Functions

- SimpleMemberCommand ()
- virtual ~SimpleMemberCommand ()

Protected Attributes

- TMemberFunctionPointer m_MemberFunction
- T * m_This

10.270.1 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.

10.270.2 Member Typedef Documentation

10.270.2.1 Self

```
template<typename T >
typedef SimpleMemberCommand gdcm::SimpleMemberCommand< T >::Self
```

Standard class typedefs.

10.270.2.2 TMemberFunctionPointer

```
template<typename T >
typedef void(T::* gdcm::SimpleMemberCommand< T >::TMemberFunctionPointer) ()
```

A method callback.

10.270.3 Constructor & Destructor Documentation

10.270.3.1 SimpleMemberCommand()

```
template<typename T >
gdcm::SimpleMemberCommand< T >::SimpleMemberCommand ( )  [inline], [protected]
```

10.270.3.2 ~SimpleMemberCommand()

```
template<typename T >
virtual gdcm::SimpleMemberCommand< T >::~SimpleMemberCommand ( )  [inline], [protected], [virtual]
```

10.270.4 Member Function Documentation

10.270.4.1 Execute() [1/2]

```
template<typename T >
virtual void gdcm::SimpleMemberCommand< T >::Execute (
            Subject * ,
            const Event &  )  [inline], [virtual]
```

Invoke the callback function.

Implements gdcm::Command.

10.270.4.2 Execute() [2/2]

```
template<typename T >
virtual void gdcm::SimpleMemberCommand< T >::Execute (
            const Subject * caller,
            const Event & event )  [inline], [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.

10.270.4.3 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.

10.270.4.4 SetCallbackFunction()

```
template<typename T >
void gdcm::SimpleMemberCommand< T >::SetCallbackFunction (
            T * object,
            TMemberFunctionPointer memberFunction )  [inline]
```

Specify the callback function.

10.270.5 Member Data Documentation

10.270.5.1 m_MemberFunction

```
template<typename T >
TMemberFunctionPointer gdcm::SimpleMemberCommand< T >::m_MemberFunction  [protected]
```

10.270.5.2 m_This

```
template<typename T >
T* gdcm::SimpleMemberCommand< T >::m_This  [protected]
```

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

- gdcmCommand.h

10.271 gdcm::SimpleSubjectWatcher Class Reference

SimpleSubjectWatcher.

```
#include <gdcmSimpleSubjectWatcher.h>
```

Public Member Functions

- SimpleSubjectWatcher (Subject *s, const char *comment="")
- virtual ~SimpleSubjectWatcher ()

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 ()

10.271.1 Detailed Description

SimpleSubjectWatcher.

This is a typical Subject Watcher class. It will observe all events.

Examples:

SimpleScanner.cxx.

10.271.2 Constructor & Destructor Documentation

10.271.2.1 SimpleSubjectWatcher()

```
gdcm::SimpleSubjectWatcher::SimpleSubjectWatcher (
            Subject * s,
            const char * comment = "" )
```

10.271.2.2 ~SimpleSubjectWatcher()

```
virtual gdcm::SimpleSubjectWatcher::~SimpleSubjectWatcher ( )  [virtual]
```

10.271.3 Member Function Documentation

10.271.3.1 EndFilter()

```
virtual void gdcm::SimpleSubjectWatcher::EndFilter ( )  [protected], [virtual]
```

10.271.3.2 ShowAbort()

```
virtual void gdcm::SimpleSubjectWatcher::ShowAbort ( )  [protected], [virtual]
```

10.271.3.3 ShowAnonymization()

```
virtual void gdcm::SimpleSubjectWatcher::ShowAnonymization (
            Subject * caller,
            const Event & evt )  [protected], [virtual]
```

10.271.3.4 ShowData()

```
virtual void gdcm::SimpleSubjectWatcher::ShowData (
            Subject * caller,
            const Event & evt )  [protected], [virtual]
```

10.271.3.5 ShowDataSet()

```
virtual void gdcm::SimpleSubjectWatcher::ShowDataSet (
            Subject * caller,
            const Event & evt )  [protected], [virtual]
```

10.271.3.6 ShowFileName()

```
virtual void gdcm::SimpleSubjectWatcher::ShowFileName (
            Subject * caller,
            const Event & evt )  [protected], [virtual]
```

Examples:

SimpleScanner.cxx.

10.271.3.7 ShowIteration()

```
virtual void gdcm::SimpleSubjectWatcher::ShowIteration ( )  [protected], [virtual]
```

10.271.3.8 ShowProgress()

```
virtual void gdcm::SimpleSubjectWatcher::ShowProgress (
            Subject * caller,
            const Event & evt )  [protected], [virtual]
```

10.271.3.9 StartFilter()

```
virtual void gdcm::SimpleSubjectWatcher::StartFilter ( )  [protected], [virtual]
```

10.271.3.10 TestAbortOff()

```
void gdcm::SimpleSubjectWatcher::TestAbortOff ( )  [protected]
```

10.271.3.11 TestAbortOn()

```
void gdcm::SimpleSubjectWatcher::TestAbortOn ( )  [protected]
```

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

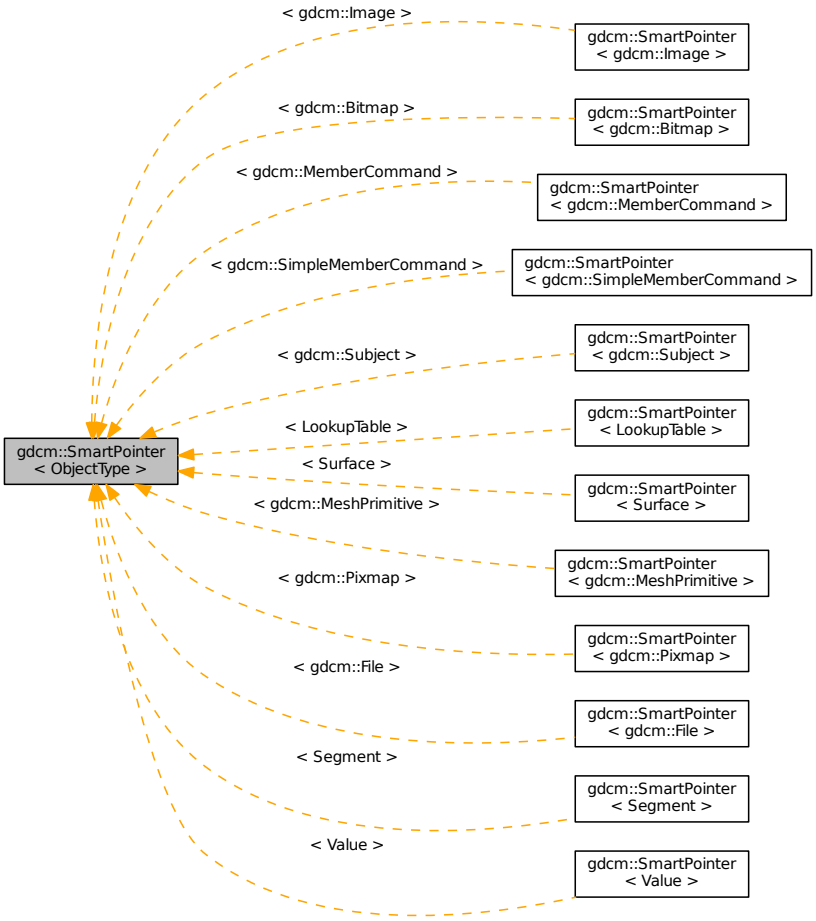
- gdcmSimpleSubjectWatcher.h

10.272 gdcm::SmartPointer< ObjectType > Class Template Reference

Class for Smart Pointer.

`#include <gdcmObject.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= (SmartPointer const &r)

  Overload operator assignment.
- SmartPointer & operator= (ObjectType *r)

  Overload operator assignment.
- SmartPointer & operator= (ObjectType const &r)

10.272.1 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:

ChangeSequenceUltrasound.cxx, CStoreQtProgress.cxx, DumpGEMSMovieGroup.cxx, DumpPhilipsECHO.cxx, DumpToshibaDTI.cxx, Extracting_All_Resolution.cxx, Fake_Image_Using_Stream_Image_Writer.cxx, FixBrokenJ2K.cxx, gdcmrtionplan.cxx, gdcmrtplan.cxx, GenAllVR.cxx, GenFakeIdentifyFile.cxx, GenLongSeqs.cxx, GenSeqs.cxx, GetSubSequenceData.cxx, LargeVRDSExplicit.cxx, ReadAndDumpDICOMDIR.cxx, ReadExplicitLengthSQIVR.cxx, and SimpleScanner.cxx.

10.272.2 Constructor & Destructor Documentation

10.272.2.1 SmartPointer() [1/4]

```
template<class ObjectType>
gdcm::SmartPointer< ObjectType >::SmartPointer ( )  [inline]
```

10.272.2.2 SmartPointer() [2/4]

```
template<class ObjectType>
gdcm::SmartPointer< ObjectType >::SmartPointer (
            const SmartPointer< ObjectType > & p )  [inline]
```

10.272.2.3 SmartPointer() [3/4]

```
template<class ObjectType>
gdcm::SmartPointer< ObjectType >::SmartPointer (
            ObjectType * p )  [inline]
```

10.272.2.4 SmartPointer() [4/4]

```
template<class ObjectType>
gdcm::SmartPointer< ObjectType >::SmartPointer (
            ObjectType const & p )  [inline]
```

10.272.2.5 ~SmartPointer()

```
template<class ObjectType>
gdcm::SmartPointer< ObjectType >::~SmartPointer ( )  [inline]
```

10.272.3 Member Function Documentation

10.272.3.1 GetPointer()

```
template<class ObjectType>
ObjectType* gdcm::SmartPointer< ObjectType >::GetPointer ( ) const  [inline]
```

Explicit function to retrieve the pointer.

10.272.3.2 operator ObjectType *()

```
template<class ObjectType>
gdcm::SmartPointer< ObjectType >::operator ObjectType * ( ) const  [inline]
```

Return pointer to object.

10.272.3.3 operator*()

```
template<class ObjectType>
ObjectType& gdcm::SmartPointer< ObjectType >::operator* ( ) const  [inline]
```

10.272.3.4 operator->()

```
template<class ObjectType>
ObjectType* gdcm::SmartPointer< ObjectType >::operator-> ( ) const  [inline]
```

Overload operator ->

10.272.3.5 operator=() [1/3]

```
template<class ObjectType>
SmartPointer& gdcm::SmartPointer< ObjectType >::operator= (
            SmartPointer< ObjectType > const & r )  [inline]
```

Overload operator assignment.

Referenced by gdcm::SmartPointer< Value >::operator=().

10.272.3.6 operator=() [2/3]

```
template<class ObjectType>
SmartPointer& gdcm::SmartPointer< ObjectType >::operator= (
            ObjectType * r )  [inline]
```

Overload operator assignment.

10.272.3.7 operator=() [3/3]

```
template<class ObjectType>
SmartPointer& gdcm::SmartPointer< ObjectType >::operator= (
            ObjectType const & r )  [inline]
```

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

- gdcmObject.h
- gdcmSmartPointer.h

10.273 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

10.273.1 Detailed Description

SOPClassExtendedNegociationSub.

PS 3.7 Table D.3-11 SOP CLASS EXTENDED NEGOTIATION SUB-ITEM FIELDS (A-ASSOCIATE-RQ and A-ASSOCIATE-AC)

10.273.2 Constructor & Destructor Documentation

10.273.2.1 SOPClassExtendedNegociationSub()

```
gdcm::network::SOPClassExtendedNegociationSub::SOPClassExtendedNegociationSub ( )
```

10.273.3 Member Function Documentation

10.273.3.1 Print()

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

10.273.3.2 Read()

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

10.273.3.3 SetTuple()

```
void gdcm::network::SOPClassExtendedNegociationSub::SetTuple (
            const char * uid,
            uint8_t levelofsupport = 3,
            uint8_t levelofdigitalsig = 0,
            uint8_t elementcoercion = 2 )
```

10.273.3.4 Size()

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

10.273.3.5 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

10.274 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 ()

10.274.1 Detailed Description

Class convert a class SOP Class UID into IOD.

Reference PS 3.4 Table B.5-1 STANDARD SOP CLASSES

10.274.2 Member Typedef Documentation

10.274.2.1 const

```
typedef const char* gdcm::SOPClassUIDToIOD::const(SOPClassUIDToIODType)[2]
```

10.274.3 Member Function Documentation

10.274.3.1 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.

10.274.3.2 GetIODFromSOPClassUID()

```
static const char* gdcm::SOPClassUIDToIOD::GetIODFromSOPClassUID (
            const char * sopclassuid )  [static]
```

10.274.3.3 GetNumberOfSOPClassToIOD()

```
static unsigned int gdcm::SOPClassUIDToIOD::GetNumberOfSOPClassToIOD ( )  [static]
```

Return the number of SOP Class UID listed internally.

10.274.3.4 GetSOPClassUIDFromIOD()

```
static const char* gdcm::SOPClassUIDToIOD::GetSOPClassUIDFromIOD (
            const char * iod )  [static]
```

10.274.3.5 GetSOPClassUIDToIOD()

```
static SOPClassUIDToIODType& gdcm::SOPClassUIDToIOD::GetSOPClassUIDToIOD (
            unsigned int i )  [static]
```

10.274.3.6 GetSOPClassUIDToIODs()

```
static SOPClassUIDToIODType* gdcm::SOPClassUIDToIOD::GetSOPClassUIDToIODs ( )  [static]
```

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

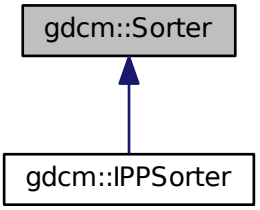
- gdcmSOPClassUIDToIOD.h

10.275 gdcm::Sorter Class Reference

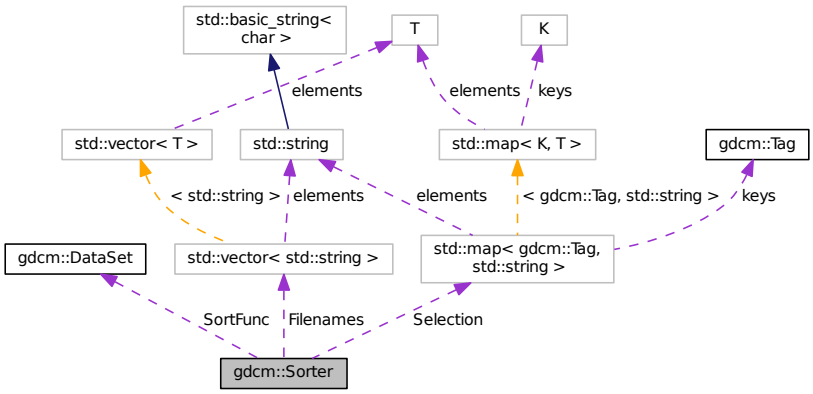
Sorter.

```
#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)
- 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

Friends

- std::ostream & operator<< (std::ostream &_os, const Sorter &s)

10.275.1 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 everytime you call Sort, all files specified as input paramater are *read*

See also

Scanner

Examples:

SortImage.cxx, and VolumeSorter.cxx.

10.275.2 Member Typedef Documentation

10.275.2.1 SelectionMap

```
typedef std::map<Tag,std::string> gdcm::Sorter::SelectionMap  [protected]
```

10.275.2.2 SortFunction

```
typedef bool(* gdcm::Sorter::SortFunction) (DataSet const &, DataSet const &)
```

Set the sort function which compares one dataset to the other.

10.275.3 Constructor & Destructor Documentation

10.275.3.1 Sorter()

```
gdcm::Sorter::Sorter ( )
```

10.275.3.2 ~Sorter()

```
virtual gdcm::Sorter::~Sorter ( )  [virtual]
```

10.275.4 Member Function Documentation

10.275.4.1 AddSelect()

```
bool gdcm::Sorter::AddSelect (
            Tag const & tag,
            const char * value )
```

UNSUPPORTED FOR NOW.

10.275.4.2 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, gdcmorthoplanes.cxx, reslicesphere.cxx, SortImage.cxx, and VolumeSorter.cxx.

10.275.4.3 Print()

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

Print.

Examples:

gdcmorthoplanes.cxx, SortImage.cxx, and VolumeSorter.cxx.

Referenced by gdcm::operator<<().

10.275.4.4 SetSortFunction()

```
void gdcm::Sorter::SetSortFunction (
            SortFunction f )
```

Examples:

SortImage.cxx, and VolumeSorter.cxx.

10.275.4.5 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.

10.275.4.6 StableSort()

```
virtual bool gdcm::Sorter::StableSort (
            std::vector< std::string > const & filenames )  [virtual]
```

Examples:

SortImage.cxx, and VolumeSorter.cxx.

10.275.5 Friends And Related Function Documentation

10.275.5.1 operator<<

```
std::ostream& operator<< (
            std::ostream & _os,
            const Sorter & s )  [friend]
```

10.275.6 Member Data Documentation

10.275.6.1 Filenames

```
std::vector<std::string> gdcm::Sorter::Filenames  [protected]
```

10.275.6.2 Selection

```
std::map<Tag,std::string> gdcm::Sorter::Selection  [protected]
```

10.275.6.3 SortFunc

```
SortFunction gdcm::Sorter::SortFunc  [protected]
```

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

- gdcmSorter.h

10.276 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)

10.276.1 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`

10.276.2 Member Enumeration Documentation

10.276.2.1 SpacingType

enum gdcm::Spacing::SpacingType

Enumerator

DETECTOR	
MAGNIFIED	
CALIBRATED	
UNKNOWN	

10.276.3 Constructor & Destructor Documentation

10.276.3.1 Spacing()

```
gdcm::Spacing::Spacing ( )
```

10.276.3.2 ~Spacing()

```
gdcm::Spacing::~Spacing ( )
```

10.276.4 Member Function Documentation

10.276.4.1 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

10.277 gdcm::Spectroscopy Class Reference

Spectroscopy class.

```
#include <gdcmSpectroscopy.h>
```

Public Member Functions

- Spectroscopy ()

10.277.1 Detailed Description

Spectroscopy class.

10.277.2 Constructor & Destructor Documentation

10.277.2.1 Spectroscopy()

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

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

- gdcmSpectroscopy.h

10.278 gdcm::SplitMosaicFilter Class Reference

SplitMosaicFilter class.

```
#include <gdcmSplitMosaicFilter.h>
```

Public Member Functions

- SplitMosaicFilter ()
- ~SplitMosaicFilter ()
- bool ComputeMOSAICDimensions (unsigned int dims[3])
- File & GetFile ()
- const File & GetFile () const
- const Image & GetImage () const
- Image & GetImage ()
- void SetFile (const File &f)
- void SetImage (const Image &image)
- bool Split ()

  Split the SIEMENS MOSAIC image.

10.278.1 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

10.278.2 Constructor & Destructor Documentation

10.278.2.1 SplitMosaicFilter()

```
gdcm::SplitMosaicFilter::SplitMosaicFilter ( )
```

10.278.2.2 ~SplitMosaicFilter()

```
gdcm::SplitMosaicFilter::~SplitMosaicFilter ( )
```

10.278.3 Member Function Documentation

10.278.3.1 ComputeMOSAICDimensions()

```
bool gdcm::SplitMosaicFilter::ComputeMOSAICDimensions (
            unsigned int dims[3] )
```

Compute the new dimensions according to private information stored in the MOSAIC header.

10.278.3.2 GetFile() [1/2]

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

10.278.3.3 GetFile() [2/2]

```
const File& gdcm::SplitMosaicFilter::GetFile ( ) const  [inline]
```

10.278.3.4 GetImage() [1/2]

```
const Image& gdcm::SplitMosaicFilter::GetImage ( ) const  [inline]
```

10.278.3.5 GetImage() [2/2]

```
Image& gdcm::SplitMosaicFilter::GetImage ( )  [inline]
```

10.278.3.6 SetFile()

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

10.278.3.7 SetImage()

```
void gdcm::SplitMosaicFilter::SetImage (
            const Image & image )
```

10.278.3.8 Split()

```
bool gdcm::SplitMosaicFilter::Split ( )
```

Split the SIEMENS MOSAIC image.

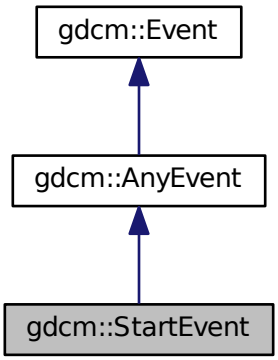
The documentation for this class was generated from the following file:

- gdcmSplitMosaicFilter.h

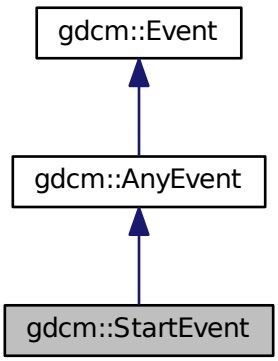
10.279 gdcm::StartEvent Class Reference

```
#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

10.280 gdcm::static_assert_test< x > Struct Template Reference

`#include <gdcmStaticAssert.h>`

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

- gdcmStaticAssert.h

10.281 gdcm::STATIC_ASSERTION_FAILURE< x > Struct Template Reference

`#include <gdcmStaticAssert.h>`

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

- gdcmStaticAssert.h

10.282 gdcm::STATIC_ASSERTION_FAILURE< true > Struct Template Reference

`#include <gdcmStaticAssert.h>`

Public Types

- enum { value = 1 }

10.282.1 Member Enumeration Documentation

10.282.1.1 anonymous enum

`anonymous enum`

Enumerator

value	

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

- gdcmStaticAssert.h

10.283 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)

10.283.1 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:

StreamImageReaderTest.cxx.

10.283.2 Constructor & Destructor Documentation

10.283.2.1 StreamImageReader()

```
gdcm::StreamImageReader::StreamImageReader ( )
```

10.283.2.2 ~StreamImageReader()

```
virtual gdcm::StreamImageReader::~StreamImageReader ( )  [virtual]
```

10.283.3 Member Function Documentation

10.283.3.1 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.

10.283.3.2 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:

StreamImageReaderTest.cxx.

10.283.3.3 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:

StreamImageReaderTest.cxx.

10.283.3.4 GetDimensionsValueForResolution()

```
std::vector<unsigned int> gdcm::StreamImageReader::GetDimensionsValueForResolution (
             unsigned int )
```

10.283.3.5 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:

StreamImageReaderTest.cxx.

10.283.3.6 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:

StreamImageReaderTest.cxx.

10.283.3.7 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:

StreamImageReaderTest.cxx.

10.283.3.8 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.

10.283.3.9 SetStream()

```
void gdcm::StreamImageReader::SetStream (
            std::istream & inStream )
```

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

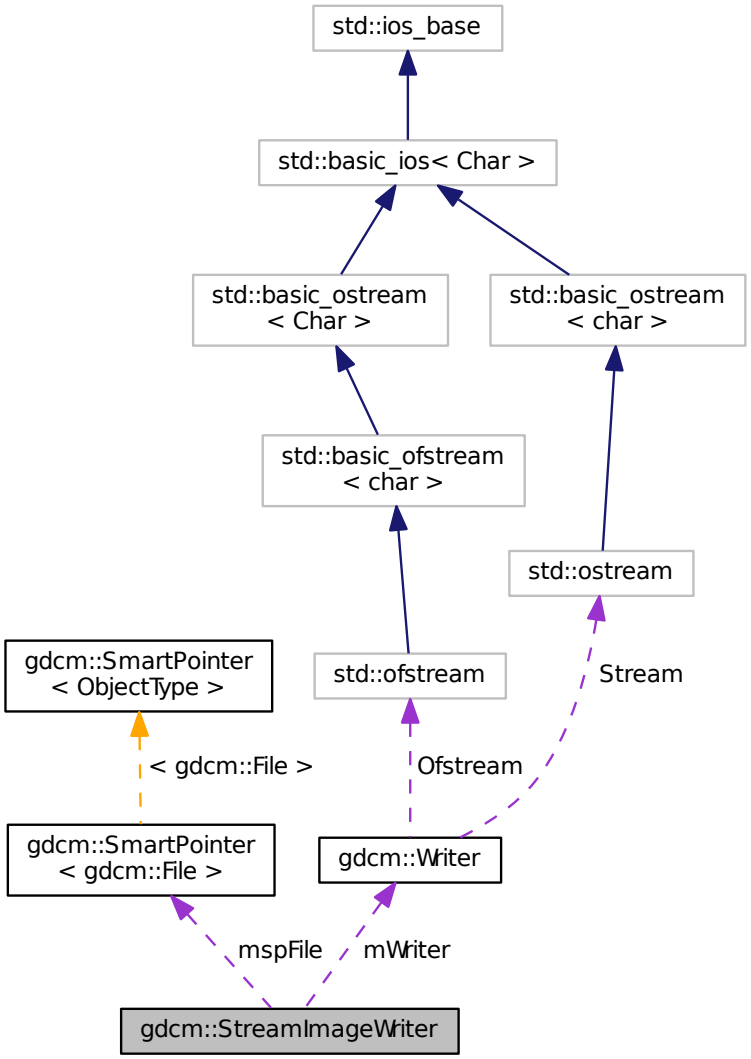
- gdcmStreamImageReader.h

10.284 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

10.284.1 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.

10.284.2 Constructor & Destructor Documentation

10.284.2.1 StreamImageWriter()

```
gdcm::StreamImageWriter::StreamImageWriter ( )
```

10.284.2.2 ~StreamImageWriter()

```
virtual gdcm::StreamImageWriter::~StreamImageWriter ( )  [virtual]
```

10.284.3 Member Function Documentation

10.284.3.1 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.

10.284.3.2 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.

10.284.3.3 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.

10.284.3.4 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.

10.284.3.5 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.

10.284.3.6 SetStream()

```
void gdcm::StreamImageWriter::SetStream (
            std::ostream & inStream )
```

Examples:

Extracting_All_Resolution.cxx, Fake_Image_Using_Stream_Image_Writer.cxx, and StreamImageReaderTest.cxx.

10.284.3.7 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 accomodate 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.

10.284.3.8 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.

10.284.3.9 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

10.284.3.10 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.

10.284.4 Member Data Documentation

10.284.4.1 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.

10.284.4.2 mElementOffsets1

```
int gdcm::StreamImageWriter::mElementOffsets1  [protected]
```

10.284.4.3 mspFile

```
SmartPointer<File> gdcm::StreamImageWriter::mspFile  [protected]
```

10.284.4.4 mWriter

```
Writer gdcm::StreamImageWriter::mWriter  [protected]
```

10.284.4.5 mXMax

```
uint16_t gdcm::StreamImageWriter::mXMax  [protected]
```

10.284.4.6 mXMin

```
uint16_t gdcm::StreamImageWriter::mXMin  [protected]
```

10.284.4.7 mYMax

```
uint16_t gdcm::StreamImageWriter::mYMax  [protected]
```

10.284.4.8 mYMin

```
uint16_t gdcm::StreamImageWriter::mYMin  [protected]
```

10.284.4.9 mZMax

```
uint16_t gdcm::StreamImageWriter::mZMax  [protected]
```

10.284.4.10 mZMin

```
uint16_t gdcm::StreamImageWriter::mZMin  [protected]
```

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

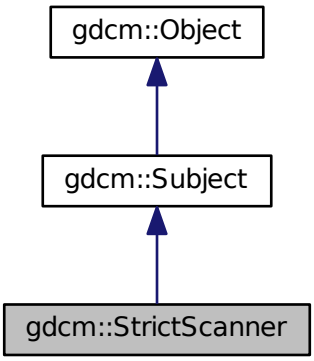
- gdcmStreamImageWriter.h

10.285 gdcm::StrictScanner Class Reference

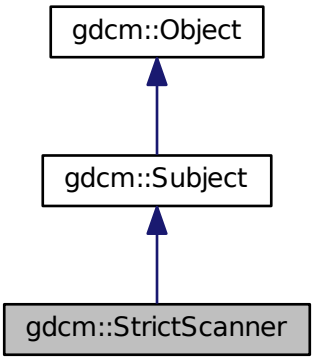
StrictScanner.

`#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 ()
- 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

  Print result.
- bool Scan (Directory::FilenamesType const &filenames)

  Start the scan !

Static Public Member Functions

- static SmartPointer< StrictScanner > New ()

  for wrapped language: instanciate a reference counted object

Protected Member Functions

- void ProcessPublicTag (StringFilter &sf, const char *filename)

Friends

- std::ostream & operator<< (std::ostream &_os, const StrictScanner &s)

10.285.1 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 mimimum 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:

SimpleScanner.cxx.

10.285.2 Member Typedef Documentation

10.285.2.1 ConstIterator

```
typedef MappingType::const_iterator gdcm::StrictScanner::ConstIterator
```

10.285.2.2 MappingType

```
typedef std::map<const char *,TagToValue, ltstr> gdcm::StrictScanner::MappingType
```

10.285.2.3 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)

10.285.2.4 TagToValueValueType

```
typedef TagToValue::value_type gdcm::StrictScanner::TagToValueValueType
```

10.285.2.5 ValuesType

```
typedef std::set< std::string > gdcm::StrictScanner::ValuesType
```

10.285.3 Constructor & Destructor Documentation

10.285.3.1 StrictScanner()

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

10.285.3.2 ~StrictScanner()

```
gdcm::StrictScanner::~StrictScanner ( )
```

10.285.4 Member Function Documentation

10.285.4.1 AddPrivateTag()

```
void gdcm::StrictScanner::AddPrivateTag (
            PrivateTag const & t )
```

10.285.4.2 AddSkipTag()

```
void gdcm::StrictScanner::AddSkipTag (
            Tag const & t )
```

Add a tag that will need to be skipped. Those are root level skip tags.

10.285.4.3 AddTag()

```
void gdcm::StrictScanner::AddTag (
            Tag const & t )
```

Add a tag that will need to be read. Those are root level skip tags.

Examples:

SimpleScanner.cxx.

10.285.4.4 Begin()

```
ConstIterator gdcm::StrictScanner::Begin ( ) const  [inline]
```

10.285.4.5 ClearSkipTags()

```
void gdcm::StrictScanner::ClearSkipTags ( )
```

10.285.4.6 ClearTags()

```
void gdcm::StrictScanner::ClearTags ( )
```

10.285.4.7 End()

```
ConstIterator gdcm::StrictScanner::End ( ) const  [inline]
```

10.285.4.8 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'

10.285.4.9 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'

10.285.4.10 GetFilenames()

```
Directory::FilenamesType const& gdcm::StrictScanner::GetFilenames ( ) const  [inline]
```

10.285.4.11 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

10.285.4.12 GetMapping()

```
TagToValue const& gdcm::StrictScanner::GetMapping (
            const char * filename ) const
```

Get the std::map mapping filenames to value for file 'filename'.

Examples:

SimpleScanner.cxx.

10.285.4.13 GetMappingFromTagToValue()

```
TagToValue const& gdcm::StrictScanner::GetMappingFromTagToValue (
            Tag const & t,
            const char * value ) const
```

See GetFilenameFromTagToValue(). This is simply GetFilenameFromTagToValue followed.

10.285.4.14 GetMappings()

```
MappingType const& gdcm::StrictScanner::GetMappings ( ) const  [inline]
```

Mappings are the mapping from a particular tag to the map, mapping filename to value:

10.285.4.15 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)

10.285.4.16 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 !

10.285.4.17 GetValues() [1/2]

```
ValuesType const& gdcm::StrictScanner::GetValues ( ) const  [inline]
```

Get all the values found (in lexicographic order)

10.285.4.18 GetValues() [2/2]

```
ValuesType gdcm::StrictScanner::GetValues (
            Tag const & t ) const
```

Get all the values found (in lexicographic order) associated with Tag 't'.

10.285.4.19 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:

SimpleScanner.cxx.

10.285.4.20 New()

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

for wrapped language: instanciate a reference counted object

10.285.4.21 Print()

```
void gdcm::StrictScanner::Print (
            std::ostream & os ) const  [virtual]
```

Print result.

Reimplemented from gdcm::Object.

Referenced by gdcm::operator<<().

10.285.4.22 ProcessPublicTag()

```
void gdcm::StrictScanner::ProcessPublicTag (
            StringFilter & sf,
            const char * filename )  [protected]
```

10.285.4.23 Scan()

```
bool gdcm::StrictScanner::Scan (
            Directory::FilenamesType const & filenames )
```

Start the scan !

Examples:

SimpleScanner.cxx.

10.285.5 Friends And Related Function Documentation

10.285.5.1 operator<<

```
std::ostream& operator<< (
            std::ostream & _os,
            const StrictScanner & s )  [friend]
```

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

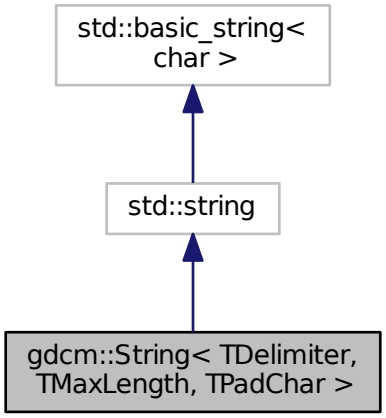
- gdcmStrictScanner.h

10.286 gdcm::String< TDelimiter, TMaxLength, TPadChar > Class Template Reference

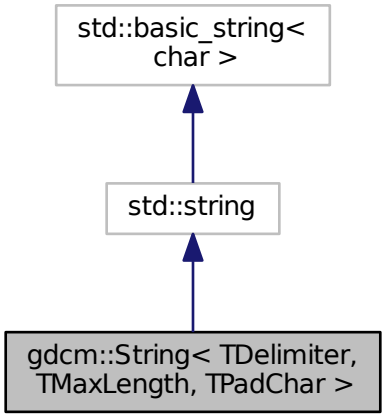
String.

`#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 value_type *s)
- String (const value_type *s, size_type n)
- String (const std::string &s, size_type pos=0, size_type n=npos)
- 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)

10.286.1 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. Noone actually respect the max length TPadChar is the string padding (0 or space)

10.286.2 Member Typedef Documentation

10.286.2.1 const_iterator

```
template<char TDelimiter = '\\', unsigned int TMaxLength = 64, char TPadChar = ' '>
typedef std::string::const_iterator gdcm::String< TDelimiter, TMaxLength, TPadChar >::const_↩
iterator
```

10.286.2.2 const_reference

```
template<char TDelimiter = '\\', unsigned int TMaxLength = 64, char TPadChar = ' '>
typedef std::string::const_reference gdcm::String< TDelimiter, TMaxLength, TPadChar >::const_↩
reference
```

10.286.2.3 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
```

10.286.2.4 difference_type

```
template<char TDelimiter = '\\', unsigned int TMaxLength = 64, char TPadChar = ' '>
typedef std::string::difference_type gdcm::String< TDelimiter, TMaxLength, TPadChar >::difference↩
_type
```

10.286.2.5 iterator

```
template<char TDelimiter = '\\', unsigned int TMaxLength = 64, char TPadChar = ' '>
typedef std::string::iterator gdcm::String< TDelimiter, TMaxLength, TPadChar >::iterator
```

10.286.2.6 pointer

```
template<char TDelimiter = '\\', unsigned int TMaxLength = 64, char TPadChar = ' '>
typedef std::string::pointer gdcm::String< TDelimiter, TMaxLength, TPadChar >::pointer
```

10.286.2.7 reference

```
template<char TDelimiter = '\\', unsigned int TMaxLength = 64, char TPadChar = ' '>
typedef std::string::reference gdcm::String< TDelimiter, TMaxLength, TPadChar >::reference
```

10.286.2.8 reverse_iterator

```
template<char TDelimiter = '\\', unsigned int TMaxLength = 64, char TPadChar = ' '>
typedef std::string::reverse_iterator gdcm::String< TDelimiter, TMaxLength, TPadChar >::reverse↩
_iterator
```

10.286.2.9 size_type

```
template<char TDelimiter = '\\', unsigned int TMaxLength = 64, char TPadChar = ' '>
typedef std::string::size_type gdcm::String< TDelimiter, TMaxLength, TPadChar >::size_type
```

10.286.2.10 value_type

```
template<char TDelimiter = '\\', unsigned int TMaxLength = 64, char TPadChar = ' '>
typedef std::string::value_type gdcm::String< TDelimiter, TMaxLength, TPadChar >::value_type
```

10.286.3 Constructor & Destructor Documentation

10.286.3.1 String() [1/4]

```
template<char TDelimiter = '\\', unsigned int TMaxLength = 64, char TPadChar = ' '>
gdcm::String< TDelimiter, TMaxLength, TPadChar >::String ( )  [inline]
```

String constructors.

10.286.3.2 String() [2/4]

```
template<char TDelimiter = '\\', unsigned int TMaxLength = 64, char TPadChar = ' '>
gdcm::String< TDelimiter, TMaxLength, TPadChar >::String (
            const value_type * s )  [inline]
```

10.286.3.3 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]
```

10.286.3.4 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]
```

10.286.4 Member Function Documentation

10.286.4.1 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::String< TDelimiter, TMaxLength, TPadChar >::Truncate().

10.286.4.2 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:

10.286.4.3 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...

10.286.4.4 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]
```

10.286.4.5 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

10.287 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, VL const &vl)
- 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 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

10.287.1 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:

ReadAndPrintAttributes.cxx.

10.287.2 Constructor & Destructor Documentation

10.287.2.1 StringFilter()

```
gdcm::StringFilter::StringFilter ( )
```

10.287.2.2 ~StringFilter()

```
gdcm::StringFilter::~StringFilter ( )
```

10.287.3 Member Function Documentation

10.287.3.1 ExecuteQuery() [1/2]

```
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

10.287.3.2 ExecuteQuery() [2/2]

```
bool gdcm::StringFilter::ExecuteQuery (
            std::string const & query,
            DataSet const & ds,
            std::string & value ) const  [protected]
```

10.287.3.3 FromString() [1/2]

```
std::string gdcm::StringFilter::FromString (
            const Tag & t,
            const char * value,
            VL const & vl )
```

10.287.3.4 FromString() [2/2]

```
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)

10.287.3.5 GetFile() [1/2]

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

10.287.3.6 GetFile() [2/2]

```
const File& gdcm::StringFilter::GetFile ( ) const  [inline]
```

10.287.3.7 SetDicts()

```
void gdcm::StringFilter::SetDicts (
            const Dicts & dicts )
```

Allow user to pass in there own dicts.

10.287.3.8 SetFile()

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

Set/Get File.

Examples:

ReadAndPrintAttributes.cxx, and SimplePrintPatientName.cs.

10.287.3.9 ToString() [1/2]

```
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:

ReadAndPrintAttributes.cxx.

10.287.3.10 ToString() [2/2]

```
std::string gdcm::StringFilter::ToString (
            const Tag & t ) const
```

Directly from a Tag:

10.287.3.11 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.

10.287.3.12 ToStringPair() [2/3]

```
std::pair<std::string, std::string> gdcm::StringFilter::ToStringPair (
            const Tag & t ) const
```

Directly from a Tag:

10.287.3.13 ToStringPair() [3/3]

```
std::pair<std::string, std::string> gdcm::StringFilter::ToStringPair (
            const Tag & t,
            DataSet const & ds ) const  [protected]
```

10.287.3.14 UseDictAlways()

```
void gdcm::StringFilter::UseDictAlways (
            bool  )  [inline]
```

References GDCM_LEGACY.

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

- gdcmStringFilter.h

10.288 gdcm::Study Class Reference

Study.

```
#include <gdcmStudy.h>
```

Public Member Functions

- Study ()

10.288.1 Detailed Description

Study.

10.288.2 Constructor & Destructor Documentation

10.288.2.1 Study()

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

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

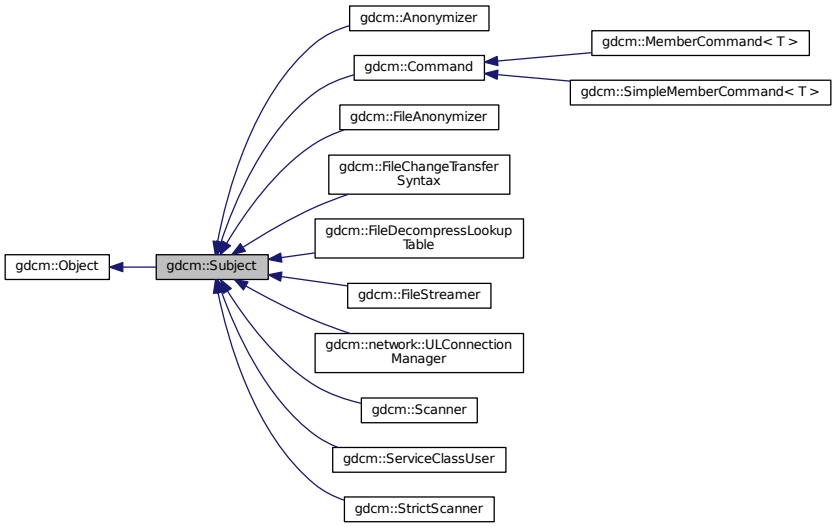
- gdcmStudy.h

10.289 gdcm::Subject Class Reference

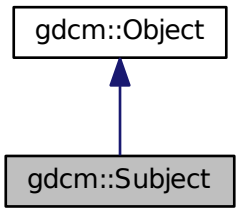
Subject.

```
#include <gdcmSubject.h>
```

Inheritance diagram for gdcm::Subject:



Collaboration diagram for gdcm::Subject:



Public Member Functions

- Subject ()
- ~Subject ()
- 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

10.289.1 Detailed Description

Subject.

See also

Command Event

Examples:

SimpleScanner.cxx.

10.289.2 Constructor & Destructor Documentation

10.289.2.1 Subject()

```
gdcm::Subject::Subject ( )
```

10.289.2.2 ~Subject()

```
gdcm::Subject::~Subject ( )
```

10.289.3 Member Function Documentation

10.289.3.1 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

10.289.3.2 AddObserver() [2/2]

```
unsigned long gdcm::Subject::AddObserver (
            const Event & event,
            Command *  ) const
```

10.289.3.3 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 asign 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.

10.289.3.4 HasObserver()

```
bool gdcm::Subject::HasObserver (
            const Event & event ) const
```

Return true if an observer is registered for this event.

10.289.3.5 InvokeEvent() [1/2]

```
void gdcm::Subject::InvokeEvent (
            const Event &  )
```

Call Execute on all the Commands observing this event id.

10.289.3.6 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.

10.289.3.7 RemoveAllObservers()

```
void gdcm::Subject::RemoveAllObservers ( )
```

Remove all observers .

10.289.3.8 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:

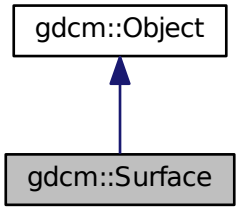
- gdcmSubject.h

10.290 gdcm::Surface Class Reference

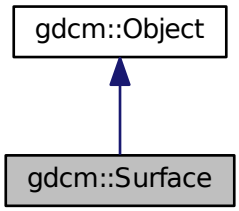
This class defines a SURFACE IE.

```
#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 ()
- virtual ~Surface ()
- SegmentHelper::BasicCodedEntry const & GetAlgorithmFamily () const
- SegmentHelper::BasicCodedEntry & GetAlgorithmFamily ()
- 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 const & GetMeshPrimitive () const
- MeshPrimitive & GetMeshPrimitive ()
- unsigned long GetNumberOfSurfacePoints () const
- unsigned long GetNumberOfVectors () const
- const DataElement & GetPointCoordinatesData () const

- DataElement & GetPointCoordinatesData ()
- const float * GetPointPositionAccuracy () const
- const float * GetPointsBoundingBoxCoordinates () const
- SegmentHelper::BasicCodedEntry const & GetProcessingAlgorithm () const
- SegmentHelper::BasicCodedEntry & GetProcessingAlgorithm ()
- 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
- const DataElement & GetVectorCoordinateData () const
- DataElement & GetVectorCoordinateData ()
- 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 unsigned short vl[3])
- void SetRecommendedDisplayCIELabValue (const unsigned short vl, const unsigned int idx=0)
- void SetRecommendedDisplayCIELabValue (const std::vector< unsigned short > &vl)
- 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

10.290.1 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

10.290.2 Member Enumeration Documentation

10.290.2.1 STATES

`enum gdcm::Surface::STATES`

Enumerator

NO	
YES	
UNKNOWN	
STATES_END	

10.290.2.2 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	

10.290.3 Constructor & Destructor Documentation

10.290.3.1 Surface()

```
gdcm::Surface::Surface ( )
```

10.290.3.2 ~Surface()

```
virtual gdcm::Surface::~Surface ( )  [virtual]
```

10.290.4 Member Function Documentation

10.290.4.1 GetAlgorithmFamily() [1/2]

```
SegmentHelper::BasicCodedEntry const& gdcm::Surface::GetAlgorithmFamily ( ) const
```

10.290.4.2 GetAlgorithmFamily() [2/2]

```
SegmentHelper::BasicCodedEntry& gdcm::Surface::GetAlgorithmFamily ( )
```

10.290.4.3 GetAlgorithmName()

```
const char* gdcm::Surface::GetAlgorithmName ( ) const
```

10.290.4.4 GetAlgorithmVersion()

```
const char* gdcm::Surface::GetAlgorithmVersion ( ) const
```

10.290.4.5 GetAxisOfRotation()

```
const float* gdcm::Surface::GetAxisOfRotation ( ) const
```

Note

Pointer is null if undefined

10.290.4.6 GetCenterOfRotation()

```
const float* gdcm::Surface::GetCenterOfRotation ( ) const
```

Note

Pointer is null if undefined

10.290.4.7 GetFiniteVolume()

```
STATES gdcm::Surface::GetFiniteVolume ( ) const
```

10.290.4.8 GetManifold()

```
STATES gdcm::Surface::GetManifold ( ) const
```

10.290.4.9 GetMaximumPointDistance()

```
float gdcm::Surface::GetMaximumPointDistance ( ) const
```

10.290.4.10 GetMeanPointDistance()

```
float gdcm::Surface::GetMeanPointDistance ( ) const
```

10.290.4.11 GetMeshPrimitive() [1/2]

```
MeshPrimitive const& gdcm::Surface::GetMeshPrimitive ( ) const
```

10.290.4.12 GetMeshPrimitive() [2/2]

```
MeshPrimitive& gdcm::Surface::GetMeshPrimitive ( )
```

10.290.4.13 GetNumberOfSurfacePoints()

```
unsigned long gdcm::Surface::GetNumberOfSurfacePoints ( ) const
```

10.290.4.14 GetNumberOfVectors()

```
unsigned long gdcm::Surface::GetNumberOfVectors ( ) const
```

10.290.4.15 GetPointCoordinatesData() [1/2]

```
const DataElement& gdcm::Surface::GetPointCoordinatesData ( ) const
```

10.290.4.16 GetPointCoordinatesData() [2/2]

```
DataElement& gdcm::Surface::GetPointCoordinatesData ( )
```

10.290.4.17 GetPointPositionAccuracy()

```
const float* gdcm::Surface::GetPointPositionAccuracy ( ) const
```

Note

Pointer is null if undefined

10.290.4.18 GetPointsBoundingBoxCoordinates()

```
const float* gdcm::Surface::GetPointsBoundingBoxCoordinates ( ) const
```

Note

Pointer is null if undefined

10.290.4.19 GetProcessingAlgorithm() [1/2]

```
SegmentHelper::BasicCodedEntry const& gdcm::Surface::GetProcessingAlgorithm ( ) const
```

10.290.4.20 GetProcessingAlgorithm() [2/2]

```
SegmentHelper::BasicCodedEntry& gdcm::Surface::GetProcessingAlgorithm ( )
```

10.290.4.21 GetRecommendedDisplayCIELabValue() [1/2]

```
const unsigned short* gdcm::Surface::GetRecommendedDisplayCIELabValue ( ) const
```

10.290.4.22 GetRecommendedDisplayCIELabValue() [2/2]

```
unsigned short gdcm::Surface::GetRecommendedDisplayCIELabValue (
            const unsigned int idx ) const
```

10.290.4.23 GetRecommendedDisplayGrayscaleValue()

```
unsigned short gdcm::Surface::GetRecommendedDisplayGrayscaleValue ( ) const
```

10.290.4.24 GetRecommendedPresentationOpacity()

```
float gdcm::Surface::GetRecommendedPresentationOpacity ( ) const
```

10.290.4.25 GetRecommendedPresentationType()

```
VIEWType gdcm::Surface::GetRecommendedPresentationType ( ) const
```

10.290.4.26 GetSTATES()

```
static STATES gdcm::Surface::GetSTATES (
            const char * state )  [static]
```

10.290.4.27 GetSTATESString()

```
static const char* gdcm::Surface::GetSTATESString (
            STATES state )  [static]
```

10.290.4.28 GetSurfaceComments()

```
const char* gdcm::Surface::GetSurfaceComments ( ) const
```

10.290.4.29 GetSurfaceNumber()

```
unsigned long gdcm::Surface::GetSurfaceNumber ( ) const
```

10.290.4.30 GetSurfaceProcessing()

```
bool gdcm::Surface::GetSurfaceProcessing ( ) const
```

10.290.4.31 GetSurfaceProcessingDescription()

```
const char* gdcm::Surface::GetSurfaceProcessingDescription ( ) const
```

10.290.4.32 GetSurfaceProcessingRatio()

```
float gdcm::Surface::GetSurfaceProcessingRatio ( ) const
```

10.290.4.33 GetVectorAccuracy()

```
const float* gdcm::Surface::GetVectorAccuracy ( ) const
```

10.290.4.34 GetVectorCoordinateData() [1/2]

```
const DataElement& gdcm::Surface::GetVectorCoordinateData ( ) const
```

10.290.4.35 GetVectorCoordinateData() [2/2]

```
DataElement& gdcm::Surface::GetVectorCoordinateData ( )
```

10.290.4.36 GetVectorDimensionality()

```
unsigned short gdcm::Surface::GetVectorDimensionality ( ) const
```

10.290.4.37 GetVIEWType()

```
static VIEWType gdcm::Surface::GetVIEWType (
            const char * type )  [static]
```

10.290.4.38 GetVIEWTypeString()

```
static const char* gdcm::Surface::GetVIEWTypeString (
            VIEWType type )  [static]
```

10.290.4.39 SetAlgorithmFamily()

```
void gdcm::Surface::SetAlgorithmFamily (
            SegmentHelper::BasicCodedEntry const & BSE )
```

10.290.4.40 SetAlgorithmName()

```
void gdcm::Surface::SetAlgorithmName (
            const char * str )
```

10.290.4.41 SetAlgorithmVersion()

```
void gdcm::Surface::SetAlgorithmVersion (
            const char * str )
```

10.290.4.42 SetAxisOfRotation()

```
void gdcm::Surface::SetAxisOfRotation (
            const float * axis )
```

10.290.4.43 SetCenterOfRotation()

```
void gdcm::Surface::SetCenterOfRotation (
            const float * center )
```

10.290.4.44 SetFiniteVolume()

```
void gdcm::Surface::SetFiniteVolume (
            STATES state )
```

10.290.4.45 SetManifold()

```
void gdcm::Surface::SetManifold (
            STATES state )
```

10.290.4.46 SetMaximumPointDistance()

```
void gdcm::Surface::SetMaximumPointDistance (
            float maximum )
```

10.290.4.47 SetMeanPointDistance()

```
void gdcm::Surface::SetMeanPointDistance (
            float average )
```

10.290.4.48 SetMeshPrimitive()

```
void gdcm::Surface::SetMeshPrimitive (
            MeshPrimitive & mp )
```

10.290.4.49 SetNumberOfSurfacePoints()

```
void gdcm::Surface::SetNumberOfSurfacePoints (
            const unsigned long nb )
```

10.290.4.50 SetNumberOfVectors()

```
void gdcm::Surface::SetNumberOfVectors (
            const unsigned long nb )
```

10.290.4.51 SetPointCoordinatesData()

```
void gdcm::Surface::SetPointCoordinatesData (
            DataElement const & de )
```

10.290.4.52 SetPointPositionAccuracy()

```
void gdcm::Surface::SetPointPositionAccuracy (
            const float * accuracies )
```

10.290.4.53 SetPointsBoundingBoxCoordinates()

```
void gdcm::Surface::SetPointsBoundingBoxCoordinates (
            const float * coordinates )
```

10.290.4.54 SetProcessingAlgorithm()

```
void gdcm::Surface::SetProcessingAlgorithm (
            SegmentHelper::BasicCodedEntry const & BSE )
```

10.290.4.55 SetRecommendedDisplayCIELabValue() [1/3]

```
void gdcm::Surface::SetRecommendedDisplayCIELabValue (
            const unsigned short vl[3] )
```

10.290.4.56 SetRecommendedDisplayCIELabValue() [2/3]

```
void gdcm::Surface::SetRecommendedDisplayCIELabValue (
            const unsigned short vl,
            const unsigned int idx = 0 )
```

10.290.4.57 SetRecommendedDisplayCIELabValue() [3/3]

```
void gdcm::Surface::SetRecommendedDisplayCIELabValue (
            const std::vector< unsigned short > & vl )
```

10.290.4.58 SetRecommendedDisplayGrayscaleValue()

```
void gdcm::Surface::SetRecommendedDisplayGrayscaleValue (
            const unsigned short vl )
```

10.290.4.59 SetRecommendedPresentationOpacity()

```
void gdcm::Surface::SetRecommendedPresentationOpacity (
            const float opacity )
```

10.290.4.60 SetRecommendedPresentationType()

```
void gdcm::Surface::SetRecommendedPresentationType (
            VIEWType type )
```

10.290.4.61 SetSurfaceComments()

```
void gdcm::Surface::SetSurfaceComments (
            const char * comment )
```

10.290.4.62 SetSurfaceNumber()

```
void gdcm::Surface::SetSurfaceNumber (
            const unsigned long nb )
```

10.290.4.63 SetSurfaceProcessing()

```
void gdcm::Surface::SetSurfaceProcessing (
            bool b )
```

10.290.4.64 SetSurfaceProcessingDescription()

```
void gdcm::Surface::SetSurfaceProcessingDescription (
            const char * description )
```

10.290.4.65 SetSurfaceProcessingRatio()

```
void gdcm::Surface::SetSurfaceProcessingRatio (
            const float ratio )
```

10.290.4.66 SetVectorAccuracy()

```
void gdcm::Surface::SetVectorAccuracy (
            const float * accuracy )
```

10.290.4.67 SetVectorCoordinateData()

```
void gdcm::Surface::SetVectorCoordinateData (
            DataElement const & de )
```

10.290.4.68 SetVectorDimensionality()

```
void gdcm::Surface::SetVectorDimensionality (
            const unsigned short dim )
```

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

- gdcmSurface.h

10.291 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).

10.291.1 Detailed Description

SurfaceHelper.

Helper class for Surface object

10.291.2 Member Typedef Documentation

10.291.2.1 ColorArray

```
typedef std::vector< unsigned short > gdcm::SurfaceHelper::ColorArray
```

10.291.3 Member Function Documentation

10.291.3.1 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.

10.291.3.2 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.

10.291.3.3 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.

10.291.3.4 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:

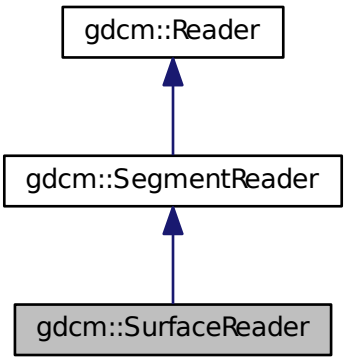
- gdcmSurfaceHelper.h

10.292 gdcm::SurfaceReader Class Reference

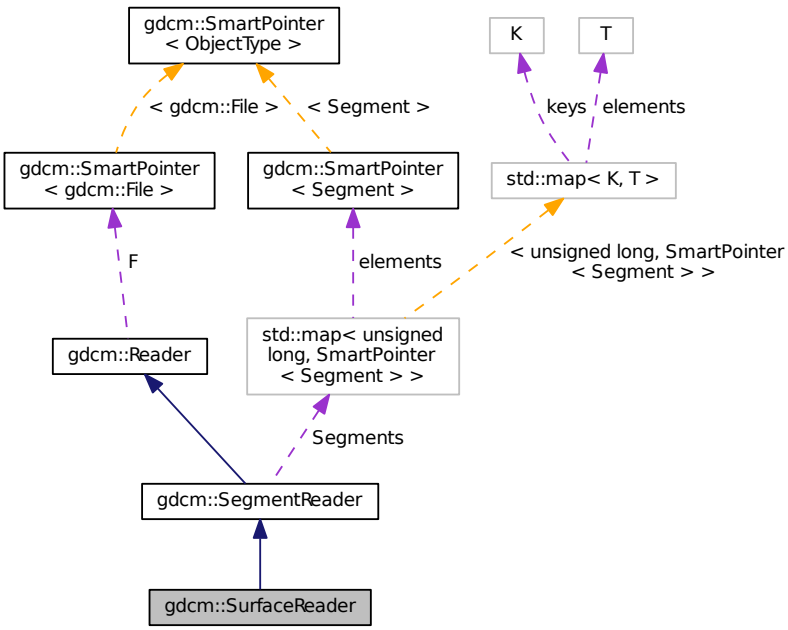
This class defines a SURFACE IE reader.

```
#include <gdcmSurfaceReader.h>
```

Inheritance diagram for gdcm::SurfaceReader:



Collaboration diagram for gdcm::SurfaceReader:



Public Member Functions

- SurfaceReader ()

- virtual ~SurfaceReader ()
- unsigned long GetNumberOfSurfaces () const
- virtual bool Read ()

  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

10.292.1 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

10.292.2 Constructor & Destructor Documentation

10.292.2.1 SurfaceReader()

```
gdcm::SurfaceReader::SurfaceReader ( )
```

10.292.2.2 ~SurfaceReader()

```
virtual gdcm::SurfaceReader::~SurfaceReader ( )  [virtual]
```

10.292.3 Member Function Documentation

10.292.3.1 GetNumberOfSurfaces()

```
unsigned long gdcm::SurfaceReader::GetNumberOfSurfaces ( ) const
```

10.292.3.2 Read()

```
virtual bool gdcm::SurfaceReader::Read ( )  [virtual]
```

Read.

Reimplemented from gdcm::SegmentReader.

10.292.3.3 ReadPointMacro()

```
bool gdcm::SurfaceReader::ReadPointMacro (
            SmartPointer< Surface > surface,
            const DataSet & surfaceDS )  [protected]
```

10.292.3.4 ReadSurface()

```
bool gdcm::SurfaceReader::ReadSurface (
            const Item & surfaceItem,
            const unsigned long idx )  [protected]
```

10.292.3.5 ReadSurfaces()

```
bool gdcm::SurfaceReader::ReadSurfaces ( )  [protected]
```

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

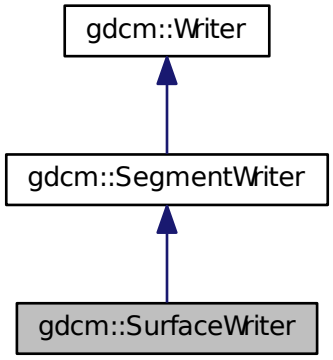
- gdcmSurfaceReader.h

10.293 gdcm::SurfaceWriter Class Reference

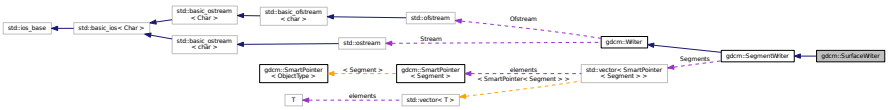
This class defines a SURFACE IE writer.

```
#include <gdcmSurfaceWriter.h>
```

Inheritance diagram for gdcm::SurfaceWriter:



Collaboration diagram for gdcm::SurfaceWriter:



Public Member Functions

- SurfaceWriter ()
- virtual ~SurfaceWriter ()
- unsigned long GetNumberOfSurfaces ()
- void SetNumberOfSurfaces (const unsigned long nb)
- bool Write ()

  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

10.293.1 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

10.293.2 Constructor & Destructor Documentation

10.293.2.1 SurfaceWriter()

```
gdcm::SurfaceWriter::SurfaceWriter ( )
```

10.293.2.2 ~SurfaceWriter()

```
virtual gdcm::SurfaceWriter::~SurfaceWriter ( )  [virtual]
```

10.293.3 Member Function Documentation

10.293.3.1 ComputeNumberOfSurfaces()

```
void gdcm::SurfaceWriter::ComputeNumberOfSurfaces ( )  [protected]
```

10.293.3.2 GetNumberOfSurfaces()

```
unsigned long gdcm::SurfaceWriter::GetNumberOfSurfaces ( )
```

10.293.3.3 PrepareWrite()

```
bool gdcm::SurfaceWriter::PrepareWrite ( )  [protected]
```

10.293.3.4 PrepareWritePointMacro()

```
bool gdcm::SurfaceWriter::PrepareWritePointMacro (
            SmartPointer< Surface > surface,
            DataSet & surfaceDS,
            const TransferSyntax & ts )  [protected]
```

10.293.3.5 SetNumberOfSurfaces()

```
void gdcm::SurfaceWriter::SetNumberOfSurfaces (
            const unsigned long nb )
```

10.293.3.6 Write()

```
bool gdcm::SurfaceWriter::Write ( )  [virtual]
```

Write.

Reimplemented from gdcm::SegmentWriter.

10.293.4 Member Data Documentation

10.293.4.1 NumberOfSurfaces

```
unsigned long gdcm::SurfaceWriter::NumberOfSurfaces  [protected]
```

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

- gdcmSurfaceWriter.h

10.294 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)

10.294.1 Detailed Description

SwapCode representation.

Examples:

TestByteSwap.cxx.

10.294.2 Member Enumeration Documentation

10.294.2.1 SwapCodeType

```
enum gdcm::SwapCode::SwapCodeType
```

Enumerator

Unknown	
LittleEndian	
BigEndian	
BadLittleEndian	
BadBigEndian	

10.294.3 Constructor & Destructor Documentation

10.294.3.1 SwapCode()

```
gdcm::SwapCode::SwapCode (
            SwapCodeType sc = Unknown )  [inline]
```

References gdcm::operator<<().

10.294.4 Member Function Documentation

10.294.4.1 GetIndex()

```
static int gdcm::SwapCode::GetIndex (
            SwapCode const & sc )  [static], [protected]
```

10.294.4.2 GetSwapCodeString()

```
static const char* gdcm::SwapCode::GetSwapCodeString (
            SwapCode const & sc )  [static]
```

Referenced by gdcm::operator<<().

10.294.4.3 operator SwapCode::SwapCodeType()

```
gdcm::SwapCode::operator SwapCode::SwapCodeType ( ) const  [inline]
```

10.294.5 Friends And Related Function Documentation

10.294.5.1 operator<<

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

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

- gdcmSwapCode.h

10.295 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)

10.295.1 Member Function Documentation

10.295.1.1 Swap()

```
template<typename T >
static T gdcm::SwapperDoOp::Swap (
            T val )  [static]
```

10.295.1.2 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

10.296 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)

10.296.1 Detailed Description

Examples:

ReadExplicitLengthSQIVR.cxx.

10.296.2 Member Function Documentation

10.296.2.1 Swap()

```
template<typename T >
static T gdcm::SwapperNoOp::Swap (
            T val )  [inline], [static]
```

10.296.2.2 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

10.297 gdcm::System Class Reference

Class to do system operation.

```
#include <gdcmSystem.h>
```

Static Public Member Functions

- 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 sytem.
- 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)

10.297.1 Detailed Description

Class to do system operation.

OS independent functionalities

10.297.2 Member Function Documentation

10.297.2.1 DeleteDirectory()

```
static bool gdcm::System::DeleteDirectory (
            const char * source )  [static]
```

remove a directory named source

10.297.2.2 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

10.297.2.3 FileExists()

```
static bool gdcm::System::FileExists (
            const char * filename )  [static]
```

Check whether the specified file exist on the sytem.

Examples:

EncapsulateFileInRawData.cxx, gdcmorthoplanes.cxx, and MagnifyFile.cxx.

10.297.2.4 FileIsDirectory()

```
static bool gdcm::System::FileIsDirectory (
            const char * name )  [static]
```

Check whether the file specified is a directory:

Examples:

gdcmorthoplanes.cxx, and threadgdcm.cxx.

10.297.2.5 FileIsSymlink()

```
static bool gdcm::System::FileIsSymlink (
            const char * name )  [static]
```

Check whether name is a symlink.

10.297.2.6 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.

10.297.2.7 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

10.297.2.8 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]

10.297.2.9 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

10.297.2.10 GetCurrentModuleFileName()

```
static const char* gdcm::System::GetCurrentModuleFileName ( )  [static]
```

Return the directory the current module is located: NOT THREAD SAFE

10.297.2.11 GetCurrentProcessFileName()

```
static const char* gdcm::System::GetCurrentProcessFileName ( )  [static]
```

Return the directory the current process (executable) is located: NOT THREAD SAFE

10.297.2.12 GetCurrentResourcesDirectory()

```
static const char* gdcm::System::GetCurrentResourcesDirectory ( )  [static]
```

On some system (Apple) return the path to the current bundled 'Resources' directory NOT THREAD SAFE

10.297.2.13 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

10.297.2.14 GetHostName()

```
static bool gdcm::System::GetHostName (
            char hostname[255] )  [static]
```

Retrieve the hostname, only the first 255 byte are copyied. This may come handy to specify the Station Name

10.297.2.15 GetLastSystemError()

```
static const char* gdcm::System::GetLastSystemError ( )  [static]
```

Return the last error.

10.297.2.16 GetLocaleCharset()

```
static const char* gdcm::System::GetLocaleCharset ( )  [static]
```

return `locale charmap`

10.297.2.17 GetPermissions()

```
static bool gdcm::System::GetPermissions (
            const char * file,
            unsigned short & mode )  [static], [protected]
```

NOT THREAD SAFE.

10.297.2.18 GetTimezoneOffsetFromUTC()

```
static const char* gdcm::System::GetTimezoneOffsetFromUTC ( )  [static]
```

Return the value for Timezone Offset From UTC as string.

Warning

not thread safe

10.297.2.19 MakeDirectory()

```
static bool gdcm::System::MakeDirectory (
            const char * path )  [static]
```

Create a directory name path.

10.297.2.20 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)

10.297.2.21 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

10.297.2.22 RemoveFile()

```
static bool gdcm::System::RemoveFile (
            const char * source )  [static]
```

remove a file named source

10.297.2.23 SetPermissions()

```
static bool gdcm::System::SetPermissions (
            const char * file,
            unsigned short mode )  [static], [protected]
```

10.297.2.24 StrCaseCmp()

```
static int gdcm::System::StrCaseCmp (
             const char * s1,
             const char * s2 )  [static]
```

consistent func for C99 spec of strcasecmp/strncasecmp

10.297.2.25 StrNCaseCmp()

```
static int gdcm::System::StrNCaseCmp (
             const char * s1,
             const char * s2,
             size_t n )  [static]
```

Precondition

n != 0

10.297.2.26 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

10.297.2.27 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

10.298 gdcm::Table Class Reference

Table.

```
#include <gdcmTable.h>
```

Public Types

- typedef std::map< Tag, TableEntry > MapTableEntry

Public Member Functions

- Table ()
- ~Table ()
- const TableEntry & GetTableEntry (const Tag &tag) const
- void InsertEntry (Tag const &tag, TableEntry const &te)

Friends

- std::ostream & operator<< (std::ostream &_os, const Table &_val)

10.298.1 Detailed Description

Table.

10.298.2 Member Typedef Documentation

10.298.2.1 MapTableEntry

```
typedef std::map<Tag, TableEntry> gdcm::Table::MapTableEntry
```

10.298.3 Constructor & Destructor Documentation

10.298.3.1 Table()

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

Referenced by GetTableEntry().

10.298.3.2 ~Table()

```
gdcm::Table::~Table ( )  [inline]
```

References operator<<.

10.298.4 Member Function Documentation

10.298.4.1 GetTableEntry()

```
const TableEntry& gdcm::Table::GetTableEntry (
            const Tag & tag ) const  [inline]
```

References Table().

10.298.4.2 InsertEntry()

```
void gdcm::Table::InsertEntry (
            Tag const & tag,
            TableEntry const & te )  [inline]
```

10.298.5 Friends And Related Function Documentation

10.298.5.1 operator<<

```
std::ostream& operator<< (
            std::ostream & _os,
            const Table & _val )  [friend]
```

Referenced by ~Table().

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

- gdcmTable.h

10.299 gdcm::TableEntry Class Reference

TableEntry.

```
#include <gdcmTableEntry.h>
```

Public Member Functions

- TableEntry (const char *attribute=0, Type const &type=Type(), const char *des=0)
- ~TableEntry ()

10.299.1 Detailed Description

TableEntry.

10.299.2 Constructor & Destructor Documentation

10.299.2.1 TableEntry()

```
gdcm::TableEntry::TableEntry (
            const char * attribute = 0,
            Type const & type = Type(),
            const char * des = 0 )  [inline]
```

10.299.2.2 ~TableEntry()

```
gdcm::TableEntry::~TableEntry ( )  [inline]
```

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

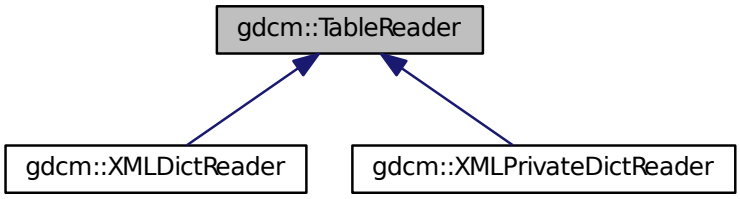
- gdcmTableEntry.h

10.300 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 ()
- 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)

10.300.1 Detailed Description

Class for representing a TableReader.

Note

This class is an empty shell meant to be derived

10.300.2 Constructor & Destructor Documentation

10.300.2.1 TableReader()

```
gdcm::TableReader::TableReader (
            Defs & defs )  [inline]
```

10.300.2.2 ~TableReader()

```
virtual gdcm::TableReader::~TableReader ( )  [inline], [virtual]
```

10.300.3 Member Function Documentation

10.300.3.1 CharacterDataHandler()

```
virtual void gdcm::TableReader::CharacterDataHandler (
            const char * data,
            int length )  [virtual]
```

Reimplemented in gdcm::XMLDictReader, and gdcm::XMLPrivateDictReader.

10.300.3.2 EndElement()

```
virtual void gdcm::TableReader::EndElement (
            const char * name )  [virtual]
```

Reimplemented in gdcm::XMLDictReader, and gdcm::XMLPrivateDictReader.

10.300.3.3 GetDefs()

```
const Defs& gdcm::TableReader::GetDefs ( ) const  [inline]
```

10.300.3.4 GetFilename()

```
const char* gdcm::TableReader::GetFilename ( )  [inline]
```

10.300.3.5 HandleIOD()

```
void gdcm::TableReader::HandleIOD (
            const char ** atts )
```

10.300.3.6 HandleIODEntry()

```
void gdcm::TableReader::HandleIODEntry (
            const char ** atts )
```

10.300.3.7 HandleMacro()

```
void gdcm::TableReader::HandleMacro (
            const char ** atts )
```

10.300.3.8 HandleMacroEntry()

```
void gdcm::TableReader::HandleMacroEntry (
            const char ** atts )
```

10.300.3.9 HandleMacroEntryDescription()

```
void gdcm::TableReader::HandleMacroEntryDescription (
            const char ** atts )
```

10.300.3.10 HandleModule()

```
void gdcm::TableReader::HandleModule (
            const char ** atts )
```

10.300.3.11 HandleModuleEntry()

```
void gdcm::TableReader::HandleModuleEntry (
            const char ** atts )
```

10.300.3.12 HandleModuleEntryDescription()

```
void gdcm::TableReader::HandleModuleEntryDescription (
            const char ** atts )
```

10.300.3.13 HandleModuleInclude()

```
void gdcm::TableReader::HandleModuleInclude (
            const char ** atts )
```

10.300.3.14 Read()

```
int gdcm::TableReader::Read ( )
```

10.300.3.15 SetFilename()

```
void gdcm::TableReader::SetFilename (
            const char * filename )  [inline]
```

10.300.3.16 StartElement()

```
virtual void gdcm::TableReader::StartElement (
            const char * name,
            const char ** atts )  [virtual]
```

Reimplemented in gdcm::XMLDictReader, and gdcm::XMLPrivateDictReader.

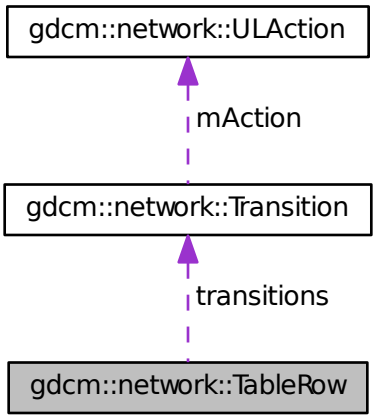
The documentation for this class was generated from the following file:

- gdcmTableReader.h

10.301 gdcm::network::TableRow Class Reference

`#include <gdcmULTransitionTable.h>`

Collaboration diagram for gdcm::network::TableRow:



Public Member Functions

- TableRow ()
- ~TableRow ()

Public Attributes

- Transition * transitions [cMaxStateID]

10.301.1 Constructor & Destructor Documentation

10.301.1.1 TableRow()

`gdcm::network::TableRow::TableRow ( ) [inline]`

References gdcm::network::cMaxStateID.

10.301.1.2 ~TableRow()

```
gdcm::network::TableRow::~TableRow ( )  [inline]
```

References gdcm::network::cMaxStateID.

10.301.2 Member Data Documentation

10.301.2.1 transitions

```
Transition* gdcm::network::TableRow::transitions[cMaxStateID]
```

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

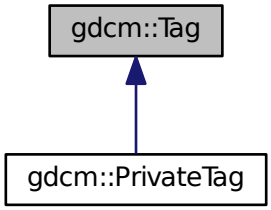
- gdcmULTransitionTable.h

10.302 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 (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.
- Tag (const Tag &_val)
- 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
- const uint16_t & operator[ ] (const unsigned int &_id) const

  Returns the Group or Element of the given Tag, depending on id (0/1)
- uint16_t & operator[ ] (const unsigned int &_id)

  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)

10.302.1 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:

ChangeSequenceUltrasound.cxx, ClinicalTrialAnnotate.cxx, CreateARGBImage.cxx, CreateCMYKImage.cxx, CreateFakeRTDOSE.cxx, CreateJPIPDataSet.cxx, DumpToSQLITE3.cxx, DuplicatePCDE.cxx, EncapsulateFileInRawData.cxx, ExtractEncryptedContent.cxx, Extracting_All_Resolution.cxx, Fake_Image_Using_Stream_Image_Writer.cxx, FixBrokenJ2K.cxx, FixJAIBugJPEGLS.cxx, gdcmrtionplan.cxx, gdcmrtplan.cxx, GenAllVR.cxx, GenFakeIdentifyFile.cxx, GenFakeImage.cxx, GenLongSeqs.cxx, GenSeqs.cxx, GetJPEGSamplePrecision.cxx, GetSequenceUltrasound.cxx, GetSubSequenceData.cxx, iU22tomultisc.cxx, LargeVRDSExplicit.cxx, MakeTemplate.cxx, MergeTwoFiles.cxx, PatchFile.cxx, pmsct_rgb1.cxx, PublicDict.cxx, ReadAndDumpDICOMDIR.cxx, ReadAndPrintAttributes.cxx, ReadExplicitLengthSQIVR.cxx, rle2img.cxx, SimpleScanner.cxx, SortImage.cxx, StreamImageReaderTest.cxx, TraverseModules.cxx, and VolumeSorter.cxx.

10.302.2 Constructor & Destructor Documentation

10.302.2.1 Tag() [1/3]

```
gdcm::Tag::Tag (
            uint16_t group,
            uint16_t element )  [inline]
```

Constructor with 2∗uint16_t.

10.302.2.2 Tag() [2/3]

```
gdcm::Tag::Tag (
            uint32_t tag = 0 )  [inline]
```

Constructor with 1∗uint32_t Prefer the cstor that takes two uint16_t.

References gdcm::operator<<(), and gdcm::operator>>().

10.302.2.3 Tag() [3/3]

```
gdcm::Tag::Tag (
            const Tag & _val )  [inline]
```

References tag.

10.302.3 Member Function Documentation

10.302.3.1 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::DataSet::ComputeGroupLength(), IsGroupXX(), gdcm::PrivateDict::PrintXML(), gdcm::Private↩ Tag::PrivateTag(), gdcm::SequenceOfFragments::ReadValue(), and SetPrivateCreator().

10.302.3.2 GetElementTag()

```
uint32_t gdcm::Tag::GetElementTag ( ) const  [inline]
```

Returns the full tag value of the given Tag.

10.302.3.3 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::CommandDataSet::Insert(), gdcm::FileMeta↩ Information::Insert(), gdcm::DataSet::Insert(), IsGroupXX(), gdcm::PrivateDict::PrintXML(), gdcm::SequenceOf↩ Fragments::ReadValue(), gdcm::Attribute< Group, Element, TVR, TVM >::SetFromDataElement(), gdcm::Attribute< Group, Element, TVR, VM::VM1 >::SetFromDataElement(), gdcm::Attribute< Group, Element, TVR, VM::VM1_n >::SetFromDataElement(), and SetPrivateCreator().

10.302.3.4 GetLength()

```
uint32_t gdcm::Tag::GetLength ( ) const  [inline]
```

return the length of tag (read: size on disk)

10.302.3.5 GetPrivateCreator()

```
Tag gdcm::Tag::GetPrivateCreator ( ) const  [inline]
```

Return the Private Creator Data Element tag of a private data element.

References SetElement().

10.302.3.6 IsGroupLength()

```
bool gdcm::Tag::IsGroupLength ( ) const  [inline]
```

return whether the tag correspond to a group length tag:

10.302.3.7 IsGroupXX()

```
bool gdcm::Tag::IsGroupXX (
             const Tag & t ) const  [inline]
```

e.g 6002,3000 belong to groupXX: 6000,3000

References GetElement(), GetGroup(), and IsPrivate().

10.302.3.8 IsIllegal()

```
bool gdcm::Tag::IsIllegal ( ) const  [inline]
```

return if the tag is considered to be an illegal tag

10.302.3.9 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().

10.302.3.10 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.

10.302.3.11 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...

10.302.3.12 operator"!=()

```
bool gdcm::Tag::operator!= (
            const Tag & _val ) const  [inline]
```

References tag.

10.302.3.13 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.

10.302.3.14 operator<=()

```
bool gdcm::Tag::operator<= (
            const Tag & t2 ) const  [inline]
```

10.302.3.15 operator=()

```
Tag& gdcm::Tag::operator= (
            const Tag & _val )  [inline]
```

References tag.

10.302.3.16 operator==()

```
bool gdcm::Tag::operator== (
            const Tag & _val ) const  [inline]
```

References tag.

10.302.3.17 operator[]() [1/2]

```
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)

10.302.3.18 operator[]() [2/2]

```
uint16_t& gdcm::Tag::operator[] (
            const unsigned int & _id )  [inline]
```

Returns the Group or Element of the given Tag, depending on id (0/1)

10.302.3.19 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)

10.302.3.20 PrintAsContinuousUpperCaseString()

```
std::string gdcm::Tag::PrintAsContinuousUpperCaseString ( ) const
```

Same as PrintAsContinuousString, but hexadecimal [a-f] are printed using upper case.

10.302.3.21 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

10.302.3.22 Read()

```
template<typename TSwap >
std::istream& gdcm::Tag::Read (
            std::istream & is )  [inline]
```

Read a tag from binary representation.

10.302.3.23 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 formated 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

10.302.3.24 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)

10.302.3.25 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

10.302.3.26 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(), and gdcm::operator>>().

10.302.3.27 SetElementTag() [1/2]

```
void gdcm::Tag::SetElementTag (
            uint16_t group,
            uint16_t element )  [inline]
```

Sets the 'Group number' & 'Element number' of the given Tag.

10.302.3.28 SetElementTag() [2/2]

```
void gdcm::Tag::SetElementTag (
            uint32_t tag )  [inline]
```

Sets the full tag value of the given Tag.

10.302.3.29 SetGroup()

```
void gdcm::Tag::SetGroup (
            uint16_t group )  [inline]
```

Sets the 'Group number' of the given Tag.

Referenced by gdcm::operator>>().

10.302.3.30 SetPrivateCreator()

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

Set private creator:

Examples:

DuplicatePCDE.cxx.

References GetElement(), GetGroup(), and IsPrivate().

10.302.3.31 Write()

```
template<typename TSwap >
const std::ostream& gdcm::Tag::Write (
            std::ostream & os ) const  [inline]
```

Write a tag in binary rep.

Referenced by gdcm::SequenceOfItems::Write(), and gdcm::Item::Write().

10.302.4 Friends And Related Function Documentation

10.302.4.1 operator<<

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

10.302.4.2 operator>>

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

10.302.5 Member Data Documentation

10.302.5.1 bytes

```
char gdcm::Tag::bytes[4]
```

10.302.5.2 tag

```
uint32_t gdcm::Tag::tag
```

Referenced by operator!=(), operator<(), operator=(), operator==(), and Tag().

10.302.5.3 tags

```
uint16_t gdcm::Tag::tags[2]
```

Referenced by operator<().

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

- gdcmTag.h

10.303 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.

10.303.1 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`

10.303.2 Constructor & Destructor Documentation

10.303.2.1 TagPath()

```
gdcm::TagPath::TagPath ( )
```

10.303.2.2 ~TagPath()

```
gdcm::TagPath::~TagPath ( )
```

10.303.3 Member Function Documentation

10.303.3.1 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

10.303.3.2 ConstructFromTagList()

```
bool gdcm::TagPath::ConstructFromTagList (
            Tag const * l,
            unsigned int n )
```

Construct from a list of tags.

10.303.3.3 IsValid()

```
static bool gdcm::TagPath::IsValid (
            const char * path )  [static]
```

Return if path is valid or not.

10.303.3.4 Print()

```
void gdcm::TagPath::Print (
            std::ostream &  ) const
```

10.303.3.5 Push() [1/2]

```
bool gdcm::TagPath::Push (
            Tag const & t )
```

10.303.3.6 Push() [2/2]

```
bool gdcm::TagPath::Push (
            unsigned int itemnum )
```

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

- gdcmTagPath.h

10.304 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 ()
- ~Testing ()
- 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, unsigned long 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=0)
- static const wchar_t * GetTempDirectoryW (const wchar_t *subdir=0)

  NOT THREAD SAFE.
- static const char * GetTempFilename (const char *filename, const char *subdir=0)

  NOT THREAD SAFE.
- static const wchar_t * GetTempFilenameW (const wchar_t *filename, const wchar_t *subdir=0)

  NOT THREAD SAFE.

10.304.1 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

10.304.2 Member Typedef Documentation

10.304.2.1 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

10.304.2.2 MediaStorageDataFilesType

```
typedef const char* const(* gdcm::Testing::MediaStorageDataFilesType)[2]
```

return the table that map the media storage (as string) of a filename (gdcmData)

10.304.3 Constructor & Destructor Documentation

10.304.3.1 Testing()

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

10.304.3.2 ~Testing()

```
gdcm::Testing::~Testing ( )  [inline]
```

10.304.4 Member Function Documentation

10.304.4.1 ComputeFileMD5()

```
static bool gdcm::Testing::ComputeFileMD5 (
            const char * filename,
            char digest_str[33] )  [static]
```

Examples:

MetaImageMD5Activiz.cs.

10.304.4.2 ComputeMD5()

```
static bool gdcm::Testing::ComputeMD5 (
            const char * buffer,
            unsigned long 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.

10.304.4.3 GetDataExtraRoot()

```
static const char* gdcm::Testing::GetDataExtraRoot ( )  [static]
```

Return the GDCM DATA EXTRA ROOT.

Examples:

DiscriminateVolume.cxx, reslicesphere.cxx, and VolumeSorter.cxx.

10.304.4.4 GetDataRoot()

```
static const char* gdcm::Testing::GetDataRoot ( )  [static]
```

Return the GDCM DATA ROOT.

Examples:

Convert16BitsTo8Bits.cxx, ConvertMultiFrameToSingleFrame.cxx, ConvertRGBToLuminance.cxx, and MagnifyFile.cxx.

10.304.4.5 GetFileName()

```
static const char* gdcm::Testing::GetFileName (
            unsigned int file )  [static]
```

Examples:

MetaImageMD5Activiz.cs.

10.304.4.6 GetFileNames()

```
static const char* const* gdcm::Testing::GetFileNames ( )  [static]
```

return the table of fullpath to gdcmData DICOM files:

Examples:

TestReader.cxx.

10.304.4.7 GetLossyFlagFromFile()

```
static int gdcm::Testing::GetLossyFlagFromFile (
            const char * filepath )  [static]
```

Return the lossy flag of the given filename -1 -> Error 0 -> Lossless 1 -> Lossy

10.304.4.8 GetMD5DataImage()

```
static const char* const* gdcm::Testing::GetMD5DataImage (
            unsigned int file )  [static]
```

10.304.4.9 GetMD5DataImages()

```
static MD5DataImagesType gdcm::Testing::GetMD5DataImages ( )  [static]
```

10.304.4.10 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.

10.304.4.11 GetMD5FromFile()

```
static const char* gdcm::Testing::GetMD5FromFile (
            const char * filepath )  [static]
```

10.304.4.12 GetMediaStorageDataFile()

```
static const char* const* gdcm::Testing::GetMediaStorageDataFile (
            unsigned int file )  [static]
```

10.304.4.13 GetMediaStorageDataFiles()

```
static MediaStorageDataFilesType gdcm::Testing::GetMediaStorageDataFiles ( )  [static]
```

10.304.4.14 GetMediaStorageFromFile()

```
static const char* gdcm::Testing::GetMediaStorageFromFile (
            const char * filepath )  [static]
```

Examples:

MetaImageMD5Activiz.cs, and TestReader.cxx.

10.304.4.15 GetNumberOfFileNames()

```
static unsigned int gdcm::Testing::GetNumberOfFileNames ( )  [static]
```

Examples:

MetaImageMD5Activiz.cs.

10.304.4.16 GetNumberOfMD5DataImages()

```
static unsigned int gdcm::Testing::GetNumberOfMD5DataImages ( )  [static]
```

10.304.4.17 GetNumberOfMediaStorageDataFiles()

```
static unsigned int gdcm::Testing::GetNumberOfMediaStorageDataFiles ( )  [static]
```

10.304.4.18 GetPixelSpacingDataRoot()

```
static const char* gdcm::Testing::GetPixelSpacingDataRoot ( )  [static]
```

Return the GDCM PIXEL SPACING DATA ROOT (See David Clunie website for dataset)

10.304.4.19 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

10.304.4.20 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

10.304.4.21 GetSourceDirectory()

```
static const char* gdcm::Testing::GetSourceDirectory ( )  [static]
```

Examples:

BasicAnonymizer.cs, and ClinicalTrialIdentificationWorkflow.cs.

10.304.4.22 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

10.304.4.23 GetTempDirectory()

```
static const char* gdcm::Testing::GetTempDirectory (
            const char * subdir = 0 )  [static]
```

NOT THREAD SAFE Returns the temp directory as used in testing needing to output data:

Examples:

MetaImageMD5Activiz.cs.

10.304.4.24 GetTempDirectoryW()

```
static const wchar_t* gdcm::Testing::GetTempDirectoryW (
            const wchar_t * subdir = 0 )  [static]
```

NOT THREAD SAFE.

10.304.4.25 GetTempFilename()

```
static const char* gdcm::Testing::GetTempFilename (
            const char * filename,
            const char * subdir = 0 )  [static]
```

NOT THREAD SAFE.

Examples:

MetaImageMD5Activiz.cs.

10.304.4.26 GetTempFilenameW()

```
static const wchar_t* gdcm::Testing::GetTempFilenameW (
            const wchar_t * filename,
            const wchar_t * subdir = 0 )  [static]
```

NOT THREAD SAFE.

10.304.4.27 Print()

```
void gdcm::Testing::Print (
            std::ostream & os = std::cout )
```

Print.

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

- gdcmTesting.h

10.305 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 ()

10.305.1 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.

10.305.2 Constructor & Destructor Documentation

10.305.2.1 Trace()

```
gdcm::Trace::Trace ( )
```

10.305.2.2 ~Trace()

```
gdcm::Trace::~Trace ( )
```

10.305.3 Member Function Documentation

10.305.3.1 DebugOff()

```
static void gdcm::Trace::DebugOff ( )  [static]
```

Examples:

MetaImageMD5Activiz.cs, and TestReader.cxx.

10.305.3.2 DebugOn()

```
static void gdcm::Trace::DebugOn ( )  [static]
```

Examples:

CreateFakePET.cxx, Fake_Image_Using_Stream_Image_Writer.cxx, and StreamImageReaderTest.cxx.

10.305.3.3 ErrorOff()

```
static void gdcm::Trace::ErrorOff ( )  [static]
```

Examples:

MetaImageMD5Activiz.cs.

10.305.3.4 ErrorOn()

```
static void gdcm::Trace::ErrorOn ( )  [static]
```

10.305.3.5 GetDebugFlag()

```
static bool gdcm::Trace::GetDebugFlag ( )  [static]
```

10.305.3.6 GetDebugStream()

```
static std::ostream& gdcm::Trace::GetDebugStream ( )  [static]
```

10.305.3.7 GetErrorFlag()

```
static bool gdcm::Trace::GetErrorFlag ( )  [static]
```

10.305.3.8 GetErrorStream()

```
static std::ostream& gdcm::Trace::GetErrorStream ( )  [static]
```

10.305.3.9 GetStream()

```
static std::ostream& gdcm::Trace::GetStream ( )  [static]
```

10.305.3.10 GetWarningFlag()

```
static bool gdcm::Trace::GetWarningFlag ( )  [static]
```

10.305.3.11 GetWarningStream()

```
static std::ostream& gdcm::Trace::GetWarningStream ( )  [static]
```

10.305.3.12 SetDebug()

```
static void gdcm::Trace::SetDebug (
            bool debug )  [static]
```

Turn debug messages on (default: false)

Examples:

DumpToSQLITE3.cxx.

10.305.3.13 SetDebugStream()

```
static void gdcm::Trace::SetDebugStream (
            std::ostream & os )  [static]
```

Explicitly set the stream which receive Debug messages:

10.305.3.14 SetError()

```
static void gdcm::Trace::SetError (
            bool debug )  [static]
```

Turn error messages on (default: true)

10.305.3.15 SetErrorStream()

```
static void gdcm::Trace::SetErrorStream (
            std::ostream & os )  [static]
```

Explicitly set the stream which receive Error messages:

Examples:

CStoreQtProgress.cxx.

10.305.3.16 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:

10.305.3.17 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)

10.305.3.18 SetWarning()

```
static void gdcm::Trace::SetWarning (
            bool debug )  [static]
```

Turn warning messages on (default: true)

Examples:

DumpToSQLITE3.cxx.

10.305.3.19 SetWarningStream()

```
static void gdcm::Trace::SetWarningStream (
            std::ostream & os )  [static]
```

Explicitly set the stream which receive Warning messages:

10.305.3.20 WarningOff()

```
static void gdcm::Trace::WarningOff ( )  [static]
```

Examples:

MetaImageMD5Activiz.cs, and TestReader.cxx.

10.305.3.21 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

10.306 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,
  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)

10.306.1 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:

GetJPEGSamplePrecision.cxx, LargeVRDSExplicit.cxx, and MakeTemplate.cxx.

10.306.2 Member Enumeration Documentation

10.306.2.1 NegociatedType

```
enum gdcm::TransferSyntax::NegociatedType
```

Enumerator

Unknown	
Explicit	
Implicit	

10.306.2.2 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	
CT_private_ELE	
JPIPReferenced	
MPEG2MainProfileHighLevel	
MPEG4AVCH264HighProfileLevel4_1	
MPEG4AVCH264BDcompatibleHighProfileLevel4_1	
TS_END	

10.306.3 Constructor & Destructor Documentation

10.306.3.1 TransferSyntax()

```
gdcm::TransferSyntax::TransferSyntax (
            TSType type = ImplicitVRLittleEndian )  [inline]
```

10.306.4 Member Function Documentation

10.306.4.1 CanStoreLossy()

```
bool gdcm::TransferSyntax::CanStoreLossy ( ) const
```

return true if TransFer Syntax Allow storing of Lossy Pixel Data

10.306.4.2 GetNegociatedType()

```
NegociatedType gdcm::TransferSyntax::GetNegociatedType ( ) const
```

10.306.4.3 GetString()

```
const char* gdcm::TransferSyntax::GetString ( ) const  [inline]
```

References GetTSString(), and gdcm::operator<<().

10.306.4.4 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.

10.306.4.5 GetTSString()

```
static const char* gdcm::TransferSyntax::GetTSString (
            TSType ts )  [static]
```

Examples:

LargeVRDSExplicit.cxx.

Referenced by GetString(), and gdcm::operator<<().

10.306.4.6 GetTSType()

```
static TSType gdcm::TransferSyntax::GetTSType (
            const char * str )  [static]
```

10.306.4.7 IsEncapsulated()

```
bool gdcm::TransferSyntax::IsEncapsulated ( ) const
```

Examples:

ExtractIconFromFile.cxx.

10.306.4.8 IsEncoded()

```
bool gdcm::TransferSyntax::IsEncoded ( ) const
```

10.306.4.9 IsExplicit()

```
bool gdcm::TransferSyntax::IsExplicit ( ) const
```

10.306.4.10 IsImplicit()

```
bool gdcm::TransferSyntax::IsImplicit ( ) const
```

10.306.4.11 IsLossless()

```
bool gdcm::TransferSyntax::IsLossless ( ) const
```

Return true if the transfer syntax algorithm is a lossless algorithm

10.306.4.12 IsLossy()

```
bool gdcm::TransferSyntax::IsLossy ( ) const
```

Return true if the transfer syntax algorithm is a lossy algorithm

10.306.4.13 IsValid()

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

10.306.4.14 operator TSType()

```
gdcm::TransferSyntax::operator TSType ( ) const  [inline]
```

10.306.5 Friends And Related Function Documentation

10.306.5.1 operator<<

```
std::ostream& operator<< (
            std::ostream & os,
            const TransferSyntax & ts )  [friend]
```

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

- gdcmTransferSyntax.h

10.307 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

10.307.1 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

10.307.2 Constructor & Destructor Documentation

10.307.2.1 TransferSyntaxSub()

```
gdcm::network::TransferSyntaxSub::TransferSyntaxSub ( )
```

10.307.3 Member Function Documentation

10.307.3.1 GetName()

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

References Print(), Read(), SetNameFromUID(), Size(), and Write().

10.307.3.2 operator==()

```
bool gdcm::network::TransferSyntaxSub::operator== (
             const TransferSyntaxSub & ts ) const  [inline]
```

10.307.3.3 Print()

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

Referenced by GetName().

10.307.3.4 Read()

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

Referenced by GetName().

10.307.3.5 SetName()

```
void gdcm::network::TransferSyntaxSub::SetName (
             const char * name )
```

10.307.3.6 SetNameFromUID()

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

Referenced by GetName().

10.307.3.7 Size()

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

Referenced by GetName().

10.307.3.8 Write()

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

Referenced by GetName().

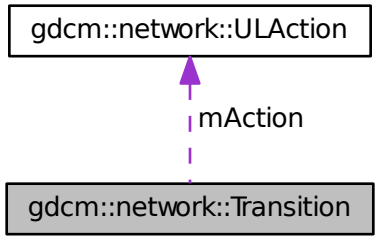
The documentation for this class was generated from the following file:

- gdcmTransferSyntaxSub.h

10.308 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

10.308.1 Constructor & Destructor Documentation

10.308.1.1 Transition() [1/2]

```
gdcm::network::Transition::Transition ( )  [inline]
```

References gdcm::network::eStaDoesNotExist.

Referenced by MakeNew().

10.308.1.2 ~Transition()

```
gdcm::network::Transition::~Transition ( )  [inline]
```

References mAction.

10.308.1.3 Transition() [2/2]

```
gdcm::network::Transition::Transition (
            int inEndState,
            ULAction * inAction )  [inline]
```

10.308.2 Member Function Documentation

10.308.2.1 MakeNew()

```
static Transition* gdcm::network::Transition::MakeNew (
            int inEndState,
            ULAction * inAction )  [inline], [static]
```

References Transition().

10.308.3 Member Data Documentation

10.308.3.1 mAction

```
ULAction* gdcm::network::Transition::mAction
```

Referenced by ~Transition().

10.308.3.2 mEnd

```
int gdcm::network::Transition::mEnd
```

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

- gdcmULTransitionTable.h

10.309 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)

10.309.1 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.

10.309.2 Member Enumeration Documentation

10.309.2.1 TypeType

```
enum gdcm::Type::TypeType
```

Enumerator

T1	
T1C	
T2	
T2C	
T3	
UNKNOWN	

10.309.3 Constructor & Destructor Documentation

10.309.3.1 Type()

```
gdcm::Type::Type (
            TypeType type = UNKNOWN )  [inline]
```

10.309.4 Member Function Documentation

10.309.4.1 GetTypeString()

```
static const char* gdcm::Type::GetTypeString (
            TypeType type )  [static]
```

Referenced by gdcm::operator<<().

10.309.4.2 GetTypeType()

```
static TypeType gdcm::Type::GetTypeType (
            const char * type )  [static]
```

Referenced by gdcm::ModuleEntry::ModuleEntry().

10.309.4.3 operator TypeType()

```
gdcm::Type::operator TypeType ( ) const  [inline]
```

References gdcm::operator<<().

10.309.5 Friends And Related Function Documentation

10.309.5.1 operator<<

```
std::ostream& operator<< (
            std::ostream & os,
            const Type & vr )  [friend]
```

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

- gdcmType.h

10.310 gdcm::UI Struct Reference

```
#include <gdcmVR.h>
```

Public Attributes

- char Internal [64+1]

Friends

- std::ostream & operator<< (std::ostream &_os, const UI &_val)

10.310.1 Friends And Related Function Documentation

10.310.1.1 operator<<

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

10.310.2 Member Data Documentation

10.310.2.1 Internal

```
char gdcm::UI::Internal[64+1]
```

Referenced by gdcm::operator<<().

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

- gdcmVR.h

10.311 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)

10.311.1 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.cxx, GenAllVR.cxx, GenFakeIdentifyFile.cxx, GenFakeImage.cxx, GetSubSequenceData.cxx, StreamImageReaderTest.cxx, and uid_unique.cxx.

10.311.2 Constructor & Destructor Documentation

10.311.2.1 UIDGenerator()

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

By default the root of a UID is a GDCM Root...

10.311.3 Member Function Documentation

10.311.3.1 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.cxx, GenAllVR.cxx, GenFakeIdentifyFile.cxx, GenFakeImage.cxx, GetSubSequenceData.cxx, StreamImageReaderTest.cxx, and uid_unique.cxx.

10.311.3.2 GenerateUUID()

```
static bool gdcm::UIDGenerator::GenerateUUID (
            unsigned char * uuid_data )  [static], [protected]
```

10.311.3.3 GetGDCMUID()

```
static const char* gdcm::UIDGenerator::GetGDCMUID ( )  [static]
```

Return the default (GDCM) root UID:

10.311.3.4 GetRoot()

```
static const char* gdcm::UIDGenerator::GetRoot ( )  [static]
```

Examples:

ClinicalTrialIdentificationWorkflow.cs, ReformatFile.cs, and StandardizeFiles.cs.

10.311.3.5 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)

10.311.3.6 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

10.312 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,
  JPEGLosslessNonHierarchicalFirstOrderPredictionProcess14SelectionValue1DefaultTransferSyntaxforLossless↩
  JPEGImageCompression = 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,
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,
EnhancedUSVolumeStorage,
SurfaceSegmentationStorage,
BreastTomosynthesisImageStorage }

- 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,
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_1_2_4_101,
uid_1_2_840_10008_1_2_4_102,
uid_1_2_840_10008_1_2_4_103 }

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)

10.312.1 Detailed Description

all known uids

Examples:

GenerateStandardSOPClasses.cxx.

10.312.2 Member Typedef Documentation

10.312.2.1 TransferSyntaxStringsType

```
typedef const char* const(* gdcm::UIDs::TransferSyntaxStringsType)[2]
```

10.312.3 Member Enumeration Documentation

10.312.3.1 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	
JPEGLosslessNonHierarchicalFirstOrderPredictionProcess14SelectionValue1DefaultTransferSyntaxfor↩ LosslessJPEGImageCompression	
JPEGLSLosslessImageCompression	
JPEGLSLossyNearLosslessImageCompression	
JPEG2000ImageCompressionLosslessOnly	
JPEG2000ImageCompression	
JPEG2000Part2MulticomponentImageCompressionLosslessOnly	
JPEG2000Part2MulticomponentImageCompression	
JPIPReferenced	
JPIPReferencedDeflate	
MPEG2MainProfileMainLevel	
RLELossless	
RFC2557MIMEencapsulation	
XMLEncoding	
MediaStorageDirectoryStorage	
TalairachBrainAtlasFrameofReference	
SPM2T1FrameofReference	

Enumerator

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	
DetachedInterpretationManagementSOPClassRetired	
StorageServiceClass	
BasicFilmSessionSOPClass	

Enumerator

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	
NuclearMedicineImageStorageRetired	
UltrasoundImageStorageRetired	
UltrasoundImageStorage	

Enumerator

SecondaryCaptureImageStorage	
MultiframeSingleBitSecondaryCaptureImageStorage	
MultiframeGrayscaleByteSecondaryCaptureImageStorage	
MultiframeGrayscaleWordSecondaryCaptureImageStorage	
MultiframeTrueColorSecondaryCaptureImageStorage	
StandaloneOverlayStorageRetired	
StandaloneCurveStorageRetired	
WaveformStorageTrialRetired	
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	
VLSlideCoordinatesMicroscopicImageStorage	
VLPhotographicImageStorage	
VideoPhotographicImageStorage	
OphthalmicPhotography8BitImageStorage	

Enumerator

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	
GeneralPurposeScheduledProcedureStepSOPClass	
GeneralPurposePerformedProcedureStepSOPClass	
GeneralPurposeWorklistManagementMetaSOPClass	
InstanceAvailabilityNotificationSOPClass	

Enumerator

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	
dicomStationName	
dicomDeviceSerialNumber	
dicomInstitutionName	
dicomInstitutionAddress	
dicomInstitutionDepartmentName	

Enumerator

dicomIssuerOfPatientID	
dicomPreferredCallingAETitle	
dicomSupportedCharacterSet	
dicomConfigurationRoot	
dicomDevicesRoot	
dicomUniqueAETitlesRegistryRoot	
dicomDevice	
dicomNetworkAE	
dicomNetworkConnection	
dicomUniqueAETitle	
dicomTransferCapability	
VLWholeSlideMicroscopyImageStorage	
EnhancedUSVolumeStorage	
SurfaceSegmentationStorage	
BreastTomosynthesisImageStorage	

10.312.3.2 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	
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	

Enumerator

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	
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	

Enumerator

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	
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	

Enumerator

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	
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	

Enumerator

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	
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	

Enumerator

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	
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	

Enumerator

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_1_2_4_101	
uid_1_2_840_10008_1_2_4_102	
uid_1_2_840_10008_1_2_4_103	

10.312.4 Member Function Documentation

10.312.4.1 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<<().

10.312.4.2 GetNumberOfTransferSyntaxStrings()

```
static unsigned int gdcm::UIDs::GetNumberOfTransferSyntaxStrings ( )  [static]
```

10.312.4.3 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<<().

10.312.4.4 GetTransferSyntaxString()

```
static const char* const* gdcm::UIDs::GetTransferSyntaxString (
            unsigned int ts )  [static]
```

10.312.4.5 GetTransferSyntaxStrings()

```
static TransferSyntaxStringsType gdcm::UIDs::GetTransferSyntaxStrings ( )  [static]
```

10.312.4.6 GetUIDName()

```
static const char* gdcm::UIDs::GetUIDName (
            unsigned int ts )  [static]
```

10.312.4.7 GetUIDString()

```
static const char* gdcm::UIDs::GetUIDString (
            unsigned int ts )  [static]
```

10.312.4.8 operator TSType()

```
gdcm::UIDs::operator TSType ( ) const  [inline]
```

10.312.4.9 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:

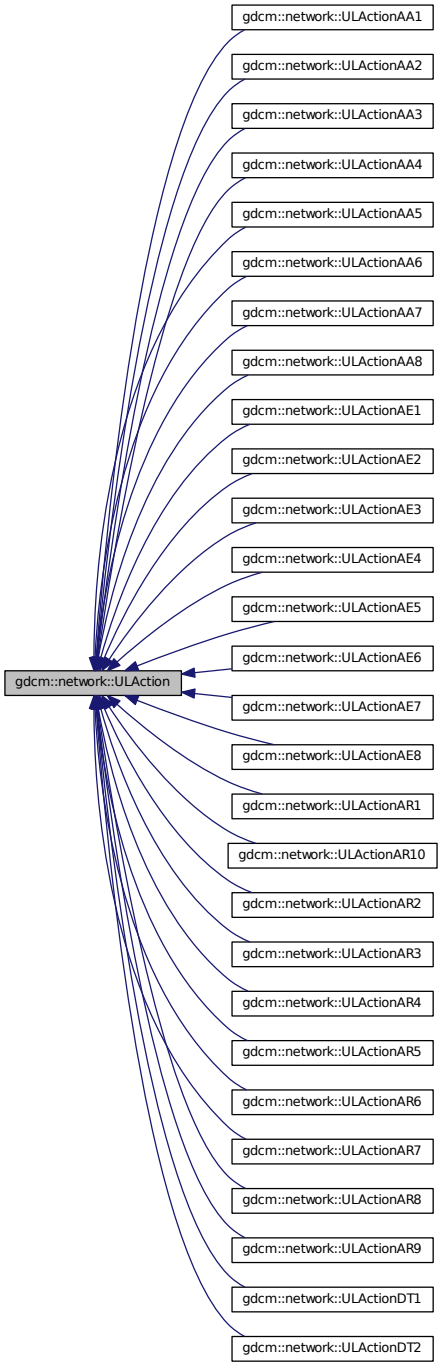
- gdcmUIDs.h

10.313 gdcm::network::ULAction Class Reference

ULAction.

```
#include <gdcmULAction.h>
```

Inheritance diagram for gdcm::network::ULAction:



Public Member Functions

- ULAction ()

- virtual ~ULAction ()
- virtual EStateID PerformAction (Subject *s, ULEvent &inEvent, ULConnection &inConnection, bool &outWaiting↩ ForEvent, EEventID &outRaisedEvent)=0

10.313.1 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 thos 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.

10.313.2 Constructor & Destructor Documentation

10.313.2.1 ULAction()

```
gdcm::network::ULAction::ULAction ( )  [inline]
```

10.313.2.2 ~ULAction()

```
virtual gdcm::network::ULAction::~ULAction ( )  [inline], [virtual]
```

References PerformAction().

10.313.3 Member Function Documentation

10.313.3.1 PerformAction()

```
virtual EStateID gdcm::network::ULAction::PerformAction (
            Subject * s,
            ULEvent & inEvent,
            ULConnection & inConnection,
            bool & outWaitingForEvent,
            EEventID & outRaisedEvent )  [pure virtual]
```

Implemented in gdcm::network::ULActionAR10, gdcm::network::ULActionAR9, gdcm::network::ULActionAE8, gdcm::network::ULActionAA8, gdcm::network::ULActionAR8, gdcm::network::ULActionAE7, gdcm::network::ULActionAA7, gdcm::network::ULActionAR7, gdcm::network::ULActionAE6, gdcm::network::ULActionAA6, gdcm::network::ULActionAR6, gdcm::network::ULActionAA5, gdcm::network::ULActionAE5, gdcm::network::ULActionAR5, gdcm::network::ULActionAA4, gdcm::network::ULActionAE4, gdcm::network::ULActionAR4, gdcm::network::ULActionAA3, gdcm::network::ULActionAE3, gdcm::network::ULActionAR3, gdcm::network::ULActionAA2, gdcm::network::ULActionAE2, gdcm::network::ULActionAR2, gdcm::network::ULActionDT2, gdcm::network::ULActionAA1, gdcm::network::ULActionAE1, gdcm::network::ULActionAR1, and gdcm::network::ULActionDT1.

Referenced by ~ULAction().

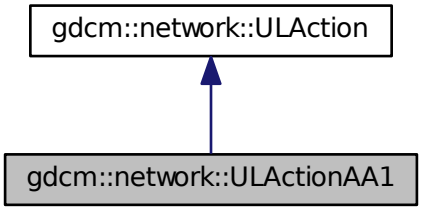
The documentation for this class was generated from the following file:

- gdcmULAction.h

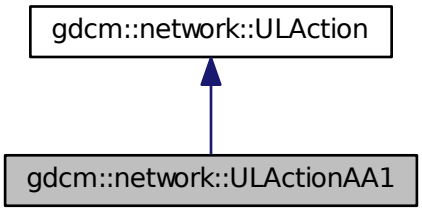
10.314 gdcm::network::ULActionAA1 Class Reference

```
#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)

10.314.1 Member Function Documentation

10.314.1.1 PerformAction()

```
EStateID gdcm::network::ULActionAA1::PerformAction (
            Subject * s,
            ULEvent & inEvent,
            ULConnection & inConnection,
            bool & outWaitingForEvent,
            EEventID & outRaisedEvent )  [virtual]
```

Implements gdcm::network::ULAction.

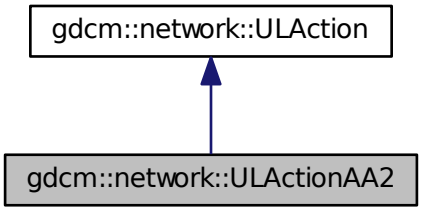
The documentation for this class was generated from the following file:

- gdcmULActionAA.h

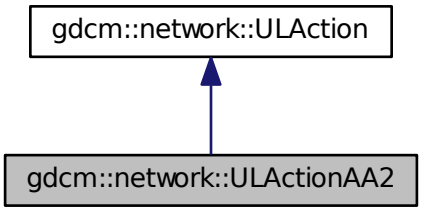
10.315 gdcm::network::ULActionAA2 Class Reference

```
#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)

10.315.1 Member Function Documentation

10.315.1.1 PerformAction()

```
EStateID gdcm::network::ULActionAA2::PerformAction (
            Subject * s,
            ULEvent & inEvent,
            ULConnection & inConnection,
            bool & outWaitingForEvent,
            EEventID & outRaisedEvent )  [virtual]
```

Implements gdcm::network::ULAction.

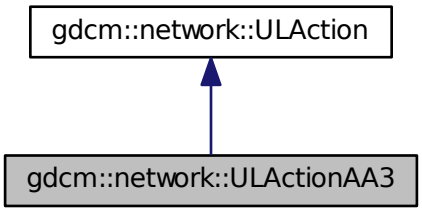
The documentation for this class was generated from the following file:

- gdcmULActionAA.h

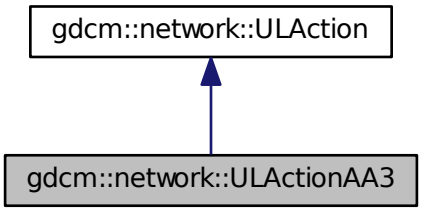
10.316 gdcm::network::ULActionAA3 Class Reference

`#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 &outWaitingForEvent, EEventID &outRaisedEvent)

10.316.1 Member Function Documentation

10.316.1.1 PerformAction()

```
EStateID gdcm::network::ULActionAA3::PerformAction (
            Subject * s,
```

```
            ULEvent & inEvent,
            ULConnection & inConnection,
            bool & outWaitingForEvent,
            EEventID & outRaisedEvent )  [virtual]
```

Implements gdcm::network::ULAction.

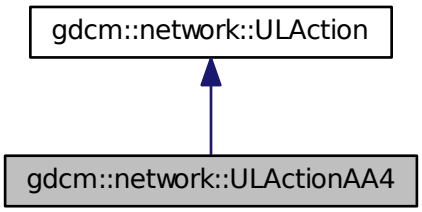
The documentation for this class was generated from the following file:

- gdcmULActionAA.h

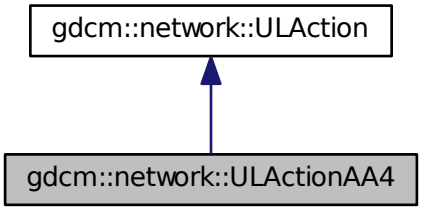
10.317 gdcm::network::ULActionAA4 Class Reference

`#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)

10.317.1 Member Function Documentation

10.317.1.1 PerformAction()

```
EStateID gdcm::network::ULActionAA4::PerformAction (
            Subject * s,
            ULEvent & inEvent,
            ULConnection & inConnection,
            bool & outWaitingForEvent,
            EEventID & outRaisedEvent )  [virtual]
```

Implements gdcm::network::ULAction.

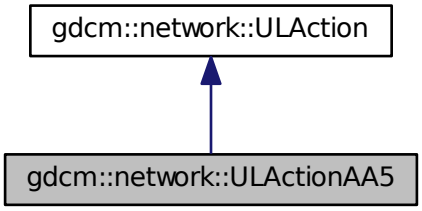
The documentation for this class was generated from the following file:

- gdcmULActionAA.h

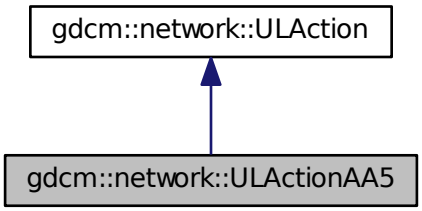
10.318 gdcm::network::ULActionAA5 Class Reference

```
#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)

10.318.1 Member Function Documentation

10.318.1.1 PerformAction()

```
EStateID gdcm::network::ULActionAA5::PerformAction (
            Subject * s,
            ULEvent & inEvent,
            ULConnection & inConnection,
            bool & outWaitingForEvent,
            EEventID & outRaisedEvent )  [virtual]
```

Implements gdcm::network::ULAction.

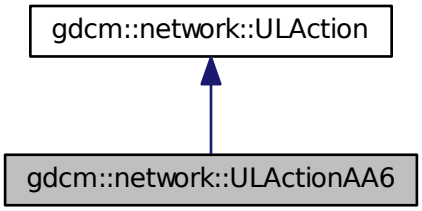
The documentation for this class was generated from the following file:

- gdcmULActionAA.h

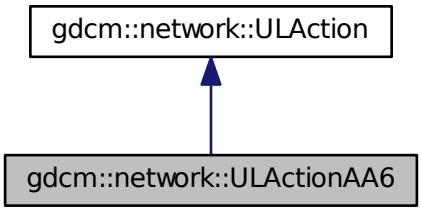
10.319 gdcm::network::ULActionAA6 Class Reference

```
#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)

10.319.1 Member Function Documentation

10.319.1.1 PerformAction()

```
EStateID gdcm::network::ULActionAA6::PerformAction (
            Subject * s,
            ULEvent & inEvent,
            ULConnection & inConnection,
            bool & outWaitingForEvent,
            EEventID & outRaisedEvent )  [virtual]
```

Implements gdcm::network::ULAction.

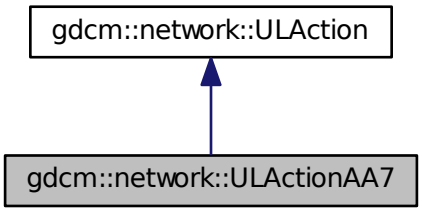
The documentation for this class was generated from the following file:

- gdcmULActionAA.h

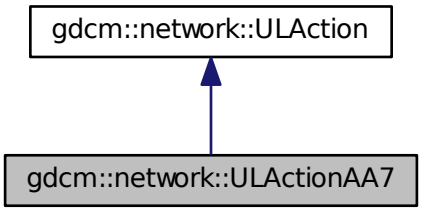
10.320 gdcm::network::ULActionAA7 Class Reference

`#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 &outWaitingFor↩ Event, EEventID &outRaisedEvent)

10.320.1 Member Function Documentation

10.320.1.1 PerformAction()

```
EStateID gdcm::network::ULActionAA7::PerformAction (
            Subject * s,
```

```
            ULEvent & inEvent,
            ULConnection & inConnection,
            bool & outWaitingForEvent,
            EEventID & outRaisedEvent )  [virtual]
```

Implements gdcm::network::ULAction.

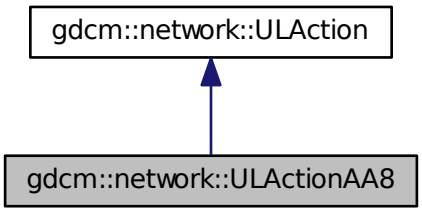
The documentation for this class was generated from the following file:

- gdcmULActionAA.h

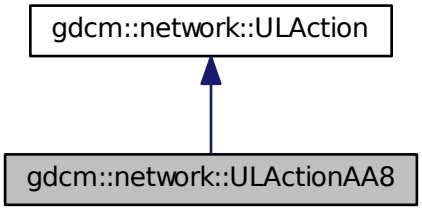
10.321 gdcm::network::ULActionAA8 Class Reference

`#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 &outWaitingForEvent, EEventID &outRaisedEvent)

10.321.1 Member Function Documentation

10.321.1.1 PerformAction()

```
EStateID gdcm::network::ULActionAA8::PerformAction (
            Subject * s,
            ULEvent & inEvent,
            ULConnection & inConnection,
            bool & outWaitingForEvent,
            EEventID & outRaisedEvent )  [virtual]
```

Implements gdcm::network::ULAction.

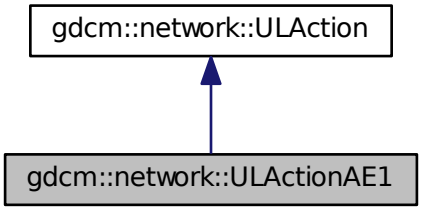
The documentation for this class was generated from the following file:

- gdcmULActionAA.h

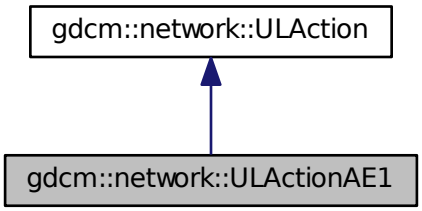
10.322 gdcm::network::ULActionAE1 Class Reference

```
#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)

10.322.1 Member Function Documentation

10.322.1.1 PerformAction()

```
EStateID gdcm::network::ULActionAE1::PerformAction (
            Subject * s,
            ULEvent & inEvent,
            ULConnection & inConnection,
            bool & outWaitingForEvent,
            EEventID & outRaisedEvent )  [virtual]
```

Implements gdcm::network::ULAction.

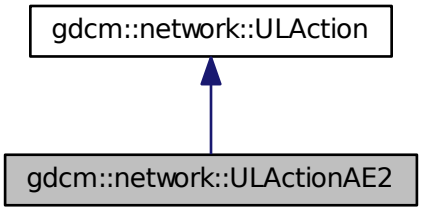
The documentation for this class was generated from the following file:

- gdcmULActionAE.h

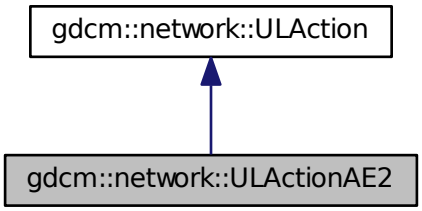
10.323 gdcm::network::ULActionAE2 Class Reference

```
#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 &outWaitingFor↩ Event, EEventID &outRaisedEvent)

10.323.1 Member Function Documentation

10.323.1.1 PerformAction()

```
EStateID gdcm::network::ULActionAE2::PerformAction (
            Subject * s,
            ULEvent & inEvent,
            ULConnection & inConnection,
            bool & outWaitingForEvent,
            EEventID & outRaisedEvent )  [virtual]
```

Implements gdcm::network::ULAction.

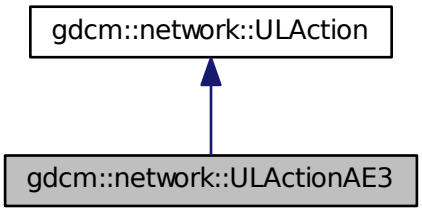
The documentation for this class was generated from the following file:

- gdcmULActionAE.h

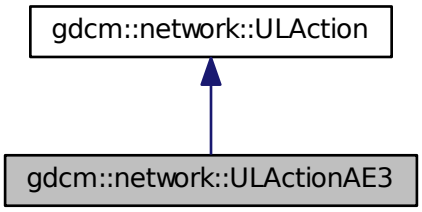
10.324 gdcm::network::ULActionAE3 Class Reference

`#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 &outWaitingForEvent, EEventID &outRaisedEvent)

10.324.1 Member Function Documentation

10.324.1.1 PerformAction()

```
EStateID gdcm::network::ULActionAE3::PerformAction (
            Subject * s,
```

```
            ULEvent & inEvent,
            ULConnection & inConnection,
            bool & outWaitingForEvent,
            EEventID & outRaisedEvent )  [virtual]
```

Implements gdcm::network::ULAction.

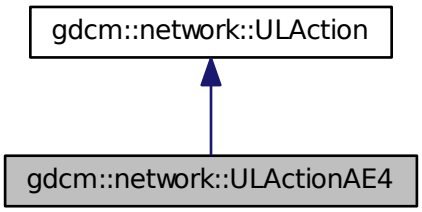
The documentation for this class was generated from the following file:

- gdcmULActionAE.h

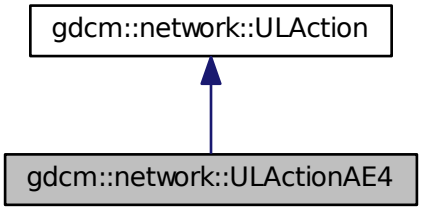
10.325 gdcm::network::ULActionAE4 Class Reference

```
#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)

10.325.1 Member Function Documentation

10.325.1.1 PerformAction()

```
EStateID gdcm::network::ULActionAE4::PerformAction (
            Subject * s,
            ULEvent & inEvent,
            ULConnection & inConnection,
            bool & outWaitingForEvent,
            EEventID & outRaisedEvent )  [virtual]
```

Implements gdcm::network::ULAction.

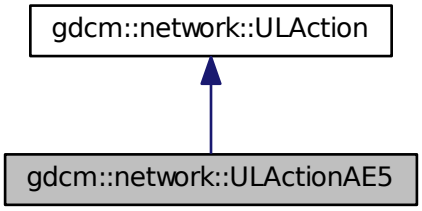
The documentation for this class was generated from the following file:

- gdcmULActionAE.h

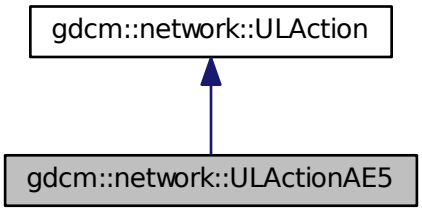
10.326 gdcm::network::ULActionAE5 Class Reference

```
#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 &outWaitingFor↩ Event, EEventID &outRaisedEvent)

10.326.1 Member Function Documentation

10.326.1.1 PerformAction()

```
EStateID gdcm::network::ULActionAE5::PerformAction (
            Subject * s,
            ULEvent & inEvent,
            ULConnection & inConnection,
            bool & outWaitingForEvent,
            EEventID & outRaisedEvent )  [virtual]
```

Implements gdcm::network::ULAction.

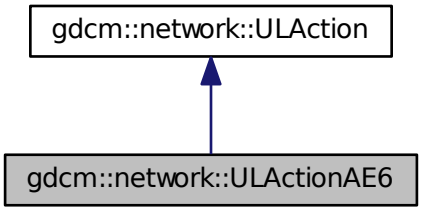
The documentation for this class was generated from the following file:

- gdcmULActionAE.h

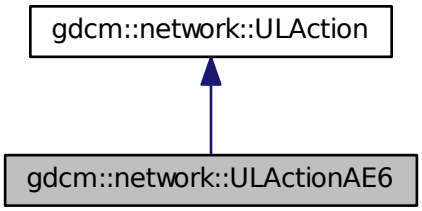
10.327 gdcm::network::ULActionAE6 Class Reference

```
#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)

10.327.1 Member Function Documentation

10.327.1.1 PerformAction()

```
EStateID gdcm::network::ULActionAE6::PerformAction (
            Subject * s,
            ULEvent & inEvent,
            ULConnection & inConnection,
            bool & outWaitingForEvent,
            EEventID & outRaisedEvent )  [virtual]
```

Implements gdcm::network::ULAction.

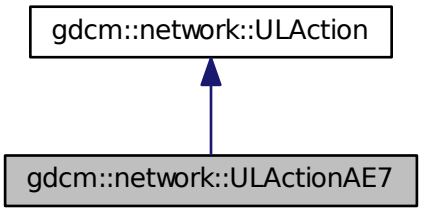
The documentation for this class was generated from the following file:

- gdcmULActionAE.h

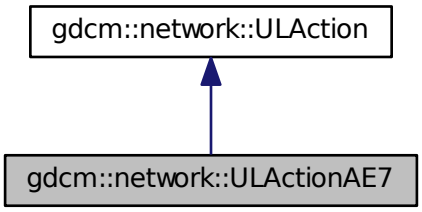
10.328 gdcm::network::ULActionAE7 Class Reference

`#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)

10.328.1 Member Function Documentation

10.328.1.1 PerformAction()

```
EStateID gdcm::network::ULActionAE7::PerformAction (
            Subject * s,
```

```
            ULEvent & inEvent,
            ULConnection & inConnection,
            bool & outWaitingForEvent,
            EEventID & outRaisedEvent )  [virtual]
```

Implements gdcm::network::ULAction.

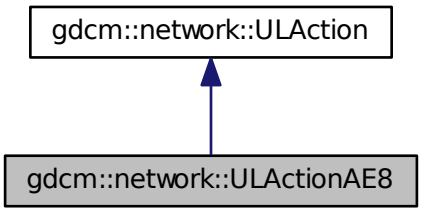
The documentation for this class was generated from the following file:

- gdcmULActionAE.h

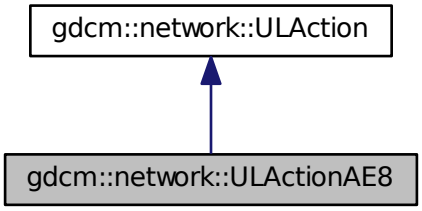
10.329 gdcm::network::ULActionAE8 Class Reference

```
#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)

10.329.1 Member Function Documentation

10.329.1.1 PerformAction()

```
EStateID gdcm::network::ULActionAE8::PerformAction (
            Subject * s,
            ULEvent & inEvent,
            ULConnection & inConnection,
            bool & outWaitingForEvent,
            EEventID & outRaisedEvent )  [virtual]
```

Implements gdcm::network::ULAction.

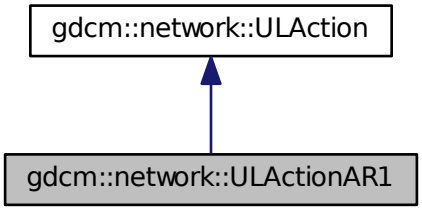
The documentation for this class was generated from the following file:

- gdcmULActionAE.h

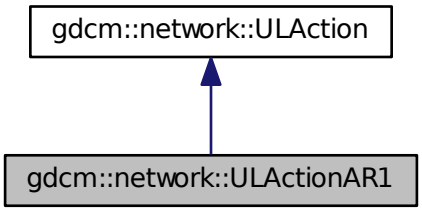
10.330 gdcm::network::ULActionAR1 Class Reference

```
#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)

10.330.1 Member Function Documentation

10.330.1.1 PerformAction()

```
EStateID gdcm::network::ULActionAR1::PerformAction (
            Subject * s,
            ULEvent & inEvent,
            ULConnection & inConnection,
            bool & outWaitingForEvent,
            EEventID & outRaisedEvent )  [virtual]
```

Implements gdcm::network::ULAction.

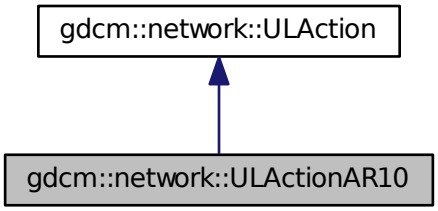
The documentation for this class was generated from the following file:

- gdcmULActionAR.h

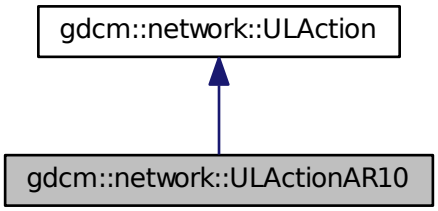
10.331 gdcm::network::ULActionAR10 Class Reference

```
#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)

10.331.1 Member Function Documentation

10.331.1.1 PerformAction()

```
EStateID gdcm::network::ULActionAR10::PerformAction (
            Subject * s,
            ULEvent & inEvent,
            ULConnection & inConnection,
            bool & outWaitingForEvent,
            EEventID & outRaisedEvent )  [virtual]
```

Implements gdcm::network::ULAction.

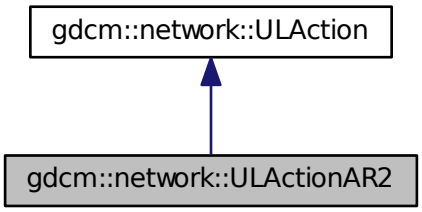
The documentation for this class was generated from the following file:

- gdcmULActionAR.h

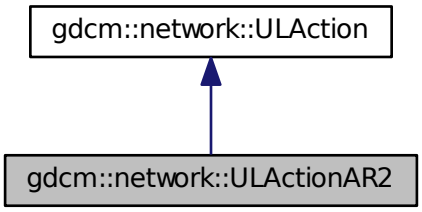
10.332 gdcm::network::ULActionAR2 Class Reference

`#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)

10.332.1 Member Function Documentation

10.332.1.1 PerformAction()

```
EStateID gdcm::network::ULActionAR2::PerformAction (
            Subject * s,
```

```
            ULEvent & inEvent,
            ULConnection & inConnection,
            bool & outWaitingForEvent,
            EEventID & outRaisedEvent )  [virtual]
```

Implements gdcm::network::ULAction.

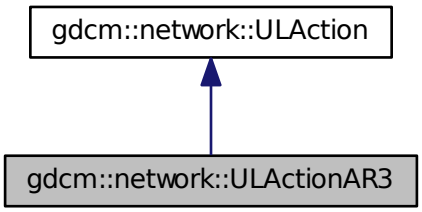
The documentation for this class was generated from the following file:

- gdcmULActionAR.h

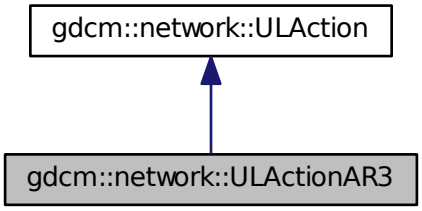
10.333 gdcm::network::ULActionAR3 Class Reference

```
#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)

10.333.1 Member Function Documentation

10.333.1.1 PerformAction()

```
EStateID gdcm::network::ULActionAR3::PerformAction (
            Subject * s,
            ULEvent & inEvent,
            ULConnection & inConnection,
            bool & outWaitingForEvent,
            EEventID & outRaisedEvent )  [virtual]
```

Implements gdcm::network::ULAction.

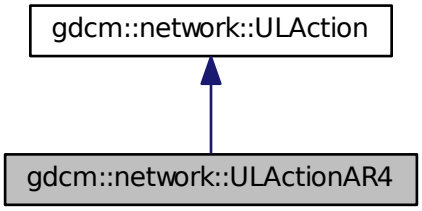
The documentation for this class was generated from the following file:

- gdcmULActionAR.h

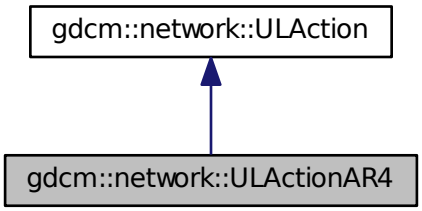
10.334 gdcm::network::ULActionAR4 Class Reference

```
#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)

10.334.1 Member Function Documentation

10.334.1.1 PerformAction()

```
EStateID gdcm::network::ULActionAR4::PerformAction (
            Subject * s,
            ULEvent & inEvent,
            ULConnection & inConnection,
            bool & outWaitingForEvent,
            EEventID & outRaisedEvent )  [virtual]
```

Implements gdcm::network::ULAction.

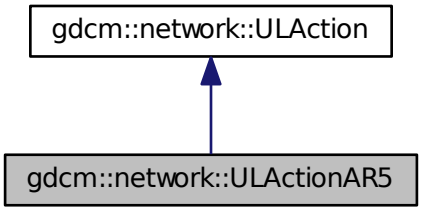
The documentation for this class was generated from the following file:

- gdcmULActionAR.h

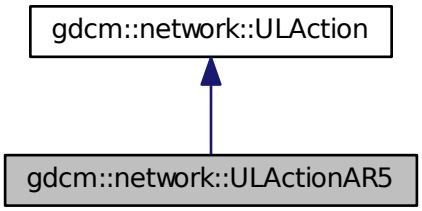
10.335 gdcm::network::ULActionAR5 Class Reference

```
#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 &outWaitingFor↩ Event, EEventID &outRaisedEvent)

10.335.1 Member Function Documentation

10.335.1.1 PerformAction()

```
EStateID gdcm::network::ULActionAR5::PerformAction (
            Subject * s,
            ULEvent & inEvent,
            ULConnection & inConnection,
            bool & outWaitingForEvent,
            EEventID & outRaisedEvent )  [virtual]
```

Implements gdcm::network::ULAction.

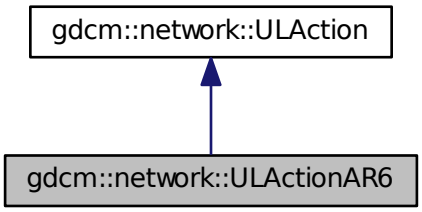
The documentation for this class was generated from the following file:

- gdcmULActionAR.h

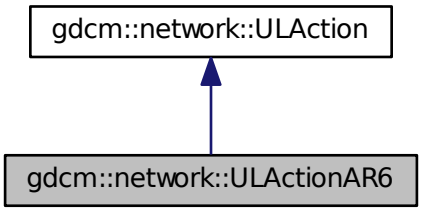
10.336 gdcm::network::ULActionAR6 Class Reference

`#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)

10.336.1 Member Function Documentation

10.336.1.1 PerformAction()

```
EStateID gdcm::network::ULActionAR6::PerformAction (
            Subject * s,
```

```
            ULEvent & inEvent,
            ULConnection & inConnection,
            bool & outWaitingForEvent,
            EEventID & outRaisedEvent )  [virtual]
```

Implements gdcm::network::ULAction.

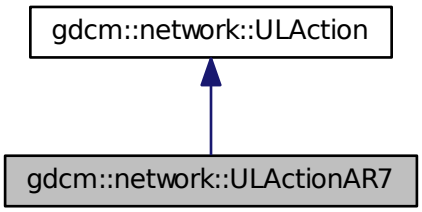
The documentation for this class was generated from the following file:

- gdcmULActionAR.h

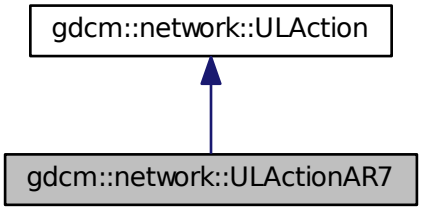
10.337 gdcm::network::ULActionAR7 Class Reference

`#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 &outWaitingForEvent, EEventID &outRaisedEvent)

10.337.1 Member Function Documentation

10.337.1.1 PerformAction()

```
EStateID gdcm::network::ULActionAR7::PerformAction (
            Subject * s,
            ULEvent & inEvent,
            ULConnection & inConnection,
            bool & outWaitingForEvent,
            EEventID & outRaisedEvent )  [virtual]
```

Implements gdcm::network::ULAction.

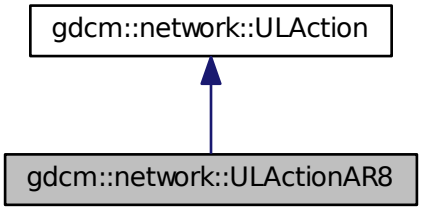
The documentation for this class was generated from the following file:

- gdcmULActionAR.h

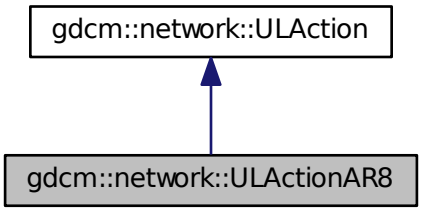
10.338 gdcm::network::ULActionAR8 Class Reference

```
#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 &outWaitingFor↩ Event, EEventID &outRaisedEvent)

10.338.1 Member Function Documentation

10.338.1.1 PerformAction()

```
EStateID gdcm::network::ULActionAR8::PerformAction (
            Subject * s,
            ULEvent & inEvent,
            ULConnection & inConnection,
            bool & outWaitingForEvent,
            EEventID & outRaisedEvent )  [virtual]
```

Implements gdcm::network::ULAction.

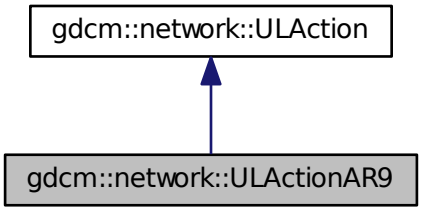
The documentation for this class was generated from the following file:

- gdcmULActionAR.h

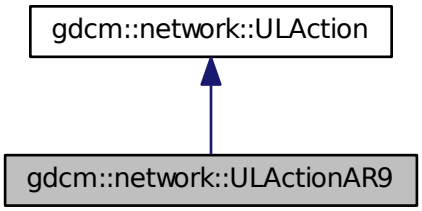
10.339 gdcm::network::ULActionAR9 Class Reference

```
#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)

10.339.1 Member Function Documentation

10.339.1.1 PerformAction()

```
EStateID gdcm::network::ULActionAR9::PerformAction (
            Subject * s,
            ULEvent & inEvent,
            ULConnection & inConnection,
            bool & outWaitingForEvent,
            EEventID & outRaisedEvent )  [virtual]
```

Implements gdcm::network::ULAction.

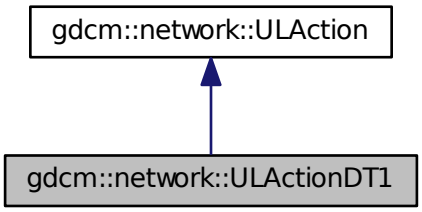
The documentation for this class was generated from the following file:

- gdcmULActionAR.h

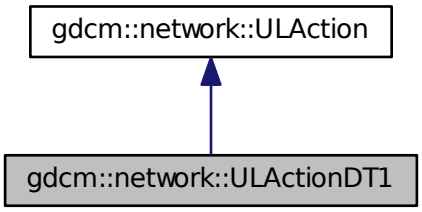
10.340 gdcm::network::ULActionDT1 Class Reference

`#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 &outWaitingForEvent, EEventID &outRaisedEvent)

10.340.1 Member Function Documentation

10.340.1.1 PerformAction()

```
EStateID gdcm::network::ULActionDT1::PerformAction (
            Subject * s,
```

```
            ULEvent & inEvent,
            ULConnection & inConnection,
            bool & outWaitingForEvent,
            EEventID & outRaisedEvent )  [virtual]
```

Implements gdcm::network::ULAction.

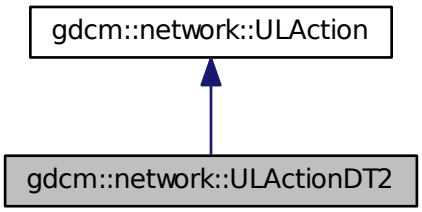
The documentation for this class was generated from the following file:

- gdcmULActionDT.h

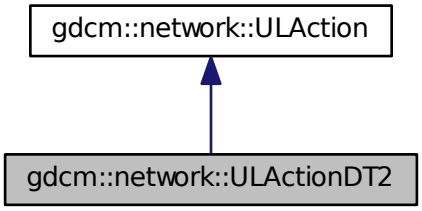
10.341 gdcm::network::ULActionDT2 Class Reference

`#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 &outWaitingFor↩ Event, EEventID &outRaisedEvent)

10.341.1 Member Function Documentation

10.341.1.1 PerformAction()

```
EStateID gdcm::network::ULActionDT2::PerformAction (
            Subject * s,
            ULEvent & inEvent,
            ULConnection & inConnection,
            bool & outWaitingForEvent,
            EEventID & outRaisedEvent )  [virtual]
```

Implements gdcm::network::ULAction.

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

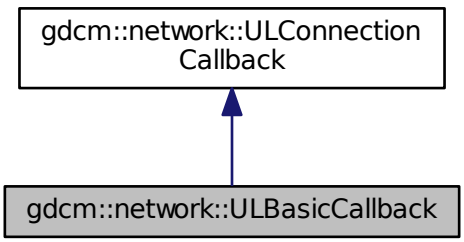
- gdcmULActionDT.h

10.342 gdcm::network::ULBasicCallback Class Reference

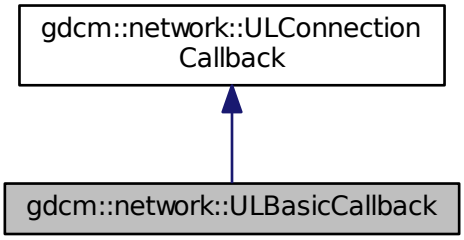
ULBasicCallback.

```
#include <gdcmULBasicCallback.h>
```

Inheritance diagram for gdcm::network::ULBasicCallback:



Collaboration diagram for gdcm::network::ULBasicCallback:



Public Member Functions

- ULBasicCallback ()
- virtual ~ULBasicCallback ()
- std::vector< DataSet > const & GetDataSets () const
- std::vector< DataSet > const & GetResponses () const
- virtual void HandleDataSet (const DataSet &inDataSet)
- virtual void HandleResponse (const DataSet &inDataSet)

Additional Inherited Members

10.342.1 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.

10.342.2 Constructor & Destructor Documentation

10.342.2.1 ULBasicCallback()

```
gdcm::network::ULBasicCallback::ULBasicCallback ( )  [inline]
```

10.342.2.2 ~ULBasicCallback()

```
virtual gdcm::network::ULBasicCallback::~ULBasicCallback ( )  [inline], [virtual]
```

10.342.3 Member Function Documentation

10.342.3.1 GetDataSets()

```
std::vector<DataSet> const& gdcm::network::ULBasicCallback::GetDataSets ( ) const
```

10.342.3.2 GetResponses()

```
std::vector<DataSet> const& gdcm::network::ULBasicCallback::GetResponses ( ) const
```

10.342.3.3 HandleDataSet()

```
virtual void gdcm::network::ULBasicCallback::HandleDataSet (
            const DataSet & inDataSet )  [virtual]
```

Implements gdcm::network::ULConnectionCallback.

10.342.3.4 HandleResponse()

```
virtual void gdcm::network::ULBasicCallback::HandleResponse (
            const DataSet & inDataSet )  [virtual]
```

Implements gdcm::network::ULConnectionCallback.

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

- gdcmULBasicCallback.h

10.343 gdcm::network::ULConnection Class Reference

ULConnection.

```
#include <gdcmULConnection.h>
```

Public Member Functions

- ULConnection (const ULConnectionInfo &inUserInformation)
- virtual ~ULConnection ()
- void AddAcceptedPresentationContext (const PresentationContextAC &inPC)
- PresentationContextRQ FindContext (const DataElement &de) const
- std::vector< PresentationContextAC > const & GetAcceptedPresentationContexts () const
- std::vector< PresentationContextAC > & GetAcceptedPresentationContexts ()
- 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 SetMaxPDUSize (uint32_t inSize)
- void SetPresentationContexts (const std::vector< PresentationContextRQ > &inContexts)
- void SetPresentationContexts (const std::vector< PresentationContext > &inContexts)
- void SetState (const EStateID &inState)
- void StopProtocol ()

Friends

- class ULActionAE6
- class ULConnectionManager

10.343.1 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.

10.343.2 Constructor & Destructor Documentation

10.343.2.1 ULConnection()

```
gdcm::network::ULConnection::ULConnection (
            const ULConnectionInfo & inUserInformation )
```

10.343.2.2 ~ULConnection()

```
virtual gdcm::network::ULConnection::~ULConnection ( )  [virtual]
```

10.343.3 Member Function Documentation

10.343.3.1 AddAcceptedPresentationContext()

```
void gdcm::network::ULConnection::AddAcceptedPresentationContext (
            const PresentationContextAC & inPC )
```

10.343.3.2 FindContext()

```
PresentationContextRQ gdcm::network::ULConnection::FindContext (
            const DataElement & de ) const
```

10.343.3.3 GetAcceptedPresentationContexts() [1/2]

```
std::vector<PresentationContextAC> const& gdcm::network::ULConnection::GetAcceptedPresentation↩
Contexts ( ) const
```

10.343.3.4 GetAcceptedPresentationContexts() [2/2]

```
std::vector<PresentationContextAC>& gdcm::network::ULConnection::GetAcceptedPresentationContexts
( )
```

10.343.3.5 GetConnectionInfo()

```
const ULConnectionInfo& gdcm::network::ULConnection::GetConnectionInfo ( ) const
```

10.343.3.6 GetMaxPDUSize()

```
uint32_t gdcm::network::ULConnection::GetMaxPDUSize ( ) const
```

10.343.3.7 GetPresentationContextACByID()

```
const PresentationContextAC* gdcm::network::ULConnection::GetPresentationContextACByID (
            uint8_t id ) const
```

10.343.3.8 GetPresentationContextIDFromPresentationContext()

```
uint8_t gdcm::network::ULConnection::GetPresentationContextIDFromPresentationContext (
            PresentationContextRQ const & pc ) const
```

return 0 upon error

10.343.3.9 GetPresentationContextRQByID()

```
const PresentationContextRQ* gdcm::network::ULConnection::GetPresentationContextRQByID (
            uint8_t id ) const
```

10.343.3.10 GetPresentationContexts()

```
std::vector<PresentationContextRQ> const& gdcm::network::ULConnection::GetPresentationContexts (
) const
```

10.343.3.11 GetProtocol()

```
std::iostream* gdcm::network::ULConnection::GetProtocol ( )
```

10.343.3.12 GetState()

```
EStateID gdcm::network::ULConnection::GetState ( ) const
```

10.343.3.13 GetTimer()

```
ARTIMTimer& gdcm::network::ULConnection::GetTimer ( )
```

10.343.3.14 InitializeConnection()

```
bool gdcm::network::ULConnection::InitializeConnection ( )
```

used to establish scu connections

10.343.3.15 InitializeIncomingConnection()

```
bool gdcm::network::ULConnection::InitializeIncomingConnection ( )
```

used to establish scp connections

10.343.3.16 SetMaxPDUSize()

```
void gdcm::network::ULConnection::SetMaxPDUSize (
            uint32_t inSize )
```

10.343.3.17 SetPresentationContexts() [1/2]

```
void gdcm::network::ULConnection::SetPresentationContexts (
            const std::vector< PresentationContextRQ > & inContexts )
```

10.343.3.18 SetPresentationContexts() [2/2]

```
void gdcm::network::ULConnection::SetPresentationContexts (
            const std::vector< PresentationContext > & inContexts )
```

10.343.3.19 SetState()

```
void gdcm::network::ULConnection::SetState (
            const EStateID & inState )
```

10.343.3.20 StopProtocol()

```
void gdcm::network::ULConnection::StopProtocol ( )
```

10.343.4 Friends And Related Function Documentation

10.343.4.1 ULActionAE6

```
friend class ULActionAE6  [friend]
```

10.343.4.2 ULConnectionManager

```
friend class ULConnectionManager  [friend]
```

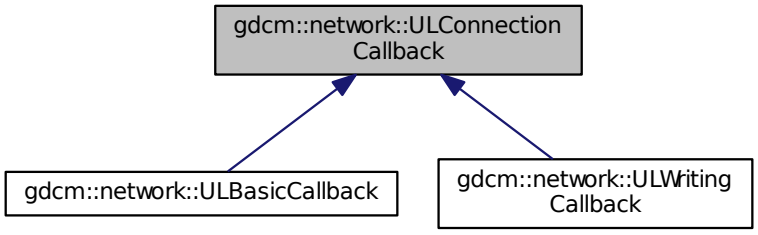
The documentation for this class was generated from the following file:

- gdcmULConnection.h

10.344 gdcm::network::ULConnectionCallback Class Reference

`#include <gdcmULConnectionCallback.h>`

Inheritance diagram for gdcm::network::ULConnectionCallback:



Public Member Functions

- ULConnectionCallback ()
- virtual ~ULConnectionCallback ()
- 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

10.344.1 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.

10.344.2 Constructor & Destructor Documentation

10.344.2.1 ULConnectionCallback()

```
gdcm::network::ULConnectionCallback::ULConnectionCallback ( )  [inline]
```

10.344.2.2 ~ULConnectionCallback()

```
virtual gdcm::network::ULConnectionCallback::~ULConnectionCallback ( )  [inline], [virtual]
```

10.344.3 Member Function Documentation

10.344.3.1 DataSetHandled()

```
void gdcm::network::ULConnectionCallback::DataSetHandled ( )  [inline], [protected]
```

10.344.3.2 DataSetHandles()

```
bool gdcm::network::ULConnectionCallback::DataSetHandles ( ) const  [inline]
```

10.344.3.3 HandleDataSet()

```
virtual void gdcm::network::ULConnectionCallback::HandleDataSet (
            const DataSet & inDataSet )  [pure virtual]
```

Implemented in gdcm::network::ULBasicCallback, and gdcm::network::ULWritingCallback.

10.344.3.4 HandleResponse()

```
virtual void gdcm::network::ULConnectionCallback::HandleResponse (
            const DataSet & inDataSet )  [pure virtual]
```

Implemented in gdcm::network::ULBasicCallback, and gdcm::network::ULWritingCallback.

10.344.3.5 ResetHandledDataSet()

```
void gdcm::network::ULConnectionCallback::ResetHandledDataSet ( )  [inline]
```

10.344.3.6 SetImplicitFlag()

```
void gdcm::network::ULConnectionCallback::SetImplicitFlag (
            const bool imp )  [inline]
```

10.344.4 Member Data Documentation

10.344.4.1 mImplicit

```
bool gdcm::network::ULConnectionCallback::mImplicit  [protected]
```

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

- gdcmULConnectionCallback.h

10.345 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 *inCallingA↩ ETitle, unsigned long inCalledIPAddress, int inCalledIPPort, std::string inCalledComputerName)
- void SetMaxPDULength (unsigned long inMaxPDULength)

10.345.1 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.

10.345.2 Constructor & Destructor Documentation

10.345.2.1 ULConnectionInfo()

```
gdcm::network::ULConnectionInfo::ULConnectionInfo ( )
```

10.345.3 Member Function Documentation

10.345.3.1 GetCalledAETitle()

```
const char* gdcm::network::ULConnectionInfo::GetCalledAETitle ( ) const
```

10.345.3.2 GetCalledComputerName()

```
std::string gdcm::network::ULConnectionInfo::GetCalledComputerName ( ) const
```

10.345.3.3 GetCalledIPAddress()

```
unsigned long gdcm::network::ULConnectionInfo::GetCalledIPAddress ( ) const
```

10.345.3.4 GetCalledIPPort()

```
int gdcm::network::ULConnectionInfo::GetCalledIPPort ( ) const
```

10.345.3.5 GetCallingAETitle()

```
const char* gdcm::network::ULConnectionInfo::GetCallingAETitle ( ) const
```

10.345.3.6 GetMaxPDULength()

```
unsigned long gdcm::network::ULConnectionInfo::GetMaxPDULength ( ) const
```

10.345.3.7 Initialize()

```
bool gdcm::network::ULConnectionInfo::Initialize (
             UserInformation const & inUserInformation,
             const char * inCalledAETitle,
             const char * inCallingAETitle,
             unsigned long inCalledIPAddress,
             int inCalledIPPort,
             std::string inCalledComputerName )
```

10.345.3.8 SetMaxPDULength()

```
void gdcm::network::ULConnectionInfo::SetMaxPDULength (
            unsigned long inMaxPDULength )
```

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

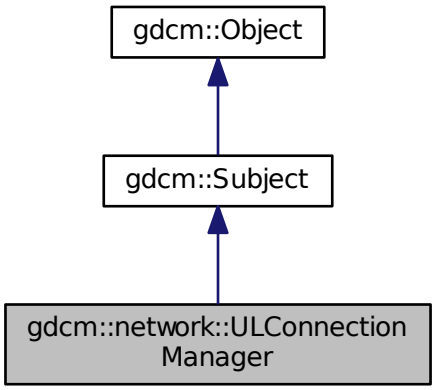
- gdcmULConnectionInfo.h

10.346 gdcm::network::ULConnectionManager Class Reference

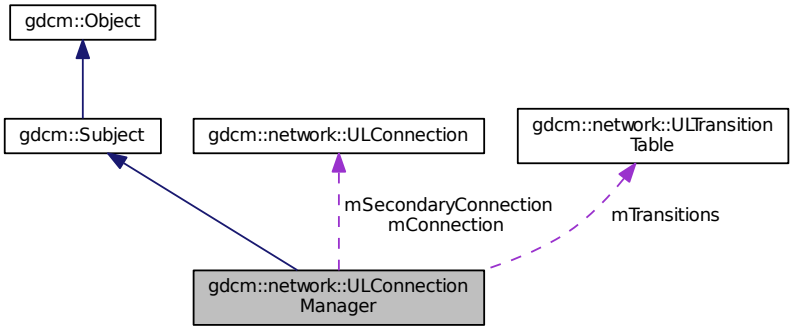
ULConnectionManager.

`#include <gdcmULConnectionManager.h>`

Inheritance diagram for gdcm::network::ULConnectionManager:



Collaboration diagram for gdcm::network::ULConnectionManager:



Public Member Functions

- ULConnectionManager ()
- virtual ~ULConnectionManager ()
- 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=NULL, std::streampos dataSetOffset=0)
- void SendStore (const File &file, ULConnectionCallback *inCallback, std::istream *pStream=NULL, 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

10.346.1 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.

10.346.2 Constructor & Destructor Documentation

10.346.2.1 ULConnectionManager() [1/2]

```
gdcm::network::ULConnectionManager::ULConnectionManager (
            const ULConnectionManager & inCM )  [protected]
```

10.346.2.2 ULConnectionManager() [2/2]

```
gdcm::network::ULConnectionManager::ULConnectionManager ( )
```

10.346.2.3 ~ULConnectionManager()

```
virtual gdcm::network::ULConnectionManager::~ULConnectionManager ( )  [virtual]
```

10.346.3 Member Function Documentation

10.346.3.1 BreakConnection()

```
bool gdcm::network::ULConnectionManager::BreakConnection (
            const double & inTimeout )
```

10.346.3.2 BreakConnectionNow()

```
void gdcm::network::ULConnectionManager::BreakConnectionNow ( )
```

10.346.3.3 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.

10.346.3.4 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

10.346.3.5 RunEventLoop()

```
EStateID gdcm::network::ULConnectionManager::RunEventLoop (
            ULEvent & inEvent,
            ULConnection * inWhichConnection,
            ULConnectionCallback * inCallback,
            const bool & startWaiting )  [protected]
```

10.346.3.6 RunMoveEventLoop()

```
EStateID gdcm::network::ULConnectionManager::RunMoveEventLoop (
            ULEvent & inEvent,
            ULConnectionCallback * inCallback )  [protected]
```

10.346.3.7 SendEcho()

```
std::vector<PresentationDataValue> gdcm::network::ULConnectionManager::SendEcho ( )
```

10.346.3.8 SendFind() [1/2]

```
std::vector<DataSet> gdcm::network::ULConnectionManager::SendFind (
            const BaseRootQuery * inRootQuery )
```

10.346.3.9 SendFind() [2/2]

```
void gdcm::network::ULConnectionManager::SendFind (
            const BaseRootQuery * inRootQuery,
            ULConnectionCallback * inCallback )
```

10.346.3.10 SendMove() [1/2]

```
std::vector<DataSet> gdcm::network::ULConnectionManager::SendMove (
            const BaseRootQuery * inRootQuery )
```

10.346.3.11 SendMove() [2/2]

```
bool gdcm::network::ULConnectionManager::SendMove (
            const BaseRootQuery * inRootQuery,
            ULConnectionCallback * inCallback )
```

return false upon error

10.346.3.12 SendNAction() [1/2]

```
std::vector<DataSet> gdcm::network::ULConnectionManager::SendNAction (
            const BaseQuery * inQuery )
```

10.346.3.13 SendNAction() [2/2]

```
void gdcm::network::ULConnectionManager::SendNAction (
            const BaseQuery * inQuery,
            ULConnectionCallback * inCallback )
```

10.346.3.14 SendNCreate() [1/2]

```
std::vector<DataSet> gdcm::network::ULConnectionManager::SendNCreate (
            const BaseQuery * inQuery )
```

10.346.3.15 SendNCreate() [2/2]

```
void gdcm::network::ULConnectionManager::SendNCreate (
            const BaseQuery * inQuery,
            ULConnectionCallback * inCallback )
```

10.346.3.16 SendNDelete() [1/2]

```
std::vector<DataSet> gdcm::network::ULConnectionManager::SendNDelete (
            const BaseQuery * inQuery )
```

10.346.3.17 SendNDelete() [2/2]

```
void gdcm::network::ULConnectionManager::SendNDelete (
            const BaseQuery * inQuery,
            ULConnectionCallback * inCallback )
```

10.346.3.18 SendNEventReport() [1/2]

```
std::vector<DataSet> gdcm::network::ULConnectionManager::SendNEventReport (
            const BaseQuery * inQuery )
```

10.346.3.19 SendNEventReport() [2/2]

```
void gdcm::network::ULConnectionManager::SendNEventReport (
            const BaseQuery * inQuery,
            ULConnectionCallback * inCallback )
```

10.346.3.20 SendNGet() [1/2]

```
std::vector<DataSet> gdcm::network::ULConnectionManager::SendNGet (
            const BaseQuery * inQuery )
```

10.346.3.21 SendNGet() [2/2]

```
void gdcm::network::ULConnectionManager::SendNGet (
            const BaseQuery * inQuery,
            ULConnectionCallback * inCallback )
```

10.346.3.22 SendNSet() [1/2]

```
std::vector<DataSet> gdcm::network::ULConnectionManager::SendNSet (
            const BaseQuery * inQuery )
```

10.346.3.23 SendNSet() [2/2]

```
void gdcm::network::ULConnectionManager::SendNSet (
            const BaseQuery * inQuery,
            ULConnectionCallback * inCallback )
```

10.346.3.24 SendStore() [1/2]

```
std::vector<DataSet> gdcm::network::ULConnectionManager::SendStore (
            const File & file,
            std::istream * pStream = NULL,
            std::streampos dataSetOffset = 0 )
```

10.346.3.25 SendStore() [2/2]

```
void gdcm::network::ULConnectionManager::SendStore (
            const File & file,
            ULConnectionCallback * inCallback,
            std::istream * pStream = NULL,
            std::streampos dataSetOffset = 0 )
```

callback based API

10.346.4 Member Data Documentation

10.346.4.1 mConnection

```
ULConnection* gdcm::network::ULConnectionManager::mConnection  [protected]
```

10.346.4.2 mSecondaryConnection

```
ULConnection* gdcm::network::ULConnectionManager::mSecondaryConnection  [protected]
```

10.346.4.3 mTransitions

```
ULTransitionTable gdcm::network::ULConnectionManager::mTransitions  [protected]
```

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

- gdcmULConnectionManager.h

10.347 gdcm::network::ULEvent Class Reference

ULEvent.

`#include <gdcmULEvent.h>`

Public Member Functions

- ULEvent (const EEventID &inEventID, std::vector< BasePDU *> inBasePDU, std::istream *iStream=NULL, std::streampos posDataSet=0)
- ULEvent (const EEventID &inEventID, BasePDU *inBasePDU, std::istream *iStream=NULL, 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)

10.347.1 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!

10.347.2 Constructor & Destructor Documentation

10.347.2.1 ULEvent() [1/2]

```
gdcm::network::ULEvent::ULEvent (
            const EEventID & inEventID,
            std::vector< BasePDU *> inBasePDU,
            std::istream * iStream = NULL,
            std::streampos posDataSet = 0 )  [inline]
```

10.347.2.2 ULEvent() [2/2]

```
gdcm::network::ULEvent::ULEvent (
            const EEventID & inEventID,
            BasePDU * inBasePDU,
            std::istream * iStream = NULL,
            std::streampos posDataSet = 0 )  [inline]
```

10.347.2.3 ~ULEvent()

```
gdcm::network::ULEvent::~ULEvent ( )  [inline]
```

10.347.3 Member Function Documentation

10.347.3.1 GetDataSetPos()

```
std::streampos gdcm::network::ULEvent::GetDataSetPos ( ) const  [inline]
```

10.347.3.2 GetEvent()

```
EEventID gdcm::network::ULEvent::GetEvent ( ) const  [inline]
```

10.347.3.3 GetIStream()

```
std::istream* gdcm::network::ULEvent::GetIStream ( ) const  [inline]
```

10.347.3.4 GetPDUs()

```
std::vector<BasePDU*> const& gdcm::network::ULEvent::GetPDUs ( ) const  [inline]
```

10.347.3.5 SetEvent()

```
void gdcm::network::ULEvent::SetEvent (
            const EEventID & inEvent )  [inline]
```

10.347.3.6 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

10.348 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

10.348.1 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 TransitionTable 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.

10.348.2 Constructor & Destructor Documentation

10.348.2.1 ULTransitionTable()

```
gdcm::network::ULTransitionTable::ULTransitionTable ( )
```

10.348.3 Member Function Documentation

10.348.3.1 HandleEvent()

```
void gdcm::network::ULTransitionTable::HandleEvent (
            Subject * s,
            ULEvent & inEvent,
            ULConnection & inConnection,
            bool & outWaitingForEvent,
            EEventID & outRaisedEvent ) const
```

10.348.3.2 PrintTable()

```
void gdcm::network::ULTransitionTable::PrintTable ( ) const
```

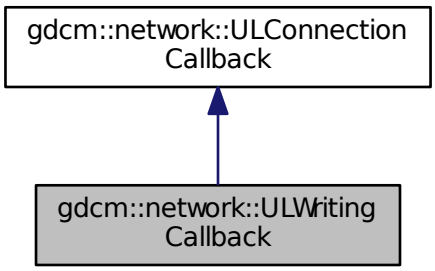
The documentation for this class was generated from the following file:

- gdcmULTransitionTable.h

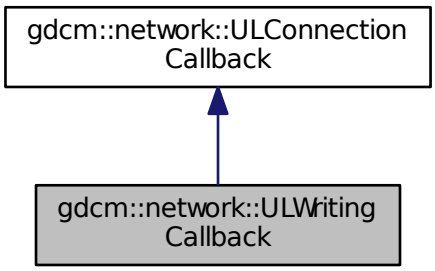
10.349 gdcm::network::ULWritingCallback Class Reference

`#include <gdcmULWritingCallback.h>`

Inheritance diagram for gdcm::network::ULWritingCallback:



Collaboration diagram for gdcm::network::ULWritingCallback:



Public Member Functions

- ULWritingCallback ()
- virtual ~ULWritingCallback ()
- virtual void HandleDataSet (const DataSet &inDataSet)
- virtual void HandleResponse (const DataSet &inDataSet)
- void SetDirectory (const std::string &inDirectoryName)

  provide the directory into which all files are written.

Additional Inherited Members

10.349.1 Constructor & Destructor Documentation

10.349.1.1 ULWritingCallback()

```
gdcm::network::ULWritingCallback::ULWritingCallback ( )  [inline]
```

10.349.1.2 ~ULWritingCallback()

```
virtual gdcm::network::ULWritingCallback::~ULWritingCallback ( )  [inline], [virtual]
```

10.349.2 Member Function Documentation

10.349.2.1 HandleDataSet()

```
virtual void gdcm::network::ULWritingCallback::HandleDataSet (
            const DataSet & inDataSet )  [virtual]
```

Implements gdcm::network::ULConnectionCallback.

10.349.2.2 HandleResponse()

```
virtual void gdcm::network::ULWritingCallback::HandleResponse (
            const DataSet & inDataSet )  [virtual]
```

Implements gdcm::network::ULConnectionCallback.

10.349.2.3 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:

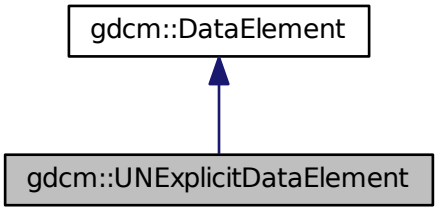
- gdcmULWritingCallback.h

10.350 gdcm::UNExplicitDataElement Class Reference

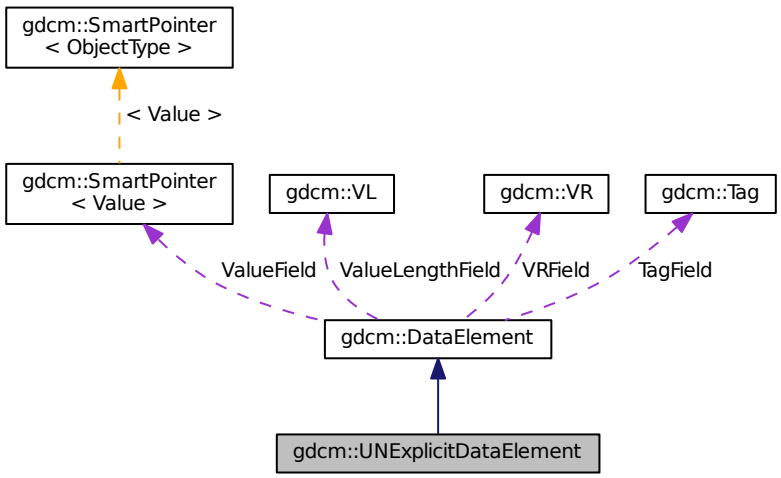
Class to read/write a DataElement as UNExplicit Data Element.

`#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

10.350.1 Detailed Description

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

Note

bla

10.350.2 Member Function Documentation

10.350.2.1 GetLength()

```
VL gdcm::UNExplicitDataElement::GetLength ( ) const
```

10.350.2.2 Read()

```
template<typename TSwap >
std::istream& gdcm::UNExplicitDataElement::Read (
             std::istream & is )
```

10.350.2.3 ReadPreValue()

```
template<typename TSwap >
std::istream& gdcm::UNExplicitDataElement::ReadPreValue (
             std::istream & is )
```

10.350.2.4 ReadValue()

```
template<typename TSwap >
std::istream& gdcm::UNExplicitDataElement::ReadValue (
             std::istream & is,
             bool readvalues = true )
```

10.350.2.5 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:

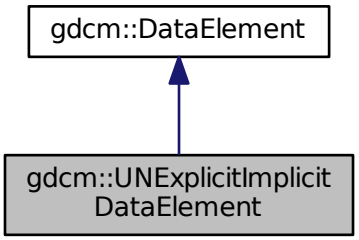
- gdcmUNExplicitDataElement.h

10.351 gdcm::UNExplicitImplicitDataElement Class Reference

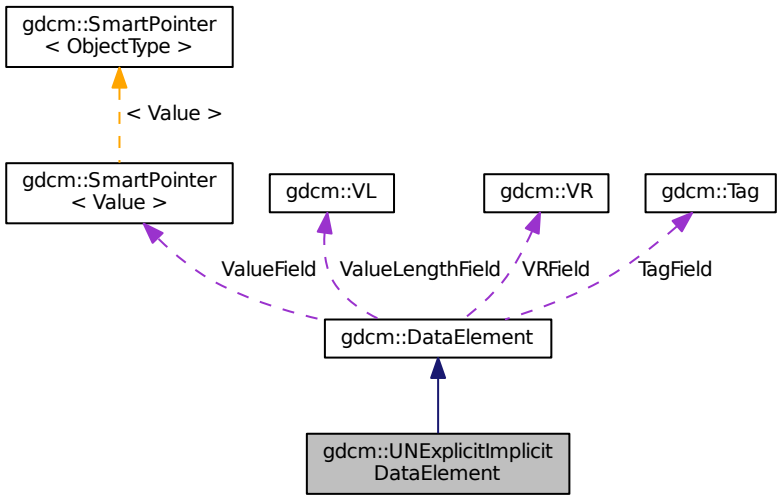
Class to read/write a DataElement as ExplicitImplicit Data Element.

```
#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

10.351.1 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

10.351.2 Member Function Documentation

10.351.2.1 GetLength()

```
VL gdcm::UNExplicitImplicitDataElement::GetLength ( ) const
```

10.351.2.2 Read()

```
template<typename TSwap >
std::istream& gdcm::UNExplicitImplicitDataElement::Read (
             std::istream & is )
```

10.351.2.3 ReadPreValue()

```
template<typename TSwap >
std::istream& gdcm::UNExplicitImplicitDataElement::ReadPreValue (
             std::istream & is )
```

10.351.2.4 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

10.352 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)

10.352.1 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

10.352.2 Member Function Documentation

10.352.2.1 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

10.352.2.2 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

10.353 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)

10.353.1 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.

10.353.2 Member Enumeration Documentation

10.353.2.1 UsageType

enum gdcm::Usage::UsageType

Enumerator

Mandatory	
Conditional	
UserOption	
Invalid	

10.353.3 Constructor & Destructor Documentation

10.353.3.1 Usage()

```
gdcm::Usage::Usage (
            UsageType type = Invalid )  [inline]
```

10.353.4 Member Function Documentation

10.353.4.1 GetUsageString()

```
static const char* gdcm::Usage::GetUsageString (
            UsageType type )  [static]
```

Referenced by gdcm::operator<<().

10.353.4.2 GetUsageType()

```
static UsageType gdcm::Usage::GetUsageType (
            const char * type )  [static]
```

10.353.4.3 operator UsageType()

```
gdcm::Usage::operator UsageType ( ) const  [inline]
```

References gdcm::operator<<().

10.353.5 Friends And Related Function Documentation

10.353.5.1 operator<<

```
std::ostream& operator<< (
            std::ostream & os,
            const Usage & vr )  [friend]
```

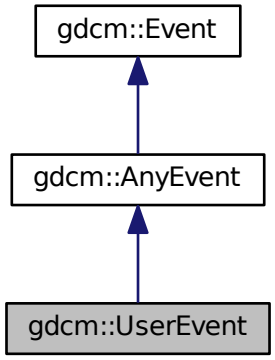
The documentation for this class was generated from the following file:

- gdcmUsage.h

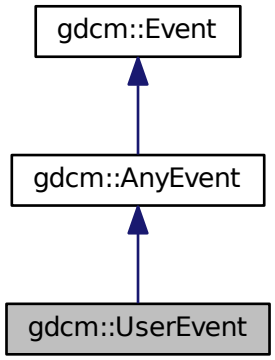
10.354 gdcm::UserEvent Class Reference

`#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

10.355 gdcm::network::UserInformation Class Reference

UserInformation.

`#include <gdcmUserInformation.h>`

Public Member Functions

- UserInformation ()
- ~UserInformation ()
- void AddRoleSelectionSub (RoleSelectionSub const &r)
- void AddSOPClassExtendedNegociationSub (SOPClassExtendedNegociationSub const &s)
- const MaximumLengthSub & GetMaximumLengthSub () const
- MaximumLengthSub & GetMaximumLengthSub ()
- 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

10.355.1 Detailed Description

UserInformation.

Table 9-16 USER INFORMATION ITEM FIELDS

TODO what is the goal of :

Table 9-20 USER INFORMATION ITEM FIELDS

10.355.2 Constructor & Destructor Documentation

10.355.2.1 UserInformation()

`gdcm::network::UserInformation::UserInformation ( )`

Referenced by GetMaximumLengthSub().

10.355.2.2 ~UserInformation()

`gdcm::network::UserInformation::~UserInformation ( )`

10.355.3 Member Function Documentation

10.355.3.1 AddRoleSelectionSub()

```
void gdcm::network::UserInformation::AddRoleSelectionSub (
            RoleSelectionSub const & r )
```

Referenced by GetMaximumLengthSub().

10.355.3.2 AddSOPClassExtendedNegociationSub()

```
void gdcm::network::UserInformation::AddSOPClassExtendedNegociationSub (
            SOPClassExtendedNegociationSub const & s )
```

Referenced by GetMaximumLengthSub().

10.355.3.3 GetMaximumLengthSub() [1/2]

```
const MaximumLengthSub& gdcm::network::UserInformation::GetMaximumLengthSub ( ) const  [inline]
```

10.355.3.4 GetMaximumLengthSub() [2/2]

```
MaximumLengthSub& gdcm::network::UserInformation::GetMaximumLengthSub ( )  [inline]
```

References AddRoleSelectionSub(), AddSOPClassExtendedNegociationSub(), operator=(), and UserInformation().

10.355.3.5 operator=()

```
UserInformation& gdcm::network::UserInformation::operator= (
            const UserInformation &  )
```

Referenced by GetMaximumLengthSub().

10.355.3.6 Print()

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

10.355.3.7 Read()

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

10.355.3.8 Size()

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

10.355.3.9 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

10.356 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.

10.356.1 Detailed Description

Class for generating unique UUID.

generate DCE 1.1 uid

10.356.2 Member Function Documentation

10.356.2.1 Generate()

```
const char* gdcm::UUIDGenerator::Generate ( )
```

Return the generated uuid NOT THREAD SAFE

10.356.2.2 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:

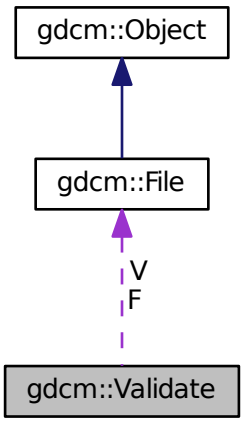
- gdcmUUIDGenerator.h

10.357 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

10.357.1 Detailed Description

Validate class.

10.357.2 Constructor & Destructor Documentation

10.357.2.1 Validate()

```
gdcm::Validate::Validate ( )
```

10.357.2.2 ~Validate()

```
gdcm::Validate::~Validate ( )
```

10.357.3 Member Function Documentation

10.357.3.1 GetValidatedFile()

```
const File& gdcm::Validate::GetValidatedFile ( )  [inline]
```

10.357.3.2 SetFile()

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

10.357.3.3 Validation()

```
void gdcm::Validate::Validation ( )
```

10.357.4 Member Data Documentation

10.357.4.1 F

```
const File* gdcm::Validate::F  [protected]
```

10.357.4.2 V

File gdcm::Validate::V [protected]

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

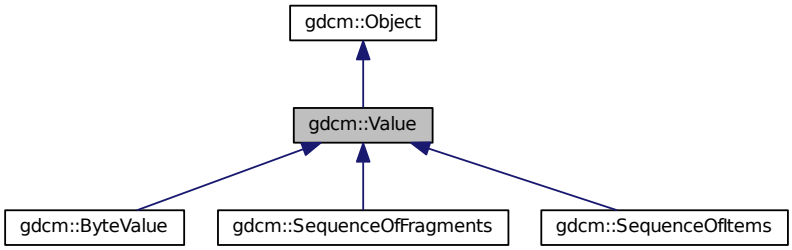
- gdcmValidate.h

10.358 gdcm::Value Class Reference

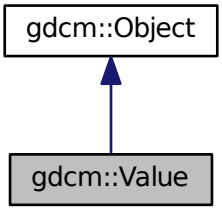
Class to represent the value of a Data Element.

#include <gdcmValue.h>

Inheritance diagram for gdcm::Value:



Collaboration diagram for gdcm::Value:



Public Member Functions

- Value ()
- ~Value ()
- 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

10.358.1 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.

10.358.2 Constructor & Destructor Documentation

10.358.2.1 Value()

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

10.358.2.2 ~Value()

```
gdcm::Value::~Value ( )  [inline]
```

References gdcm::operator==().

10.358.3 Member Function Documentation

10.358.3.1 Clear()

```
virtual void gdcm::Value::Clear ( )  [pure virtual]
```

Implemented in gdcm::ByteValue, gdcm::SequenceOfItems, and gdcm::SequenceOfFragments.

10.358.3.2 GetLength()

```
virtual VL gdcm::Value::GetLength ( ) const  [pure virtual]
```

Implemented in gdcm::ByteValue, gdcm::SequenceOfItems, and gdcm::SequenceOfFragments.

Referenced by gdcm::DataSet::InsertDataElement(), and gdcm::DataElement::SetValue().

10.358.3.3 operator==()

```
virtual bool gdcm::Value::operator== (
            const Value & val ) const  [pure virtual]
```

Implemented in gdcm::SequenceOfFragments, gdcm::SequenceOfItems, and gdcm::ByteValue.

10.358.3.4 SetLength()

```
virtual void gdcm::Value::SetLength (
            VL l )  [pure virtual]
```

Implemented in gdcm::ByteValue, gdcm::SequenceOfItems, and gdcm::SequenceOfFragments.

10.358.3.5 SetLengthOnly()

```
virtual void gdcm::Value::SetLengthOnly (
            VL l )  [protected], [virtual]
```

Reimplemented in gdcm::ByteValue.

10.358.4 Friends And Related Function Documentation

10.358.4.1 DataElement

```
friend class DataElement  [friend]
```

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

- gdcmValue.h

10.359 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)

10.359.1 Detailed Description

template<typename TDE, typename TSwap, typename TType = uint8_t>
class gdcm::ValueIO< TDE, TSwap, TType >

Class to dispatch template calls.

10.359.2 Member Function Documentation

10.359.2.1 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]
```

10.359.2.2 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

10.360 gdcm::Version Class Reference

major/minor and build version

```
#include <gdcmVersion.h>
```

Public Member Functions

- Version ()
- ~Version ()
- 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)

10.360.1 Detailed Description

major/minor and build version

10.360.2 Constructor & Destructor Documentation

10.360.2.1 Version()

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

10.360.2.2 ~Version()

```
gdcm::Version::~Version ( )  [inline]
```

10.360.3 Member Function Documentation

10.360.3.1 GetBuildVersion()

```
static int gdcm::Version::GetBuildVersion ( )  [static]
```

10.360.3.2 GetMajorVersion()

```
static int gdcm::Version::GetMajorVersion ( )  [static]
```

10.360.3.3 GetMinorVersion()

```
static int gdcm::Version::GetMinorVersion ( )  [static]
```

10.360.3.4 GetVersion()

```
static const char* gdcm::Version::GetVersion ( )  [static]
```

10.360.3.5 Print()

```
void gdcm::Version::Print (
            std::ostream & os = std::cout ) const
```

Referenced by gdcm::operator<<().

10.360.4 Friends And Related Function Documentation

10.360.4.1 operator<<

```
std::ostream& operator<< (
            std::ostream & _os,
            const Version & v )  [friend]
```

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

- gdcmVersion.h

10.361 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)

10.361.1 Detailed Description

Value Length.

Warning

this is a 4bytes value ! Do not try to use it for 2bytes value length

Examples:

rle2img.cxx.

10.361.2 Member Typedef Documentation

10.361.2.1 Type

```
typedef uint32_t gdcm::VL::Type
```

10.361.3 Constructor & Destructor Documentation

10.361.3.1 VL()

```
gdcm::VL::VL (
            uint32_t vl = 0 )  [inline]
```

10.361.4 Member Function Documentation

10.361.4.1 GetLength()

```
VL gdcm::VL::GetLength ( ) const  [inline]
```

References gdcm::operator<<().

Referenced by gdcm::FileMetaInformation::GetFullLength(), and gdcm::Item::Write().

10.361.4.2 GetVL16Max()

```
static uint16_t gdcm::VL::GetVL16Max ( )  [inline], [static]
```

10.361.4.3 GetVL32Max()

```
static uint32_t gdcm::VL::GetVL32Max ( )  [inline], [static]
```

10.361.4.4 IsOdd()

```
bool gdcm::VL::IsOdd ( ) const  [inline]
```

Return whether or not the VL is odd or not.

10.361.4.5 IsUndefined()

```
bool gdcm::VL::IsUndefined ( ) const  [inline]
```

10.361.4.6 operator uint32_t()

```
gdcm::VL::operator uint32_t ( ) const  [inline]
```

10.361.4.7 operator++() [1/2]

```
VL& gdcm::VL::operator++ ( )  [inline]
```

10.361.4.8 operator++() [2/2]

```
VL gdcm::VL::operator++ (
             int  )  [inline]
```

10.361.4.9 operator+=()

```
VL& gdcm::VL::operator+= (
             VL const & vl )  [inline]
```

+= operator

10.361.4.10 Read()

```
template<typename TSwap >
std::istream& gdcm::VL::Read (
             std::istream & is )  [inline]
```

10.361.4.11 Read16()

```
template<typename TSwap >
std::istream& gdcm::VL::Read16 (
             std::istream & is )  [inline]
```

10.361.4.12 SetToUndefined()

```
void gdcm::VL::SetToUndefined ( )  [inline]
```

10.361.4.13 Write()

```
template<typename TSwap >
const std::ostream& gdcm::VL::Write (
             std::ostream & os ) const  [inline]
```

Referenced by gdcm::Fragment::Write(), gdcm::SequenceOfItems::Write(), gdcm::Item::Write(), and gdcm::Sequence↩
OfFragments::Write().

10.361.4.14 Write16()

```
template<typename TSwap >
const std::ostream& gdcm::VL::Write16 (
            std::ostream & os ) const  [inline]
```

10.361.5 Friends And Related Function Documentation

10.361.5.1 operator<<

```
std::ostream& operator<< (
            std::ostream & os,
            const VL & vl )  [friend]
```

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

- gdcmVL.h

10.362 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,
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 unsigned int GetNumberOfElementsFromArray (const char *array, unsigned int length)
- static const char * GetVMString (VMType vm)
- static VMType GetVMType (const char *vm)
- static VMType GetVMTypeFromLength (unsigned int 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)

10.362.1 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

10.362.2 Member Enumeration Documentation

10.362.2.1 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	

Enumerator

VM2_n	
VM3_4	
VM3_3n	
VM3_n	
VM4_4n	
VM6_6n	
VM7_7n	
VM30_30n	
VM47_47n	
VM_END	

10.362.3 Constructor & Destructor Documentation

10.362.3.1 VM()

```
gdcm::VM::VM (
            VMType type = VM0 )  [inline]
```

10.362.4 Member Function Documentation

10.362.4.1 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.

10.362.4.2 GetIndex()

```
static unsigned int gdcm::VM::GetIndex (
            VMType vm )  [static], [protected]
```

10.362.4.3 GetLength()

```
unsigned int gdcm::VM::GetLength ( ) const
```

10.362.4.4 GetNumberOfElementsFromArray()

```
static unsigned int gdcm::VM::GetNumberOfElementsFromArray (
            const char * array,
            unsigned int length )  [static]
```

10.362.4.5 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

Referenced by gdcm::operator<<().

10.362.4.6 GetVMType()

```
static VMType gdcm::VM::GetVMType (
            const char * vm )  [static]
```

10.362.4.7 GetVMTypeFromLength()

```
static VMType gdcm::VM::GetVMTypeFromLength (
            unsigned int length,
            unsigned int size )  [static]
```

10.362.4.8 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

10.362.4.9 operator VMType()

```
gdcm::VM::operator VMType ( ) const  [inline]
```

References gdcm::operator<<().

10.362.5 Friends And Related Function Documentation

10.362.5.1 operator<<

```
std::ostream& operator<< (
            std::ostream & os,
            const VM & vm )  [friend]
```

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

- gdcmVM.h

10.363 gdcm::VMToLength< T > Struct Template Reference

`#include <gdcmVM.h>`

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

- gdcmVM.h

10.364 gdcm::VR Class Reference

VR class.

`#include <gdcmVR.h>`

Public Types

- enum VRType {
  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,
  OW = 16384,
  PN = 32768,
  SH = 65536,
  SL = 131072,
  SQ = 262144,
  SS = 524288,
  ST = 1048576,
  TM = 2097152,
  UI = 4194304,
  UL = 8388608,
  UN = 16777216,
  US = 33554432,
  UT = 67108864,
  OB_OW = OB | OW,
  US_SS = US | SS,
  US_SS_OW = US | SS | 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 | SQ | UN | UT,
VRASCII = AE | AS | CS | DA | DS | DT | IS | LO | LT | PN | SH | ST | TM | UI | UT,
VRBINARY = AT | FL | FD | OB | OD | OF | OW | SL | SQ | SS | UL | UN | US,
VR_VM1 = AS | LT | ST | UT | SQ | OF | OD | OW | OB | UN,
VRALL = VRASCII | VRBINARY,
VR_END = UT+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)

10.364.1 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, and GenFakeIdentifyFile.cxx.

10.364.2 Member Enumeration Documentation

10.364.2.1 VRType

`enum gdcm::VR::VRType`

Enumerator

INVALID	
AE	
AS	
AT	
CS	
DA	
DS	
DT	
FD	
FL	
IS	
LO	
LT	
OB	
OD	
OF	
OW	
PN	
SH	
SL	
SQ	
SS	
ST	
TM	

Enumerator

UI	
UL	
UN	
US	
UT	
OB_OW	
US_SS	
US_SS_OW	
VL16	
VL32	
VRASCII	
VRBINARY	
VR_VM1	
VRALL	
VR_END	

Examples:

NewSequence.cs.

10.364.3 Constructor & Destructor Documentation

10.364.3.1 VR()

```
gdcm::VR::VR (
            VRType vr = INVALID )  [inline]
```

10.364.4 Member Function Documentation

10.364.4.1 CanDisplay()

```
static bool gdcm::VR::CanDisplay (
            VRType vr )  [static]
```

10.364.4.2 Compatible()

```
bool gdcm::VR::Compatible (
            VR const & vr ) const
```

10.364.4.3 GetLength() [1/2]

```
int gdcm::VR::GetLength ( ) const  [inline]
```

10.364.4.4 GetLength() [2/2]

```
static uint32_t gdcm::VR::GetLength (
            VRType vr )  [inline], [static]
```

10.364.4.5 GetSize()

```
unsigned int gdcm::VR::GetSize ( ) const  [inline]
```

References US_SS, and VRTypeTemplateCase.

10.364.4.6 GetSizeof()

```
unsigned int gdcm::VR::GetSizeof ( ) const
```

10.364.4.7 GetVRString()

```
static const char* gdcm::VR::GetVRString (
            VRType vr )  [static]
```

Referenced by gdcm::operator<<().

10.364.4.8 GetVRStringFromFile()

```
static const char* gdcm::VR::GetVRStringFromFile (
            VRType vr )  [static]
```

10.364.4.9 GetVRType()

```
static VRType gdcm::VR::GetVRType (
            const char * vr )  [static]
```

10.364.4.10 GetVRTypeFromFile()

```
static VRType gdcm::VR::GetVRTypeFromFile (
            const char * vr )  [static]
```

10.364.4.11 IsASCII()

```
static bool gdcm::VR::IsASCII (
            VRType vr )  [static]
```

10.364.4.12 IsASCII2()

```
static bool gdcm::VR::IsASCII2 (
            VRType vr )  [static]
```

10.364.4.13 IsBinary()

```
static bool gdcm::VR::IsBinary (
            VRType vr )  [static]
```

10.364.4.14 IsBinary2()

```
static bool gdcm::VR::IsBinary2 (
            VRType vr )  [static]
```

10.364.4.15 IsDual()

```
bool gdcm::VR::IsDual ( ) const
```

10.364.4.16 IsSwap()

```
static bool gdcm::VR::IsSwap (
            const char * vr )  [static]
```

10.364.4.17 IsValid() [1/2]

```
static bool gdcm::VR::IsValid (
            const char * vr )  [static]
```

10.364.4.18 IsValid() [2/2]

```
static bool gdcm::VR::IsValid (
            const char * vr1,
            VRType vr2 )  [static]
```

10.364.4.19 IsVRFile()

```
bool gdcm::VR::IsVRFile ( ) const
```

Referenced by gdcm::DataElement::SetVR().

10.364.4.20 operator VRType()

```
gdcm::VR::operator VRType ( ) const  [inline]
```

10.364.4.21 Read()

```
std::istream& gdcm::VR::Read (
            std::istream & is )  [inline]
```

References gdcmDebugMacro, INVALID, and VR_END.

10.364.4.22 Write()

```
const std::ostream& gdcm::VR::Write (
            std::ostream & os ) const  [inline]
```

References gdcmAssertAlwaysMacro, INVALID, and gdcm::operator<<().

10.364.5 Friends And Related Function Documentation

10.364.5.1 operator<<

```
std::ostream& operator<< (
            std::ostream & os,
            const VR & vr )  [friend]
```

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

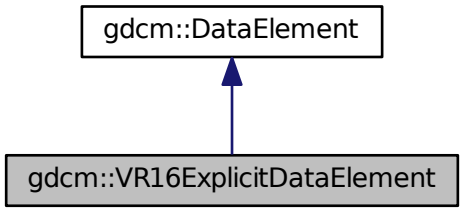
- gdcmVR.h

10.365 gdcm::VR16ExplicitDataElement Class Reference

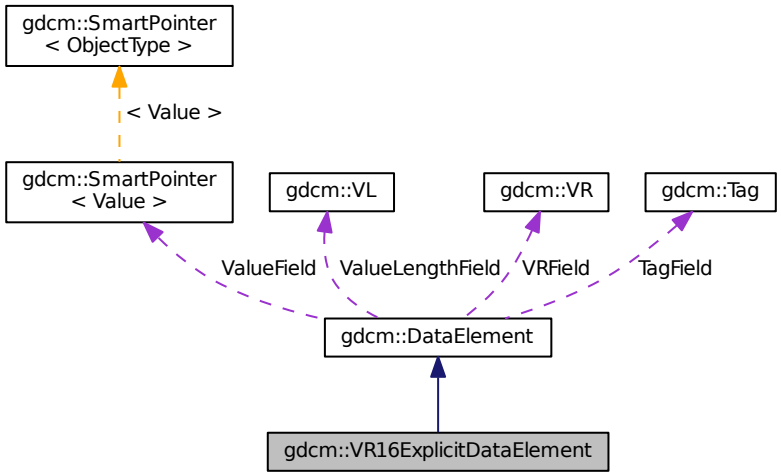
Class to read/write a DataElement as Explicit Data Element.

```
#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

10.365.1 Detailed Description

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

Note

This class support 16 bits when finding an unkown VR: For instance: Siemens_CT_Sensation64_has_VR_RT.dcm

10.365.2 Member Function Documentation

10.365.2.1 GetLength()

```
VL gdcm::VR16ExplicitDataElement::GetLength ( ) const
```

10.365.2.2 Read()

```
template<typename TSwap >
std::istream& gdcm::VR16ExplicitDataElement::Read (
             std::istream & is )
```

10.365.2.3 ReadPreValue()

```
template<typename TSwap >
std::istream& gdcm::VR16ExplicitDataElement::ReadPreValue (
             std::istream & is )
```

10.365.2.4 ReadValue()

```
template<typename TSwap >
std::istream& gdcm::VR16ExplicitDataElement::ReadValue (
             std::istream & is,
             bool readvalues = true )
```

10.365.2.5 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

10.366 gdcm::VRToEncoding< T > Struct Template Reference

`#include <gdcmVR.h>`

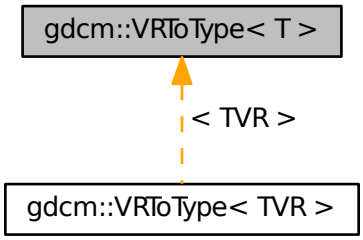
The documentation for this struct was generated from the following file:

- gdcmVR.h

10.367 gdcm::VRToType< T > Struct Template Reference

`#include <gdcmVR.h>`

Inheritance diagram for gdcm::VRToType< T >:



10.367.1 Detailed Description

template<int T>
struct gdcm::VRToType< T >

Examples:

DumpGEMSMovieGroup.cxx.

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

- gdcmVR.h

10.368 gdcm::VRVLSize< T > Class Template Reference

`#include <gdcmAttribute.h>`

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

- gdcmAttribute.h

10.369 gdcm::VRVLSize< 0 > Class Template Reference

`#include <gdcmAttribute.h>`

Static Public Member Functions

- static uint16_t Read (std::istream &_is)
- static void Write (std::ostream &os)

10.369.1 Member Function Documentation

10.369.1.1 Read()

```
static uint16_t gdcm::VRVLSize< 0 >::Read (
            std::istream & _is )  [inline], [static]
```

10.369.1.2 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

10.370 gdcm::VRVLSize< 1 > Class Template Reference

`#include <gdcmAttribute.h>`

Static Public Member Functions

- static uint32_t Read (std::istream &_is)
- static void Write (std::ostream &os)

10.370.1 Member Function Documentation

10.370.1.1 Read()

```
static uint32_t gdcm::VRVLSize< 1 >::Read (
            std::istream & _is )  [inline], [static]
```

10.370.1.2 Write()

```
static void gdcm::VRVLSize< 1 >::Write (
            std::ostream & os )  [inline], [static]
```

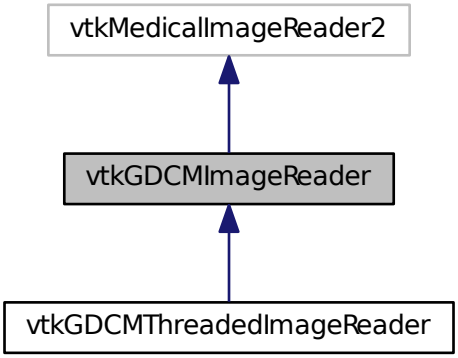
The documentation for this class was generated from the following file:

- gdcmAttribute.h

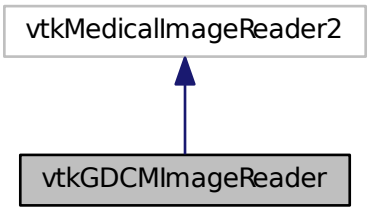
10.371 vtkGDCMImageReader Class Reference

```
#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 (LoadOverlays, int)
- vtkBooleanMacro (LoadIconImage, int)
- vtkBooleanMacro (LossyFlag, int)
- vtkBooleanMacro (ApplyLookupTable, int)
- int vtkBooleanMacro (ApplyYBRToRGB, int)
- vtkGetMacro (LoadOverlays, int)
- vtkGetMacro (LoadIconImage, int)
- vtkGetMacro (LossyFlag, int)
- vtkGetMacro (NumberOfOverlays, int)
- vtkGetMacro (NumberOfIconImages, int)
- vtkGetMacro (ApplyLookupTable, int)
- vtkGetMacro (ApplyYBRToRGB, int) vtkSetMacro(ApplyYBRToRGB
- vtkGetMacro (ImageFormat, int)
- vtkGetMacro (PlanarConfiguration, int)
- vtkGetMacro (Shift, double)
- vtkGetMacro (Scale, double)
- vtkGetObjectMacro (DirectionCosines, vtkMatrix4x4)
- vtkGetObjectMacro (MedicalImageProperties, vtkMedicalImageProperties)
- vtkGetObjectMacro (FileNames, vtkStringArray)
- vtkGetObjectMacro (Curve, vtkPolyData)
- vtkGetVector3Macro (ImagePositionPatient, double)
- vtkGetVector6Macro (ImageOrientationPatient, double)
- vtkSetMacro (LoadOverlays, int)

- vtkSetMacro (LoadIconImage, int)
- vtkSetMacro (LossyFlag, int)
- vtkSetMacro (ApplyLookupTable, int)
- vtkTypeRevisionMacro (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 (FilePrefix)
- vtkGetStringMacro (FilePattern)
- 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

10.371.1 Detailed Description

Examples:

AWTMedical3.java, Convert16BitsTo8Bits.cxx, ConvertMultiFrameToSingleFrame.cxx, ConvertRGBTo↩ Luminance.cxx, ConvertSingleBitTo8Bits.cxx, gdcmorthoplanes.cxx, gdcmreslice.cxx, gdcmtexture.cxx, gdcmvolume.cxx, HelloActiviz.cs, HelloActiviz2.cs, HelloActiviz3.cs, HelloActiviz4.cs, HelloActiviz5.cs, Hello↩ VTKWorld.cs, HelloVTKWorld.java, MagnifyFile.cxx, MetaImageMD5Activiz.cs, MIPViewer.java, MPRViewer.java, MPRViewer2.java, offscreenimage.cxx, ReadSeriesIntoVTK.java, RefCounting.cs, and reslicesphere.cxx.

10.371.2 Constructor & Destructor Documentation

10.371.2.1 vtkGDCMImageReader()

```
vtkGDCMImageReader::vtkGDCMImageReader ( )  [protected]
```

Examples:

HelloActiviz2.cs.

10.371.2.2 ~vtkGDCMImageReader()

```
vtkGDCMImageReader::~vtkGDCMImageReader ( )  [protected]
```

10.371.3 Member Function Documentation

10.371.3.1 CanReadFile()

```
virtual int vtkGDCMImageReader::CanReadFile (
            const char * fname )  [virtual]
```

Examples:

MetaImageMD5Activiz.cs.

10.371.3.2 ExecuteData()

```
void vtkGDCMImageReader::ExecuteData (
            vtkDataObject * out )  [protected]
```

10.371.3.3 ExecuteInformation()

```
void vtkGDCMImageReader::ExecuteInformation ( )  [protected]
```

10.371.3.4 FillMedicalImageInformation()

```
void vtkGDCMImageReader::FillMedicalImageInformation (
            const gdcm::ImageReader & reader )  [protected]
```

10.371.3.5 GetDescriptiveName()

```
virtual const char* vtkGDCMImageReader::GetDescriptiveName ( )  [inline], [virtual]
```

10.371.3.6 GetFileExtensions()

```
virtual const char* vtkGDCMImageReader::GetFileExtensions ( )  [inline], [virtual]
```

10.371.3.7 GetIconImage()

```
vtkImageData* vtkGDCMImageReader::GetIconImage ( )
```

10.371.3.8 GetOverlay()

```
vtkImageData* vtkGDCMImageReader::GetOverlay (
            int i )
```

10.371.3.9 LoadSingleFile()

```
int vtkGDCMImageReader::LoadSingleFile (
            const char * filename,
            char * pointer,
            unsigned long & outlen )  [protected]
```

10.371.3.10 New()

```
static vtkGDCMImageReader* vtkGDCMImageReader::New ( )  [static]
```

Examples:

Convert16BitsTo8Bits.cxx, ConvertMultiFrameToSingleFrame.cxx, ConvertRGBToLuminance.cxx, ConvertSingleBitTo8Bits.cxx, gdcmorthoplanes.cxx, gdcmreslice.cxx, gdcmtexture.cxx, gdcmvolume.cxx, HelloActiviz.cs, HelloActiviz3.cs, HelloActiviz4.cs, HelloActiviz5.cs, HelloVTKWorld.cs, MagnifyFile.cxx, MetaImageMD5Activiz.cs, offscreenimage.cxx, RefCounting.cs, and reslicesphere.cxx.

10.371.3.11 PrintSelf()

```
virtual void vtkGDCMImageReader::PrintSelf (
            ostream & os,
            vtkIndent indent )  [virtual]
```

Reimplemented in vtkGDCMThreadedImageReader.

10.371.3.12 RequestDataCompat()

```
int vtkGDCMImageReader::RequestDataCompat ( )  [protected]
```

10.371.3.13 RequestInformationCompat()

```
int vtkGDCMImageReader::RequestInformationCompat ( )  [protected]
```

10.371.3.14 SetCurve()

```
virtual void vtkGDCMImageReader::SetCurve (
            vtkPolyData * pd )  [virtual]
```

10.371.3.15 SetFileNames()

```
virtual void vtkGDCMImageReader::SetFileNames (
            vtkStringArray *  )  [virtual]
```

Examples:

gdcmorthoplanes.cxx, HelloActiviz3.cs, HelloActiviz4.cs, HelloActiviz5.cs, MIPViewer.java, MPRViewer.java, MPRViewer2.java, and ReadSeriesIntoVTK.java.

10.371.3.16 SetFilePattern()

```
void vtkGDCMImageReader::SetFilePattern (
            const char *  )  [inline], [protected]
```

10.371.3.17 SetFilePrefix()

```
void vtkGDCMImageReader::SetFilePrefix (
            const char *  )  [inline], [protected]
```

10.371.3.18 SetMedicalImageProperties()

```
virtual void vtkGDCMImageReader::SetMedicalImageProperties (
            vtkMedicalImageProperties * pd )  [virtual]
```

10.371.3.19 vtkBooleanMacro() [1/5]

```
vtkGDCMImageReader::vtkBooleanMacro (
            LoadOverlays ,
            int  )
```

10.371.3.20 vtkBooleanMacro() [2/5]

```
vtkGDCMImageReader::vtkBooleanMacro (
            LoadIconImage ,
            int  )
```

10.371.3.21 vtkBooleanMacro() [3/5]

```
vtkGDCMImageReader::vtkBooleanMacro (
            LossyFlag ,
            int  )
```

10.371.3.22 vtkBooleanMacro() [4/5]

```
vtkGDCMImageReader::vtkBooleanMacro (
            ApplyLookupTable ,
            int  )
```

10.371.3.23 vtkBooleanMacro() [5/5]

```
int vtkGDCMImageReader::vtkBooleanMacro (
            ApplyYBRToRGB ,
            int  )
```

10.371.3.24 vtkGetMacro() [1/11]

```
vtkGDCMImageReader::vtkGetMacro (
            LoadOverlays ,
            int  )
```

10.371.3.25 vtkGetMacro() [2/11]

```
vtkGDCMImageReader::vtkGetMacro (
            LoadIconImage ,
            int  )
```

10.371.3.26 vtkGetMacro() [3/11]

```
vtkGDCMImageReader::vtkGetMacro (
            LossyFlag ,
            int  )
```

10.371.3.27 vtkGetMacro() [4/11]

```
vtkGDCMImageReader::vtkGetMacro (
            NumberOfOverlays ,
            int  )
```

10.371.3.28 vtkGetMacro() [5/11]

```
vtkGDCMImageReader::vtkGetMacro (
            NumberOfIconImages ,
            int  )
```

10.371.3.29 vtkGetMacro() [6/11]

```
vtkGDCMImageReader::vtkGetMacro (
            ApplyLookupTable ,
            int  )
```

10.371.3.30 vtkGetMacro() [7/11]

```
vtkGDCMImageReader::vtkGetMacro (
            ApplyYBRToRGB ,
            int  )
```

10.371.3.31 vtkGetMacro() [8/11]

```
vtkGDCMImageReader::vtkGetMacro (
            ImageFormat ,
            int  )
```

10.371.3.32 vtkGetMacro() [9/11]

```
vtkGDCMImageReader::vtkGetMacro (
            PlanarConfiguration ,
            int  )
```

10.371.3.33 vtkGetMacro() [10/11]

```
vtkGDCMImageReader::vtkGetMacro (
            Shift ,
            double  )
```

10.371.3.34 vtkGetMacro() [11/11]

```
vtkGDCMImageReader::vtkGetMacro (
            Scale ,
            double  )
```

10.371.3.35 vtkGetObjectMacro() [1/4]

```
vtkGDCMImageReader::vtkGetObjectMacro (
            DirectionCosines ,
            vtkMatrix4x4  )
```

10.371.3.36 vtkGetObjectMacro() [2/4]

```
vtkGDCMImageReader::vtkGetObjectMacro (
            MedicalImageProperties ,
            vtkMedicalImageProperties  )
```

10.371.3.37 vtkGetObjectMacro() [3/4]

```
vtkGDCMImageReader::vtkGetObjectMacro (
            FileNames ,
            vtkStringArray  )
```

10.371.3.38 vtkGetObjectMacro() [4/4]

```
vtkGDCMImageReader::vtkGetObjectMacro (
            Curve ,
            vtkPolyData  )
```

10.371.3.39 vtkGetStringMacro() [1/2]

```
vtkGDCMImageReader::vtkGetStringMacro (
            FilePrefix  )  [protected]
```

10.371.3.40 vtkGetStringMacro() [2/2]

```
vtkGDCMImageReader::vtkGetStringMacro (
            FilePattern  )  [protected]
```

10.371.3.41 vtkGetVector3Macro()

```
vtkGDCMImageReader::vtkGetVector3Macro (
            ImagePositionPatient ,
            double  )
```

10.371.3.42 vtkGetVector6Macro()

```
vtkGDCMImageReader::vtkGetVector6Macro (
            ImageOrientationPatient ,
            double  )
```

10.371.3.43 vtkSetMacro() [1/4]

```
vtkGDCMImageReader::vtkSetMacro (
            LoadOverlays ,
            int  )
```

10.371.3.44 vtkSetMacro() [2/4]

```
vtkGDCMImageReader::vtkSetMacro (
            LoadIconImage ,
            int  )
```

10.371.3.45 vtkSetMacro() [3/4]

```
vtkGDCMImageReader::vtkSetMacro (
            LossyFlag ,
            int  )
```

10.371.3.46 vtkSetMacro() [4/4]

```
vtkGDCMImageReader::vtkSetMacro (
            ApplyLookupTable ,
            int  )
```

10.371.3.47 vtkSetVector6Macro()

```
vtkGDCMImageReader::vtkSetVector6Macro (
            ImageOrientationPatient ,
            double  )  [protected]
```

10.371.3.48 vtkTypeRevisionMacro()

```
vtkGDCMImageReader::vtkTypeRevisionMacro (
            vtkGDCMImageReader ,
            vtkMedicalImageReader2  )
```

10.371.4 Member Data Documentation

10.371.4.1 ApplyInverseVideo

```
int vtkGDCMImageReader::ApplyInverseVideo  [protected]
```

10.371.4.2 ApplyLookupTable

```
int vtkGDCMImageReader::ApplyLookupTable  [protected]
```

10.371.4.3 ApplyPlanarConfiguration

```
int vtkGDCMImageReader::ApplyPlanarConfiguration  [protected]
```

10.371.4.4 ApplyShiftScale

```
int vtkGDCMImageReader::ApplyShiftScale  [protected]
```

10.371.4.5 ApplyYBRToRGB

```
int vtkGDCMImageReader::ApplyYBRToRGB  [protected]
```

10.371.4.6 Curve

```
vtkPolyData* vtkGDCMImageReader::Curve  [protected]
```

10.371.4.7 DirectionCosines

```
vtkMatrix4x4* vtkGDCMImageReader::DirectionCosines  [protected]
```

10.371.4.8 FileNames

```
vtkStringArray* vtkGDCMImageReader::FileNames  [protected]
```

10.371.4.9 ForceRescale

```
int vtkGDCMImageReader::ForceRescale  [protected]
```

10.371.4.10 IconDataScalarType

```
int vtkGDCMImageReader::IconDataScalarType  [protected]
```

10.371.4.11 IconImageDataExtent

```
int vtkGDCMImageReader::IconImageDataExtent[6]  [protected]
```

10.371.4.12 IconNumberOfScalarComponents

```
int vtkGDCMImageReader::IconNumberOfScalarComponents  [protected]
```

10.371.4.13 ImageFormat

```
int vtkGDCMImageReader::ImageFormat  [protected]
```

10.371.4.14 ImageOrientationPatient

```
double vtkGDCMImageReader::ImageOrientationPatient[6]  [protected]
```

10.371.4.15 ImagePositionPatient

```
double vtkGDCMImageReader::ImagePositionPatient[3]  [protected]
```

10.371.4.16 LoadIconImage

```
int vtkGDCMImageReader::LoadIconImage  [protected]
```

10.371.4.17 LoadOverlays

```
int vtkGDCMImageReader::LoadOverlays  [protected]
```

10.371.4.18 LossyFlag

```
int vtkGDCMImageReader::LossyFlag  [protected]
```

10.371.4.19 MedicalImageProperties

```
vtkMedicalImageProperties* vtkGDCMImageReader::MedicalImageProperties  [protected]
```

10.371.4.20 NumberOfIconImages

```
int vtkGDCMImageReader::NumberOfIconImages  [protected]
```

10.371.4.21 NumberOfOverlays

```
int vtkGDCMImageReader::NumberOfOverlays  [protected]
```

10.371.4.22 PlanarConfiguration

```
int vtkGDCMImageReader::PlanarConfiguration  [protected]
```

10.371.4.23 Scale

```
double vtkGDCMImageReader::Scale  [protected]
```

10.371.4.24 Shift

```
double vtkGDCMImageReader::Shift  [protected]
```

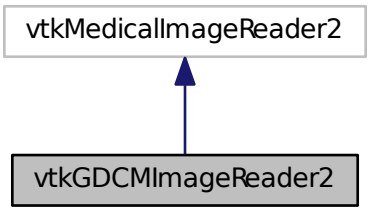
The documentation for this class was generated from the following file:

- vtkGDCMImageReader.h

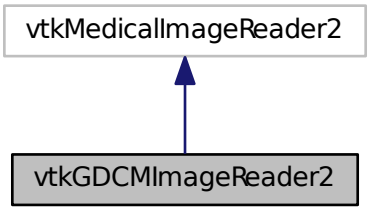
10.372 vtkGDCMImageReader2 Class Reference

`#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 (LoadOverlays, int)
- vtkBooleanMacro (LoadIconImage, int)
- vtkBooleanMacro (LossyFlag, int)
- vtkBooleanMacro (ApplyLookupTable, int)
- int vtkBooleanMacro (ApplyYBRToRGB, int)
- vtkGetMacro (LoadOverlays, int)
- vtkGetMacro (LoadIconImage, int)
- vtkGetMacro (LossyFlag, int)
- vtkGetMacro (NumberOfOverlays, int)
- vtkGetMacro (NumberOfIconImages, int)
- vtkGetMacro (ApplyLookupTable, int)
- vtkGetMacro (ApplyYBRToRGB, int) vtkSetMacro(ApplyYBRToRGB
- vtkGetMacro (ImageFormat, int)
- vtkGetMacro (PlanarConfiguration, int)
- vtkGetMacro (Shift, double)
- vtkGetMacro (Scale, double)
- vtkGetObjectMacro (DirectionCosines, vtkMatrix4x4)
- vtkGetObjectMacro (Curve, vtkPolyData)
- vtkGetVector3Macro (ImagePositionPatient, double)
- vtkGetVector6Macro (ImageOrientationPatient, double)
- vtkSetMacro (LoadOverlays, int)
- vtkSetMacro (LoadIconImage, int)
- vtkSetMacro (LossyFlag, int)
- vtkSetMacro (ApplyLookupTable, int)
- vtkTypeRevisionMacro (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 (FilePrefix)
- vtkGetStringMacro (FilePattern)
- 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

10.372.1 Detailed Description

Examples:

Compute3DSpacing.cxx.

10.372.2 Constructor & Destructor Documentation

10.372.2.1 vtkGDCMImageReader2()

```
vtkGDCMImageReader2::vtkGDCMImageReader2 ( )  [protected]
```

10.372.2.2 ~vtkGDCMImageReader2()

```
vtkGDCMImageReader2::~vtkGDCMImageReader2 ( )  [protected]
```

10.372.3 Member Function Documentation

10.372.3.1 CanReadFile()

```
virtual int vtkGDCMImageReader2::CanReadFile (
            const char * fname )  [virtual]
```

10.372.3.2 FillMedicalImageInformation()

```
void vtkGDCMImageReader2::FillMedicalImageInformation (
            const gdcm::ImageReader & reader )  [protected]
```

10.372.3.3 GetDescriptiveName()

```
virtual const char* vtkGDCMImageReader2::GetDescriptiveName ( )  [inline], [virtual]
```

10.372.3.4 GetFileExtensions()

```
virtual const char* vtkGDCMImageReader2::GetFileExtensions ( )  [inline], [virtual]
```

10.372.3.5 GetIconImage()

```
vtkImageData* vtkGDCMImageReader2::GetIconImage ( )
```

10.372.3.6 GetIconImagePort()

```
vtkAlgorithmOutput* vtkGDCMImageReader2::GetIconImagePort ( )
```

10.372.3.7 GetOverlay()

```
vtkImageData* vtkGDCMImageReader2::GetOverlay (
            int i )
```

10.372.3.8 GetOverlayPort()

```
vtkAlgorithmOutput* vtkGDCMImageReader2::GetOverlayPort (
            int index )
```

10.372.3.9 LoadSingleFile()

```
int vtkGDCMImageReader2::LoadSingleFile (
            const char * filename,
            char * pointer,
            unsigned long & outlen )  [protected]
```

10.372.3.10 New()

```
static vtkGDCMImageReader2* vtkGDCMImageReader2::New ( )  [static]
```

Examples:

Compute3DSpacing.cxx.

10.372.3.11 PrintSelf()

```
virtual void vtkGDCMImageReader2::PrintSelf (
            ostream & os,
            vtkIndent indent )  [virtual]
```

10.372.3.12 ProcessRequest()

```
int vtkGDCMImageReader2::ProcessRequest (
            vtkInformation * request,
            vtkInformationVector ** inputVector,
            vtkInformationVector * outputVector )  [protected]
```

10.372.3.13 RequestData()

```
int vtkGDCMImageReader2::RequestData (
            vtkInformation * request,
            vtkInformationVector ** inputVector,
            vtkInformationVector * outputVector )  [protected]
```

10.372.3.14 RequestDataCompat()

```
int vtkGDCMImageReader2::RequestDataCompat ( )  [protected]
```

10.372.3.15 RequestInformation()

```
int vtkGDCMImageReader2::RequestInformation (
            vtkInformation * request,
            vtkInformationVector ** inputVector,
            vtkInformationVector * outputVector )  [protected]
```

10.372.3.16 RequestInformationCompat()

```
int vtkGDCMImageReader2::RequestInformationCompat ( )  [protected]
```

10.372.3.17 SetCurve()

```
virtual void vtkGDCMImageReader2::SetCurve (
            vtkPolyData * pd )  [virtual]
```

10.372.3.18 SetFilePattern()

```
void vtkGDCMImageReader2::SetFilePattern (
            const char *  )  [inline], [protected]
```

10.372.3.19 SetFilePrefix()

```
void vtkGDCMImageReader2::SetFilePrefix (
            const char *  )  [inline], [protected]
```

10.372.3.20 SetMedicalImageProperties()

```
virtual void vtkGDCMImageReader2::SetMedicalImageProperties (
            vtkMedicalImageProperties * pd )  [virtual]
```

10.372.3.21 vtkBooleanMacro() [1/5]

```
vtkGDCMImageReader2::vtkBooleanMacro (
            LoadOverlays ,
            int  )
```

10.372.3.22 vtkBooleanMacro() [2/5]

```
vtkGDCMImageReader2::vtkBooleanMacro (
            LoadIconImage ,
            int  )
```

10.372.3.23 vtkBooleanMacro() [3/5]

```
vtkGDCMImageReader2::vtkBooleanMacro (
            LossyFlag ,
            int  )
```

10.372.3.24 vtkBooleanMacro() [4/5]

```
vtkGDCMImageReader2::vtkBooleanMacro (
            ApplyLookupTable ,
            int  )
```

10.372.3.25 vtkBooleanMacro() [5/5]

```
int vtkGDCMImageReader2::vtkBooleanMacro (
            ApplyYBRToRGB ,
            int  )
```

10.372.3.26 vtkGetMacro() [1/11]

```
vtkGDCMImageReader2::vtkGetMacro (
            LoadOverlays ,
            int  )
```

10.372.3.27 vtkGetMacro() [2/11]

```
vtkGDCMImageReader2::vtkGetMacro (
            LoadIconImage ,
            int  )
```

10.372.3.28 vtkGetMacro() [3/11]

```
vtkGDCMImageReader2::vtkGetMacro (
            LossyFlag ,
            int  )
```

10.372.3.29 vtkGetMacro() [4/11]

```
vtkGDCMImageReader2::vtkGetMacro (
            NumberOfOverlays ,
            int  )
```

10.372.3.30 vtkGetMacro() [5/11]

```
vtkGDCMImageReader2::vtkGetMacro (
            NumberOfIconImages ,
            int  )
```

10.372.3.31 vtkGetMacro() [6/11]

```
vtkGDCMImageReader2::vtkGetMacro (
            ApplyLookupTable ,
            int  )
```

10.372.3.32 vtkGetMacro() [7/11]

```
vtkGDCMImageReader2::vtkGetMacro (
            ApplyYBRToRGB ,
            int  )
```

10.372.3.33 vtkGetMacro() [8/11]

```
vtkGDCMImageReader2::vtkGetMacro (
            ImageFormat ,
            int  )
```

10.372.3.34 vtkGetMacro() [9/11]

```
vtkGDCMImageReader2::vtkGetMacro (
            PlanarConfiguration ,
            int  )
```

10.372.3.35 vtkGetMacro() [10/11]

```
vtkGDCMImageReader2::vtkGetMacro (
            Shift ,
            double  )
```

10.372.3.36 vtkGetMacro() [11/11]

```
vtkGDCMImageReader2::vtkGetMacro (
            Scale ,
            double  )
```

10.372.3.37 vtkGetObjectMacro() [1/2]

```
vtkGDCMImageReader2::vtkGetObjectMacro (
            DirectionCosines ,
            vtkMatrix4x4  )
```

10.372.3.38 vtkGetObjectMacro() [2/2]

```
vtkGDCMImageReader2::vtkGetObjectMacro (
            Curve ,
            vtkPolyData  )
```

10.372.3.39 vtkGetStringMacro() [1/2]

```
vtkGDCMImageReader2::vtkGetStringMacro (
            FilePrefix  )  [protected]
```

10.372.3.40 vtkGetStringMacro() [2/2]

```
vtkGDCMImageReader2::vtkGetStringMacro (
            FilePattern  )  [protected]
```

10.372.3.41 vtkGetVector3Macro()

```
vtkGDCMImageReader2::vtkGetVector3Macro (
            ImagePositionPatient ,
            double  )
```

10.372.3.42 vtkGetVector6Macro()

```
vtkGDCMImageReader2::vtkGetVector6Macro (
            ImageOrientationPatient ,
            double  )
```

10.372.3.43 vtkSetMacro() [1/4]

```
vtkGDCMImageReader2::vtkSetMacro (
            LoadOverlays ,
            int  )
```

10.372.3.44 vtkSetMacro() [2/4]

```
vtkGDCMImageReader2::vtkSetMacro (
            LoadIconImage ,
            int  )
```

10.372.3.45 vtkSetMacro() [3/4]

```
vtkGDCMImageReader2::vtkSetMacro (
            LossyFlag ,
            int  )
```

10.372.3.46 vtkSetMacro() [4/4]

```
vtkGDCMImageReader2::vtkSetMacro (
            ApplyLookupTable ,
            int  )
```

10.372.3.47 vtkSetVector6Macro()

```
vtkGDCMImageReader2::vtkSetVector6Macro (
            ImageOrientationPatient ,
            double  )  [protected]
```

10.372.3.48 vtkTypeRevisionMacro()

```
vtkGDCMImageReader2::vtkTypeRevisionMacro (
            vtkGDCMImageReader2 ,
            vtkMedicalImageReader2  )
```

10.372.4 Member Data Documentation

10.372.4.1 ApplyInverseVideo

```
int vtkGDCMImageReader2::ApplyInverseVideo  [protected]
```

10.372.4.2 ApplyLookupTable

```
int vtkGDCMImageReader2::ApplyLookupTable  [protected]
```

10.372.4.3 ApplyPlanarConfiguration

```
int vtkGDCMImageReader2::ApplyPlanarConfiguration  [protected]
```

10.372.4.4 ApplyShiftScale

```
int vtkGDCMImageReader2::ApplyShiftScale  [protected]
```

10.372.4.5 ApplyYBRToRGB

```
int vtkGDCMImageReader2::ApplyYBRToRGB  [protected]
```

10.372.4.6 Curve

```
vtkPolyData* vtkGDCMImageReader2::Curve  [protected]
```

10.372.4.7 DirectionCosines

```
vtkMatrix4x4* vtkGDCMImageReader2::DirectionCosines  [protected]
```

10.372.4.8 ForceRescale

```
int vtkGDCMImageReader2::ForceRescale  [protected]
```

10.372.4.9 IconDataScalarType

```
int vtkGDCMImageReader2::IconDataScalarType  [protected]
```

10.372.4.10 IconImageDataExtent

```
int vtkGDCMImageReader2::IconImageDataExtent[6]  [protected]
```

10.372.4.11 IconNumberOfScalarComponents

```
int vtkGDCMImageReader2::IconNumberOfScalarComponents  [protected]
```

10.372.4.12 ImageFormat

```
int vtkGDCMImageReader2::ImageFormat  [protected]
```

10.372.4.13 ImageOrientationPatient

```
double vtkGDCMImageReader2::ImageOrientationPatient[6]  [protected]
```

10.372.4.14 ImagePositionPatient

```
double vtkGDCMImageReader2::ImagePositionPatient[3]  [protected]
```

10.372.4.15 LoadIconImage

```
int vtkGDCMImageReader2::LoadIconImage  [protected]
```

10.372.4.16 LoadOverlays

```
int vtkGDCMImageReader2::LoadOverlays  [protected]
```

10.372.4.17 LossyFlag

```
int vtkGDCMImageReader2::LossyFlag  [protected]
```

10.372.4.18 NumberOfIconImages

```
int vtkGDCMImageReader2::NumberOfIconImages  [protected]
```

10.372.4.19 NumberOfOverlays

```
int vtkGDCMImageReader2::NumberOfOverlays  [protected]
```

10.372.4.20 PlanarConfiguration

```
int vtkGDCMImageReader2::PlanarConfiguration  [protected]
```

10.372.4.21 Scale

```
double vtkGDCMImageReader2::Scale  [protected]
```

10.372.4.22 Shift

```
double vtkGDCMImageReader2::Shift  [protected]
```

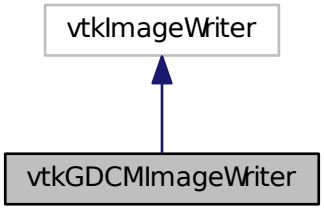
The documentation for this class was generated from the following file:

- vtkGDCMImageReader2.h

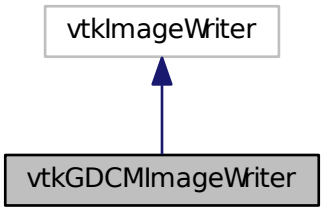
10.373 vtkGDCMImageWriter Class Reference

`#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 (LossyFlag, int)
- vtkBooleanMacro (FileLowerLeft, int)
- vtkGetMacro (LossyFlag, int)
- vtkGetMacro (Shift, double)
- vtkGetMacro (Scale, double)
- vtkGetMacro (ImageFormat, int)
- vtkGetMacro (FileLowerLeft, int)
- vtkGetMacro (PlanarConfiguration, int)
- vtkGetMacro (CompressionType, int)
- vtkGetObjectMacro (MedicalImageProperties, vtkMedicalImageProperties)
- vtkGetObjectMacro (FileNames, vtkStringArray)
- vtkGetObjectMacro (DirectionCosines, vtkMatrix4x4)
- vtkGetStringMacro (StudyUID)
- vtkGetStringMacro (SeriesUID)
- vtkSetMacro (LossyFlag, int)
- vtkSetMacro (Shift, double)
- vtkSetMacro (Scale, double)
- vtkSetMacro (ImageFormat, int)
- vtkSetMacro (FileLowerLeft, int)
- vtkSetMacro (PlanarConfiguration, int)
- vtkSetMacro (CompressionType, int)
- vtkSetStringMacro (StudyUID)
- vtkSetStringMacro (SeriesUID)
- vtkTypeRevisionMacro (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)

10.373.1 Detailed Description

Examples:

Convert16BitsTo8Bits.cxx, ConvertMultiFrameToSingleFrame.cxx, ConvertRGBToLuminance.cxx, ConvertSingleBitTo8Bits.cxx, CreateFakePET.cxx, CreateFakeRTDOSE.cxx, gdcmorthoplanes.cxx, HelloActiviz.cs, HelloActiviz2.cs, HelloVTKWorld.cs, HelloVTKWorld.java, HelloVTKWorld2.cs, MagnifyFile.cxx, and RefCounting.cs.

10.373.2 Member Enumeration Documentation

10.373.2.1 CompressionTypes

```
enum vtkGDCMImageWriter::CompressionTypes
```

Enumerator

NO_COMPRESSION	
JPEG_COMPRESSION	
JPEG2000_COMPRESSION	
JPEGLS_COMPRESSION	
RLE_COMPRESSION	

10.373.3 Constructor & Destructor Documentation

10.373.3.1 vtkGDCMImageWriter()

```
vtkGDCMImageWriter::vtkGDCMImageWriter ( )  [protected]
```

10.373.3.2 ~vtkGDCMImageWriter()

```
vtkGDCMImageWriter::~vtkGDCMImageWriter ( )  [protected]
```

10.373.4 Member Function Documentation

10.373.4.1 GetDescriptiveName()

```
virtual const char* vtkGDCMImageWriter::GetDescriptiveName ( )  [inline], [virtual]
```

10.373.4.2 GetFileExtensions()

```
virtual const char* vtkGDCMImageWriter::GetFileExtensions ( )  [inline], [virtual]
```

10.373.4.3 GetFileName()

```
virtual char* vtkGDCMImageWriter::GetFileName ( )  [protected], [virtual]
```

10.373.4.4 New()

```
static vtkGDCMImageWriter* vtkGDCMImageWriter::New ( )  [static]
```

Examples:

Convert16BitsTo8Bits.cxx, ConvertMultiFrameToSingleFrame.cxx, ConvertRGBToLuminance.cxx, ConvertSingleBitTo8Bits.cxx, CreateFakePET.cxx, CreateFakeRTDOSE.cxx, gdcmorthoplanes.cxx, HelloActiviz.cs, HelloVTKWorld.cs, HelloVTKWorld2.cs, MagnifyFile.cxx, and RefCounting.cs.

10.373.4.5 PrintSelf()

```
virtual void vtkGDCMImageWriter::PrintSelf (
            ostream & os,
            vtkIndent indent )  [virtual]
```

10.373.4.6 SetDirectionCosines()

```
virtual void vtkGDCMImageWriter::SetDirectionCosines (
            vtkMatrix4x4 * matrix )  [virtual]
```

Examples:

Convert16BitsTo8Bits.cxx, ConvertRGBToLuminance.cxx, ConvertSingleBitTo8Bits.cxx, gdcmorthoplanes.cxx, HelloActiviz2.cs, HelloVTKWorld.cs, HelloVTKWorld.java, and MagnifyFile.cxx.

10.373.4.7 SetDirectionCosinesFromImageOrientationPatient()

```
virtual void vtkGDCMImageWriter::SetDirectionCosinesFromImageOrientationPatient (
            const double dircos[6] )  [virtual]
```

10.373.4.8 SetFileNames()

```
virtual void vtkGDCMImageWriter::SetFileNames (
            vtkStringArray *  )  [virtual]
```

Examples:

ConvertMultiFrameToSingleFrame.cxx, and CreateFakePET.cxx.

10.373.4.9 SetMedicalImageProperties()

```
virtual void vtkGDCMImageWriter::SetMedicalImageProperties (
            vtkMedicalImageProperties *  )  [virtual]
```

Examples:

Convert16BitsTo8Bits.cxx, ConvertRGBToLuminance.cxx, ConvertSingleBitTo8Bits.cxx, gdcmorthoplanes.cxx, HelloActiviz.cs, HelloActiviz2.cs, HelloVTKWorld.cs, HelloVTKWorld.java, and MagnifyFile.cxx.

10.373.4.10 vtkBooleanMacro() [1/2]

```
vtkGDCMImageWriter::vtkBooleanMacro (
            LossyFlag ,
            int  )
```

10.373.4.11 vtkBooleanMacro() [2/2]

```
vtkGDCMImageWriter::vtkBooleanMacro (
            FileLowerLeft ,
            int  )
```

10.373.4.12 vtkGetMacro() [1/7]

```
vtkGDCMImageWriter::vtkGetMacro (
            LossyFlag ,
            int  )
```

10.373.4.13 vtkGetMacro() [2/7]

```
vtkGDCMImageWriter::vtkGetMacro (
            Shift ,
            double  )
```

10.373.4.14 vtkGetMacro() [3/7]

```
vtkGDCMImageWriter::vtkGetMacro (
            Scale ,
            double  )
```

10.373.4.15 vtkGetMacro() [4/7]

```
vtkGDCMImageWriter::vtkGetMacro (
            ImageFormat ,
            int  )
```

10.373.4.16 vtkGetMacro() [5/7]

```
vtkGDCMImageWriter::vtkGetMacro (
            FileLowerLeft ,
            int  )
```

10.373.4.17 vtkGetMacro() [6/7]

```
vtkGDCMImageWriter::vtkGetMacro (
            PlanarConfiguration ,
            int  )
```

10.373.4.18 vtkGetMacro() [7/7]

```
vtkGDCMImageWriter::vtkGetMacro (
            CompressionType ,
            int  )
```

10.373.4.19 vtkGetObjectMacro() [1/3]

```
vtkGDCMImageWriter::vtkGetObjectMacro (
            MedicalImageProperties ,
            vtkMedicalImageProperties  )
```

10.373.4.20 vtkGetObjectMacro() [2/3]

```
vtkGDCMImageWriter::vtkGetObjectMacro (
            FileNames ,
            vtkStringArray  )
```

10.373.4.21 vtkGetObjectMacro() [3/3]

```
vtkGDCMImageWriter::vtkGetObjectMacro (
            DirectionCosines ,
            vtkMatrix4x4  )
```

10.373.4.22 vtkGetStringMacro() [1/2]

```
vtkGDCMImageWriter::vtkGetStringMacro (
            StudyUID  )
```

10.373.4.23 vtkGetStringMacro() [2/2]

```
vtkGDCMImageWriter::vtkGetStringMacro (
            SeriesUID  )
```

10.373.4.24 vtkSetMacro() [1/7]

```
vtkGDCMImageWriter::vtkSetMacro (
            LossyFlag ,
            int  )
```

10.373.4.25 vtkSetMacro() [2/7]

```
vtkGDCMImageWriter::vtkSetMacro (
            Shift ,
            double  )
```

10.373.4.26 vtkSetMacro() [3/7]

```
vtkGDCMImageWriter::vtkSetMacro (
            Scale ,
            double  )
```

10.373.4.27 vtkSetMacro() [4/7]

```
vtkGDCMImageWriter::vtkSetMacro (
            ImageFormat ,
            int  )
```

10.373.4.28 vtkSetMacro() [5/7]

```
vtkGDCMImageWriter::vtkSetMacro (
            FileLowerLeft ,
            int  )
```

10.373.4.29 vtkSetMacro() [6/7]

```
vtkGDCMImageWriter::vtkSetMacro (
            PlanarConfiguration ,
            int  )
```

10.373.4.30 vtkSetMacro() [7/7]

```
vtkGDCMImageWriter::vtkSetMacro (
            CompressionType ,
            int  )
```

10.373.4.31 vtkSetStringMacro() [1/2]

```
vtkGDCMImageWriter::vtkSetStringMacro (
            StudyUID  )
```

10.373.4.32 vtkSetStringMacro() [2/2]

```
vtkGDCMImageWriter::vtkSetStringMacro (
            SeriesUID  )
```

10.373.4.33 vtkTypeRevisionMacro()

```
vtkGDCMImageWriter::vtkTypeRevisionMacro (
            vtkGDCMImageWriter ,
            vtkImageWriter  )
```

10.373.4.34 Write()

```
virtual void vtkGDCMImageWriter::Write ( )  [virtual]
```

Examples:

Convert16BitsTo8Bits.cxx, ConvertMultiFrameToSingleFrame.cxx, ConvertRGBToLuminance.cxx, ConvertSingleBitTo8Bits.cxx, CreateFakePET.cxx, CreateFakeRTDOSE.cxx, gdcmorthoplanes.cxx, HelloActiviz.cs, HelloActiviz2.cs, HelloVTKWorld.cs, HelloVTKWorld.java, HelloVTKWorld2.cs, and MagnifyFile.cxx.

10.373.4.35 WriteGDCMData()

```
int vtkGDCMImageWriter::WriteGDCMData (
            vtkImageData * data,
            int timeStep )  [protected]
```

10.373.4.36 WriteSlice()

```
void vtkGDCMImageWriter::WriteSlice (
            vtkImageData * data )  [protected]
```

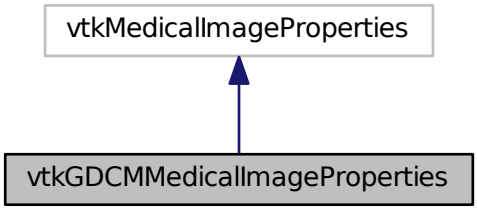
The documentation for this class was generated from the following file:

- vtkGDCMImageWriter.h

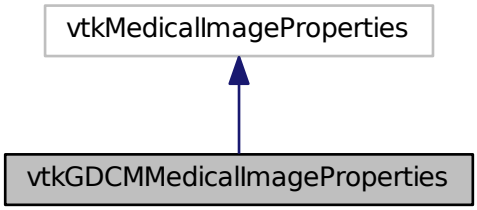
10.374 vtkGDCMMedicalImageProperties Class Reference

```
#include <vtkGDCMMedicalImageProperties.h>
```

Inheritance diagram for vtkGDCMMedicalImageProperties:



Collaboration diagram for vtkGDCMMedicalImageProperties:



Public Member Functions

- virtual void Clear ()
- void PrintSelf (ostream &os, vtkIndent indent)
- vtkTypeRevisionMacro (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

10.374.1 Constructor & Destructor Documentation

10.374.1.1 vtkGDCMMedicalImageProperties()

```
vtkGDCMMedicalImageProperties::vtkGDCMMedicalImageProperties ( )  [protected]
```

10.374.1.2 ~vtkGDCMMedicalImageProperties()

```
vtkGDCMMedicalImageProperties::~vtkGDCMMedicalImageProperties ( )  [protected]
```

10.374.2 Member Function Documentation

10.374.2.1 Clear()

```
virtual void vtkGDCMMedicalImageProperties::Clear ( )  [virtual]
```

10.374.2.2 GetFile()

```
gdcm::File const& vtkGDCMMedicalImageProperties::GetFile (
            unsigned int t )  [protected]
```

10.374.2.3 New()

```
static vtkGDCMMedicalImageProperties* vtkGDCMMedicalImageProperties::New ( )  [static]
```

10.374.2.4 PrintSelf()

```
void vtkGDCMMedicalImageProperties::PrintSelf (
            ostream & os,
            vtkIndent indent )
```

10.374.2.5 PushBackFile()

```
void vtkGDCMMedicalImageProperties::PushBackFile (
            gdcm::File const & f )  [protected]
```

10.374.2.6 vtkTypeRevisionMacro()

```
vtkGDCMMedicalImageProperties::vtkTypeRevisionMacro (
            vtkGDCMMedicalImageProperties ,
            vtkMedicalImageProperties  )
```

10.374.3 Friends And Related Function Documentation

10.374.3.1 vtkGDCMImageReader

```
friend class vtkGDCMImageReader  [friend]
```

10.374.3.2 vtkGDCMImageReader2

```
friend class vtkGDCMImageReader2  [friend]
```

10.374.3.3 vtkGDCMImageWriter

```
friend class vtkGDCMImageWriter  [friend]
```

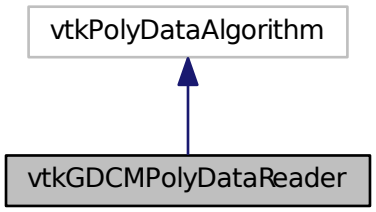
The documentation for this class was generated from the following file:

- vtkGDCMMedicalImageProperties.h

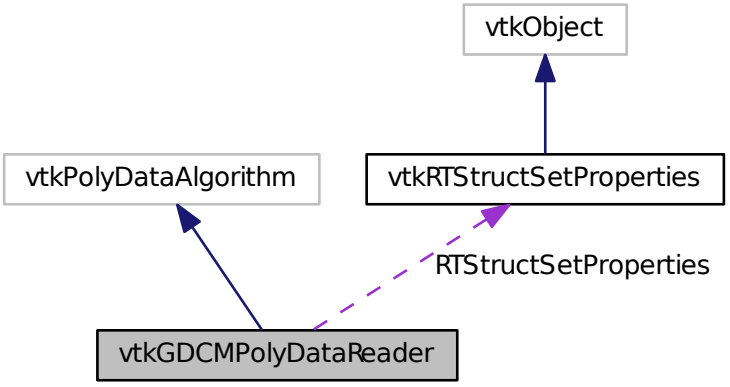
10.375 vtkGDCMPolyDataReader Class Reference

`#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)
- vtkTypeRevisionMacro (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

10.375.1 Detailed Description

Examples:

gdcmscene.cxx, GenerateRTSTRUCT.cxx, and rtstructapp.cxx.

10.375.2 Constructor & Destructor Documentation

10.375.2.1 vtkGDCMPolyDataReader()

```
vtkGDCMPolyDataReader::vtkGDCMPolyDataReader ( )  [protected]
```

10.375.2.2 ~vtkGDCMPolyDataReader()

```
vtkGDCMPolyDataReader::~vtkGDCMPolyDataReader ( )  [protected]
```

10.375.3 Member Function Documentation

10.375.3.1 FillMedicalImageInformation()

```
void vtkGDCMPolyDataReader::FillMedicalImageInformation (
            const gdcm::Reader & reader )  [protected]
```

10.375.3.2 New()

```
static vtkGDCMPolyDataReader* vtkGDCMPolyDataReader::New ( )  [static]
```

Examples:

gdcmscene.cxx, GenerateRTSTRUCT.cxx, and rtstructapp.cxx.

10.375.3.3 PrintSelf()

```
virtual void vtkGDCMPolyDataReader::PrintSelf (
            ostream & os,
            vtkIndent indent )  [virtual]
```

10.375.3.4 RequestData()

```
int vtkGDCMPolyDataReader::RequestData (
            vtkInformation * ,
            vtkInformationVector ** ,
            vtkInformationVector *  )  [protected]
```

10.375.3.5 RequestData_HemodynamicWaveformStorage()

```
int vtkGDCMPolyDataReader::RequestData_HemodynamicWaveformStorage (
            gdcm::Reader const & reader,
            vtkInformationVector * outputVector )  [protected]
```

10.375.3.6 RequestData_RTStructureSetStorage()

```
int vtkGDCMPolyDataReader::RequestData_RTStructureSetStorage (
            gdcm::Reader const & reader,
            vtkInformationVector * outputVector )  [protected]
```

10.375.3.7 RequestInformation()

```
int vtkGDCMPolyDataReader::RequestInformation (
            vtkInformation * vtkNotUsedrequest,
            vtkInformationVector ** vtkNotUsedinputVector,
            vtkInformationVector * outputVector )  [protected]
```

10.375.3.8 RequestInformation_HemodynamicWaveformStorage()

```
int vtkGDCMPolyDataReader::RequestInformation_HemodynamicWaveformStorage (
            gdcm::Reader const & reader )  [protected]
```

10.375.3.9 RequestInformation_RTStructureSetStorage()

```
int vtkGDCMPolyDataReader::RequestInformation_RTStructureSetStorage (
            gdcm::Reader const & reader )  [protected]
```

10.375.3.10 vtkGetObjectMacro() [1/2]

```
vtkGDCMPolyDataReader::vtkGetObjectMacro (
            MedicalImageProperties ,
            vtkMedicalImageProperties  )
```

10.375.3.11 vtkGetObjectMacro() [2/2]

```
vtkGDCMPolyDataReader::vtkGetObjectMacro (
            RTStructSetProperties ,
            vtkRTStructSetProperties  )
```

10.375.3.12 vtkGetStringMacro()

```
vtkGDCMPolyDataReader::vtkGetStringMacro (
            FileName  )
```

10.375.3.13 vtkSetStringMacro()

```
vtkGDCMPolyDataReader::vtkSetStringMacro (
            FileName  )
```

10.375.3.14 vtkTypeRevisionMacro()

```
vtkGDCMPolyDataReader::vtkTypeRevisionMacro (
            vtkGDCMPolyDataReader ,
            vtkPolyDataAlgorithm  )
```

10.375.4 Member Data Documentation

10.375.4.1 FileName

```
char* vtkGDCMPolyDataReader::FileName  [protected]
```

10.375.4.2 MedicalImageProperties

```
vtkMedicalImageProperties* vtkGDCMPolyDataReader::MedicalImageProperties  [protected]
```

10.375.4.3 RTStructSetProperties

```
vtkRTStructSetProperties* vtkGDCMPolyDataReader::RTStructSetProperties  [protected]
```

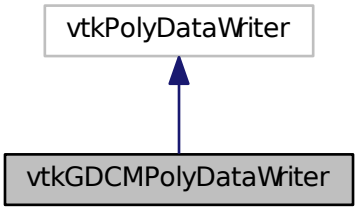
The documentation for this class was generated from the following file:

- vtkGDCMPolyDataReader.h

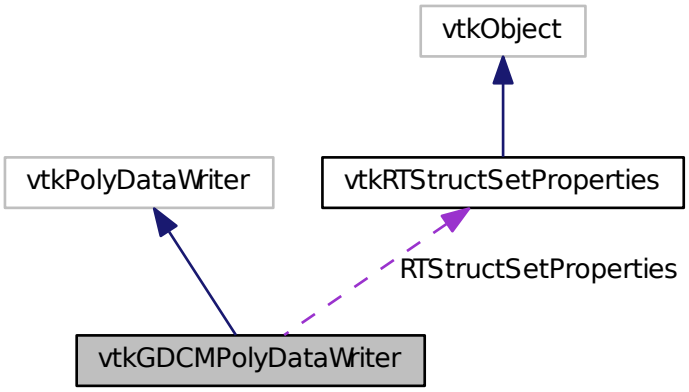
10.376 vtkGDCMPolyDataWriter Class Reference

```
#include <vtkGDCMPolyDataWriter.h>
```

Inheritance diagram for vtkGDCMPolyDataWriter:



Collaboration diagram for vtkGDCMPolyDataWriter:



Public Member Functions

- void InitializeRTStructSet (vtkStdString inDirectory, vtkStdString inStructLabel, vtkStdString inStructName, vtkStringArray *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)
- vtkTypeRevisionMacro (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

10.376.1 Detailed Description

Examples:

GenerateRTSTRUCT.cxx, and rtstructapp.cxx.

10.376.2 Constructor & Destructor Documentation

10.376.2.1 vtkGDCMPolyDataWriter()

```
vtkGDCMPolyDataWriter::vtkGDCMPolyDataWriter ( )  [protected]
```

10.376.2.2 ~vtkGDCMPolyDataWriter()

```
vtkGDCMPolyDataWriter::~vtkGDCMPolyDataWriter ( )  [protected]
```

10.376.3 Member Function Documentation

10.376.3.1 InitializeRTStructSet()

```
void vtkGDCMPolyDataWriter::InitializeRTStructSet (
            vtkStdString inDirectory,
            vtkStdString inStructLabel,
            vtkStdString inStructName,
            vtkStringArray * inROINames,
            vtkStringArray * inROIAlgorithmName,
            vtkStringArray * inROIType )
```

Examples:

GenerateRTSTRUCT.cxx.

10.376.3.2 New()

```
static vtkGDCMPolyDataWriter* vtkGDCMPolyDataWriter::New ( )  [static]
```

Examples:

GenerateRTSTRUCT.cxx, and rtstructapp.cxx.

10.376.3.3 PrintSelf()

```
virtual void vtkGDCMPolyDataWriter::PrintSelf (
            ostream & os,
            vtkIndent indent )  [virtual]
```

10.376.3.4 SetMedicalImageProperties()

```
virtual void vtkGDCMPolyDataWriter::SetMedicalImageProperties (
            vtkMedicalImageProperties * pd )  [virtual]
```

Examples:

GenerateRTSTRUCT.cxx, and rtstructapp.cxx.

10.376.3.5 SetNumberOfInputPorts()

```
void vtkGDCMPolyDataWriter::SetNumberOfInputPorts (
            int n )
```

Examples:

GenerateRTSTRUCT.cxx, and rtstructapp.cxx.

10.376.3.6 SetRTStructSetProperties()

```
virtual void vtkGDCMPolyDataWriter::SetRTStructSetProperties (
            vtkRTStructSetProperties * pd )  [virtual]
```

Examples:

GenerateRTSTRUCT.cxx, and rtstructapp.cxx.

10.376.3.7 vtkTypeRevisionMacro()

```
vtkGDCMPolyDataWriter::vtkTypeRevisionMacro (
            vtkGDCMPolyDataWriter ,
            vtkPolyDataWriter  )
```

10.376.3.8 WriteData()

```
void vtkGDCMPolyDataWriter::WriteData ( )  [protected]
```

10.376.3.9 WriteRTSTRUCTData()

```
void vtkGDCMPolyDataWriter::WriteRTSTRUCTData (
            gdcm::File & file,
            int num )  [protected]
```

10.376.3.10 WriteRTSTRUCTInfo()

```
void vtkGDCMPolyDataWriter::WriteRTSTRUCTInfo (
            gdcm::File & file )  [protected]
```

10.376.4 Member Data Documentation

10.376.4.1 MedicalImageProperties

```
vtkMedicalImageProperties* vtkGDCMPolyDataWriter::MedicalImageProperties  [protected]
```

10.376.4.2 RTStructSetProperties

```
vtkRTStructSetProperties* vtkGDCMPolyDataWriter::RTStructSetProperties  [protected]
```

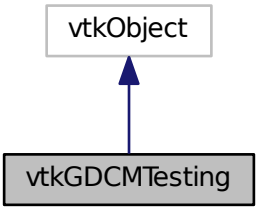
The documentation for this class was generated from the following file:

- vtkGDCMPolyDataWriter.h

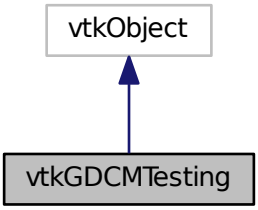
10.377 vtkGDCMTesting Class Reference

```
#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)
- vtkTypeRevisionMacro (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 ()

10.377.1 Detailed Description

Examples:

HelloActiviz5.cs, HelloVTKWorld2.cs, MetaImageMD5Activiz.cs, ReadSeriesIntoVTK.java, and RefCounting.cs.

10.377.2 Member Typedef Documentation

10.377.2.1 MD5MetaImagesType

```
typedef const char* const(* vtkGDCMTesting::MD5MetaImagesType)[3]
```

10.377.3 Constructor & Destructor Documentation

10.377.3.1 vtkGDCMTesting()

```
vtkGDCMTesting::vtkGDCMTesting ( )  [protected]
```

10.377.3.2 ~vtkGDCMTesting()

```
vtkGDCMTesting::~vtkGDCMTesting ( )  [protected]
```

10.377.4 Member Function Documentation

10.377.4.1 GetGDCMDataRoot()

```
static const char* vtkGDCMTesting::GetGDCMDataRoot ( )  [static]
```

Examples:

HelloActiviz5.cs, and ReadSeriesIntoVTK.java.

10.377.4.2 GetMD5MetaImage()

```
static const char* const* vtkGDCMTesting::GetMD5MetaImage (
            unsigned int file )  [static]
```

10.377.4.3 GetMHDMD5FromFile()

```
static const char* vtkGDCMTesting::GetMHDMD5FromFile (
            const char * filepath )  [static]
```

Examples:

MetaImageMD5Activiz.cs.

10.377.4.4 GetNumberOfMD5MetaImages()

```
static unsigned int vtkGDCMTesting::GetNumberOfMD5MetaImages ( )  [static]
```

10.377.4.5 GetRAWMD5FromFile()

```
static const char* vtkGDCMTesting::GetRAWMD5FromFile (
            const char * filepath )  [static]
```

Examples:

MetaImageMD5Activiz.cs.

10.377.4.6 GetVTKDataRoot()

```
static const char* vtkGDCMTesting::GetVTKDataRoot ( )  [static]
```

Examples:

HelloActiviz5.cs, and HelloVTKWorld2.cs.

10.377.4.7 New()

```
static vtkGDCMTesting* vtkGDCMTesting::New ( )  [static]
```

Examples:

RefCounting.cs.

10.377.4.8 PrintSelf()

```
void vtkGDCMTesting::PrintSelf (
            ostream & os,
            vtkIndent indent )
```

10.377.4.9 vtkTypeRevisionMacro()

```
vtkGDCMTesting::vtkTypeRevisionMacro (
            vtkGDCMTesting ,
            vtkObject  )
```

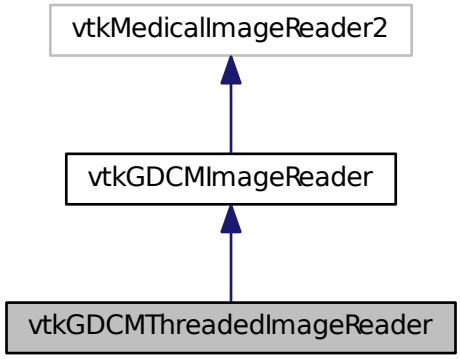
The documentation for this class was generated from the following file:

- vtkGDCMTesting.h

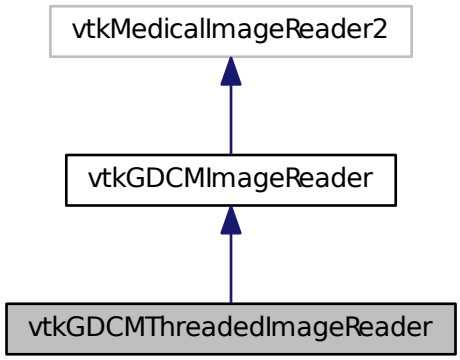
10.378 vtkGDCMThreadedImageReader Class Reference

`#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 (Shift, double)
- vtkSetMacro (Scale, double)
- vtkSetMacro (UseShiftScale, int)
- vtkTypeRevisionMacro (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

10.378.1 Constructor & Destructor Documentation

10.378.1.1 vtkGDCMThreadedImageReader()

```
vtkGDCMThreadedImageReader::vtkGDCMThreadedImageReader ( )  [protected]
```

10.378.1.2 ~vtkGDCMThreadedImageReader()

```
vtkGDCMThreadedImageReader::~vtkGDCMThreadedImageReader ( )  [protected]
```

10.378.2 Member Function Documentation

10.378.2.1 ExecuteData()

```
void vtkGDCMThreadedImageReader::ExecuteData (
            vtkDataObject * out )  [protected]
```

10.378.2.2 ExecuteInformation()

```
void vtkGDCMThreadedImageReader::ExecuteInformation ( )  [protected]
```

10.378.2.3 New()

```
static vtkGDCMThreadedImageReader* vtkGDCMThreadedImageReader::New ( )  [static]
```

10.378.2.4 PrintSelf()

```
virtual void vtkGDCMThreadedImageReader::PrintSelf (
            ostream & os,
            vtkIndent indent )  [virtual]
```

Reimplemented from vtkGDCMImageReader.

10.378.2.5 ReadFiles()

```
void vtkGDCMThreadedImageReader::ReadFiles (
            unsigned int nfiles,
            const char * filenames[] )  [protected]
```

10.378.2.6 RequestDataCompat()

```
void vtkGDCMThreadedImageReader::RequestDataCompat ( )  [protected]
```

10.378.2.7 vtkBooleanMacro()

```
vtkGDCMThreadedImageReader::vtkBooleanMacro (
            UseShiftScale ,
            int  )
```

10.378.2.8 vtkGetMacro()

```
vtkGDCMThreadedImageReader::vtkGetMacro (
            UseShiftScale ,
            int  )
```

10.378.2.9 vtkSetMacro() [1/3]

```
vtkGDCMThreadedImageReader::vtkSetMacro (
            Shift ,
            double  )
```

10.378.2.10 vtkSetMacro() [2/3]

```
vtkGDCMThreadedImageReader::vtkSetMacro (
            Scale ,
            double  )
```

10.378.2.11 vtkSetMacro() [3/3]

```
vtkGDCMThreadedImageReader::vtkSetMacro (
            UseShiftScale ,
            int  )
```

10.378.2.12 vtkTypeRevisionMacro()

```
vtkGDCMThreadedImageReader::vtkTypeRevisionMacro (
            vtkGDCMThreadedImageReader ,
            vtkGDCMImageReader  )
```

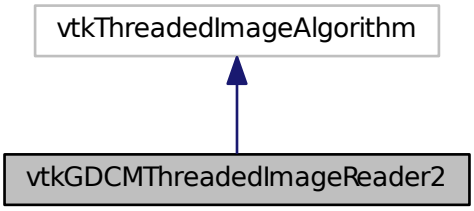
The documentation for this class was generated from the following file:

- vtkGDCMThreadedImageReader.h

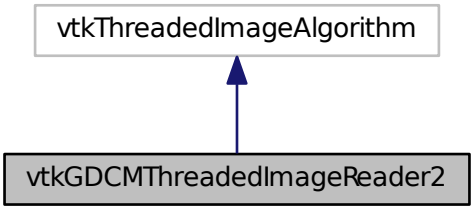
10.379 vtkGDCMThreadedImageReader2 Class Reference

```
#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 (FileLowerLeft, int)
- vtkGetMacro (NumberOfOverlays, int)
- vtkGetMacro (DataScalarType, int)
- vtkGetMacro (NumberOfScalarComponents, int)
- vtkGetMacro (LoadOverlays, int)
- vtkGetMacro (Shift, double)
- vtkGetMacro (Scale, double)
- vtkGetMacro (UseShiftScale, int)
- vtkGetObjectMacro (FileNames, vtkStringArray)
- vtkGetVector3Macro (DataOrigin, double)
- vtkGetVector3Macro (DataSpacing, double)
- vtkGetVector6Macro (DataExtent, int)
- vtkSetMacro (FileLowerLeft, int)
- vtkSetMacro (DataScalarType, int)
- vtkSetMacro (NumberOfScalarComponents, int)
- vtkSetMacro (LoadOverlays, int)
- vtkSetMacro (Shift, double)
- vtkSetMacro (Scale, double)
- vtkSetMacro (UseShiftScale, int)
- vtkSetVector3Macro (DataOrigin, double)
- vtkSetVector3Macro (DataSpacing, double)
- vtkSetVector6Macro (DataExtent, int)
- vtkTypeRevisionMacro (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)

10.379.1 Constructor & Destructor Documentation

10.379.1.1 vtkGDCMThreadedImageReader2()

```
vtkGDCMThreadedImageReader2::vtkGDCMThreadedImageReader2 ( )  [protected]
```

10.379.1.2 ~vtkGDCMThreadedImageReader2()

```
vtkGDCMThreadedImageReader2::~vtkGDCMThreadedImageReader2 ( )  [protected]
```

10.379.2 Member Function Documentation

10.379.2.1 GetFileName()

```
virtual const char* vtkGDCMThreadedImageReader2::GetFileName (
            int i = 0 )  [virtual]
```

10.379.2.2 New()

```
static vtkGDCMThreadedImageReader2* vtkGDCMThreadedImageReader2::New ( )  [static]
```

10.379.2.3 PrintSelf()

```
virtual void vtkGDCMThreadedImageReader2::PrintSelf (
            ostream & os,
            vtkIndent indent )  [virtual]
```

10.379.2.4 RequestInformation()

```
int vtkGDCMThreadedImageReader2::RequestInformation (
            vtkInformation * request,
            vtkInformationVector ** inputVector,
            vtkInformationVector * outputVector )  [protected]
```

10.379.2.5 SetFileName()

```
virtual void vtkGDCMThreadedImageReader2::SetFileName (
            const char * filename )  [virtual]
```

10.379.2.6 SetFileNames()

```
virtual void vtkGDCMThreadedImageReader2::SetFileNames (
            vtkStringArray *  )  [virtual]
```

10.379.2.7 SplitExtent()

```
int vtkGDCMThreadedImageReader2::SplitExtent (
            int splitExt[6],
            int startExt[6],
            int num,
            int total )
```

10.379.2.8 ThreadedRequestData()

```
void vtkGDCMThreadedImageReader2::ThreadedRequestData (
            vtkInformation * request,
            vtkInformationVector ** inputVector,
            vtkInformationVector * outputVector,
            vtkImageData *** inData,
            vtkImageData ** outData,
            int outExt[6],
            int id )  [protected]
```

10.379.2.9 vtkBooleanMacro() [1/3]

```
vtkGDCMThreadedImageReader2::vtkBooleanMacro (
            FileLowerLeft ,
            int  )
```

10.379.2.10 vtkBooleanMacro() [2/3]

```
vtkGDCMThreadedImageReader2::vtkBooleanMacro (
            LoadOverlays ,
            int  )
```

10.379.2.11 vtkBooleanMacro() [3/3]

```
vtkGDCMThreadedImageReader2::vtkBooleanMacro (
            UseShiftScale ,
            int  )
```

10.379.2.12 vtkGetMacro() [1/8]

```
vtkGDCMThreadedImageReader2::vtkGetMacro (
            FileLowerLeft ,
            int  )
```

10.379.2.13 vtkGetMacro() [2/8]

```
vtkGDCMThreadedImageReader2::vtkGetMacro (
            NumberOfOverlays ,
            int  )
```

10.379.2.14 vtkGetMacro() [3/8]

```
vtkGDCMThreadedImageReader2::vtkGetMacro (
            DataScalarType ,
            int  )
```

10.379.2.15 vtkGetMacro() [4/8]

```
vtkGDCMThreadedImageReader2::vtkGetMacro (
            NumberOfScalarComponents ,
            int  )
```

10.379.2.16 vtkGetMacro() [5/8]

```
vtkGDCMThreadedImageReader2::vtkGetMacro (
            LoadOverlays ,
            int  )
```

10.379.2.17 vtkGetMacro() [6/8]

```
vtkGDCMThreadedImageReader2::vtkGetMacro (
            Shift ,
            double  )
```

10.379.2.18 vtkGetMacro() [7/8]

```
vtkGDCMThreadedImageReader2::vtkGetMacro (
            Scale ,
            double  )
```

10.379.2.19 vtkGetMacro() [8/8]

```
vtkGDCMThreadedImageReader2::vtkGetMacro (
            UseShiftScale ,
            int  )
```

10.379.2.20 vtkGetObjectMacro()

```
vtkGDCMThreadedImageReader2::vtkGetObjectMacro (
            FileNames ,
            vtkStringArray  )
```

10.379.2.21 vtkGetVector3Macro() [1/2]

```
vtkGDCMThreadedImageReader2::vtkGetVector3Macro (
            DataOrigin ,
            double  )
```

10.379.2.22 vtkGetVector3Macro() [2/2]

```
vtkGDCMThreadedImageReader2::vtkGetVector3Macro (
            DataSpacing ,
            double  )
```

10.379.2.23 vtkGetVector6Macro()

```
vtkGDCMThreadedImageReader2::vtkGetVector6Macro (
            DataExtent ,
            int  )
```

10.379.2.24 vtkSetMacro() [1/7]

```
vtkGDCMThreadedImageReader2::vtkSetMacro (
            FileLowerLeft ,
            int  )
```

10.379.2.25 vtkSetMacro() [2/7]

```
vtkGDCMThreadedImageReader2::vtkSetMacro (
            DataScalarType ,
            int  )
```

10.379.2.26 vtkSetMacro() [3/7]

```
vtkGDCMThreadedImageReader2::vtkSetMacro (
            NumberOfScalarComponents ,
            int  )
```

10.379.2.27 vtkSetMacro() [4/7]

```
vtkGDCMThreadedImageReader2::vtkSetMacro (
            LoadOverlays ,
            int  )
```

10.379.2.28 vtkSetMacro() [5/7]

```
vtkGDCMThreadedImageReader2::vtkSetMacro (
            Shift ,
            double  )
```

10.379.2.29 vtkSetMacro() [6/7]

```
vtkGDCMThreadedImageReader2::vtkSetMacro (
            Scale ,
            double  )
```

10.379.2.30 vtkSetMacro() [7/7]

```
vtkGDCMThreadedImageReader2::vtkSetMacro (
            UseShiftScale ,
            int  )
```

10.379.2.31 vtkSetVector3Macro() [1/2]

```
vtkGDCMThreadedImageReader2::vtkSetVector3Macro (
            DataOrigin ,
            double  )
```

10.379.2.32 vtkSetVector3Macro() [2/2]

```
vtkGDCMThreadedImageReader2::vtkSetVector3Macro (
            DataSpacing ,
            double  )
```

10.379.2.33 vtkSetVector6Macro()

```
vtkGDCMThreadedImageReader2::vtkSetVector6Macro (
            DataExtent ,
            int  )
```

10.379.2.34 vtkTypeRevisionMacro()

```
vtkGDCMThreadedImageReader2::vtkTypeRevisionMacro (
            vtkGDCMThreadedImageReader2 ,
            vtkThreadedImageAlgorithm  )
```

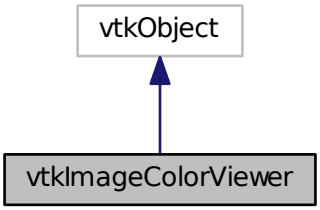
The documentation for this class was generated from the following file:

- vtkGDCMThreadedImageReader2.h

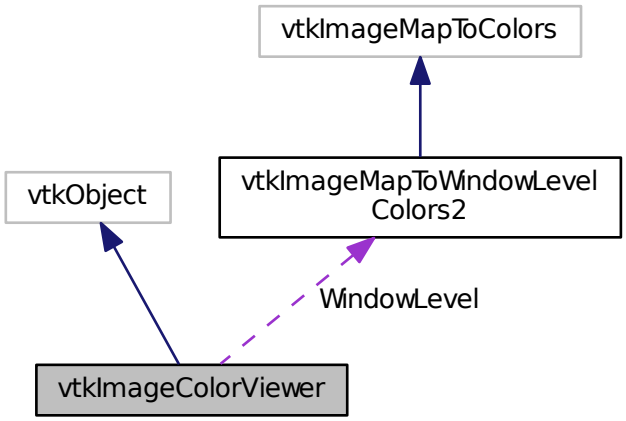
10.380 vtkImageColorViewer Class Reference

```
#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 void GetSliceRange (int range[2])
- virtual void GetSliceRange (int &min, int &max)
- virtual int * GetSliceRange ()
- 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 GetWholeZMin())
- VTK_LEGACY (int GetWholeZMax())
- VTK_LEGACY (int GetZSlice())
- VTK_LEGACY (void SetZSlice(int))
- vtkBooleanMacro (OffScreenRendering, int)
- vtkGetMacro (SliceOrientation, int)
- vtkGetMacro (Slice, int)
- vtkGetObjectMacro (RenderWindow, vtkRenderWindow)
- vtkGetObjectMacro (Renderer, vtkRenderer)
- vtkGetObjectMacro (ImageActor, vtkImageActor)
- vtkGetObjectMacro (WindowLevel, vtkImageMapToWindowLevelColors2)
- vtkGetObjectMacro (InteractorStyle, vtkInteractorStyleImage)
- vtkTypeRevisionMacro (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

10.380.1 Detailed Description

Examples:

gdcmrtionplan.cxx, and gdcmrtplan.cxx.

10.380.2 Member Enumeration Documentation

10.380.2.1 anonymous enum

```
anonymous enum
```

Enumerator

SLICE_ORIENTATION_YZ	
SLICE_ORIENTATION_XZ	
SLICE_ORIENTATION_XY	

10.380.3 Constructor & Destructor Documentation

10.380.3.1 vtkImageColorViewer()

```
vtkImageColorViewer::vtkImageColorViewer ( )  [protected]
```

10.380.3.2 ~vtkImageColorViewer()

```
vtkImageColorViewer::~vtkImageColorViewer ( )  [protected]
```

10.380.4 Member Function Documentation

10.380.4.1 AddInput()

```
virtual void vtkImageColorViewer::AddInput (
            vtkImageData * input )  [virtual]
```

10.380.4.2 AddInputConnection()

```
virtual void vtkImageColorViewer::AddInputConnection (
            vtkAlgorithmOutput * input )  [virtual]
```

10.380.4.3 GetColorLevel()

```
virtual double vtkImageColorViewer::GetColorLevel ( )  [virtual]
```

10.380.4.4 GetColorWindow()

```
virtual double vtkImageColorViewer::GetColorWindow ( )  [virtual]
```

10.380.4.5 GetInput()

```
virtual vtkImageData* vtkImageColorViewer::GetInput ( )  [virtual]
```

10.380.4.6 GetOffScreenRendering()

```
virtual int vtkImageColorViewer::GetOffScreenRendering ( )  [virtual]
```

10.380.4.7 GetOverlayVisibility()

```
double vtkImageColorViewer::GetOverlayVisibility ( )
```

10.380.4.8 GetPosition()

```
virtual int* vtkImageColorViewer::GetPosition ( )  [virtual]
```

10.380.4.9 GetSize()

```
virtual int* vtkImageColorViewer::GetSize ( )  [virtual]
```

10.380.4.10 GetSliceMax()

```
virtual int vtkImageColorViewer::GetSliceMax ( )  [virtual]
```

10.380.4.11 GetSliceMin()

```
virtual int vtkImageColorViewer::GetSliceMin ( )  [virtual]
```

10.380.4.12 GetSliceRange() [1/3]

```
virtual void vtkImageColorViewer::GetSliceRange (
            int range[2] )  [inline], [virtual]
```

10.380.4.13 GetSliceRange() [2/3]

```
virtual void vtkImageColorViewer::GetSliceRange (
            int & min,
            int & max )  [virtual]
```

10.380.4.14 GetSliceRange() [3/3]

```
virtual int* vtkImageColorViewer::GetSliceRange ( )  [virtual]
```

10.380.4.15 GetWindowName()

```
virtual const char* vtkImageColorViewer::GetWindowName ( )  [virtual]
```

10.380.4.16 InstallPipeline()

```
virtual void vtkImageColorViewer::InstallPipeline ( )  [protected], [virtual]
```

10.380.4.17 New()

```
static vtkImageColorViewer* vtkImageColorViewer::New ( )  [static]
```

Examples:

gdcmrtionplan.cxx, and gdcmrtplan.cxx.

10.380.4.18 PrintSelf()

```
void vtkImageColorViewer::PrintSelf (
            ostream & os,
            vtkIndent indent )
```

10.380.4.19 Render()

```
virtual void vtkImageColorViewer::Render (
            void  )  [virtual]
```

Examples:

gdcmrtionplan.cxx, and gdcmrtplan.cxx.

10.380.4.20 SetColorLevel()

```
virtual void vtkImageColorViewer::SetColorLevel (
            double s )  [virtual]
```

10.380.4.21 SetColorWindow()

```
virtual void vtkImageColorViewer::SetColorWindow (
            double s )  [virtual]
```

10.380.4.22 SetDisplayId()

```
virtual void vtkImageColorViewer::SetDisplayId (
            void * a )  [virtual]
```

10.380.4.23 SetInput()

```
virtual void vtkImageColorViewer::SetInput (
            vtkImageData * in )  [virtual]
```

Examples:

gdcmrtionplan.cxx, and gdcmrtplan.cxx.

10.380.4.24 SetInputConnection()

```
virtual void vtkImageColorViewer::SetInputConnection (
            vtkAlgorithmOutput * input )  [virtual]
```

10.380.4.25 SetOffScreenRendering()

```
virtual void vtkImageColorViewer::SetOffScreenRendering (
            int  )  [virtual]
```

10.380.4.26 SetOverlayVisibility()

```
void vtkImageColorViewer::SetOverlayVisibility (
            double vis )
```

10.380.4.27 SetParentId()

```
virtual void vtkImageColorViewer::SetParentId (
            void * a )  [virtual]
```

10.380.4.28 SetPosition() [1/2]

```
virtual void vtkImageColorViewer::SetPosition (
            int a,
            int b )  [virtual]
```

10.380.4.29 SetPosition() [2/2]

```
virtual void vtkImageColorViewer::SetPosition (
            int a[2] )  [inline], [virtual]
```

References SetPosition().

Referenced by SetPosition().

10.380.4.30 SetRenderer()

```
virtual void vtkImageColorViewer::SetRenderer (
            vtkRenderer * arg )  [virtual]
```

10.380.4.31 SetRenderWindow()

```
virtual void vtkImageColorViewer::SetRenderWindow (
            vtkRenderWindow * arg )  [virtual]
```

10.380.4.32 SetSize() [1/2]

```
virtual void vtkImageColorViewer::SetSize (
            int a,
            int b )  [virtual]
```

Examples:

gdcmrtionplan.cxx, and gdcmrtplan.cxx.

10.380.4.33 SetSize() [2/2]

```
virtual void vtkImageColorViewer::SetSize (
            int a[2] )  [inline], [virtual]
```

References SetSize().

Referenced by SetSize().

10.380.4.34 SetSlice()

```
virtual void vtkImageColorViewer::SetSlice (
            int s )  [virtual]
```

10.380.4.35 SetSliceOrientation()

```
virtual void vtkImageColorViewer::SetSliceOrientation (
            int orientation )  [virtual]
```

10.380.4.36 SetSliceOrientationToXY()

```
virtual void vtkImageColorViewer::SetSliceOrientationToXY ( )  [inline], [virtual]
```

References SLICE_ORIENTATION_XY.

10.380.4.37 SetSliceOrientationToXZ()

```
virtual void vtkImageColorViewer::SetSliceOrientationToXZ ( )  [inline], [virtual]
```

References SLICE_ORIENTATION_XZ.

10.380.4.38 SetSliceOrientationToYZ()

```
virtual void vtkImageColorViewer::SetSliceOrientationToYZ ( )  [inline], [virtual]
```

References SLICE_ORIENTATION_YZ.

10.380.4.39 SetupInteractor()

```
virtual void vtkImageColorViewer::SetupInteractor (
            vtkRenderWindowInteractor *  )  [virtual]
```

Examples:

gdcmrtionplan.cxx, and gdcmrtplan.cxx.

10.380.4.40 SetWindowId()

```
virtual void vtkImageColorViewer::SetWindowId (
            void * a )  [virtual]
```

10.380.4.41 UnInstallPipeline()

```
virtual void vtkImageColorViewer::UnInstallPipeline ( )  [protected], [virtual]
```

10.380.4.42 UpdateDisplayExtent()

```
virtual void vtkImageColorViewer::UpdateDisplayExtent ( )  [virtual]
```

10.380.4.43 UpdateOrientation()

```
virtual void vtkImageColorViewer::UpdateOrientation ( )  [protected], [virtual]
```

10.380.4.44 VTK_LEGACY() [1/4]

```
vtkImageColorViewer::VTK_LEGACY (
            int  GetWholeZMin() )
```

10.380.4.45 VTK_LEGACY() [2/4]

```
vtkImageColorViewer::VTK_LEGACY (
            int  GetWholeZMax() )
```

10.380.4.46 VTK_LEGACY() [3/4]

```
vtkImageColorViewer::VTK_LEGACY (
            int  GetZSlice() )
```

10.380.4.47 VTK_LEGACY() [4/4]

```
vtkImageColorViewer::VTK_LEGACY (
            void  SetZSliceint )
```

10.380.4.48 vtkBooleanMacro()

```
vtkImageColorViewer::vtkBooleanMacro (
            OffScreenRendering ,
            int  )
```

10.380.4.49 vtkGetMacro() [1/2]

```
vtkImageColorViewer::vtkGetMacro (
            SliceOrientation ,
            int  )
```

10.380.4.50 vtkGetMacro() [2/2]

```
vtkImageColorViewer::vtkGetMacro (
            Slice ,
            int  )
```

10.380.4.51 vtkGetObjectMacro() [1/5]

```
vtkImageColorViewer::vtkGetObjectMacro (
            RenderWindow ,
            vtkRenderWindow  )
```

10.380.4.52 vtkGetObjectMacro() [2/5]

```
vtkImageColorViewer::vtkGetObjectMacro (
            Renderer ,
            vtkRenderer  )
```

10.380.4.53 vtkGetObjectMacro() [3/5]

```
vtkImageColorViewer::vtkGetObjectMacro (
             ImageActor ,
             vtkImageActor  )
```

10.380.4.54 vtkGetObjectMacro() [4/5]

```
vtkImageColorViewer::vtkGetObjectMacro (
             WindowLevel ,
             vtkImageMapToWindowLevelColors2  )
```

10.380.4.55 vtkGetObjectMacro() [5/5]

```
vtkImageColorViewer::vtkGetObjectMacro (
             InteractorStyle ,
             vtkInteractorStyleImage  )
```

10.380.4.56 vtkTypeRevisionMacro()

```
vtkImageColorViewer::vtkTypeRevisionMacro (
             vtkImageColorViewer ,
             vtkObject  )
```

10.380.5 Friends And Related Function Documentation

10.380.5.1 vtkImageColorViewerCallback

```
friend class vtkImageColorViewerCallback  [friend]
```

10.380.6 Member Data Documentation

10.380.6.1 FirstRender

```
int vtkImageColorViewer::FirstRender  [protected]
```

10.380.6.2 ImageActor

```
vtkImageActor* vtkImageColorViewer::ImageActor  [protected]
```

10.380.6.3 Interactor

```
vtkRenderWindowInteractor* vtkImageColorViewer::Interactor  [protected]
```

10.380.6.4 InteractorStyle

```
vtkInteractorStyleImage* vtkImageColorViewer::InteractorStyle  [protected]
```

10.380.6.5 OverlayImageActor

```
vtkImageActor* vtkImageColorViewer::OverlayImageActor  [protected]
```

10.380.6.6 Renderer

```
vtkRenderer* vtkImageColorViewer::Renderer  [protected]
```

10.380.6.7 RenderWindow

```
vtkRenderWindow* vtkImageColorViewer::RenderWindow  [protected]
```

10.380.6.8 Slice

```
int vtkImageColorViewer::Slice  [protected]
```

10.380.6.9 SliceOrientation

```
int vtkImageColorViewer::SliceOrientation  [protected]
```

10.380.6.10 WindowLevel

```
vtkImageMapToWindowLevelColors2* vtkImageColorViewer::WindowLevel  [protected]
```

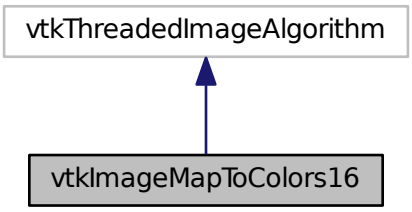
The documentation for this class was generated from the following file:

- vtkImageColorViewer.h

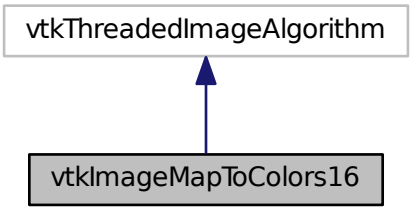
10.381 vtkImageMapToColors16 Class Reference

`#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 (OutputFormat, int)
- vtkGetMacro (ActiveComponent, int)

- vtkGetMacro (PassAlphaToOutput, int)
- vtkGetObjectMacro (LookupTable, vtkScalarsToColors)
- vtkSetMacro (OutputFormat, int)
- vtkSetMacro (ActiveComponent, int)
- vtkSetMacro (PassAlphaToOutput, int)
- vtkTypeRevisionMacro (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

10.381.1 Constructor & Destructor Documentation

10.381.1.1 vtkImageMapToColors16()

```
vtkImageMapToColors16::vtkImageMapToColors16 ( )  [protected]
```

10.381.1.2 ~vtkImageMapToColors16()

```
vtkImageMapToColors16::~vtkImageMapToColors16 ( )  [protected]
```

10.381.2 Member Function Documentation

10.381.2.1 GetMTime()

```
virtual unsigned long vtkImageMapToColors16::GetMTime ( )  [virtual]
```

10.381.2.2 New()

```
static vtkImageMapToColors16* vtkImageMapToColors16::New ( )  [static]
```

10.381.2.3 PrintSelf()

```
void vtkImageMapToColors16::PrintSelf (
            ostream & os,
            vtkIndent indent )
```

10.381.2.4 RequestData()

```
virtual int vtkImageMapToColors16::RequestData (
            vtkInformation * request,
            vtkInformationVector ** inputVector,
            vtkInformationVector * outputVector )  [protected], [virtual]
```

10.381.2.5 RequestInformation()

```
virtual int vtkImageMapToColors16::RequestInformation (
            vtkInformation * ,
            vtkInformationVector ** ,
            vtkInformationVector *  )  [protected], [virtual]
```

10.381.2.6 SetLookupTable()

```
virtual void vtkImageMapToColors16::SetLookupTable (
            vtkScalarsToColors *  )  [virtual]
```

10.381.2.7 SetOutputFormatToLuminance()

```
void vtkImageMapToColors16::SetOutputFormatToLuminance ( )  [inline]
```

10.381.2.8 SetOutputFormatToLuminanceAlpha()

```
void vtkImageMapToColors16::SetOutputFormatToLuminanceAlpha ( )  [inline]
```

10.381.2.9 SetOutputFormatToRGB()

```
void vtkImageMapToColors16::SetOutputFormatToRGB ( )  [inline]
```

10.381.2.10 SetOutputFormatToRGBA()

```
void vtkImageMapToColors16::SetOutputFormatToRGBA ( )  [inline]
```

10.381.2.11 ThreadedRequestData()

```
void vtkImageMapToColors16::ThreadedRequestData (
            vtkInformation * request,
            vtkInformationVector ** inputVector,
            vtkInformationVector * outputVector,
            vtkImageData *** inData,
            vtkImageData ** outData,
            int extent[6],
            int id )  [protected]
```

10.381.2.12 vtkBooleanMacro()

```
vtkImageMapToColors16::vtkBooleanMacro (
            PassAlphaToOutput ,
            int  )
```

10.381.2.13 vtkGetMacro() [1/3]

```
vtkImageMapToColors16::vtkGetMacro (
            OutputFormat ,
            int  )
```

10.381.2.14 vtkGetMacro() [2/3]

```
vtkImageMapToColors16::vtkGetMacro (
            ActiveComponent ,
            int  )
```

10.381.2.15 vtkGetMacro() [3/3]

```
vtkImageMapToColors16::vtkGetMacro (
            PassAlphaToOutput ,
            int  )
```

10.381.2.16 vtkGetObjectMacro()

```
vtkImageMapToColors16::vtkGetObjectMacro (
            LookupTable ,
            vtkScalarsToColors  )
```

10.381.2.17 vtkSetMacro() [1/3]

```
vtkImageMapToColors16::vtkSetMacro (
            OutputFormat ,
            int  )
```

10.381.2.18 vtkSetMacro() [2/3]

```
vtkImageMapToColors16::vtkSetMacro (
            ActiveComponent ,
            int  )
```

10.381.2.19 vtkSetMacro() [3/3]

```
vtkImageMapToColors16::vtkSetMacro (
            PassAlphaToOutput ,
            int  )
```

10.381.2.20 vtkTypeRevisionMacro()

```
vtkImageMapToColors16::vtkTypeRevisionMacro (
            vtkImageMapToColors16 ,
            vtkThreadedImageAlgorithm  )
```

10.381.3 Member Data Documentation

10.381.3.1 ActiveComponent

```
int vtkImageMapToColors16::ActiveComponent  [protected]
```

10.381.3.2 DataWasPassed

```
int vtkImageMapToColors16::DataWasPassed  [protected]
```

10.381.3.3 LookupTable

```
vtkScalarsToColors* vtkImageMapToColors16::LookupTable  [protected]
```

10.381.3.4 OutputFormat

```
int vtkImageMapToColors16::OutputFormat  [protected]
```

10.381.3.5 PassAlphaToOutput

```
int vtkImageMapToColors16::PassAlphaToOutput  [protected]
```

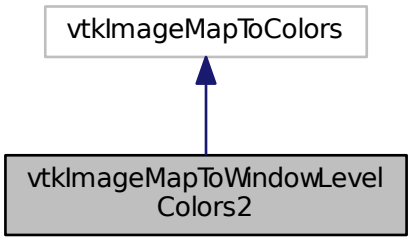
The documentation for this class was generated from the following file:

- vtkImageMapToColors16.h

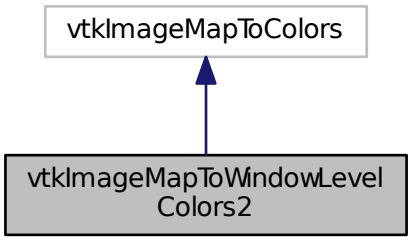
10.382 vtkImageMapToWindowLevelColors2 Class Reference

```
#include <vtkImageMapToWindowLevelColors2.h>
```

Inheritance diagram for vtkImageMapToWindowLevelColors2:



Collaboration diagram for vtkImageMapToWindowLevelColors2:



Public Member Functions

- void PrintSelf (ostream &os, vtkIndent indent)
- vtkGetMacro (Window, double)
- vtkGetMacro (Level, double)
- vtkSetMacro (Window, double)
- vtkSetMacro (Level, double)
- vtkTypeRevisionMacro (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

10.382.1 Constructor & Destructor Documentation

10.382.1.1 vtkImageMapToWindowLevelColors2()

```
vtkImageMapToWindowLevelColors2::vtkImageMapToWindowLevelColors2 ( )  [protected]
```

10.382.1.2 ~vtkImageMapToWindowLevelColors2()

```
vtkImageMapToWindowLevelColors2::~vtkImageMapToWindowLevelColors2 ( )  [protected]
```

10.382.2 Member Function Documentation

10.382.2.1 New()

```
static vtkImageMapToWindowLevelColors2* vtkImageMapToWindowLevelColors2::New ( )  [static]
```

10.382.2.2 PrintSelf()

```
void vtkImageMapToWindowLevelColors2::PrintSelf (
            ostream & os,
            vtkIndent indent )
```

10.382.2.3 RequestData()

```
virtual int vtkImageMapToWindowLevelColors2::RequestData (
            vtkInformation * request,
            vtkInformationVector ** inputVector,
            vtkInformationVector * outputVector )  [protected], [virtual]
```

10.382.2.4 RequestInformation()

```
virtual int vtkImageMapToWindowLevelColors2::RequestInformation (
            vtkInformation * ,
            vtkInformationVector ** ,
            vtkInformationVector *  )  [protected], [virtual]
```

10.382.2.5 ThreadedRequestData()

```
void vtkImageMapToWindowLevelColors2::ThreadedRequestData (
            vtkInformation * request,
            vtkInformationVector ** inputVector,
            vtkInformationVector * outputVector,
            vtkImageData *** inData,
            vtkImageData ** outData,
            int extent[6],
            int id )  [protected]
```

10.382.2.6 vtkGetMacro() [1/2]

```
vtkImageMapToWindowLevelColors2::vtkGetMacro (
            Window ,
            double  )
```

10.382.2.7 vtkGetMacro() [2/2]

```
vtkImageMapToWindowLevelColors2::vtkGetMacro (
            Level ,
            double  )
```

10.382.2.8 vtkSetMacro() [1/2]

```
vtkImageMapToWindowLevelColors2::vtkSetMacro (
            Window ,
            double  )
```

10.382.2.9 vtkSetMacro() [2/2]

```
vtkImageMapToWindowLevelColors2::vtkSetMacro (
            Level ,
            double  )
```

10.382.2.10 vtkTypeRevisionMacro()

```
vtkImageMapToWindowLevelColors2::vtkTypeRevisionMacro (
            vtkImageMapToWindowLevelColors2 ,
            vtkImageMapToColors  )
```

10.382.3 Member Data Documentation

10.382.3.1 Level

```
double vtkImageMapToWindowLevelColors2::Level  [protected]
```

10.382.3.2 Window

```
double vtkImageMapToWindowLevelColors2::Window  [protected]
```

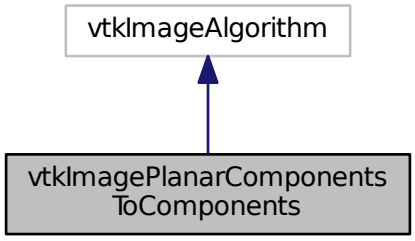
The documentation for this class was generated from the following file:

- vtkImageMapToWindowLevelColors2.h

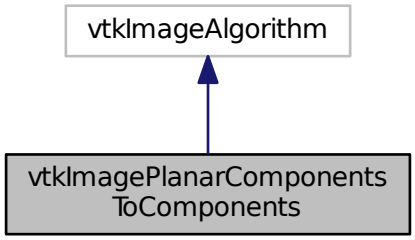
10.383 vtkImagePlanarComponentsToComponents Class Reference

`#include <vtkImagePlanarComponentsToComponents.h>`

Inheritance diagram for vtkImagePlanarComponentsToComponents:



Collaboration diagram for vtkImagePlanarComponentsToComponents:



Public Member Functions

- void PrintSelf (ostream &os, vtkIndent indent)
- vtkTypeRevisionMacro (vtkImagePlanarComponentsToComponents, vtkImageAlgorithm)

Static Public Member Functions

- static vtkImagePlanarComponentsToComponents * New ()

Protected Member Functions

- vtkImagePlanarComponentsToComponents ()
- ~vtkImagePlanarComponentsToComponents ()
- virtual int RequestData (vtkInformation *, vtkInformationVector **, vtkInformationVector *)

10.383.1 Constructor & Destructor Documentation

10.383.1.1 vtkImagePlanarComponentsToComponents()

```
vtkImagePlanarComponentsToComponents::vtkImagePlanarComponentsToComponents ( )  [protected]
```

10.383.1.2 ~vtkImagePlanarComponentsToComponents()

```
vtkImagePlanarComponentsToComponents::~vtkImagePlanarComponentsToComponents ( )  [inline], [protected]
```

10.383.2 Member Function Documentation

10.383.2.1 New()

```
static vtkImagePlanarComponentsToComponents* vtkImagePlanarComponentsToComponents::New ( )  [static]
```

10.383.2.2 PrintSelf()

```
void vtkImagePlanarComponentsToComponents::PrintSelf (
            ostream & os,
            vtkIndent indent )
```

10.383.2.3 RequestData()

```
virtual int vtkImagePlanarComponentsToComponents::RequestData (
            vtkInformation * ,
            vtkInformationVector ** ,
            vtkInformationVector *  )  [protected], [virtual]
```

10.383.2.4 vtkTypeRevisionMacro()

```
vtkImagePlanarComponentsToComponents::vtkTypeRevisionMacro (
            vtkImagePlanarComponentsToComponents ,
            vtkImageAlgorithm  )
```

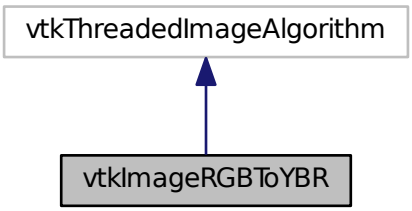
The documentation for this class was generated from the following file:

- vtkImagePlanarComponentsToComponents.h

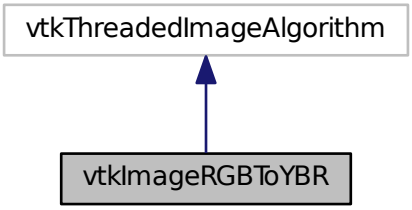
10.384 vtkImageRGBToYBR Class Reference

`#include <vtkImageRGBToYBR.h>`

Inheritance diagram for vtkImageRGBToYBR:



Collaboration diagram for vtkImageRGBToYBR:



Public Member Functions

- void PrintSelf (ostream &os, vtkIndent indent)
- vtkTypeRevisionMacro (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)

10.384.1 Constructor & Destructor Documentation

10.384.1.1 vtkImageRGBToYBR()

```
vtkImageRGBToYBR::vtkImageRGBToYBR ( )  [protected]
```

10.384.1.2 ~vtkImageRGBToYBR()

```
vtkImageRGBToYBR::~vtkImageRGBToYBR ( )  [inline], [protected]
```

10.384.2 Member Function Documentation

10.384.2.1 New()

```
static vtkImageRGBToYBR* vtkImageRGBToYBR::New ( )  [static]
```

10.384.2.2 PrintSelf()

```
void vtkImageRGBToYBR::PrintSelf (
            ostream & os,
            vtkIndent indent )
```

10.384.2.3 ThreadedExecute()

```
void vtkImageRGBToYBR::ThreadedExecute (
            vtkImageData * inData,
            vtkImageData * outData,
            int ext[6],
            int id )  [protected]
```

10.384.2.4 vtkTypeRevisionMacro()

```
vtkImageRGBToYBR::vtkTypeRevisionMacro (
            vtkImageRGBToYBR ,
            vtkThreadedImageAlgorithm  )
```

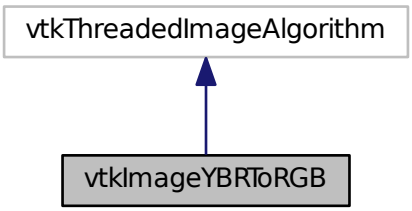
The documentation for this class was generated from the following file:

- vtkImageRGBToYBR.h

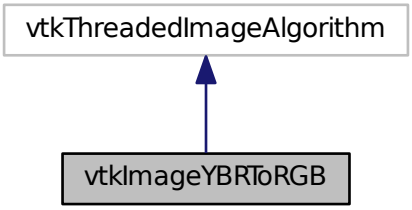
10.385 vtkImageYBRToRGB Class Reference

`#include <vtkImageYBRToRGB.h>`

Inheritance diagram for vtkImageYBRToRGB:



Collaboration diagram for vtkImageYBRToRGB:



Public Member Functions

- void PrintSelf (ostream &os, vtkIndent indent)
- vtkTypeRevisionMacro (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)

10.385.1 Constructor & Destructor Documentation

10.385.1.1 vtkImageYBRToRGB()

```
vtkImageYBRToRGB::vtkImageYBRToRGB ( )  [protected]
```

10.385.1.2 ~vtkImageYBRToRGB()

```
vtkImageYBRToRGB::~vtkImageYBRToRGB ( )  [inline], [protected]
```

10.385.2 Member Function Documentation

10.385.2.1 New()

```
static vtkImageYBRToRGB* vtkImageYBRToRGB::New ( )  [static]
```

10.385.2.2 PrintSelf()

```
void vtkImageYBRToRGB::PrintSelf (
            ostream & os,
            vtkIndent indent )
```

10.385.2.3 ThreadedExecute()

```
void vtkImageYBRToRGB::ThreadedExecute (
            vtkImageData * inData,
            vtkImageData * outData,
            int ext[6],
            int id )  [protected]
```

10.385.2.4 vtkTypeRevisionMacro()

```
vtkImageYBRToRGB::vtkTypeRevisionMacro (
            vtkImageYBRToRGB ,
            vtkThreadedImageAlgorithm  )
```

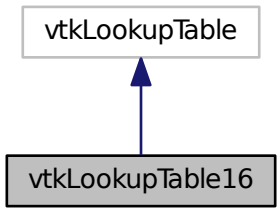
The documentation for this class was generated from the following file:

- vtkImageYBRToRGB.h

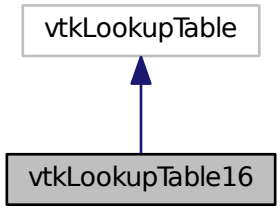
10.386 vtkLookupTable16 Class Reference

`#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)
- vtkTypeRevisionMacro (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

10.386.1 Constructor & Destructor Documentation

10.386.1.1 vtkLookupTable16()

```
vtkLookupTable16::vtkLookupTable16 (
            int sze = 256,
            int ext = 256 )  [protected]
```

10.386.1.2 ~vtkLookupTable16()

```
vtkLookupTable16::~vtkLookupTable16 ( )  [protected]
```

10.386.2 Member Function Documentation

10.386.2.1 Build()

```
void vtkLookupTable16::Build ( )
```

10.386.2.2 GetPointer()

```
unsigned short* vtkLookupTable16::GetPointer (
            const vtkIdType id )  [inline]
```

10.386.2.3 MapScalarsThroughTable2()

```
void vtkLookupTable16::MapScalarsThroughTable2 (
            void * input,
            unsigned char * output,
            int inputDataType,
            int numberOfValues,
            int inputIncrement,
            int outputFormat )  [protected]
```

10.386.2.4 New()

```
static vtkLookupTable16* vtkLookupTable16::New ( )  [static]
```

10.386.2.5 PrintSelf()

```
void vtkLookupTable16::PrintSelf (
            ostream & os,
            vtkIndent indent )
```

10.386.2.6 SetNumberOfTableValues()

```
void vtkLookupTable16::SetNumberOfTableValues (
            vtkIdType number )
```

10.386.2.7 vtkTypeRevisionMacro()

```
vtkLookupTable16::vtkTypeRevisionMacro (
            vtkLookupTable16 ,
            vtkLookupTable  )
```

10.386.2.8 WritePointer()

```
unsigned char * vtkLookupTable16::WritePointer (
            const vtkIdType id,
            const int number )  [inline]
```

10.386.3 Member Data Documentation

10.386.3.1 Table16

```
vtkUnsignedShortArray* vtkLookupTable16::Table16  [protected]
```

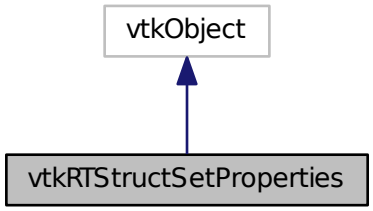
The documentation for this class was generated from the following file:

- vtkLookupTable16.h

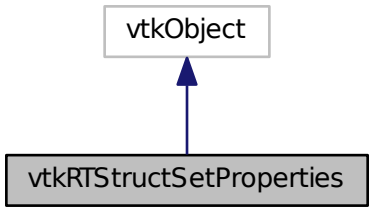
10.387 vtkRTStructSetProperties Class Reference

`#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 ∗refframerefuid, const char ∗roiname, const char ∗ROI↩ GenerationAlgorithm, 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 (StructureSetLabel)
- vtkGetStringMacro (StructureSetName)
- vtkGetStringMacro (StructureSetDate)
- vtkGetStringMacro (StructureSetTime)
- vtkGetStringMacro (SOPInstanceUID)
- vtkGetStringMacro (StudyInstanceUID)
- vtkGetStringMacro (SeriesInstanceUID)
- vtkGetStringMacro (ReferenceSeriesInstanceUID)
- vtkGetStringMacro (ReferenceFrameOfReferenceUID)
- vtkSetStringMacro (StructureSetLabel)
- vtkSetStringMacro (StructureSetName)
- vtkSetStringMacro (StructureSetDate)
- vtkSetStringMacro (StructureSetTime)
- vtkSetStringMacro (SOPInstanceUID)
- vtkSetStringMacro (StudyInstanceUID)
- vtkSetStringMacro (SeriesInstanceUID)
- vtkSetStringMacro (ReferenceSeriesInstanceUID)
- vtkSetStringMacro (ReferenceFrameOfReferenceUID)
- vtkTypeRevisionMacro (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

10.387.1 Detailed Description

Examples:

GenerateRTSTRUCT.cxx.

10.387.2 Constructor & Destructor Documentation

10.387.2.1 vtkRTStructSetProperties()

```
vtkRTStructSetProperties::vtkRTStructSetProperties ( )  [protected]
```

10.387.2.2 ~vtkRTStructSetProperties()

```
vtkRTStructSetProperties::~vtkRTStructSetProperties ( )  [protected]
```

10.387.3 Member Function Documentation

10.387.3.1 AddContourReferencedFrameOfReference()

```
void vtkRTStructSetProperties::AddContourReferencedFrameOfReference (
            vtkIdType pdnum,
            const char * classuid,
            const char * instanceuid )
```

10.387.3.2 AddReferencedFrameOfReference()

```
void vtkRTStructSetProperties::AddReferencedFrameOfReference (
            const char * classuid,
            const char * instanceuid )
```

10.387.3.3 AddStructureSetROI()

```
void vtkRTStructSetProperties::AddStructureSetROI (
            int roinumber,
            const char * reframerefuid,
            const char * roiname,
            const char * ROIGenerationAlgorithm,
            const char * ROIDescription = 0 )
```

10.387.3.4 AddStructureSetROIObservation()

```
void vtkRTStructSetProperties::AddStructureSetROIObservation (
            int refnumber,
            int observationnumber,
            const char * rtroiinterpretedtype,
            const char * roiinterpreter,
            const char * roiobservationlabel = 0 )
```

10.387.3.5 Clear()

```
virtual void vtkRTStructSetProperties::Clear ( )  [virtual]
```

10.387.3.6 DeepCopy()

```
virtual void vtkRTStructSetProperties::DeepCopy (
            vtkRTStructSetProperties * p )  [virtual]
```

10.387.3.7 GetContourReferencedFrameOfReferenceClassUID()

```
const char* vtkRTStructSetProperties::GetContourReferencedFrameOfReferenceClassUID (
            vtkIdType pdnum,
            vtkIdType id )
```

10.387.3.8 GetContourReferencedFrameOfReferenceInstanceUID()

```
const char* vtkRTStructSetProperties::GetContourReferencedFrameOfReferenceInstanceUID (
            vtkIdType pdnum,
            vtkIdType id )
```

10.387.3.9 GetNumberOfContourReferencedFrameOfReferences() [1/2]

```
vtkIdType vtkRTStructSetProperties::GetNumberOfContourReferencedFrameOfReferences ( )
```

10.387.3.10 GetNumberOfContourReferencedFrameOfReferences() [2/2]

```
vtkIdType vtkRTStructSetProperties::GetNumberOfContourReferencedFrameOfReferences (
            vtkIdType pdnum )
```

10.387.3.11 GetNumberOfReferencedFrameOfReferences()

```
vtkIdType vtkRTStructSetProperties::GetNumberOfReferencedFrameOfReferences ( )
```

10.387.3.12 GetNumberOfStructureSetROIs()

```
vtkIdType vtkRTStructSetProperties::GetNumberOfStructureSetROIs ( )
```

10.387.3.13 GetReferencedFrameOfReferenceClassUID()

```
const char* vtkRTStructSetProperties::GetReferencedFrameOfReferenceClassUID (
            vtkIdType id )
```

10.387.3.14 GetReferencedFrameOfReferenceInstanceUID()

```
const char* vtkRTStructSetProperties::GetReferencedFrameOfReferenceInstanceUID (
            vtkIdType id )
```

10.387.3.15 GetStructureSetObservationNumber()

```
int vtkRTStructSetProperties::GetStructureSetObservationNumber (
            vtkIdType id )
```

10.387.3.16 GetStructureSetROIDescription()

```
const char* vtkRTStructSetProperties::GetStructureSetROIDescription (
            vtkIdType id )
```

10.387.3.17 GetStructureSetROIGenerationAlgorithm()

```
const char* vtkRTStructSetProperties::GetStructureSetROIGenerationAlgorithm (
            vtkIdType  )
```

10.387.3.18 GetStructureSetROIName()

```
const char* vtkRTStructSetProperties::GetStructureSetROIName (
            vtkIdType  )
```

10.387.3.19 GetStructureSetROINumber()

```
int vtkRTStructSetProperties::GetStructureSetROINumber (
            vtkIdType id )
```

10.387.3.20 GetStructureSetROIObservationLabel()

```
const char* vtkRTStructSetProperties::GetStructureSetROIObservationLabel (
            vtkIdType id )
```

10.387.3.21 GetStructureSetROIRefFrameRefUID()

```
const char* vtkRTStructSetProperties::GetStructureSetROIRefFrameRefUID (
            vtkIdType  )
```

10.387.3.22 GetStructureSetRTROIInterpretedType()

```
const char* vtkRTStructSetProperties::GetStructureSetRTROIInterpretedType (
            vtkIdType id )
```

10.387.3.23 New()

```
static vtkRTStructSetProperties* vtkRTStructSetProperties::New ( )  [static]
```

Examples:

GenerateRTSTRUCT.cxx.

10.387.3.24 PrintSelf()

```
void vtkRTStructSetProperties::PrintSelf (
            ostream & os,
            vtkIndent indent )
```

10.387.3.25 vtkGetStringMacro() [1/9]

```
vtkRTStructSetProperties::vtkGetStringMacro (
            StructureSetLabel  )
```

10.387.3.26 vtkGetStringMacro() [2/9]

```
vtkRTStructSetProperties::vtkGetStringMacro (
            StructureSetName  )
```

10.387.3.27 vtkGetStringMacro() [3/9]

```
vtkRTStructSetProperties::vtkGetStringMacro (
            StructureSetDate  )
```

10.387.3.28 vtkGetStringMacro() [4/9]

```
vtkRTStructSetProperties::vtkGetStringMacro (
            StructureSetTime  )
```

10.387.3.29 vtkGetStringMacro() [5/9]

```
vtkRTStructSetProperties::vtkGetStringMacro (
            SOPInstanceUID  )
```

10.387.3.30 vtkGetStringMacro() [6/9]

```
vtkRTStructSetProperties::vtkGetStringMacro (
            StudyInstanceUID  )
```

10.387.3.31 vtkGetStringMacro() [7/9]

```
vtkRTStructSetProperties::vtkGetStringMacro (
            SeriesInstanceUID  )
```

10.387.3.32 vtkGetStringMacro() [8/9]

```
vtkRTStructSetProperties::vtkGetStringMacro (
            ReferenceSeriesInstanceUID  )
```

10.387.3.33 vtkGetStringMacro() [9/9]

```
vtkRTStructSetProperties::vtkGetStringMacro (
            ReferenceFrameOfReferenceUID  )
```

10.387.3.34 vtkSetStringMacro() [1/9]

```
vtkRTStructSetProperties::vtkSetStringMacro (
            StructureSetLabel  )
```

10.387.3.35 vtkSetStringMacro() [2/9]

```
vtkRTStructSetProperties::vtkSetStringMacro (
            StructureSetName  )
```

10.387.3.36 vtkSetStringMacro() [3/9]

```
vtkRTStructSetProperties::vtkSetStringMacro (
            StructureSetDate  )
```

10.387.3.37 vtkSetStringMacro() [4/9]

```
vtkRTStructSetProperties::vtkSetStringMacro (
            StructureSetTime  )
```

10.387.3.38 vtkSetStringMacro() [5/9]

```
vtkRTStructSetProperties::vtkSetStringMacro (
            SOPInstanceUID  )
```

10.387.3.39 vtkSetStringMacro() [6/9]

```
vtkRTStructSetProperties::vtkSetStringMacro (
            StudyInstanceUID  )
```

10.387.3.40 vtkSetStringMacro() [7/9]

```
vtkRTStructSetProperties::vtkSetStringMacro (
            SeriesInstanceUID  )
```

10.387.3.41 vtkSetStringMacro() [8/9]

```
vtkRTStructSetProperties::vtkSetStringMacro (
            ReferenceSeriesInstanceUID  )
```

10.387.3.42 vtkSetStringMacro() [9/9]

```
vtkRTStructSetProperties::vtkSetStringMacro (
            ReferenceFrameOfReferenceUID  )
```

10.387.3.43 vtkTypeRevisionMacro()

```
vtkRTStructSetProperties::vtkTypeRevisionMacro (
            vtkRTStructSetProperties ,
            vtkObject  )
```

10.387.4 Member Data Documentation

10.387.4.1 Internals

```
vtkRTStructSetPropertiesInternals* vtkRTStructSetProperties::Internals  [protected]
```

10.387.4.2 ReferenceFrameOfReferenceUID

```
char* vtkRTStructSetProperties::ReferenceFrameOfReferenceUID  [protected]
```

10.387.4.3 ReferenceSeriesInstanceUID

```
char* vtkRTStructSetProperties::ReferenceSeriesInstanceUID  [protected]
```

10.387.4.4 SeriesInstanceUID

```
char* vtkRTStructSetProperties::SeriesInstanceUID  [protected]
```

10.387.4.5 SOPInstanceUID

```
char* vtkRTStructSetProperties::SOPInstanceUID  [protected]
```

10.387.4.6 StructureSetDate

```
char* vtkRTStructSetProperties::StructureSetDate  [protected]
```

10.387.4.7 StructureSetLabel

```
char* vtkRTStructSetProperties::StructureSetLabel  [protected]
```

10.387.4.8 StructureSetName

```
char* vtkRTStructSetProperties::StructureSetName  [protected]
```

10.387.4.9 StructureSetTime

```
char* vtkRTStructSetProperties::StructureSetTime  [protected]
```

10.387.4.10 StudyInstanceUID

```
char* vtkRTStructSetProperties::StudyInstanceUID  [protected]
```

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

- vtkRTStructSetProperties.h

10.388 gdcm::Waveform Class Reference

Waveform class.

```
#include <gdcmWaveform.h>
```

Public Member Functions

- Waveform ()

10.388.1 Detailed Description

Waveform class.

10.388.2 Constructor & Destructor Documentation

10.388.2.1 Waveform()

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

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

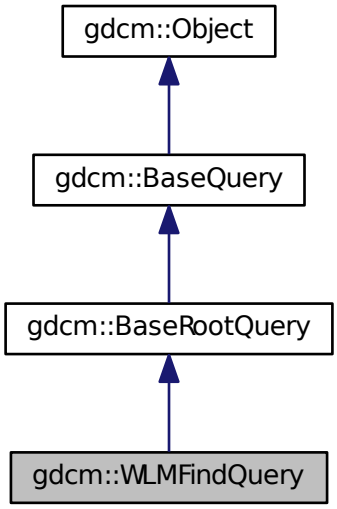
- gdcmWaveform.h

10.389 gdcm::WLMFindQuery Class Reference

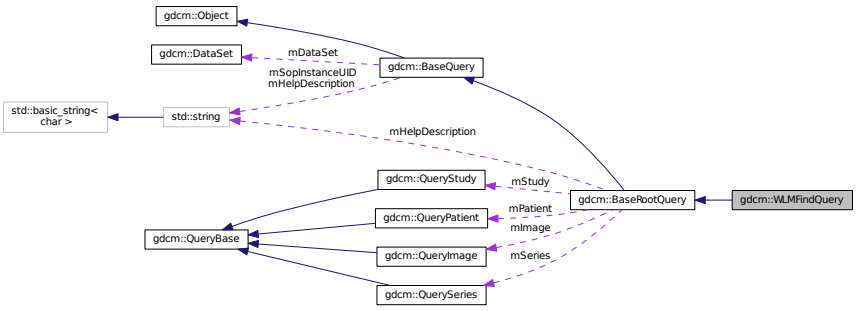
PatientRootQuery.

```
#include <gdcmWLMFindQuery.h>
```

Inheritance diagram for gdcm::WLMFindQuery:



Collaboration diagram for gdcm::WLMFindQuery:



Public Member Functions

- WLMFindQuery ()
- UIDs::TSName GetAbstractSyntaxUID () const
- std::vector< Tag > GetTagListByLevel (const EQueryLevel &inQueryLevel)
- void InitializeDataSet (const EQueryLevel &inQueryLevel)
- bool ValidateQuery (bool inStrict=true) const

Protected Member Functions

- DataSet GetValidDataSet () const

Friends

- class QueryFactory

Additional Inherited Members

10.389.1 Detailed Description

PatientRootQuery.

contains: the class which will produce a dataset for c-find with patient root

10.389.2 Constructor & Destructor Documentation

10.389.2.1 WLMFindQuery()

```
gdcm::WLMFindQuery::WLMFindQuery ( )
```

10.389.3 Member Function Documentation

10.389.3.1 GetAbstractSyntaxUID()

```
UIDs::TSName gdcm::WLMFindQuery::GetAbstractSyntaxUID ( ) const  [virtual]
```

Implements gdcm::BaseQuery.

10.389.3.2 GetTagListByLevel()

```
std::vector<Tag> gdcm::WLMFindQuery::GetTagListByLevel (
            const EQueryLevel & inQueryLevel )  [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.

10.389.3.3 GetValidDataSet()

```
DataSet gdcm::WLMFindQuery::GetValidDataSet ( ) const  [protected]
```

10.389.3.4 InitializeDataSet()

```
void gdcm::WLMFindQuery::InitializeDataSet (
            const EQueryLevel & inQueryLevel )  [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.

10.389.3.5 ValidateQuery()

```
bool gdcm::WLMFindQuery::ValidateQuery (
            bool inStrict = true ) const  [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.

10.389.4 Friends And Related Function Documentation

10.389.4.1 QueryFactory

```
friend class QueryFactory  [friend]
```

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

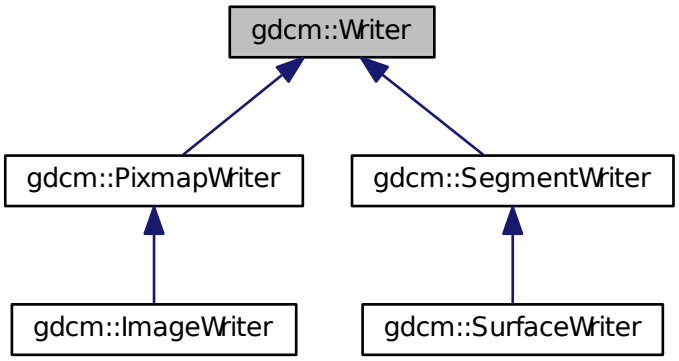
- gdcmWLMFindQuery.h

10.390 gdcm::Writer Class Reference

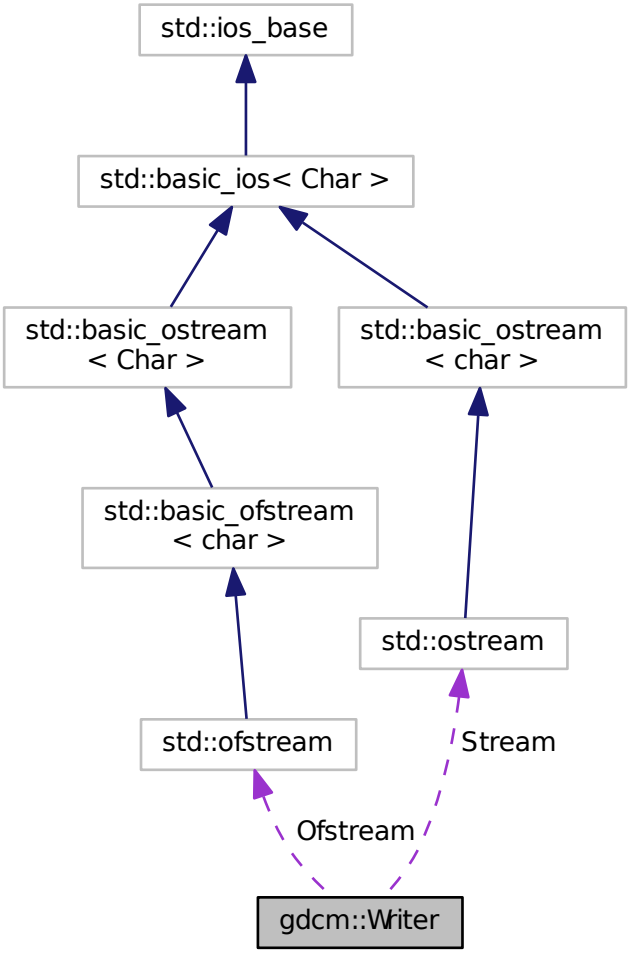
Writer ala DOM (Document Object Model)

```
#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

10.390.1 Detailed Description

Writer ala DOM (Document Object Model)

This class is a non-validating writer, it will only performs well- formedness check only.

Detailled 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 (garanteed 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:

ChangePrivateTags.cxx, ChangeSequenceUltrasound.cxx, ClinicalTrialAnnotate.cxx, CreateFakeRTDOSE.cxx, CreateJPIPDataSet.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, HelloWorld.cxx, LargeVRDSExplicit.cxx, MakeTemplate.cxx, PatchFile.cxx, pmsct_rgb1.cxx, QIDO-RS.cxx, rle2img.cxx, and StreamImageReaderTest.cxx.

10.390.2 Constructor & Destructor Documentation

10.390.2.1 Writer()

```
gdcm::Writer::Writer ( )
```

10.390.2.2 ~Writer()

```
virtual gdcm::Writer::~Writer ( )  [virtual]
```

10.390.3 Member Function Documentation

10.390.3.1 CheckFileMetaInformationOff()

```
void gdcm::Writer::CheckFileMetaInformationOff ( )  [inline]
```

Examples:

CreateFakeRTDOSE.cxx, FixBrokenJ2K.cxx, and HelloWorld.cxx.

10.390.3.2 CheckFileMetaInformationOn()

```
void gdcm::Writer::CheckFileMetaInformationOn ( )  [inline]
```

10.390.3.3 GetCheckFileMetaInformation()

```
bool gdcm::Writer::GetCheckFileMetaInformation ( ) const  [inline], [protected]
```

10.390.3.4 GetFile()

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

Examples:

CreateJPIPDataSet.cxx, EncapsulateFileInRawData.cxx, Extracting_All_Resolution.cxx, Fake_Image_Using_↩ Stream_Image_Writer.cxx, GenAllVR.cxx, GenFakeIdentifyFile.cxx, GetSubSequenceData.cxx, iU22tomultisc.↩ cxx, pmsct_rgb1.cxx, QIDO-RS.cxx, rle2img.cxx, and StreamImageReaderTest.cxx.

10.390.3.5 GetStreamPtr()

```
std::ostream* gdcm::Writer::GetStreamPtr ( ) const  [inline], [protected]
```

10.390.3.6 SetCheckFileMetaInformation()

```
void gdcm::Writer::SetCheckFileMetaInformation (
            bool b )  [inline]
```

Undocumented function, do not use (= leave default)

Examples:

GenAllVR.cxx, GenFakeIdentifyFile.cxx, and PatchFile.cxx.

10.390.3.7 SetFile()

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

Set/Get the DICOM file (DataSet + Header)

Examples:

ChangePrivateTags.cxx, ChangeSequenceUltrasound.cxx, ClinicalTrialAnnotate.cxx, CompressImage.cxx, CreateFakeRTDOSE.cxx, DuplicatePCDE.cxx, FixBrokenJ2K.cxx, FixJAIBugJPEGLS.cxx, FixOrientation.cxx, GenFakeImage.cxx, GenLongSeqs.cxx, GenSeqs.cxx, HelloWorld.cxx, LargeVRDSExplicit.cxx, MakeTemplate.↩ cxx, MergeTwoFiles.cxx, NewSequence.cs, PatchFile.cxx, pmsct_rgb1.cxx, and rle2img.cxx.

10.390.3.8 SetFileName()

```
void gdcm::Writer::SetFileName (
            const char * filename_native )
```

Set the filename of DICOM file to write:

Examples:

ChangePrivateTags.cxx, ChangeSequenceUltrasound.cxx, ClinicalTrialAnnotate.cxx, CompressImage.cxx, CreateARGBImage.cxx, CreateCMYKImage.cxx, CreateFakeRTDOSE.cxx, CreateJPIPDataSet.cxx, csa2img.cxx, DuplicatePCDE.cxx, EncapsulateFileInRawData.cxx, FixBrokenJ2K.cxx, FixJAIBugJPEGLS.cxx, FixOrientation.cxx, GenAllVR.cxx, GenFakeIdentifyFile.cxx, GenFakeImage.cxx, GenLongSeqs.cxx, GenSeqs.cxx, GetSubSequenceData.cxx, HelloVizWorld.cxx, HelloWorld.cxx, iU22tomultisc.cxx, LargeVRDSExplicit.cxx, MakeTemplate.cxx, MergeTwoFiles.cxx, PatchFile.cxx, pmsct_rgb1.cxx, QIDO-RS.cxx, and rle2img.cxx.

10.390.3.9 SetStream()

```
void gdcm::Writer::SetStream (
            std::ostream & output_stream )  [inline]
```

Set user ostream buffer.

10.390.3.10 SetWriteDataSetOnly()

```
void gdcm::Writer::SetWriteDataSetOnly (
            bool b )  [inline], [protected]
```

10.390.3.11 Write()

```
virtual bool gdcm::Writer::Write ( )  [virtual]
```

Main function to tell the writer to write.

Reimplemented in gdcm::PixmapWriter, gdcm::ImageWriter, gdcm::SurfaceWriter, and gdcm::SegmentWriter.

Examples:

ChangePrivateTags.cxx, ChangeSequenceUltrasound.cxx, ClinicalTrialAnnotate.cxx, CreateFakeRTDOSE.cxx, CreateJPIPDataSet.cxx, DuplicatePCDE.cxx, EncapsulateFileInRawData.cxx, FixBrokenJ2K.cxx, FixJAIBugJPEGLS.cxx, FixOrientation.cxx, GenAllVR.cxx, GenFakeIdentifyFile.cxx, GenLongSeqs.cxx, GenSeqs.cxx, HelloWorld.cxx, LargeVRDSExplicit.cxx, MakeTemplate.cxx, PatchFile.cxx, pmsct_rgb1.cxx, QIDO-RS.cxx, and rle2img.cxx.

10.390.4 Friends And Related Function Documentation

10.390.4.1 StreamImageWriter

```
friend class StreamImageWriter  [friend]
```

10.390.5 Member Data Documentation

10.390.5.1 Ofstream

```
std::ofstream* gdcm::Writer::Ofstream  [protected]
```

10.390.5.2 Stream

```
std::ostream* gdcm::Writer::Stream  [protected]
```

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

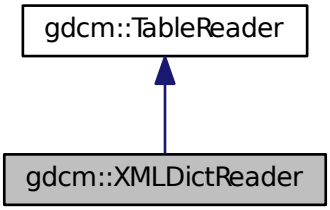
- gdcmWriter.h

10.391 gdcm::XMLDictReader Class Reference

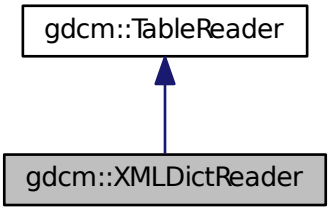
Class for representing a XMLDictReader.

```
#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)

10.391.1 Detailed Description

Class for representing a XMLDictReader.

Note

bla Will read the DICOMV3.xml file

10.391.2 Constructor & Destructor Documentation

10.391.2.1 XMLDictReader()

```
gdcm::XMLDictReader::XMLDictReader ( )
```

10.391.2.2 ~XMLDictReader()

```
gdcm::XMLDictReader::~XMLDictReader ( )  [inline]
```

10.391.3 Member Function Documentation

10.391.3.1 CharacterDataHandler()

```
void gdcm::XMLDictReader::CharacterDataHandler (
            const char * data,
            int length )  [virtual]
```

Reimplemented from gdcm::TableReader.

10.391.3.2 EndElement()

```
void gdcm::XMLDictReader::EndElement (
            const char * name )  [virtual]
```

Reimplemented from gdcm::TableReader.

10.391.3.3 GetDict()

```
const Dict& gdcm::XMLDictReader::GetDict ( )  [inline]
```

10.391.3.4 HandleDescription()

```
void gdcm::XMLDictReader::HandleDescription (
            const char ** atts )  [protected]
```

10.391.3.5 HandleEntry()

```
void gdcm::XMLDictReader::HandleEntry (
            const char ** atts )  [protected]
```

10.391.3.6 StartElement()

```
void gdcm::XMLDictReader::StartElement (
            const char * name,
            const char ** atts )  [virtual]
```

Reimplemented from gdcm::TableReader.

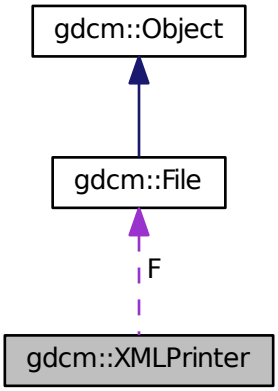
The documentation for this class was generated from the following file:

- gdcmXMLDictReader.h

10.392 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

10.392.1 Member Enumeration Documentation

10.392.1.1 PrintStyles

```
enum gdcm::XMLPrinter::PrintStyles
```

Enumerator

OnlyUUID	
LOADBULKDATA	

10.392.2 Constructor & Destructor Documentation

10.392.2.1 XMLPrinter()

```
gdcm::XMLPrinter::XMLPrinter ( )
```

10.392.2.2 ~XMLPrinter()

```
virtual gdcm::XMLPrinter::~XMLPrinter ( )  [virtual]
```

10.392.3 Member Function Documentation

10.392.3.1 GetPrintStyle()

```
PrintStyles gdcm::XMLPrinter::GetPrintStyle ( ) const  [inline]
```

10.392.3.2 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

10.392.3.3 Print()

```
void gdcm::XMLPrinter::Print (
            std::ostream & os )
```

10.392.3.4 PrintDataElement()

```
VR gdcm::XMLPrinter::PrintDataElement (
            std::ostream & os,
            const Dicts & dicts,
            const DataSet & ds,
            const DataElement & de,
            const TransferSyntax & ts )  [protected]
```

10.392.3.5 PrintDataSet()

```
void gdcm::XMLPrinter::PrintDataSet (
            const DataSet & ds,
            const TransferSyntax & ts,
            std::ostream & os )
```

10.392.3.6 PrintSQ()

```
void gdcm::XMLPrinter::PrintSQ (
            const SequenceOfItems * sqi,
            const TransferSyntax & ts,
            std::ostream & os )  [protected]
```

10.392.3.7 SetFile()

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

10.392.3.8 SetStyle()

```
void gdcm::XMLPrinter::SetStyle (
            PrintStyles ps )  [inline]
```

10.392.4 Member Data Documentation

10.392.4.1 F

```
const File* gdcm::XMLPrinter::F  [protected]
```

10.392.4.2 PrintStyle

```
PrintStyles gdcm::XMLPrinter::PrintStyle  [protected]
```

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

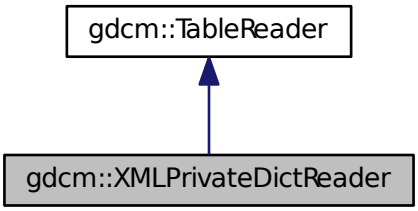
- gdcmXMLPrinter.h

10.393 gdcm::XMLPrivateDictReader Class Reference

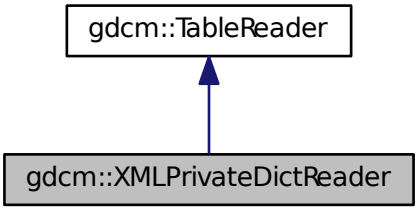
Class for representing a XMLPrivateDictReader.

```
#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)

10.393.1 Detailed Description

Class for representing a XMLPrivateDictReader.

Note

bla Will read the Private.xml file

10.393.2 Constructor & Destructor Documentation

10.393.2.1 XMLPrivateDictReader()

```
gdcm::XMLPrivateDictReader::XMLPrivateDictReader ( )
```

10.393.2.2 ~XMLPrivateDictReader()

```
gdcm::XMLPrivateDictReader::~XMLPrivateDictReader ( )  [inline]
```

10.393.3 Member Function Documentation

10.393.3.1 CharacterDataHandler()

```
void gdcm::XMLPrivateDictReader::CharacterDataHandler (
            const char * data,
            int length )  [virtual]
```

Reimplemented from gdcm::TableReader.

10.393.3.2 EndElement()

```
void gdcm::XMLPrivateDictReader::EndElement (
            const char * name )  [virtual]
```

Reimplemented from gdcm::TableReader.

10.393.3.3 GetPrivateDict()

```
const PrivateDict& gdcm::XMLPrivateDictReader::GetPrivateDict ( )  [inline]
```

10.393.3.4 HandleDescription()

```
void gdcm::XMLPrivateDictReader::HandleDescription (
            const char ** atts )  [protected]
```

10.393.3.5 HandleEntry()

```
void gdcm::XMLPrivateDictReader::HandleEntry (
            const char ** atts )  [protected]
```

10.393.3.6 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

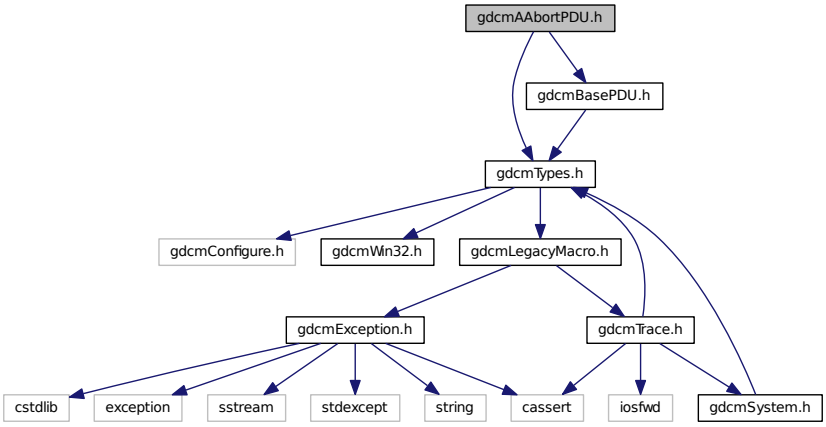
Chapter 11

File Documentation

11.1 gdcmAAbortPDU.h File Reference

```
#include "gdcmTypes.h"
#include "gdcmBasePDU.h"
```

Include dependency graph for gdcmAAbortPDU.h:



Classes

- class gdcm::network::AAbortPDU

  AAbortPDU.

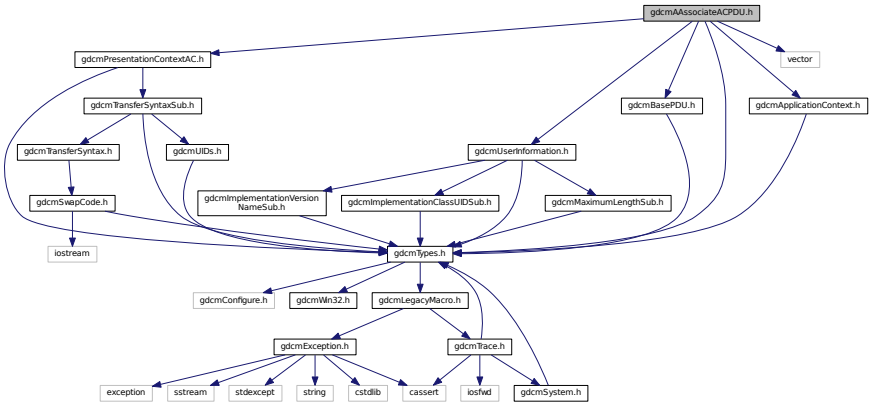
Namespaces

- gdcm
- gdcm::network

11.2 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.

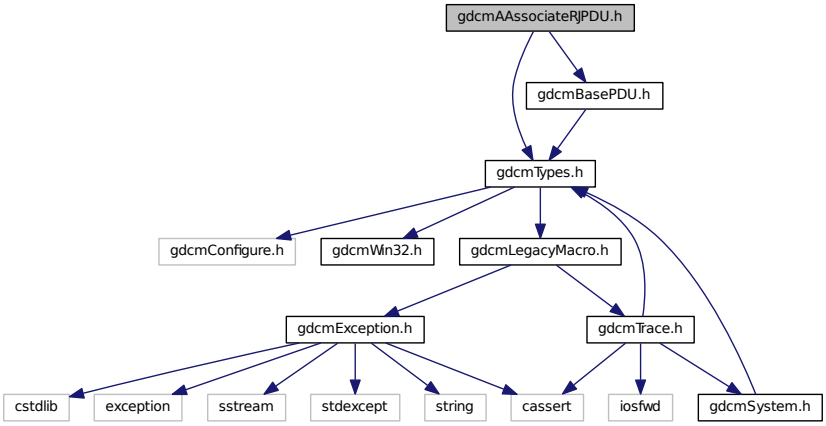
Namespaces

- gdcm
- gdcm::network

11.3 gdcmAAssociateRJPDU.h File Reference

```
#include "gdcmTypes.h"
#include "gdcmBasePDU.h"
```

Include dependency graph for gdcmAAssociateRJPDU.h:



Classes

- class gdcm::network::AAssociateRJPDU
  AAssociateRJPDU.

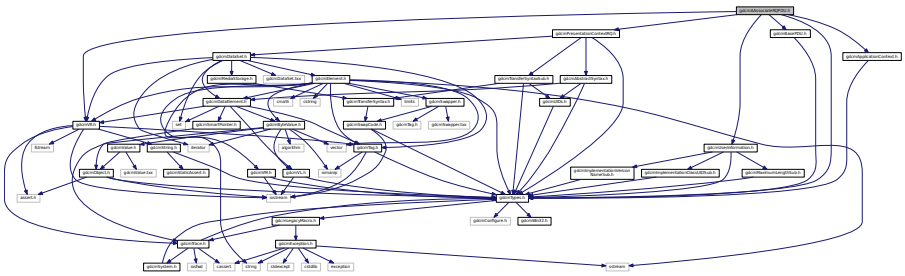
Namespaces

- gdcm
- gdcm::network

11.4 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.

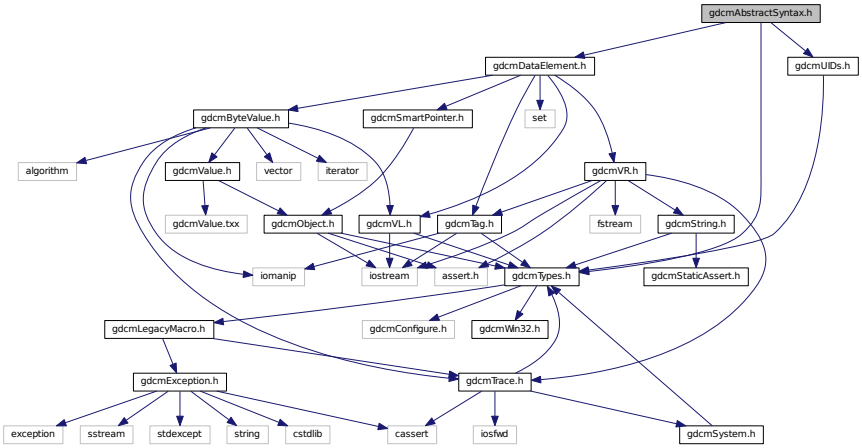
Namespaces

- gdcm
- gdcm::network

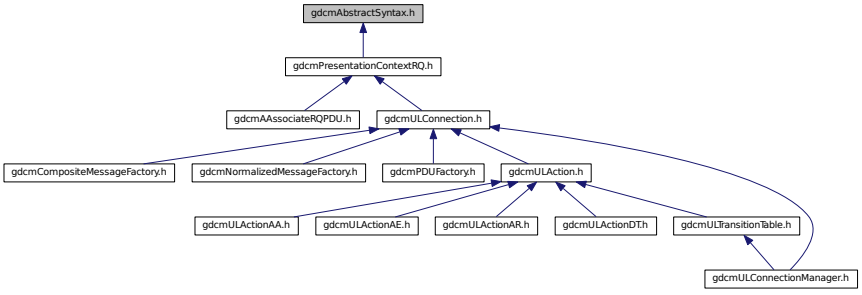
11.5 gdcmAbstractSyntax.h File Reference

```
#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.

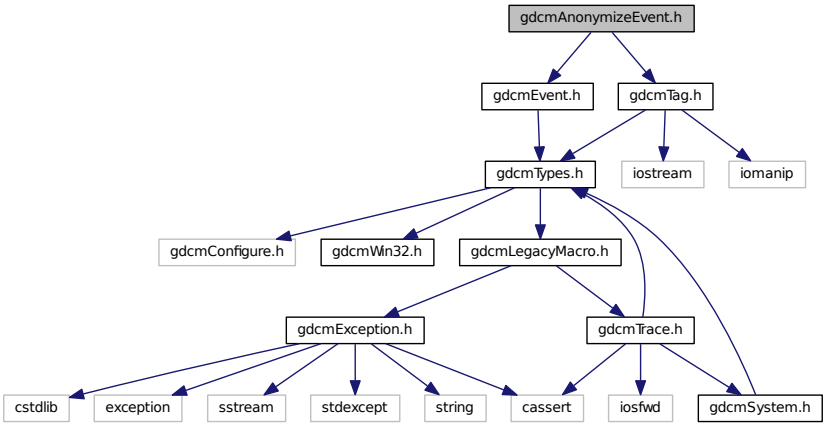
Namespaces

- gdcm
- gdcm::network

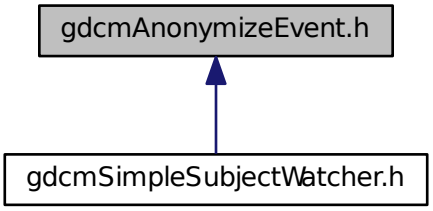
11.6 gdcmAnonymizeEvent.h File Reference

```
#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.

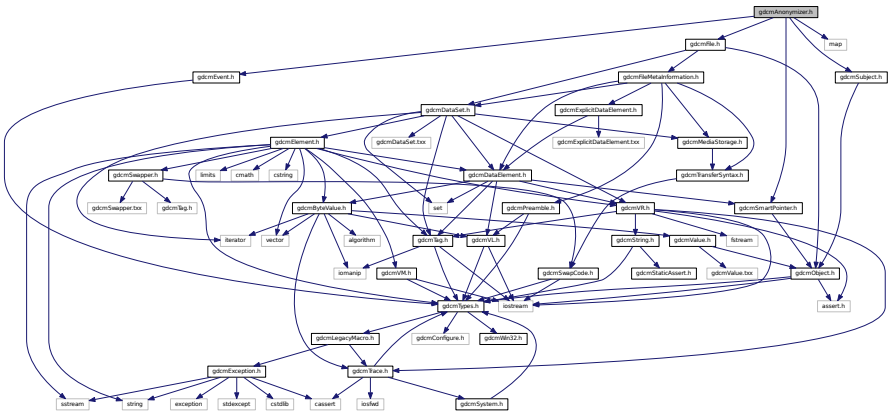
Namespaces

- gdcm

11.7 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.

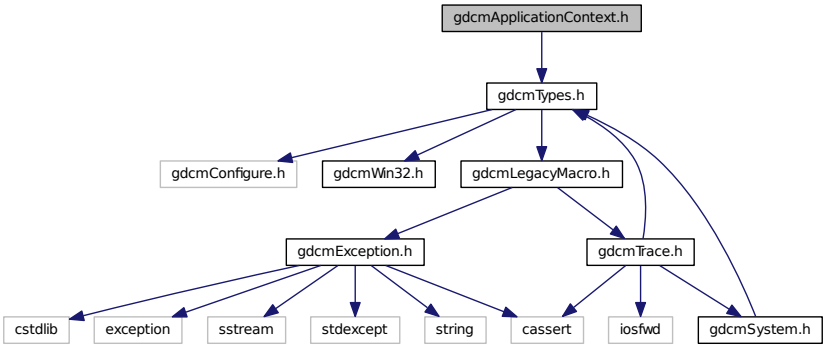
Namespaces

- gdcm

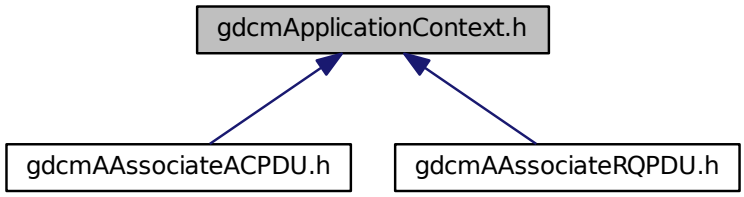
11.8 gdcmApplicationContext.h File Reference

```
#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.

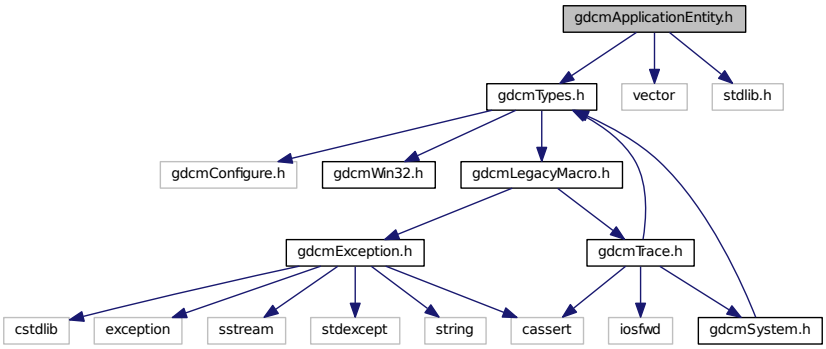
Namespaces

- gdcm
- gdcm::network

11.9 gdcmApplicationEntity.h File Reference

```
#include "gdcmTypes.h"
#include <vector>
#include <stdlib.h>
```

Include dependency graph for gdcmApplicationEntity.h:



Classes

- class gdcm::ApplicationEntity

  ApplicationEntity.

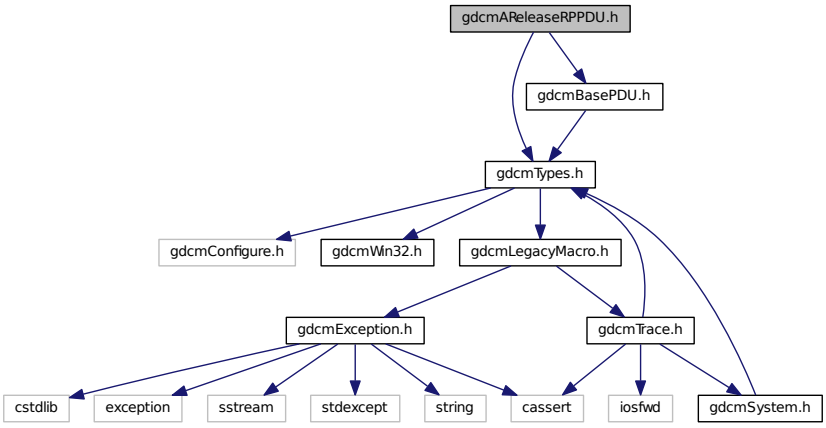
Namespaces

- gdcm

11.10 gdcmAReleaseRPPDU.h File Reference

```
#include "gdcmTypes.h"
#include "gdcmBasePDU.h"
```

Include dependency graph for gdcmAReleaseRPPDU.h:



Classes

- class gdcm::network::AReleaseRPPDU

  AReleaseRPPDU.

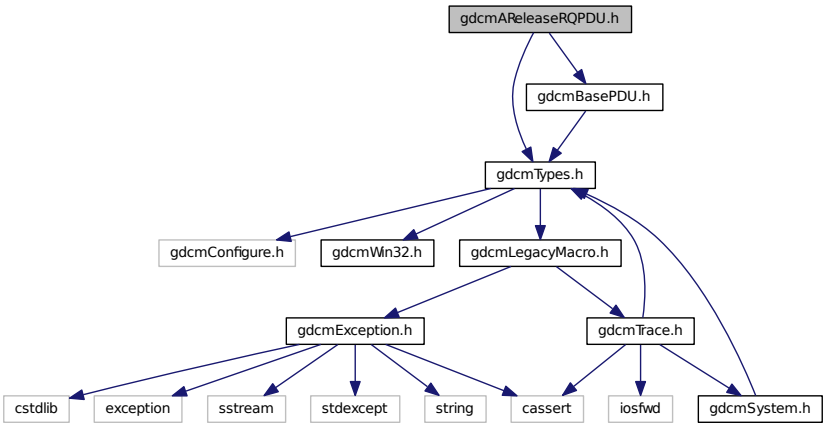
Namespaces

- gdcm
- gdcm::network

11.11 gdcmAReleaseRQPDU.h File Reference

```
#include "gdcmTypes.h"
#include "gdcmBasePDU.h"
```

Include dependency graph for gdcmAReleaseRQPDU.h:



Classes

- class gdcm::network::AReleaseRQPDU

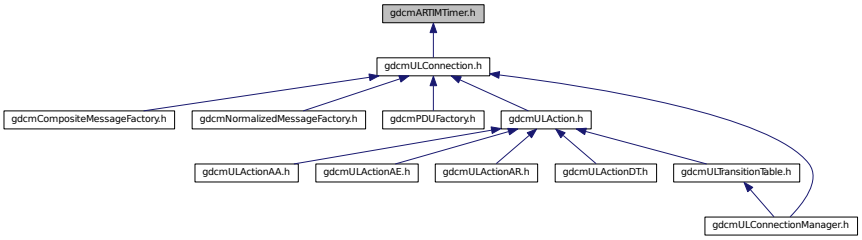
  AReleaseRQPDU.

Namespaces

- gdcm
- gdcm::network

11.12 gdcmARTIMTimer.h File Reference

This graph shows which files directly or indirectly include this file:



Classes

- class gdcm::network::ARTIMTimer

  ARTIMTimer.

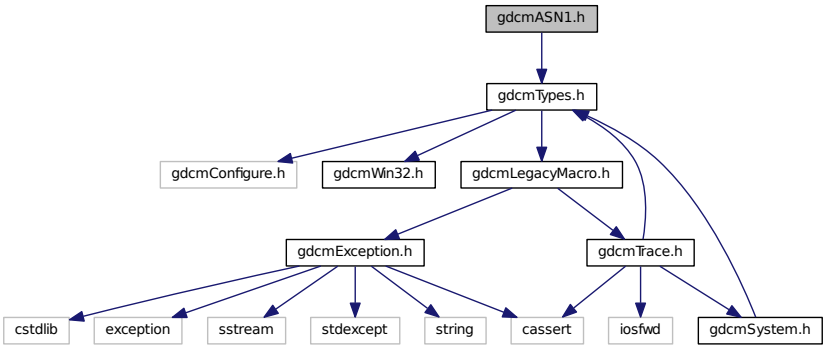
Namespaces

- gdcm
- gdcm::network

11.13 gdcmASN1.h File Reference

```
#include "gdcmTypes.h"
```

Include dependency graph for gdcmASN1.h:



Classes

- class gdcm::ASN1

  Class for ASN1.

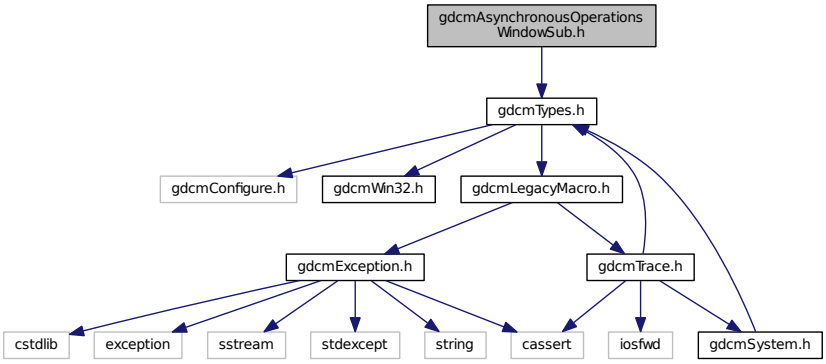
Namespaces

- gdcm

11.14 gdcmAsynchronousOperationsWindowSub.h File Reference

```
#include "gdcmTypes.h"
```

Include dependency graph for gdcmAsynchronousOperationsWindowSub.h:



Classes

- class gdcm::network::AsynchronousOperationsWindowSub

  AsynchronousOperationsWindowSub.

Namespaces

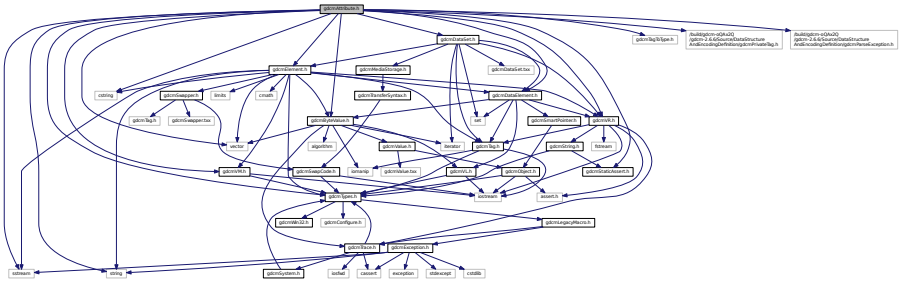
- gdcm
- gdcm::network

11.15 gdcmAttribute.h File Reference

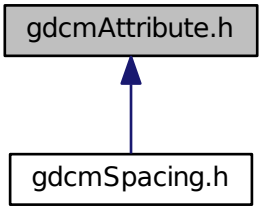
```
#include "gdcmTypes.h"
#include "gdcmVR.h"
#include "gdcmTagToType.h"
#include "gdcmVM.h"
#include "gdcmElement.h"
#include "gdcmDataElement.h"
#include "gdcmDataSet.h"
#include "gdcmStaticAssert.h"
#include <string>
#include <vector>
```

```
#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 instanciation 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< T >
- class gdcm::VRVLSize< 0 >
- class gdcm::VRVLSize< 1 >

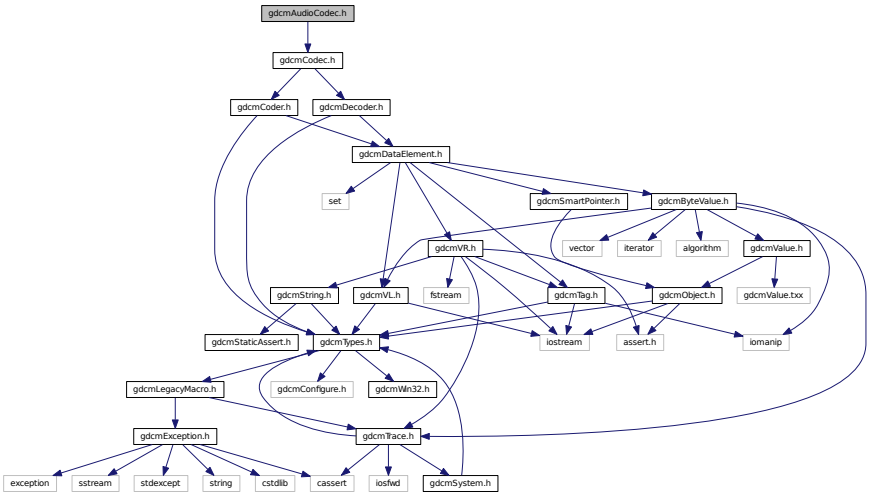
Namespaces

- gdcm

11.16 gdcmAudioCodec.h File Reference

```
#include "gdcmCodec.h"
```

Include dependency graph for gdcmAudioCodec.h:



Classes

- class gdcm::AudioCodec

  AudioCodec.

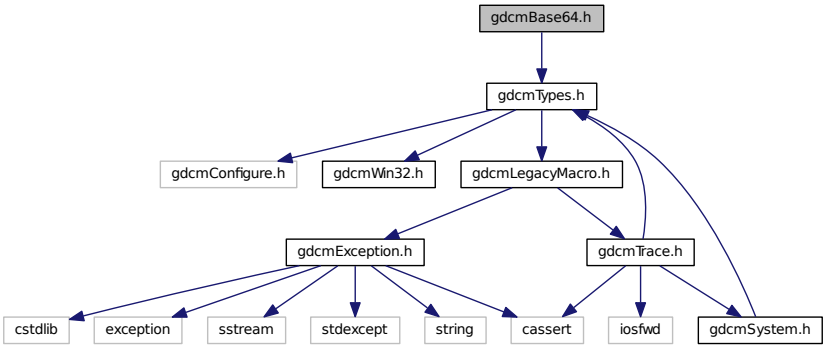
Namespaces

- gdcm

11.17 gdcmBase64.h File Reference

```
#include "gdcmTypes.h"
```

Include dependency graph for gdcmBase64.h:



Classes

- class gdcm::Base64

  Class for Base64.

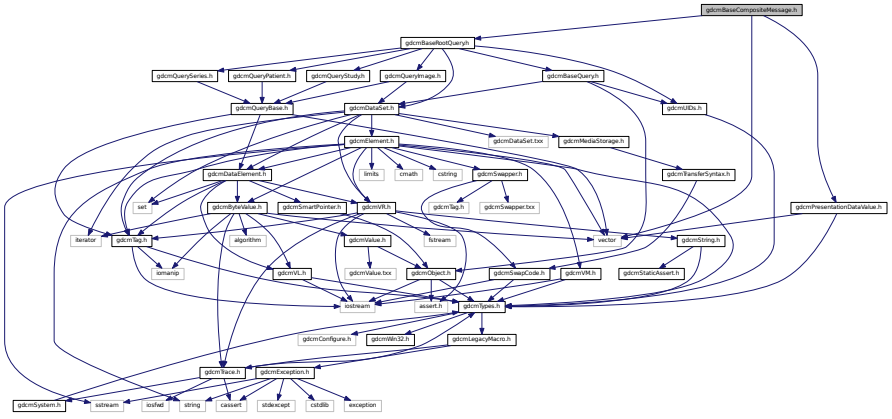
Namespaces

- gdcm

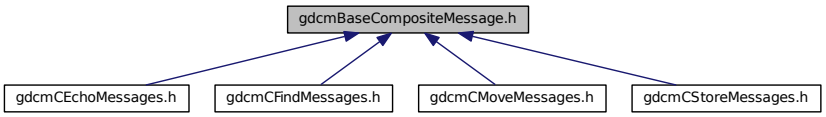
11.18 gdcmBaseCompositeMessage.h File Reference

```
#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.

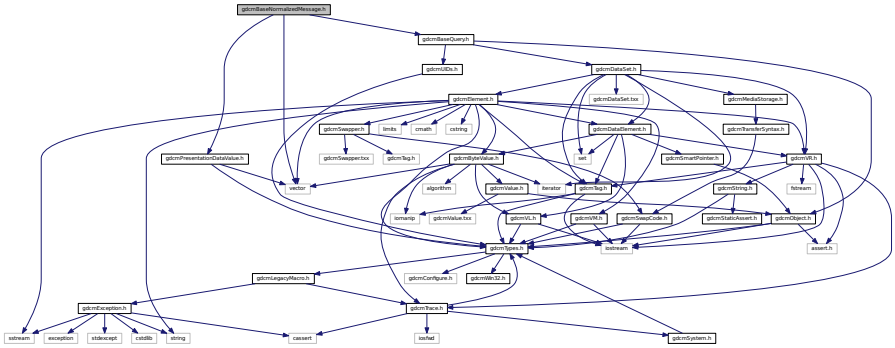
Namespaces

- gdcm
- gdcm::network

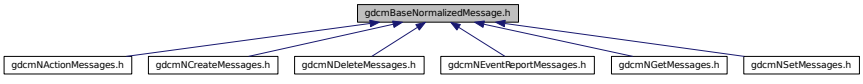
11.19 gdcmBaseNormalizedMessage.h File Reference

```
#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.

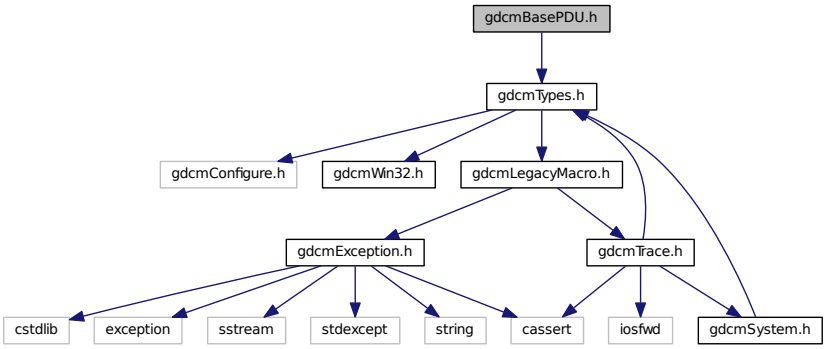
Namespaces

- gdcm
- gdcm::network

11.20 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.

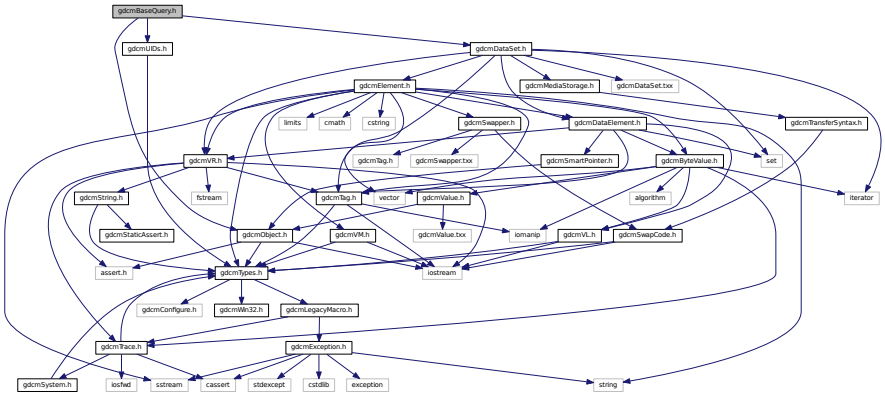
Namespaces

- gdcm
- gdcm::network

11.21 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

- gdcm

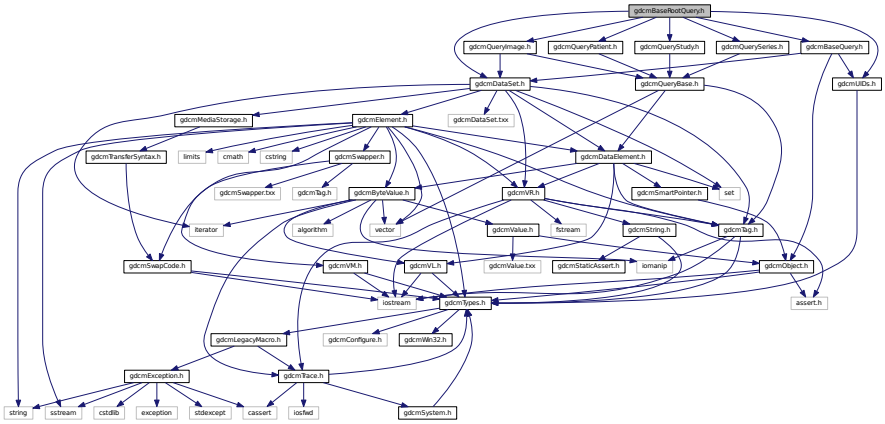
Enumerations

- enum gdcm::ENQueryType {
  gdcm::eCreateMMPS = 0,
  gdcm::eSetMMPS }

11.22 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

- gdcm

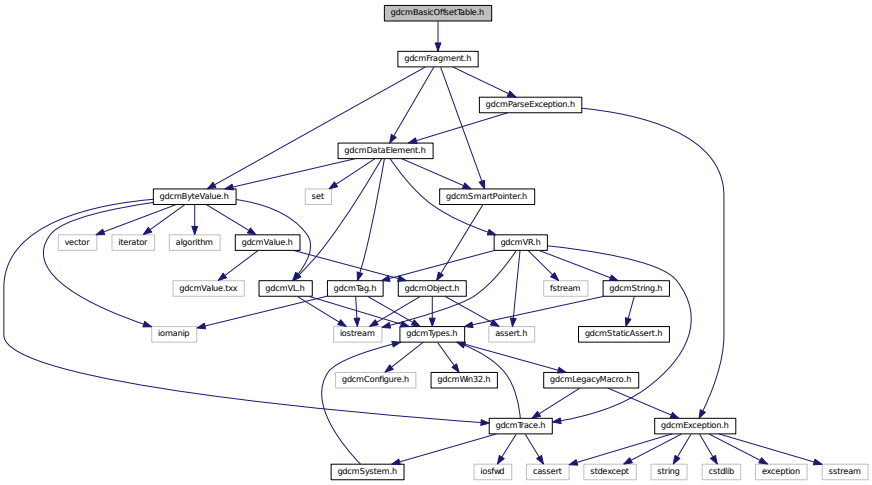
Enumerations

- 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 }

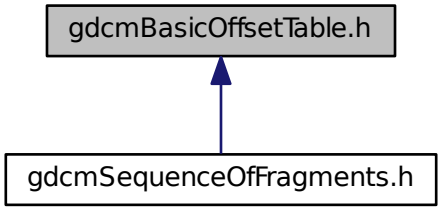
11.23 gdcmBasicOffsetTable.h File Reference

```
#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

- gdcm

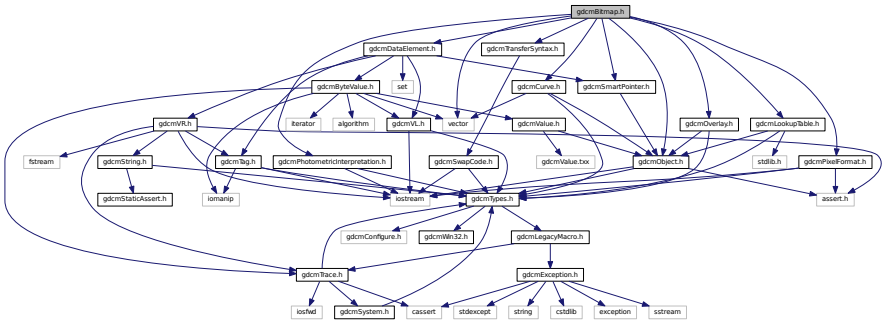
Functions

- std::ostream & gdcm::operator<< (std::ostream &os, const BasicOffsetTable &val)

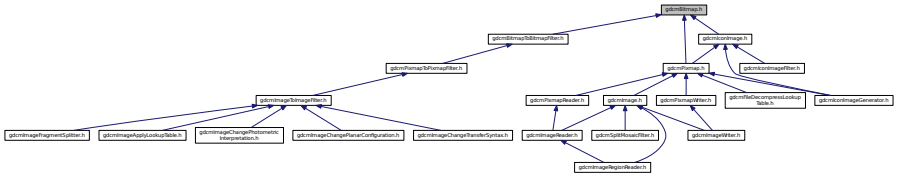
11.24 gdcmBitmap.h File Reference

```
#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.

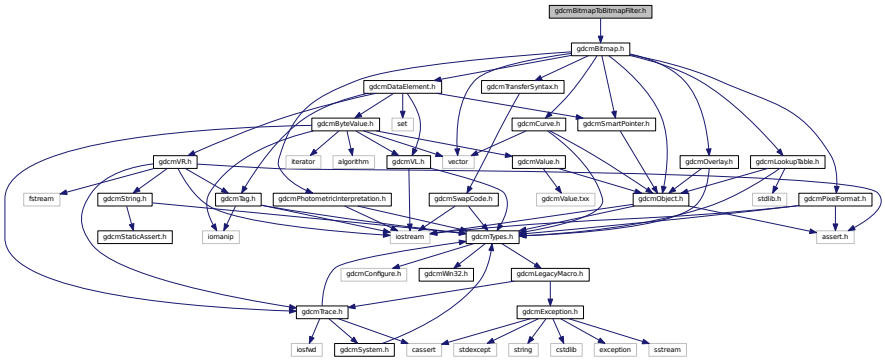
Namespaces

- gdcm

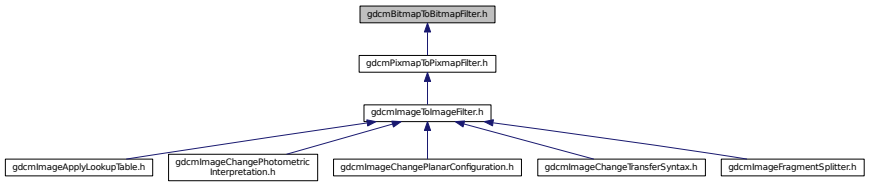
11.25 gdcmBitmapToBitmapFilter.h File Reference

```
#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.

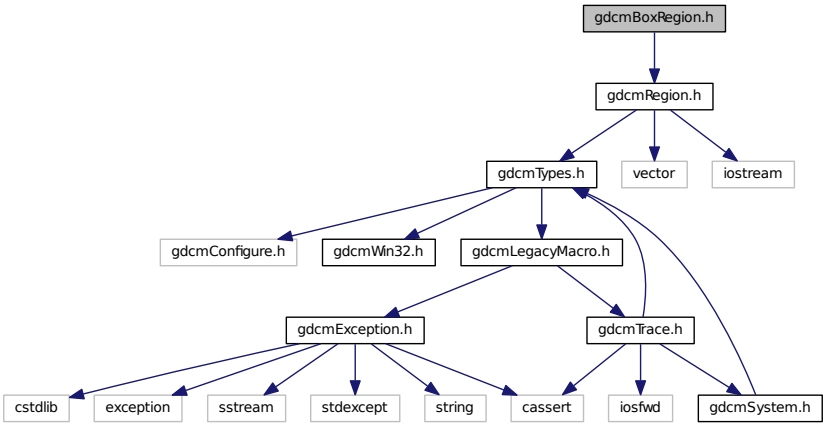
Namespaces

- gdcm

11.26 gdcmBoxRegion.h File Reference

```
#include "gdcmRegion.h"
```

Include dependency graph for gdcmBoxRegion.h:



Classes

- class gdcm::BoxRegion

  Class for manipulation box region.

Namespaces

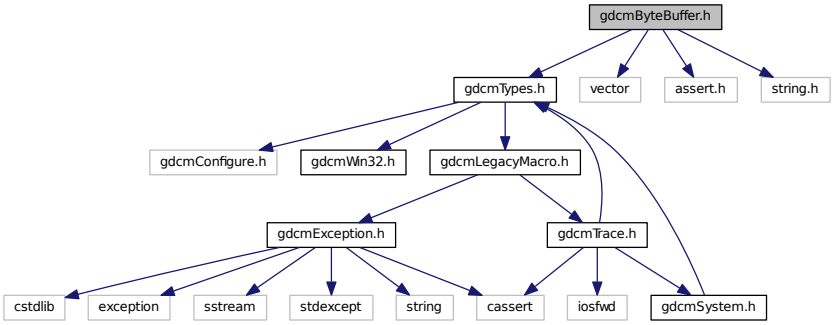
- gdcm

11.27 gdcmByteBuffer.h File Reference

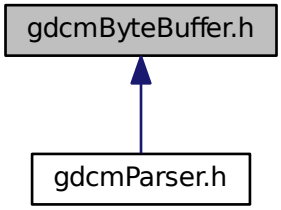
```
#include "gdcmTypes.h"
#include <vector>
#include <assert.h>
```

```
#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.

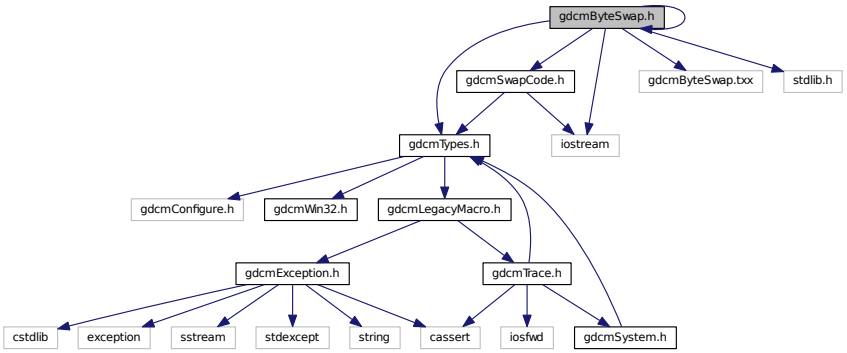
Namespaces

- gdcm

11.28 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.

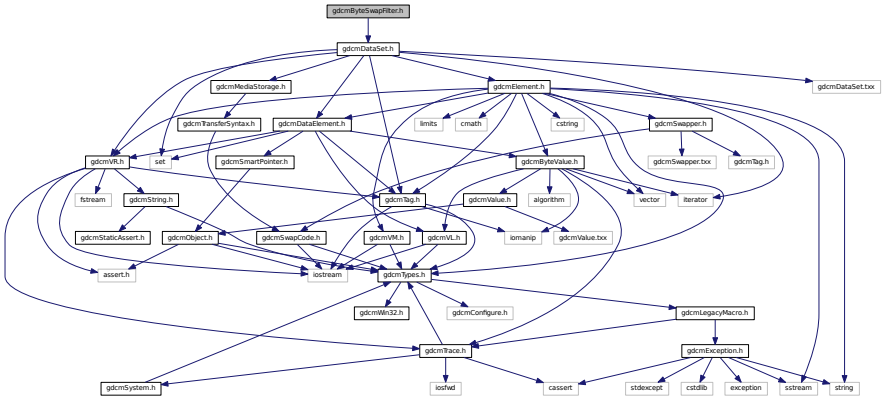
Namespaces

- gdcm

11.29 gdcmByteSwapFilter.h File Reference

```
#include "gdcmDataSet.h"
```

Include dependency graph for gdcmByteSwapFilter.h:



Classes

- class gdcm::ByteSwapFilter

  ByteSwapFilter.

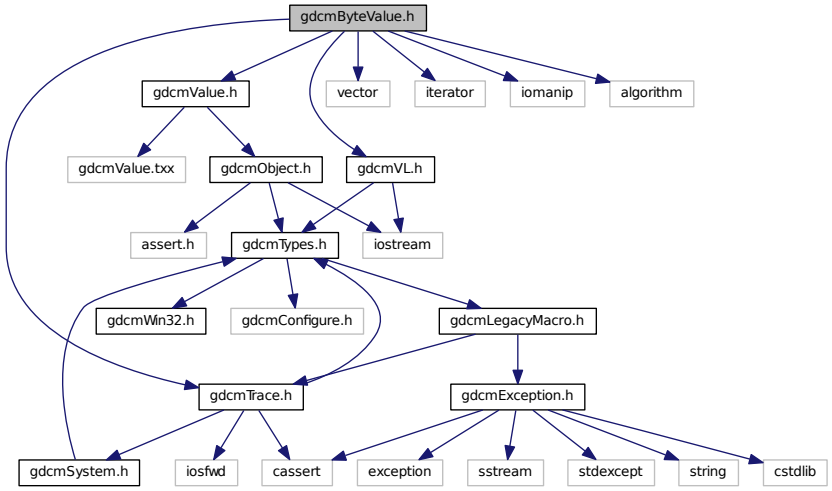
Namespaces

- gdcm

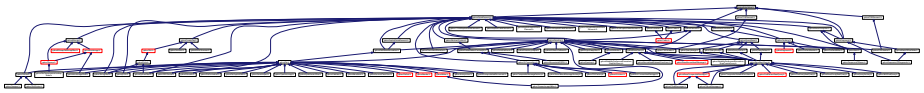
11.30 gdcmByteValue.h File Reference

```
#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)

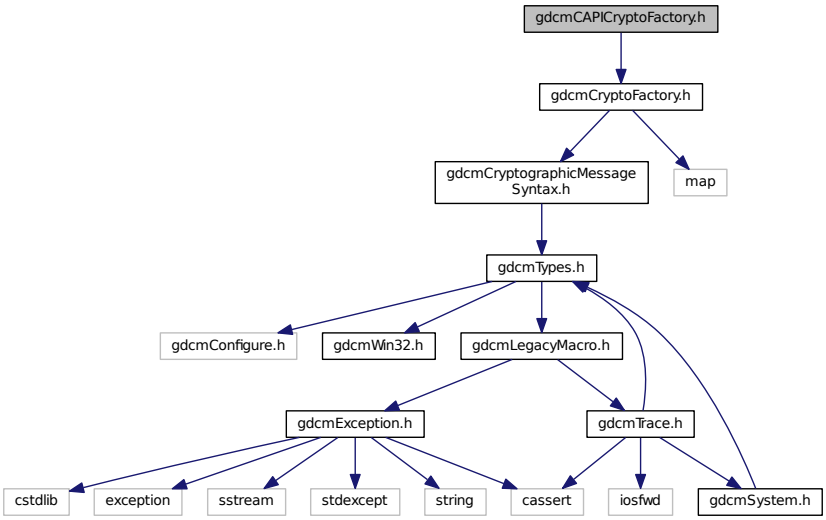
Namespaces

- gdcm

11.31 gdcmCAPICryptoFactory.h File Reference

```
#include "gdcmCryptoFactory.h"
```

Include dependency graph for gdcmCAPICryptoFactory.h:



Classes

- class gdcm::CAPICryptoFactory

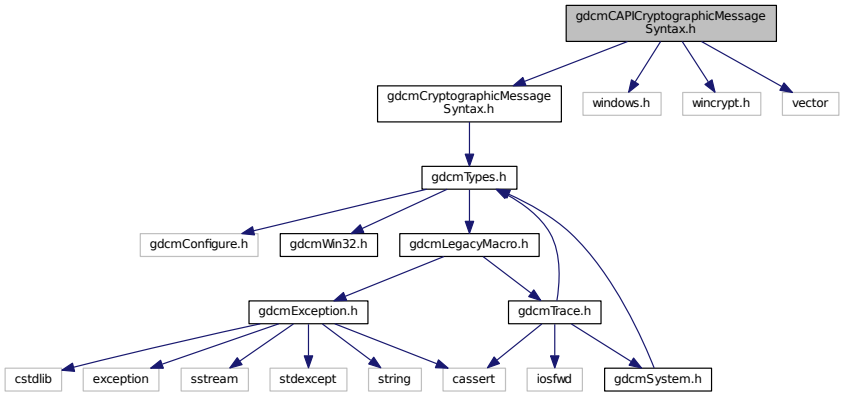
Namespaces

- gdcm

11.32 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

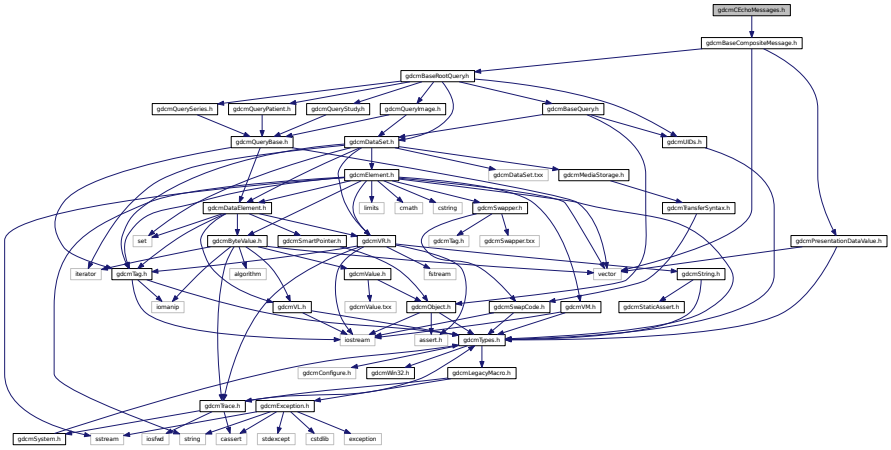
Namespaces

- gdcm

11.33 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.

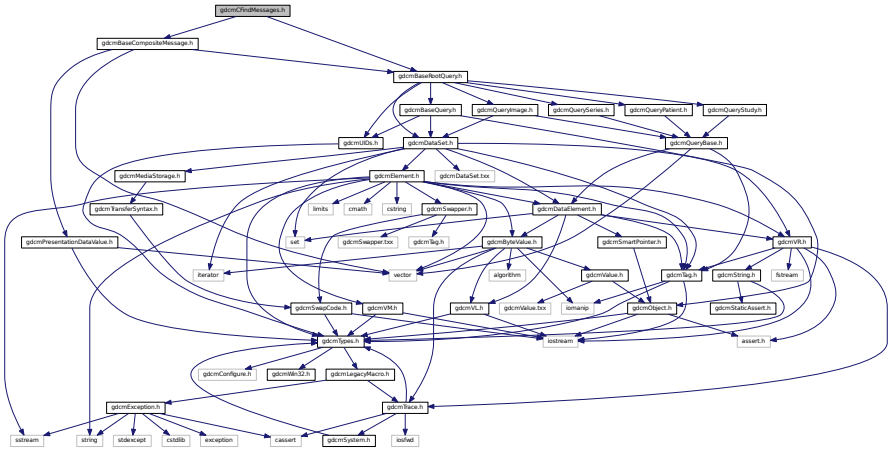
Namespaces

- gdcm
- gdcm::network

11.34 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.

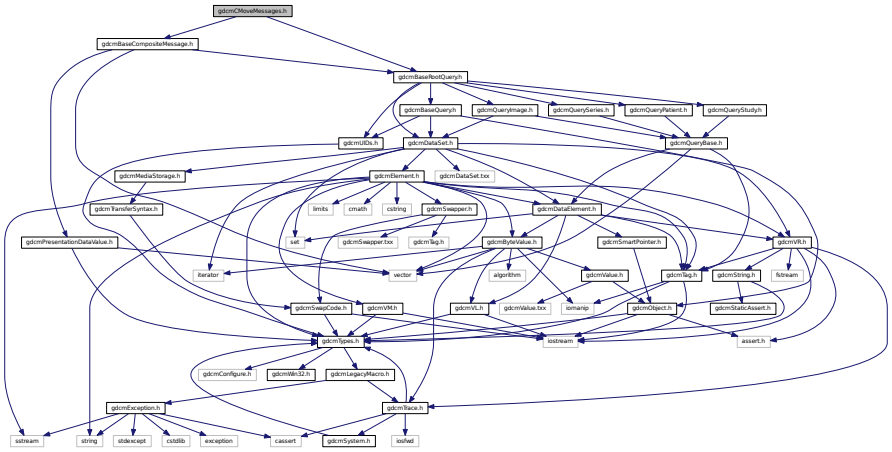
Namespaces

- gdcm
- gdcm::network

11.35 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.

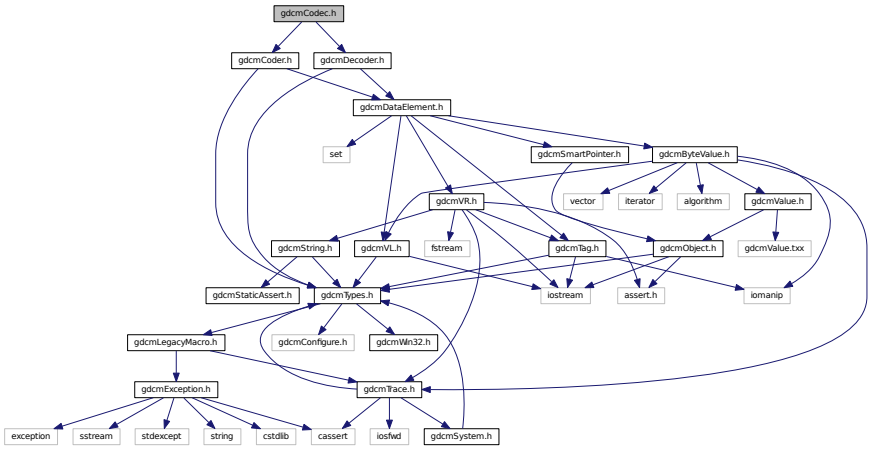
Namespaces

- gdcm
- gdcm::network

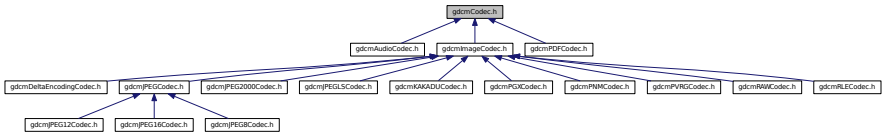
11.36 gdcmCodec.h File Reference

```
#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.

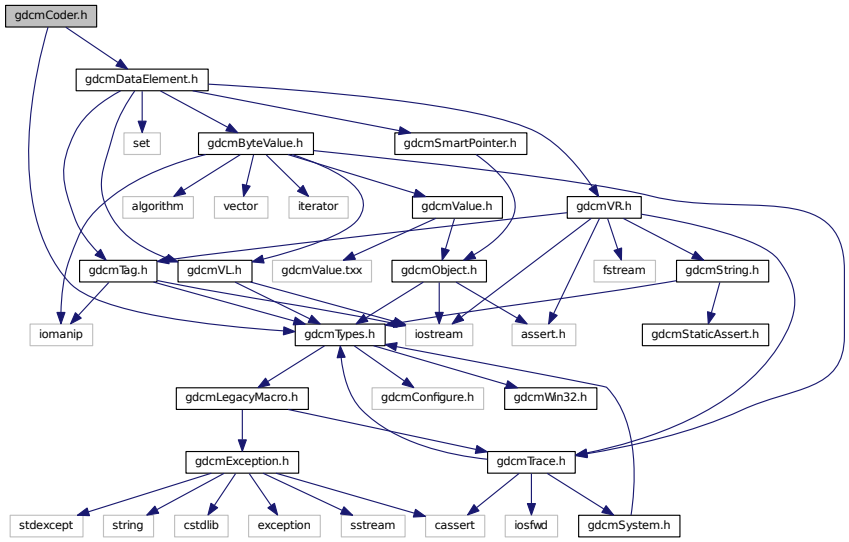
Namespaces

- gdcm

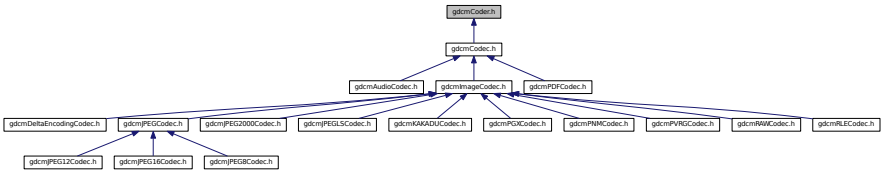
11.37 gdcmCoder.h File Reference

```
#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.

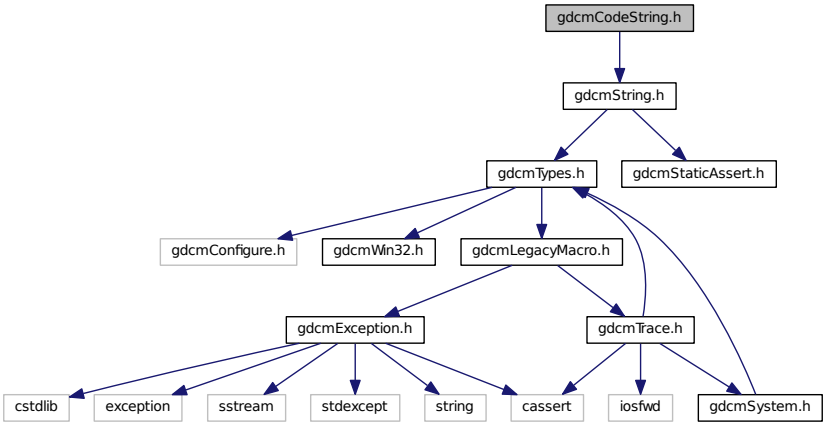
Namespaces

- gdcm

11.38 gdcmCodeString.h File Reference

```
#include "gdcmString.h"
```

Include dependency graph for gdcmCodeString.h:



Classes

- class gdcm::CodeString

  CodeString.

Namespaces

- gdcm

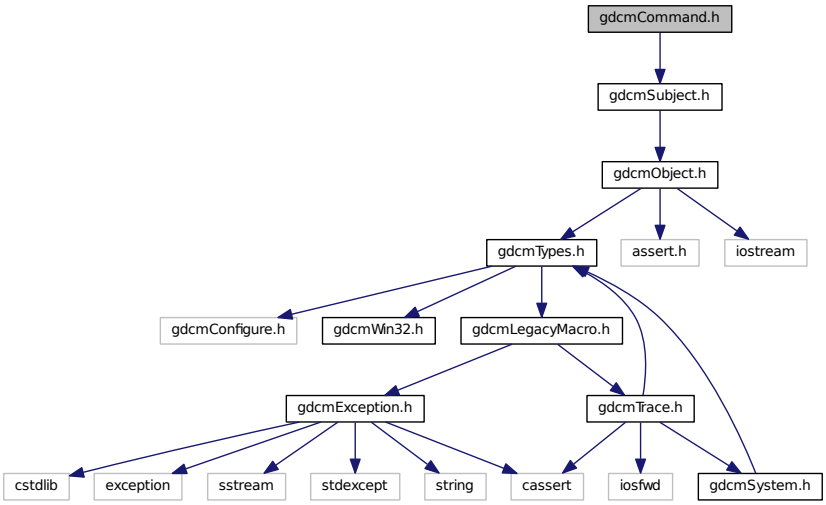
Functions

- 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)

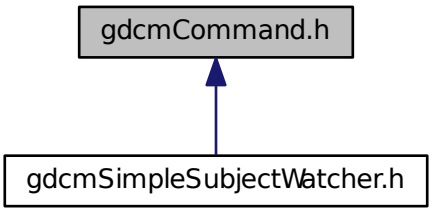
11.39 gdcmCommand.h File Reference

```
#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.

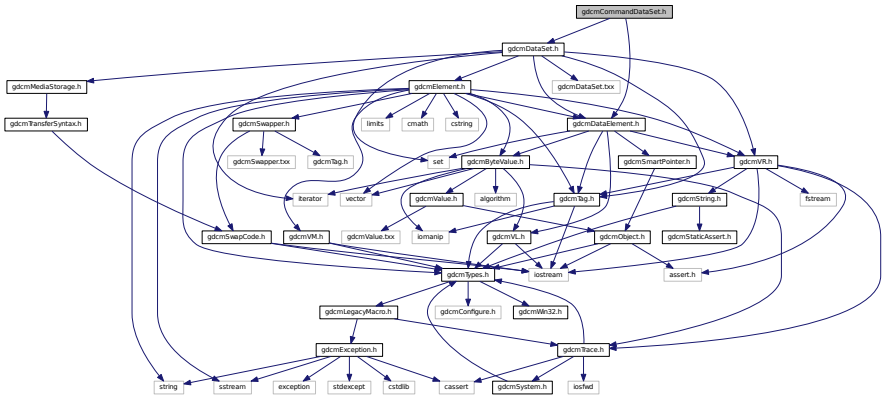
Namespaces

- gdcm

11.40 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

- gdcm

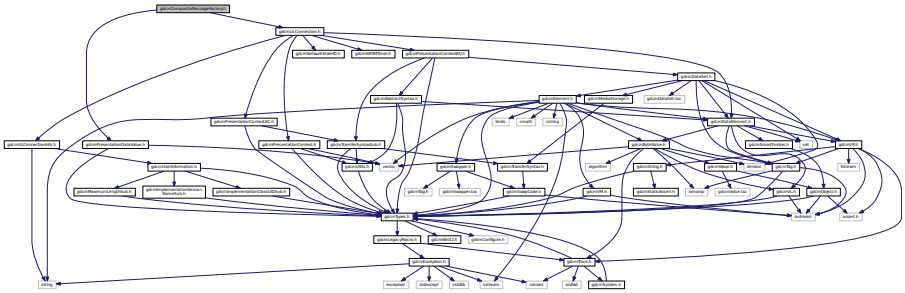
Functions

- std::ostream & gdcm::operator<< (std::ostream &os, const CommandDataSet &val)

11.41 gdcmCompositeMessageFactory.h File Reference

```
#include "gdcmPresentationDataValue.h"
#include "gdcmULConnection.h"
```

Include dependency graph for gdcmCompositeMessageFactory.h:



Classes

- class gdcm::network::CompositeMessageFactory

  CompositeMessageFactory.

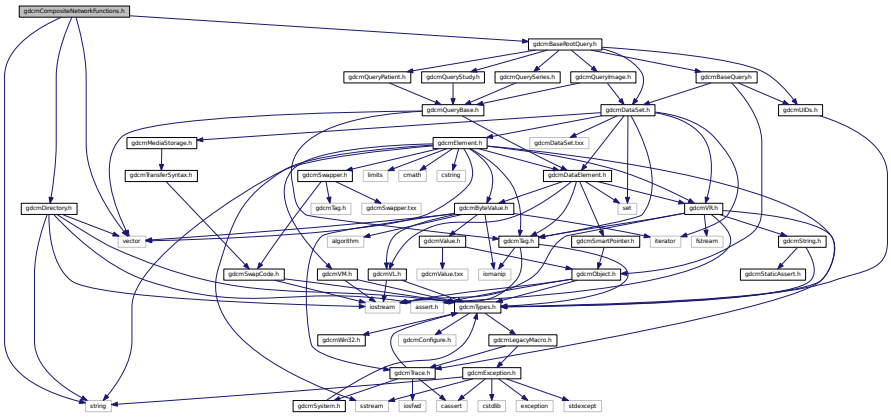
Namespaces

- gdcm
- gdcm::network

11.42 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

- gdcm

11.43 gdcmConstCharWrapper.h File Reference

Classes

- class gdcm::ConstCharWrapper

  Do not use me.

Namespaces

- gdcm

11.44 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.

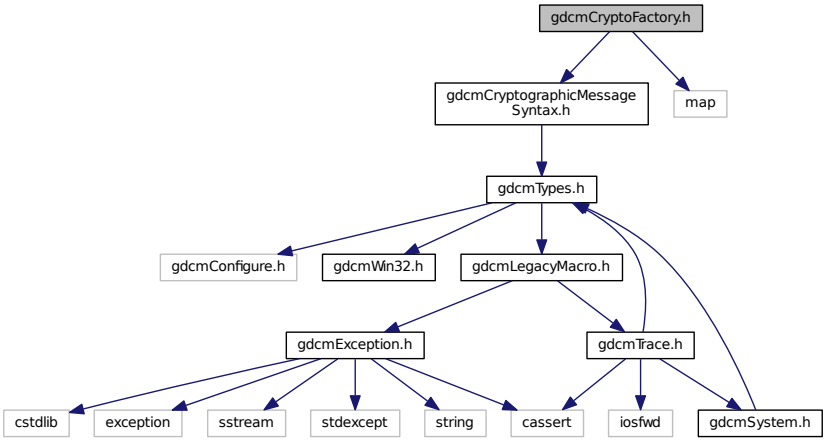
Namespaces

- gdcm

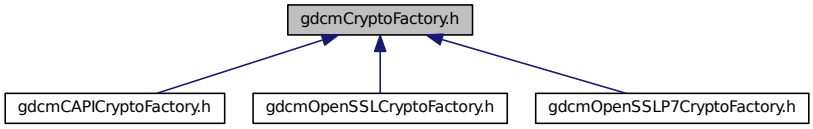
11.45 gdcmCryptoFactory.h File Reference

```
#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.

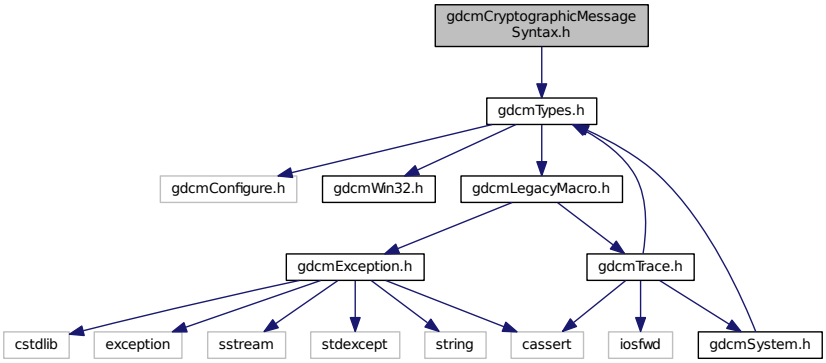
Namespaces

- gdcm

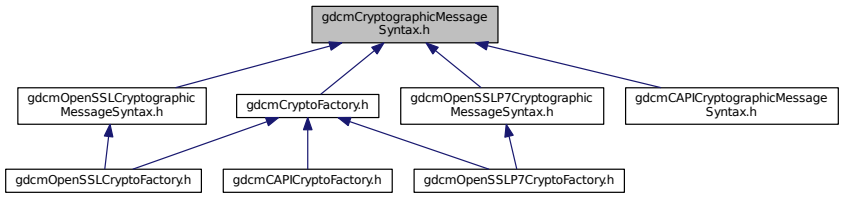
11.46 gdcmCryptographicMessageSyntax.h File Reference

```
#include "gdcmTypes.h"
```

Include dependency graph for gdcmCryptographicMessageSyntax.h:



This graph shows which files directly or indirectly include this file:



Classes

- class gdcm::CryptographicMessageSyntax

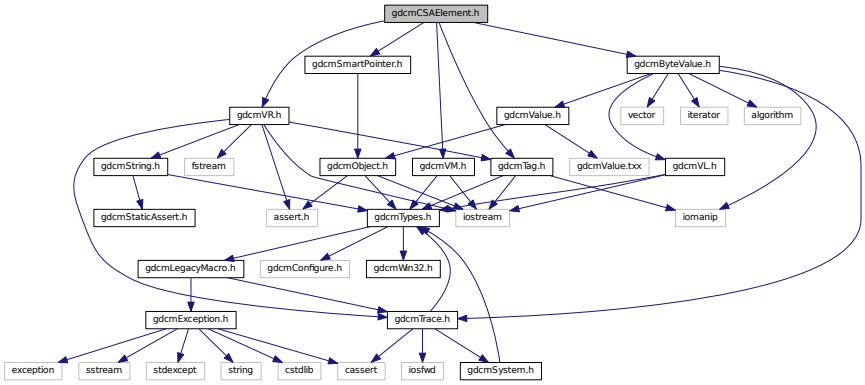
Namespaces

- gdcm

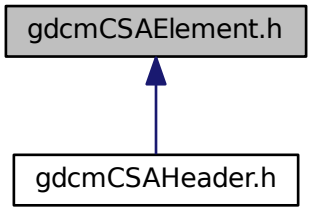
11.47 gdcmCSAElement.h File Reference

```
#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

- gdcm

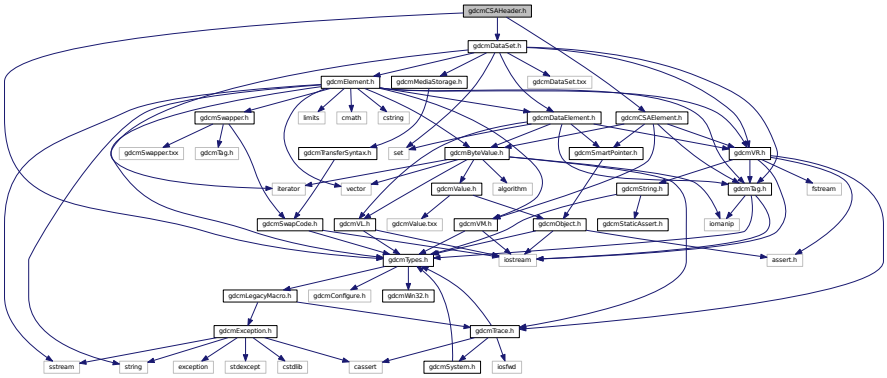
Functions

- std::ostream & gdcm::operator<< (std::ostream &os, const CSAElement &val)

11.48 gdcmCSAHeader.h File Reference

```
#include "gdcmTypes.h"
#include "gdcmDataSet.h"
#include "gdcmCSAElement.h"
```

Include dependency graph for gdcmCSAHeader.h:



Classes

- class gdcm::CSAHeader

  Class for CSAHeader.

Namespaces

- gdcm

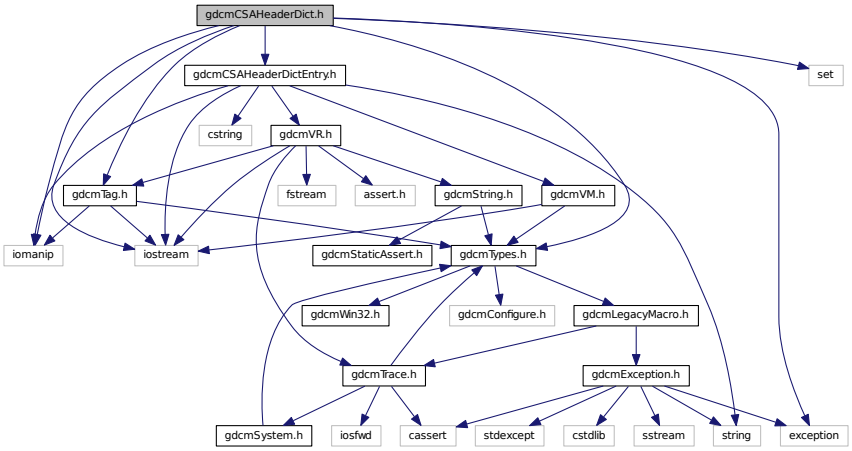
Functions

- std::ostream & gdcm::operator<< (std::ostream &os, const CSAHeader &d)

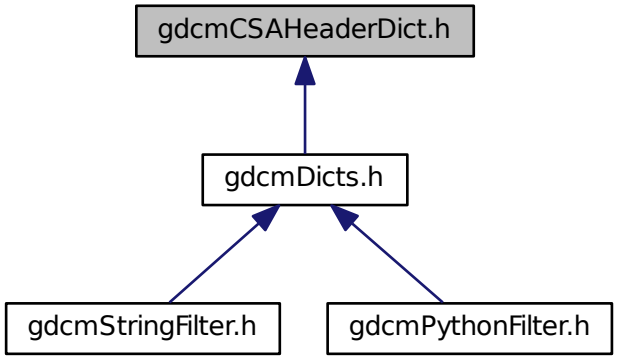
11.49 gdcmCSAHeaderDict.h File Reference

```
#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

- gdcm

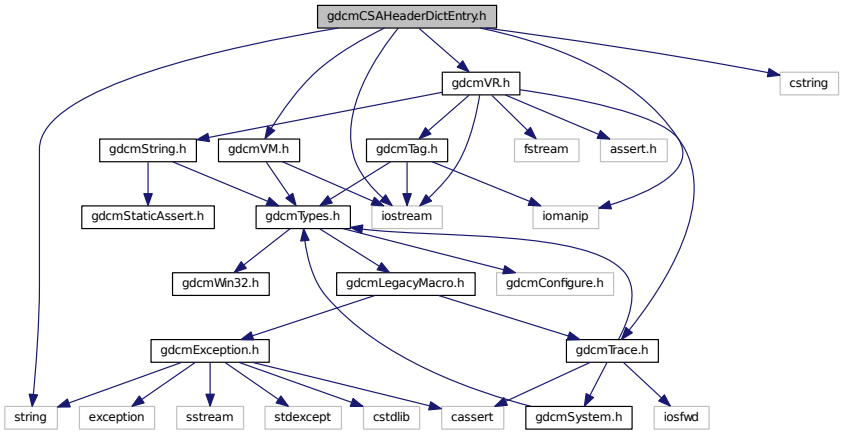
Functions

- std::ostream & gdcm::operator<< (std::ostream &os, const CSAHeaderDict &val)

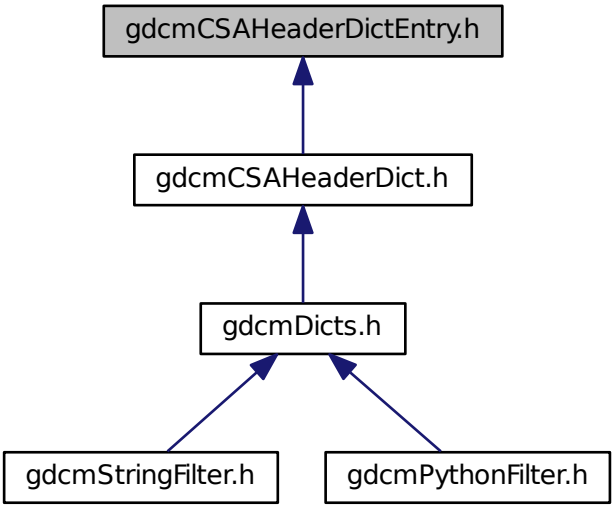
11.50 gdcmCSAHeaderDictEntry.h File Reference

```
#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

- gdcm

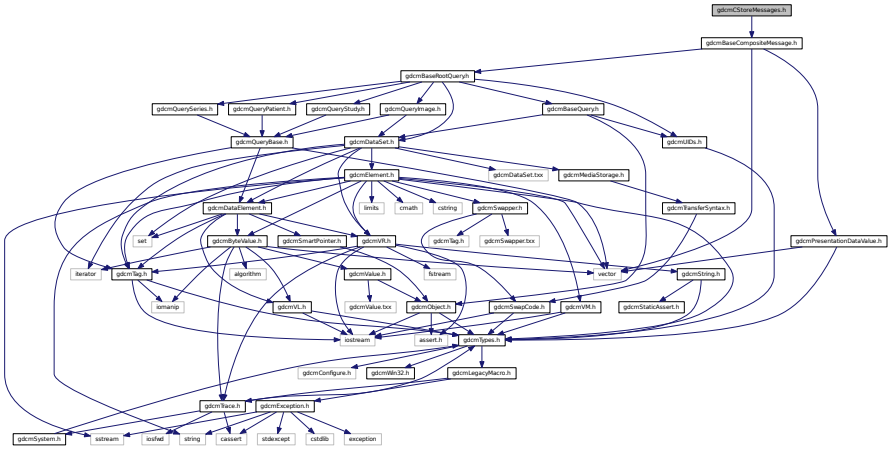
Functions

- std::ostream & gdcm::operator<< (std::ostream &os, const CSAHeaderDictEntry &val)

11.51 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.

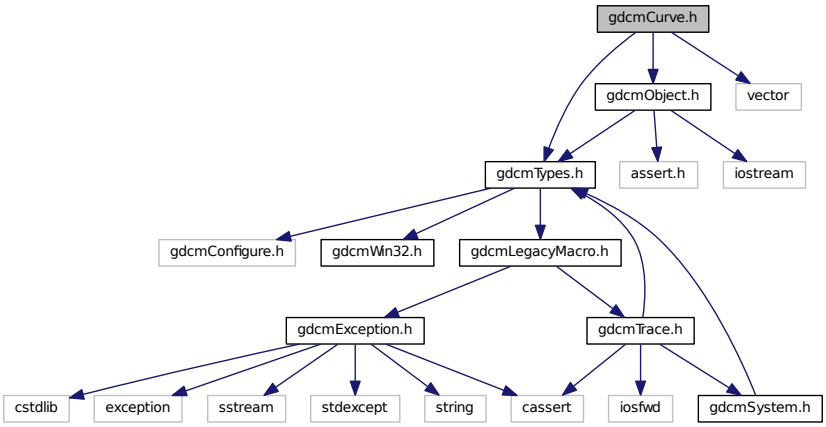
Namespaces

- gdcm
- gdcm::network

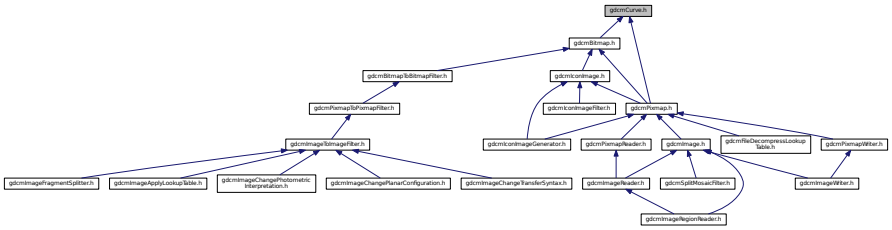
11.52 gdcmCurve.h File Reference

```
#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

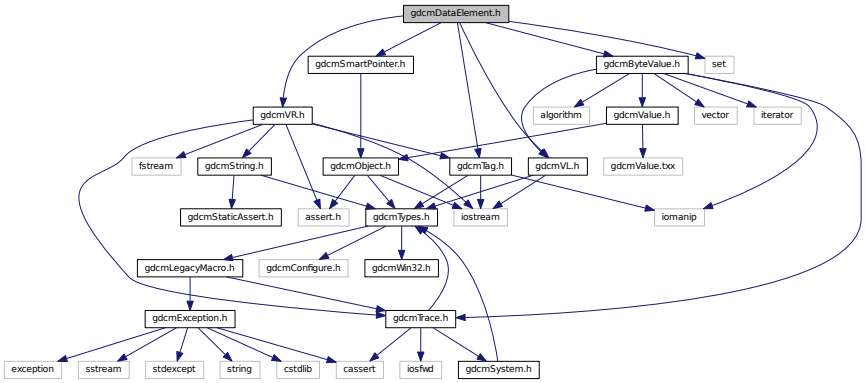
- gdcm

11.53 gdcmDataElement.h File Reference

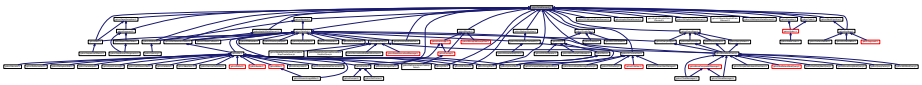
```
#include "gdcmTag.h"
#include "gdcmVL.h"
#include "gdcmVR.h"
```

```
#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

- gdcm

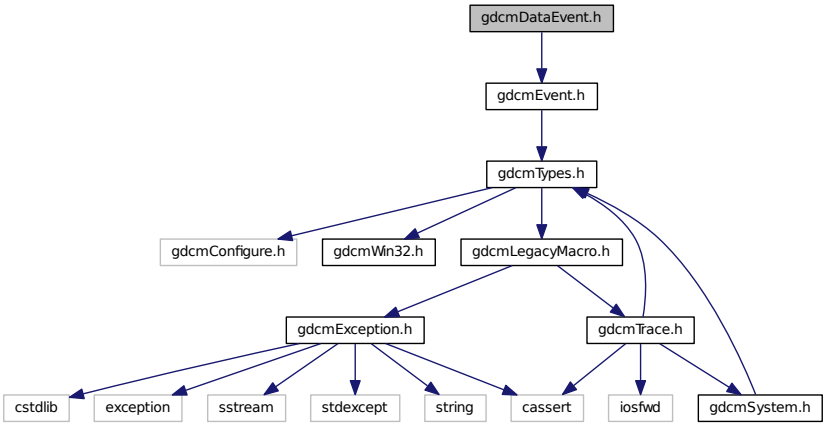
Functions

- bool gdcm::operator!= (const DataElement &lhs, const DataElement &rhs)
- std::ostream & gdcm::operator<< (std::ostream &os, const DataElement &val)

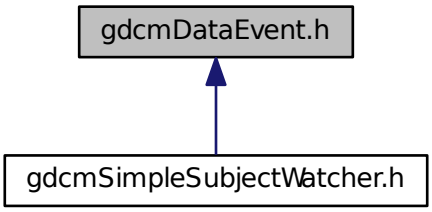
11.54 gdcmDataEvent.h File Reference

```
#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.

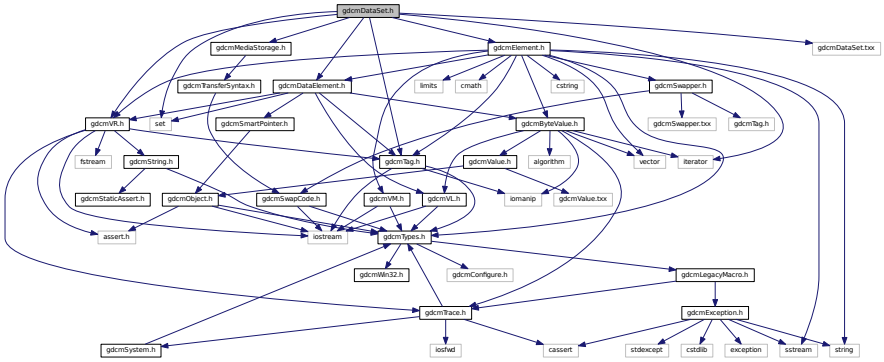
Namespaces

- gdcm

11.55 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

- gdcm

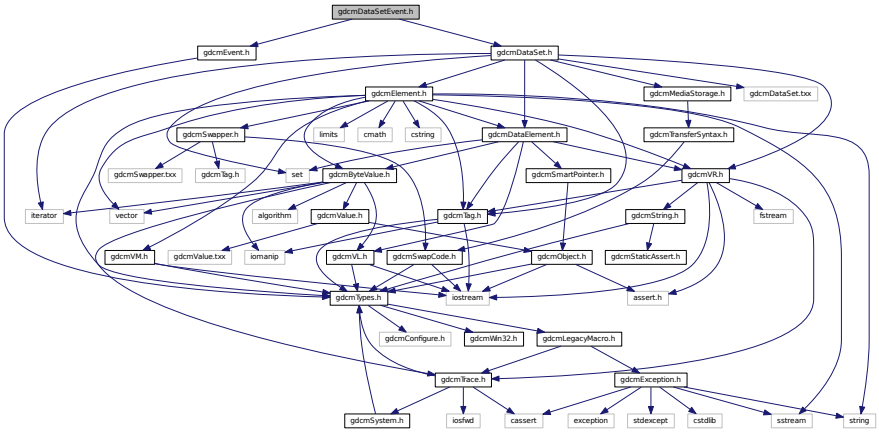
Functions

- std::ostream & gdcm::operator<< (std::ostream &os, const DataSet &val)

11.56 gdcmDataSetEvent.h File Reference

```
#include "gdcmEvent.h"
#include "gdcmDataSet.h"
```

Include dependency graph for gdcmDataSetEvent.h:



Classes

- class gdcm::DataSetEvent

  DataSetEvent.

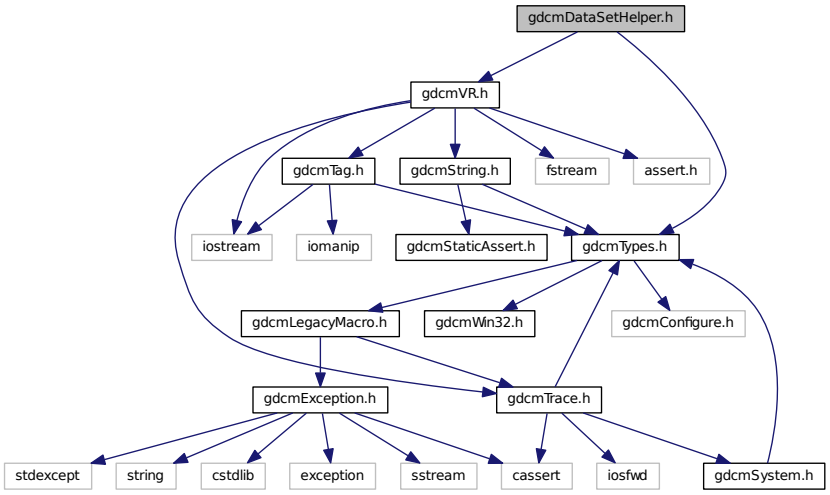
Namespaces

- gdcm

11.57 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)

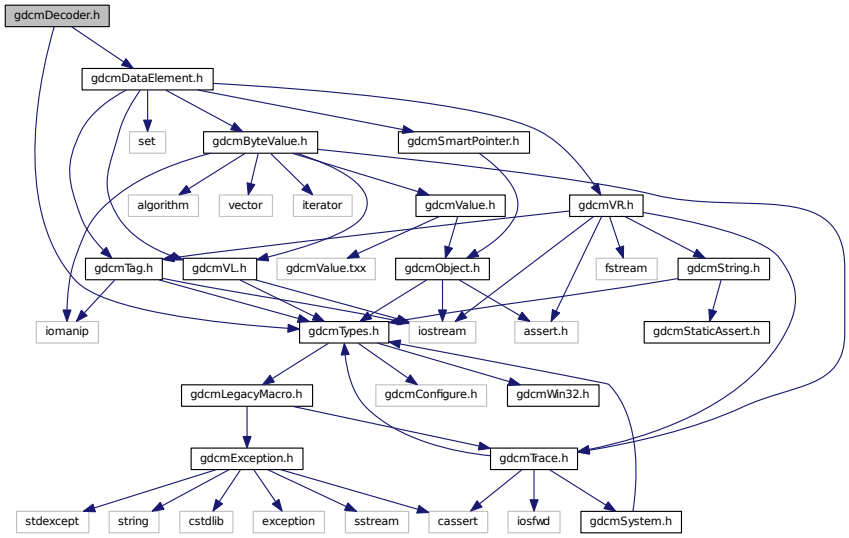
Namespaces

- gdcm

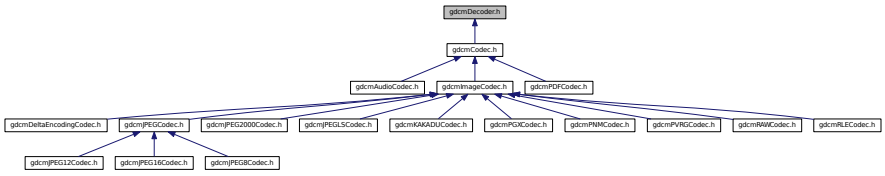
11.58 gdcmDecoder.h File Reference

```
#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.

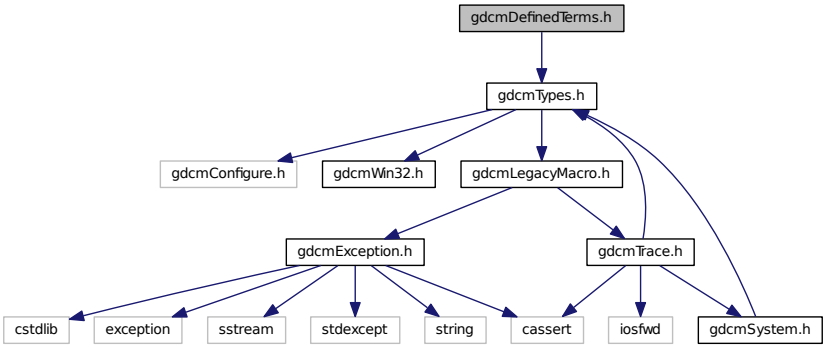
Namespaces

- gdcm

11.59 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.

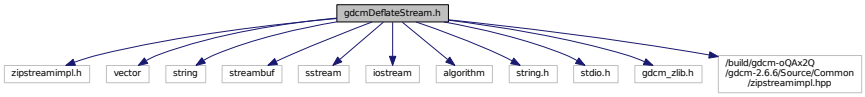
Namespaces

- gdcm

11.60 gdcmDeflateStream.h File Reference

```
#include "zipstreamimpl.h"
```

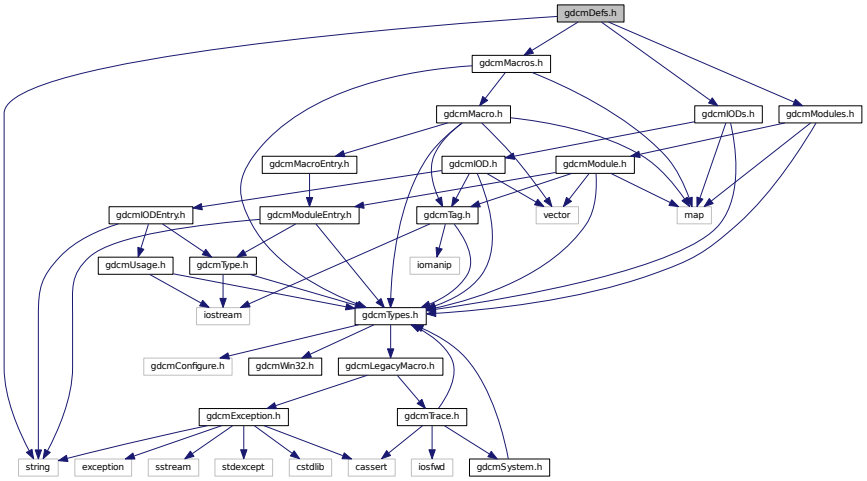
Include dependency graph for gdcmDeflateStream.h:



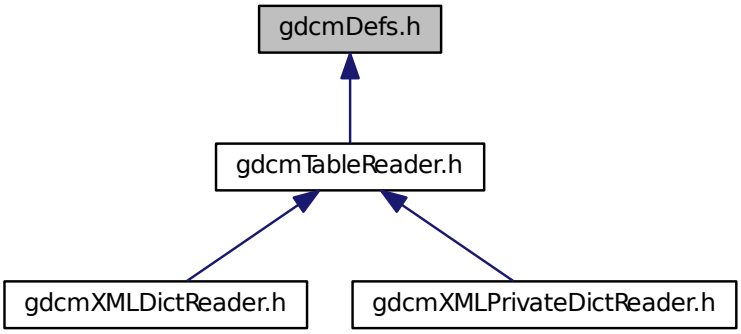
11.61 gdcmDefs.h File Reference

```
#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'.

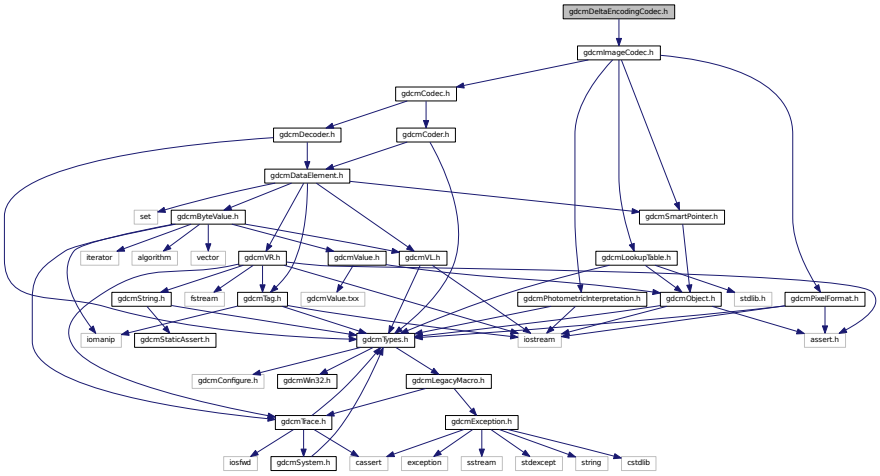
Namespaces

- gdcm

11.62 gdcmDeltaEncodingCodec.h File Reference

```
#include "gdcmImageCodec.h"
```

Include dependency graph for gdcmDeltaEncodingCodec.h:



Classes

- class gdcm::DeltaEncodingCodec

  DeltaEncodingCodec compression used by some private vendor.

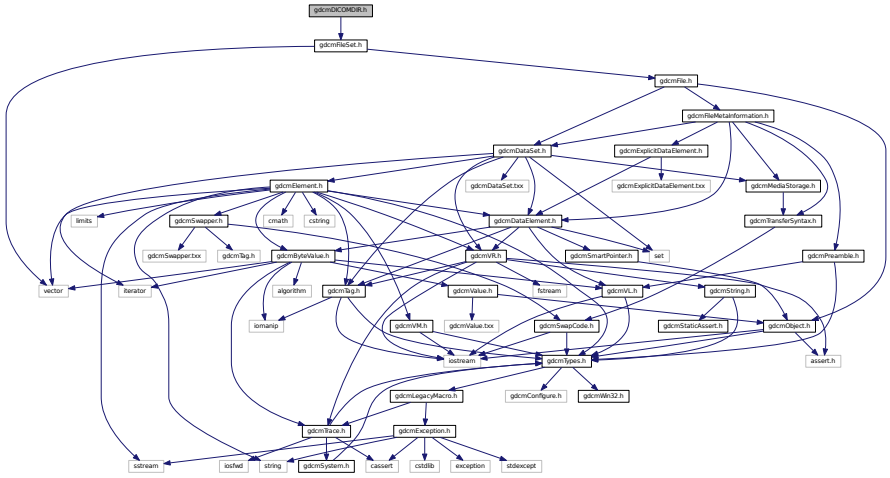
Namespaces

- gdcm

11.63 gdcmDICOMDIR.h File Reference

```
#include "gdcmFileSet.h"
```

Include dependency graph for gdcmDICOMDIR.h:



Classes

- class gdcm::DICOMDIR

  DICOMDIR class.

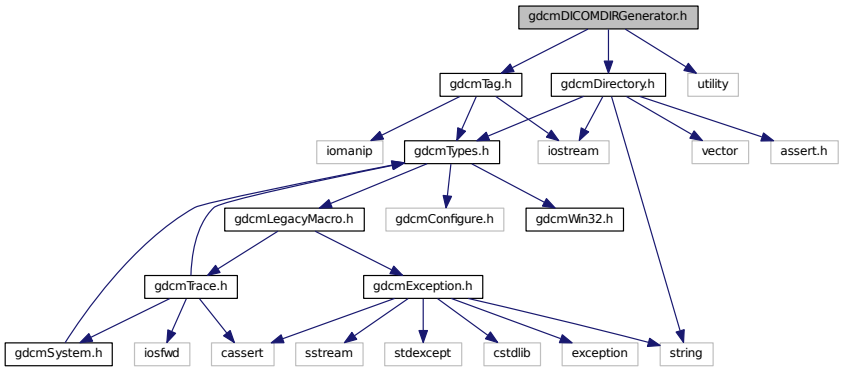
Namespaces

- gdcm

11.64 gdcmDICOMDIRGenerator.h File Reference

```
#include "gdcmDirectory.h"
#include "gdcmTag.h"
#include <utility>
```

Include dependency graph for gdcmDICOMDIRGenerator.h:



Classes

- class gdcm::DICOMDIRGenerator

  DICOMDIRGenerator class.

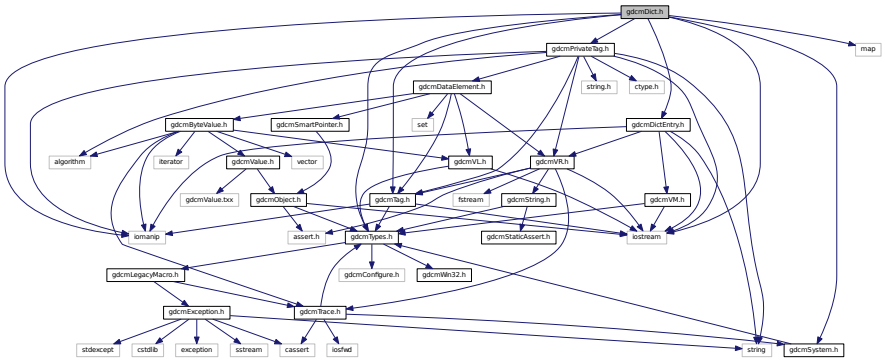
Namespaces

- gdcm

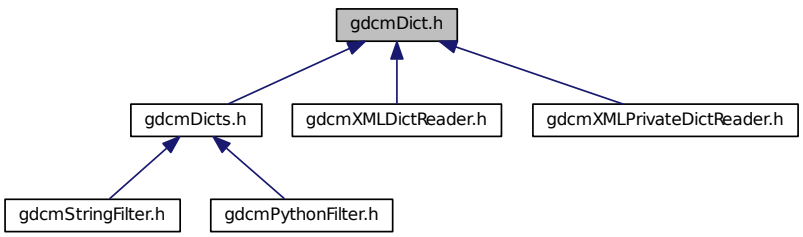
11.65 gdcmDict.h File Reference

```
#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

- gdcm

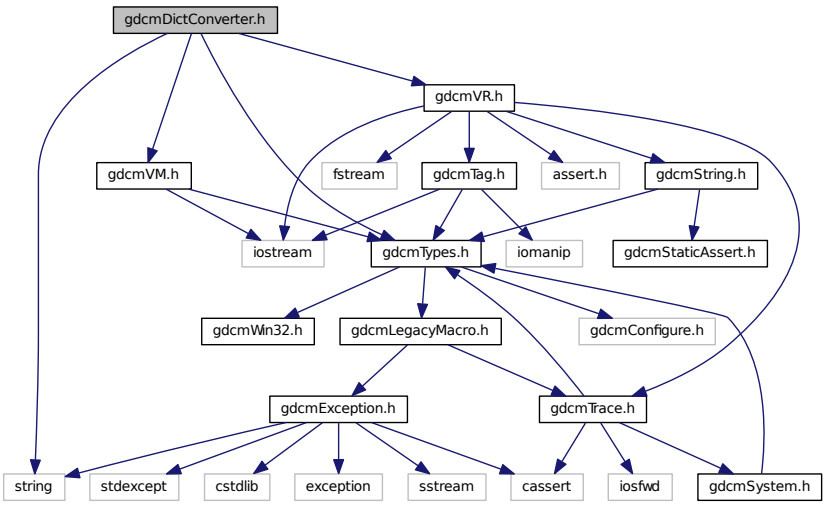
Functions

- std::ostream & gdcm::operator<< (std::ostream &os, const Dict &val)
- std::ostream & gdcm::operator<< (std::ostream &os, const PrivateDict &val)

11.66 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:

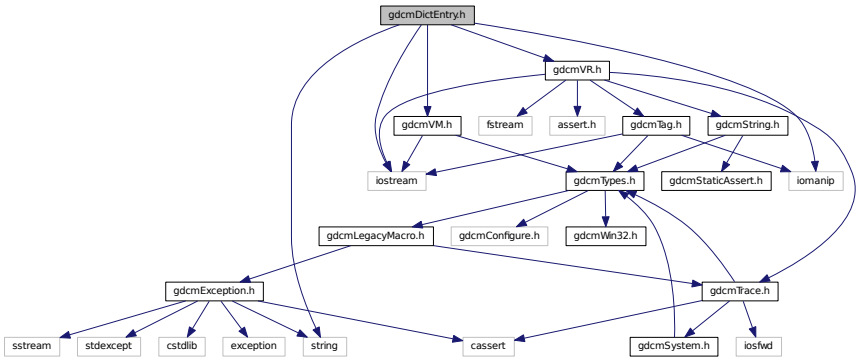
Namespaces

- gdcm

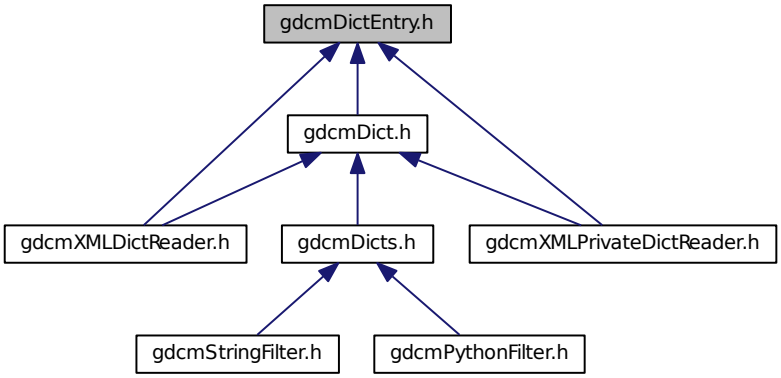
11.67 gdcmDictEntry.h File Reference

```
#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

- gdcm

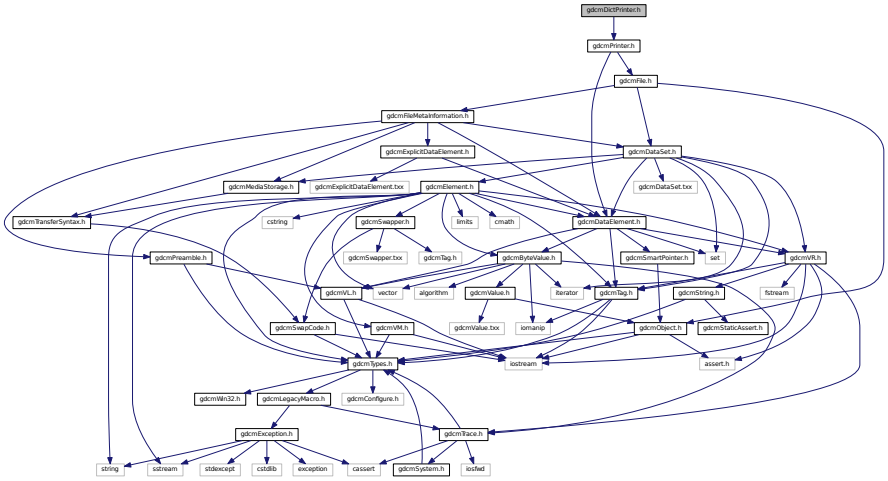
Functions

- std::ostream & gdcm::operator<< (std::ostream &os, const DictEntry &val)

11.68 gdcmDictPrinter.h File Reference

```
#include "gdcmPrinter.h"
```

Include dependency graph for gdcmDictPrinter.h:



Classes

- class gdcm::DictPrinter

  DictPrinter class.

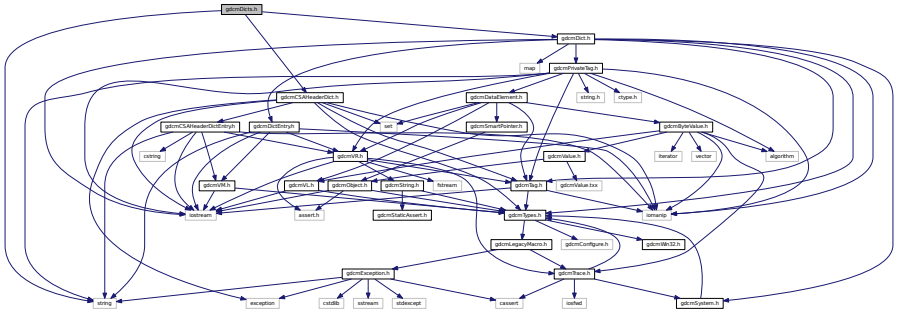
Namespaces

- gdcm

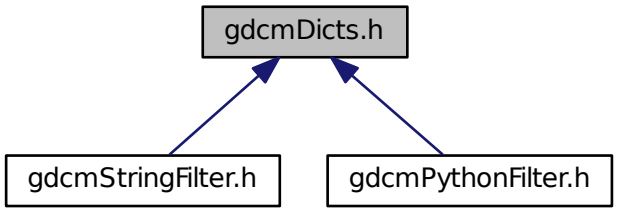
11.69 gdcmDicts.h File Reference

```
#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

- gdcm

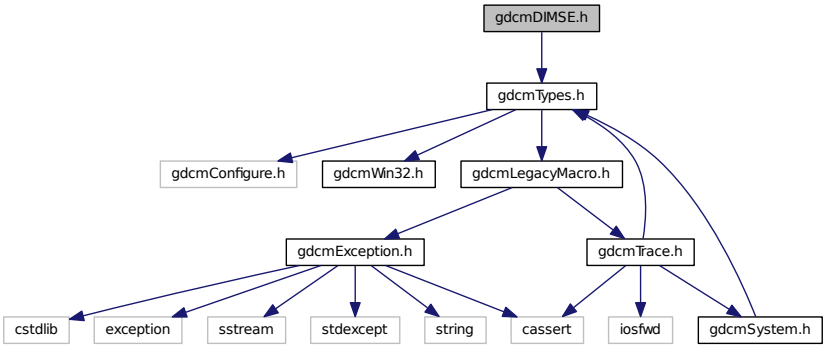
Functions

- std::ostream & gdcm::operator<< (std::ostream &os, const Dicts &d)

11.70 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.

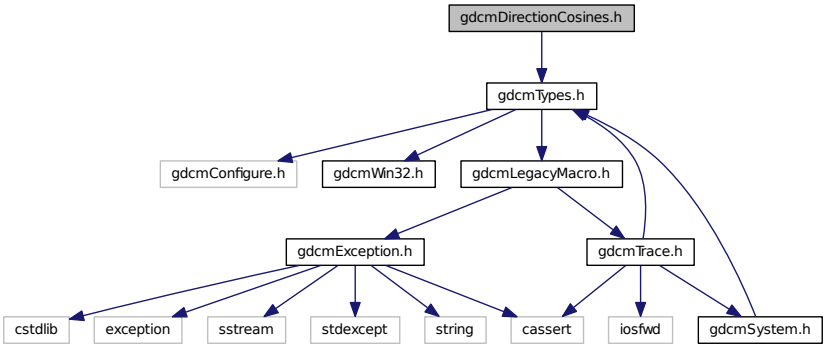
Namespaces

- gdcm
- gdcm::network

11.71 gdcmDirectionCosines.h File Reference

```
#include "gdcmTypes.h"
```

Include dependency graph for gdcmDirectionCosines.h:



Classes

- class gdcm::DirectionCosines

  class to handle DirectionCosines

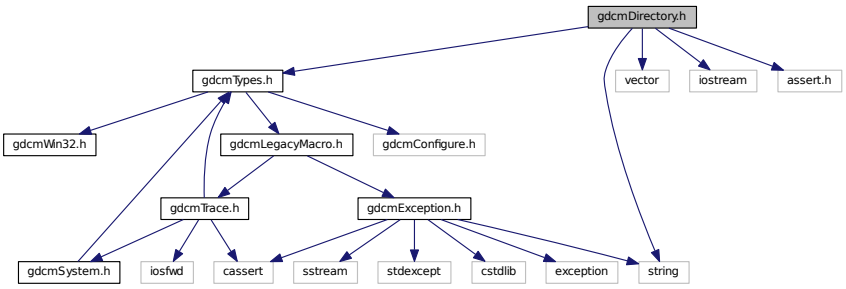
Namespaces

- gdcm

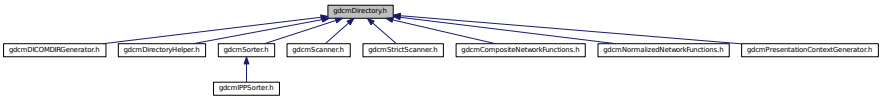
11.72 gdcmDirectory.h File Reference

```
#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

- gdcm

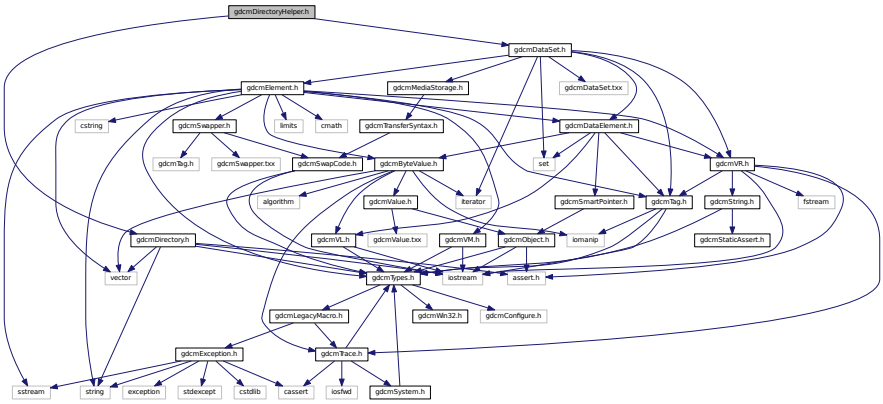
Functions

- std::ostream & gdcm::operator<< (std::ostream &os, const Directory &d)

11.73 gdcmDirectoryHelper.h File Reference

```
#include "gdcmDirectory.h"
#include "gdcmDataSet.h"
```

Include dependency graph for gdcmDirectoryHelper.h:



Classes

- class gdcm::DirectoryHelper

  DirectoryHelper.

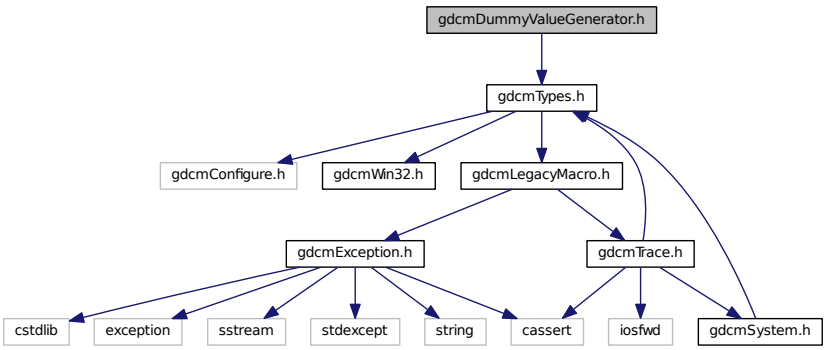
Namespaces

- gdcm

11.74 gdcmDummyValueGenerator.h File Reference

```
#include "gdcmTypes.h"
```

Include dependency graph for gdcmDummyValueGenerator.h:



Classes

- class gdcm::DummyValueGenerator

  Class for generating dummy value.

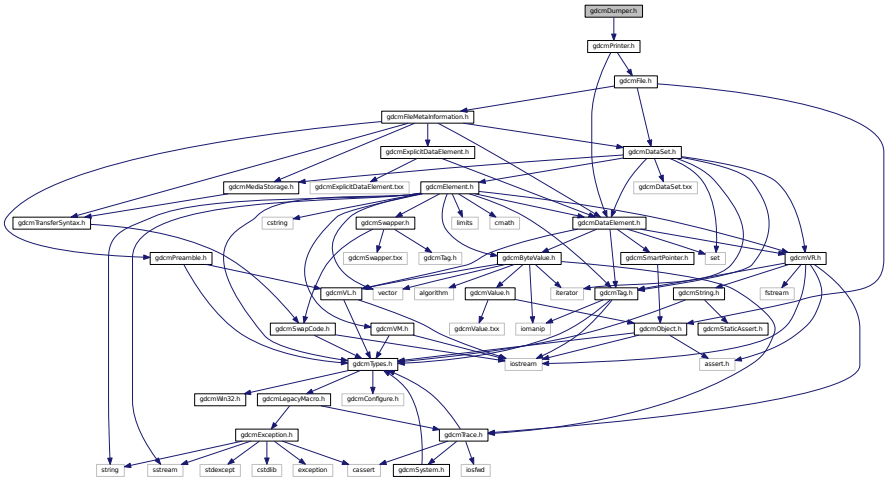
Namespaces

- gdcm

11.75 gdcmDumper.h File Reference

```
#include "gdcmPrinter.h"
```

Include dependency graph for gdcmDumper.h:



Classes

- class gdcm::Dumper

  Codec class.

Namespaces

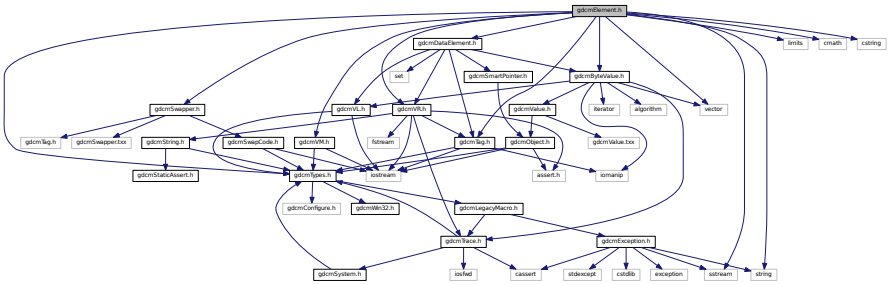
- gdcm

11.76 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_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< T >

  EncodingImplementation.
- class gdcm::EncodingImplementation< VR::VRASCII >
- class gdcm::EncodingImplementation< VR::VRBINARY >
- struct gdcm::ignore_char

Namespaces

- gdcm

Macros

- #define VRDS16ILLEGAL

Functions

- ignore_char const gdcm::backslash ('\\')
- std::istream & gdcm::operator>> (std::istream &in, ignore_char const &ic)
- template<typename Float >
  std::string gdcm::to_string (Float data)

11.76.1 Macro Definition Documentation

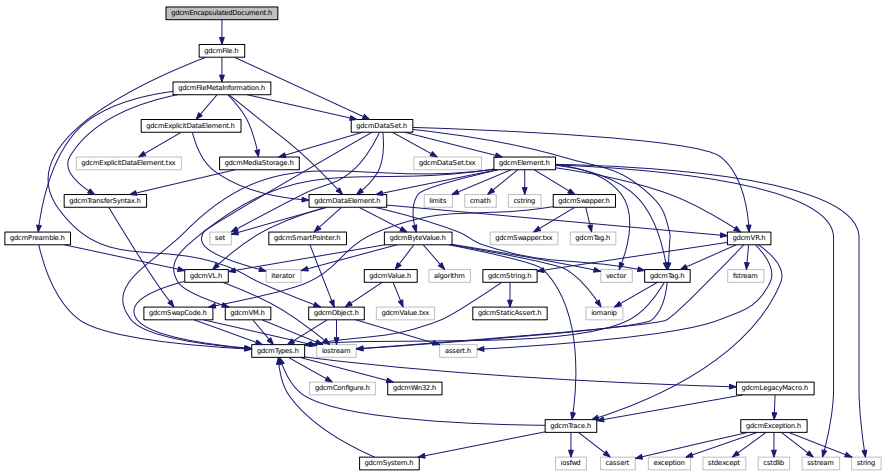
11.76.1.1 VRDS16ILLEGAL

```
#define VRDS16ILLEGAL
```

11.77 gdcmEncapsulatedDocument.h File Reference

```
#include "gdcmFile.h"
```

Include dependency graph for gdcmEncapsulatedDocument.h:



Classes

- class gdcm::EncapsulatedDocument

  EncapsulatedDocument.

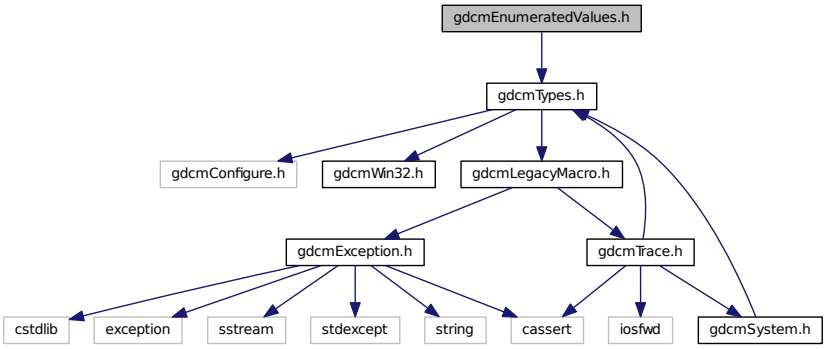
Namespaces

- gdcm

11.78 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:

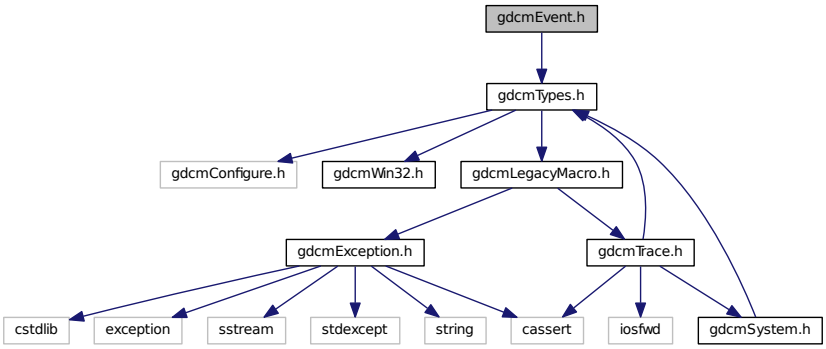
Namespaces

- gdcm

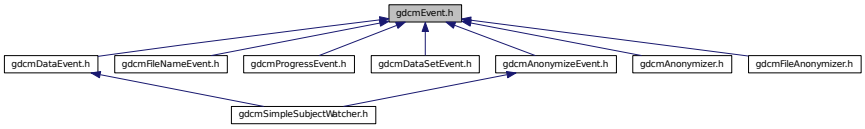
11.79 gdcmEvent.h File Reference

```
#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

- gdcm

Macros

- #define gdcmEventMacro(classname, super)

Functions

- std::ostream & gdcm::operator<< (std::ostream &os, Event &e)

  Generic inserter operator for Event and its subclasses.

11.79.1 Macro Definition Documentation

11.79.1.1 gdcmEventMacro

```
#define gdcmEventMacro(
              classname,
              super )
```

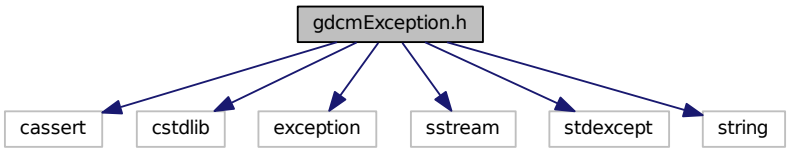
Value:

```
 \
class  classname : public super { \
  public: \
    typedef classname Self; \
    typedef super Superclass; \
    classname() {} \
    virtual ~classname() {} \
    virtual const char * GetEventName() const { return #classname; } \
    virtual bool CheckEvent(const ::gdcm::Event* e) const \
      { return dynamic_cast<const Self*>(e) ? true : false; } \
    virtual ::gdcm::Event* MakeObject() const \
      { return new Self; } \
    classname(const Self&s) : super(s){}; \
  private: \
    void operator=(const Self&); \
}
```

11.80 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.

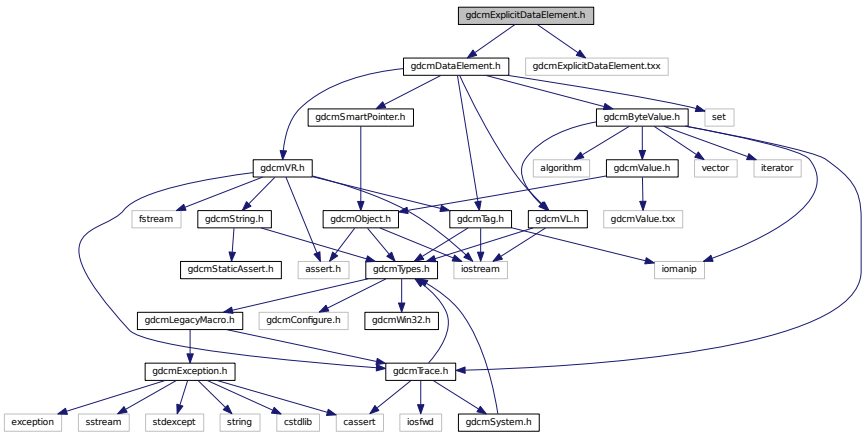
Namespaces

- gdcm

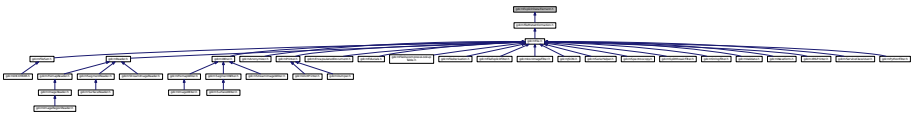
11.81 gdcmExplicitDataElement.h File Reference

```
#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.

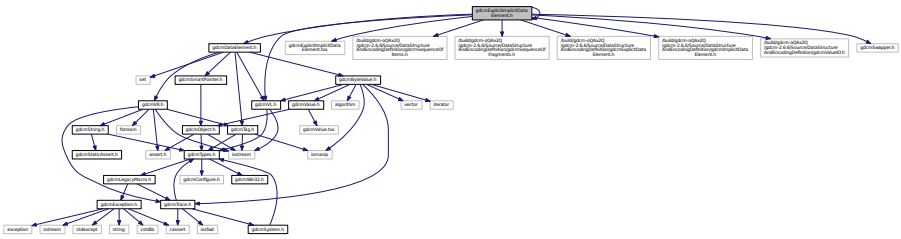
Namespaces

- gdcm

11.82 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.

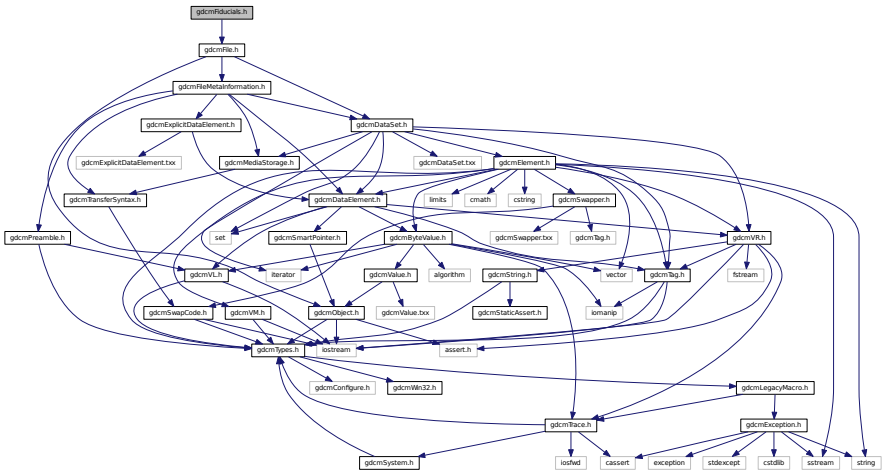
Namespaces

- gdcm

11.83 gdcmFiducials.h File Reference

```
#include "gdcmFile.h"
```

Include dependency graph for gdcmFiducials.h:



Classes

- class gdcm::Fiducials

  Fiducials.

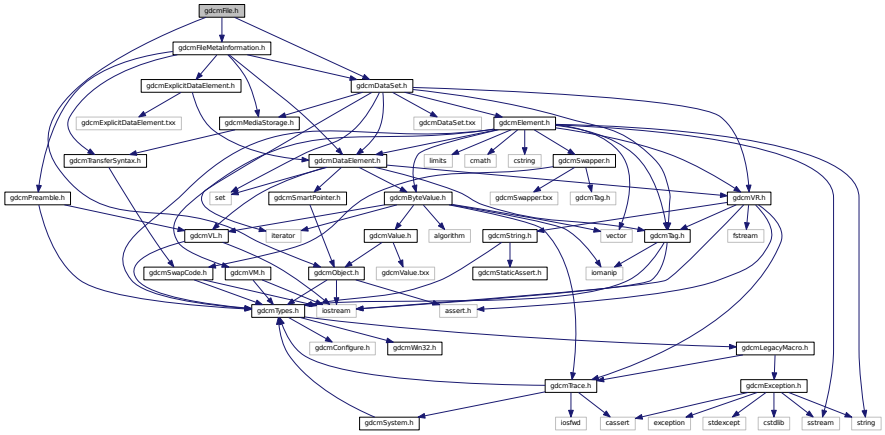
Namespaces

- gdcm

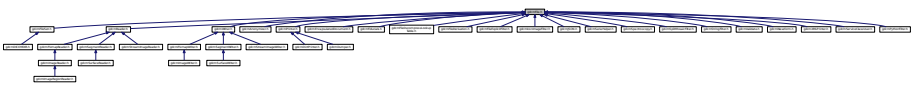
11.84 gdcmFile.h File Reference

```
#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

- gdcm

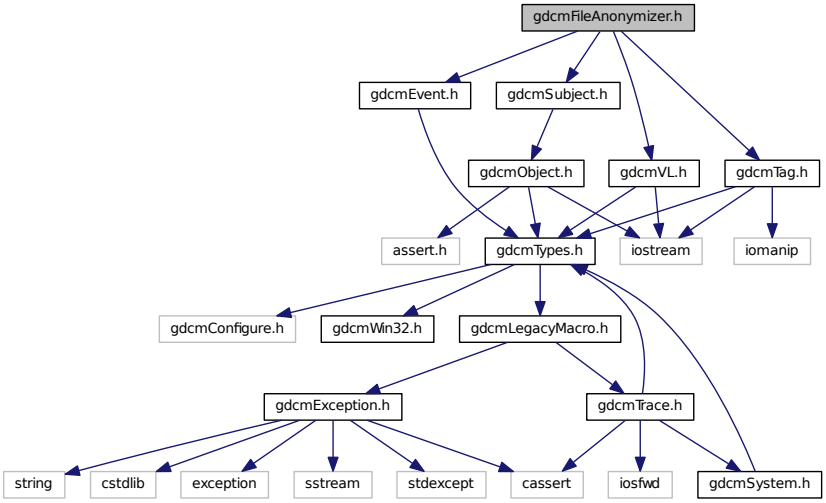
Functions

- std::ostream & gdcm::operator<< (std::ostream &os, const File &val)

11.85 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.

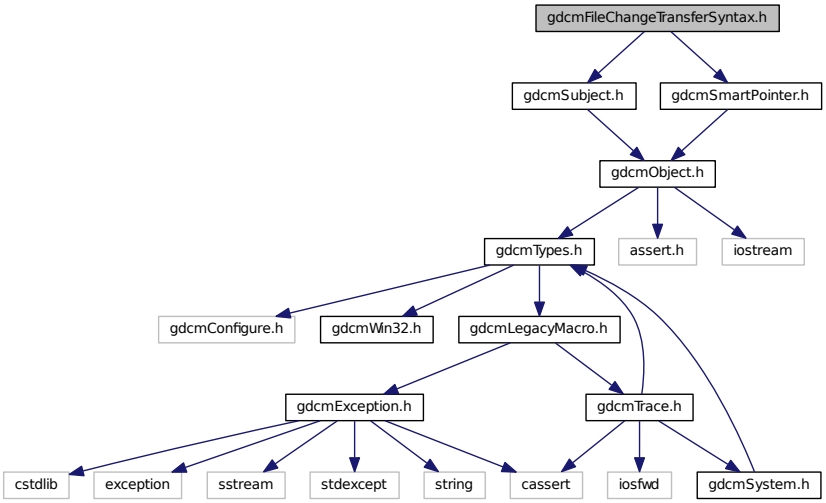
Namespaces

- gdcm

11.86 gdcmFileChangeTransferSyntax.h File Reference

```
#include "gdcmSubject.h"
#include "gdcmSmartPointer.h"
```

Include dependency graph for gdcmFileChangeTransferSyntax.h:



Classes

- class gdcm::FileChangeTransferSyntax

  FileChangeTransferSyntax.

Namespaces

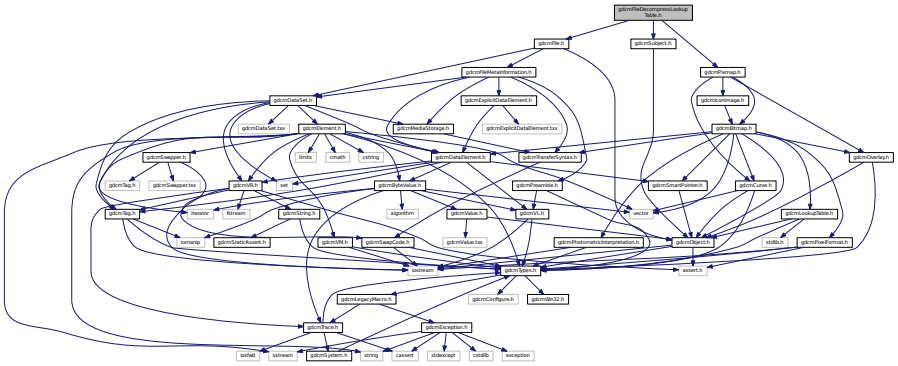
- gdcm

11.87 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.

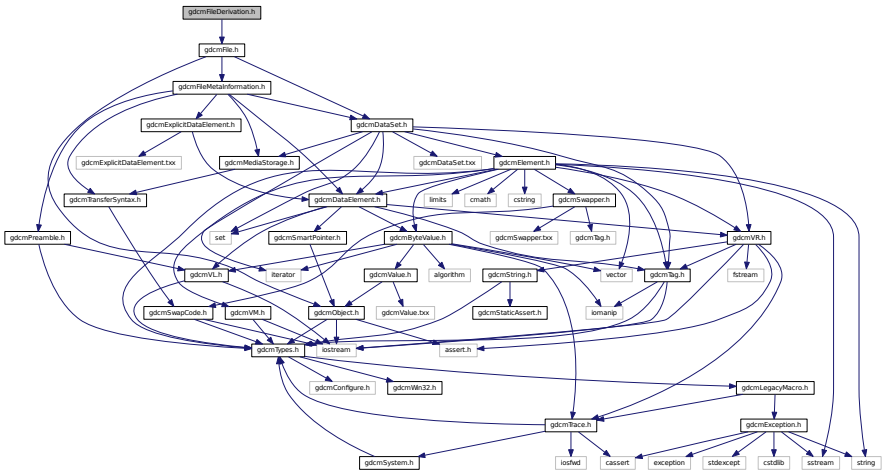
Namespaces

- gdcm

11.88 gdcmFileDerivation.h File Reference

```
#include "gdcmFile.h"
```

Include dependency graph for gdcmFileDerivation.h:



Classes

- class gdcm::FileDerivation

  FileDerivation class.

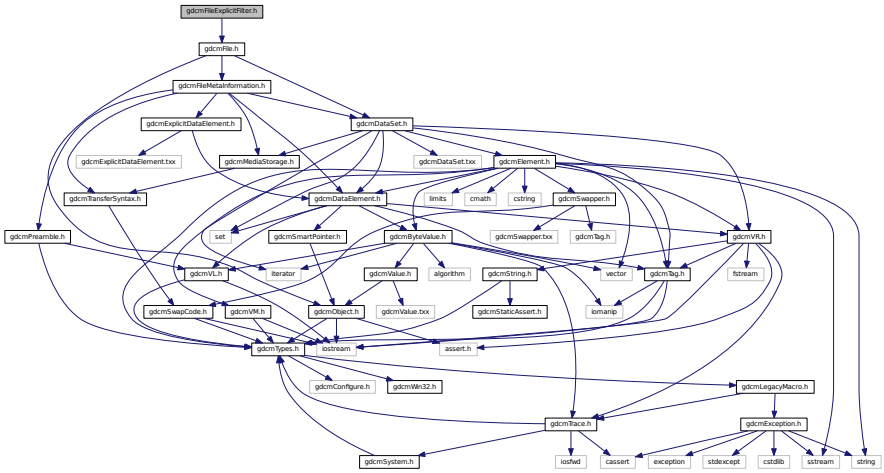
Namespaces

- gdcm

11.89 gdcmFileExplicitFilter.h File Reference

```
#include "gdcmFile.h"
```

Include dependency graph for gdcmFileExplicitFilter.h:



Classes

- class gdcm::FileExplicitFilter

  FileExplicitFilter class.

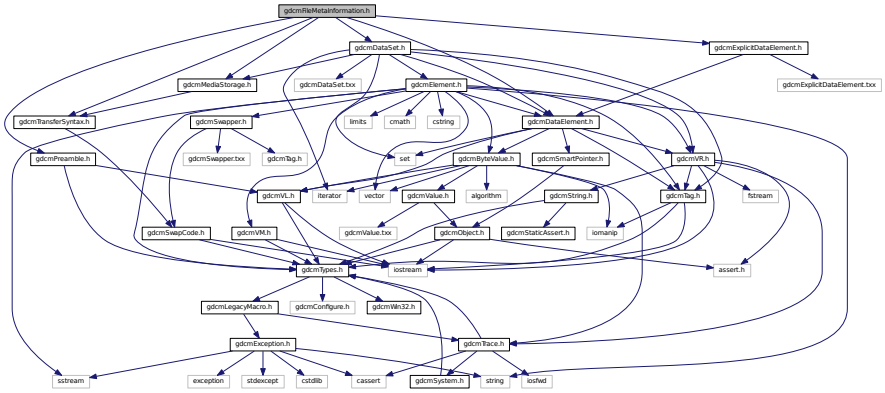
Namespaces

- gdcm

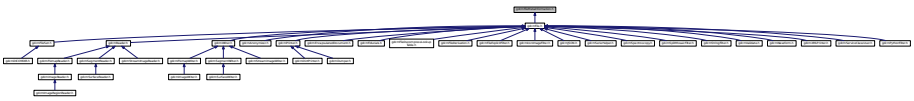
11.90 gdcmFileMetaInformation.h File Reference

```
#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

- gdcm

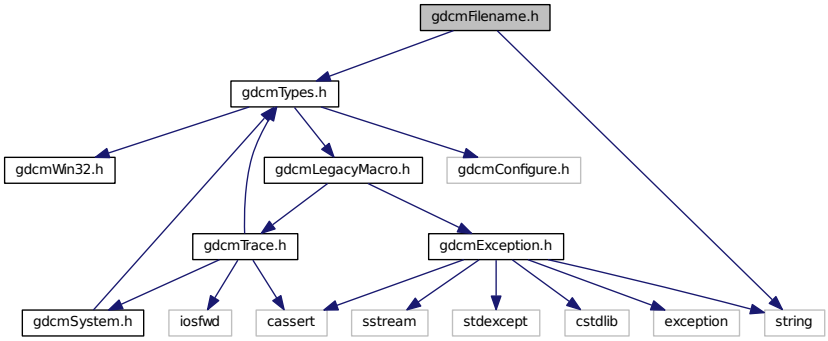
Functions

- std::ostream & gdcm::operator<< (std::ostream &os, const FileMetaInformation &val)

11.91 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.

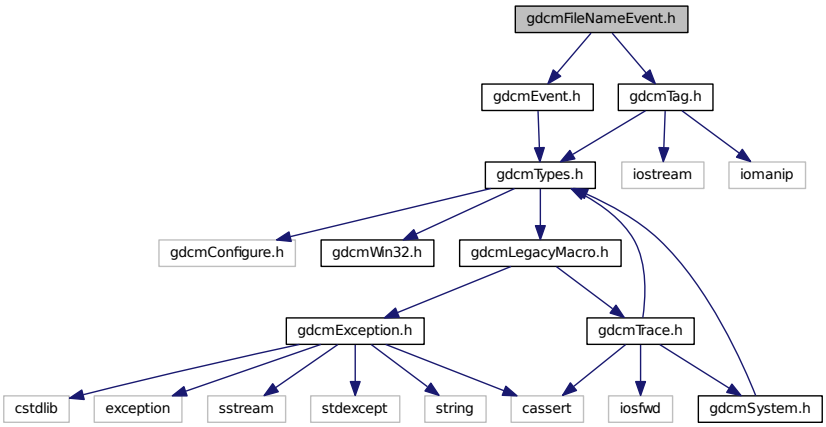
Namespaces

- gdcm

11.92 gdcmFileNameEvent.h File Reference

```
#include "gdcmEvent.h"
#include "gdcmTag.h"
```

Include dependency graph for gdcmFileNameEvent.h:



Classes

- class gdcm::FileNameEvent
  FileNameEvent.

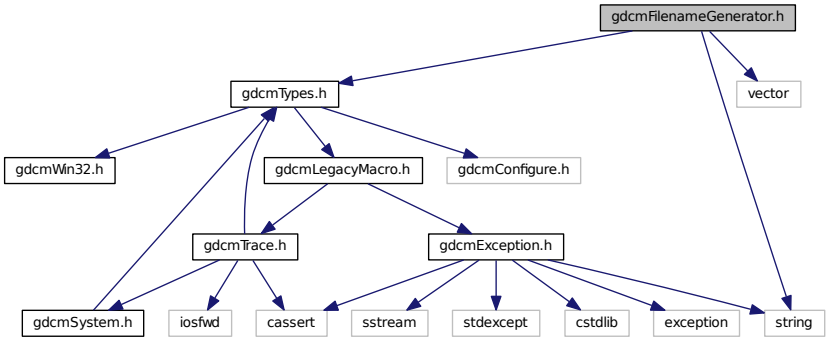
Namespaces

- gdcm

11.93 gdcmFilenameGenerator.h File Reference

```
#include "gdcmTypes.h"
#include <string>
#include <vector>
```

Include dependency graph for gdcmFilenameGenerator.h:



Classes

- class gdcm::FilenameGenerator

  FilenameGenerator.

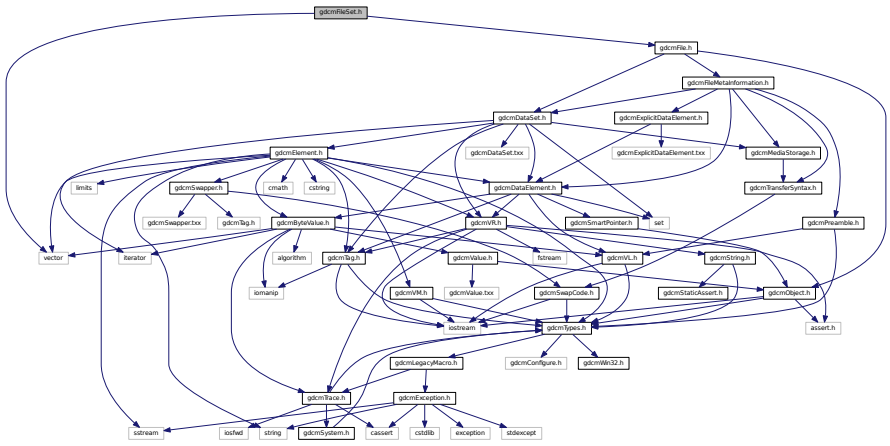
Namespaces

- gdcm

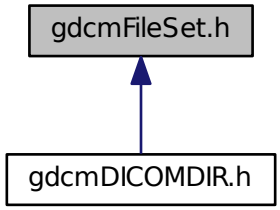
11.94 gdcmFileSet.h File Reference

```
#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

- gdcm

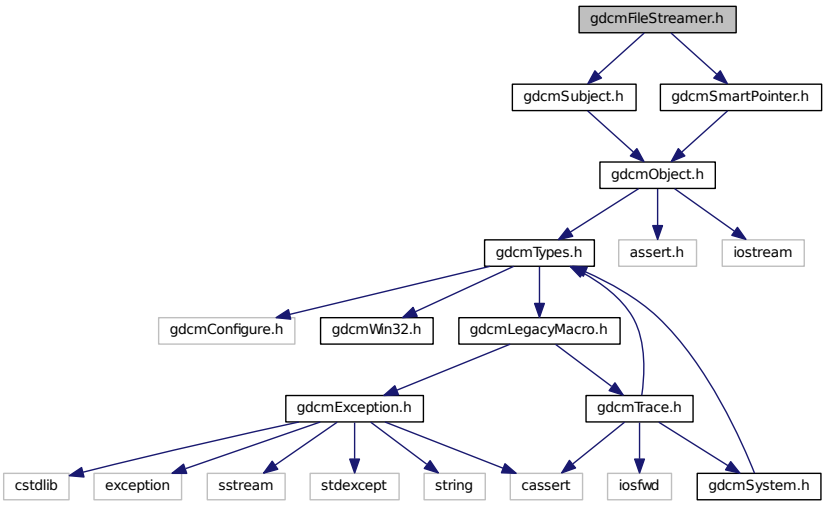
Functions

- std::ostream & gdcm::operator<< (std::ostream &os, const FileSet &f)

11.95 gdcmFileStreamer.h File Reference

```
#include "gdcmSubject.h"
#include "gdcmSmartPointer.h"
```

Include dependency graph for gdcmFileStreamer.h:



Classes

- class gdcm::FileStreamer

  FileStreamer.

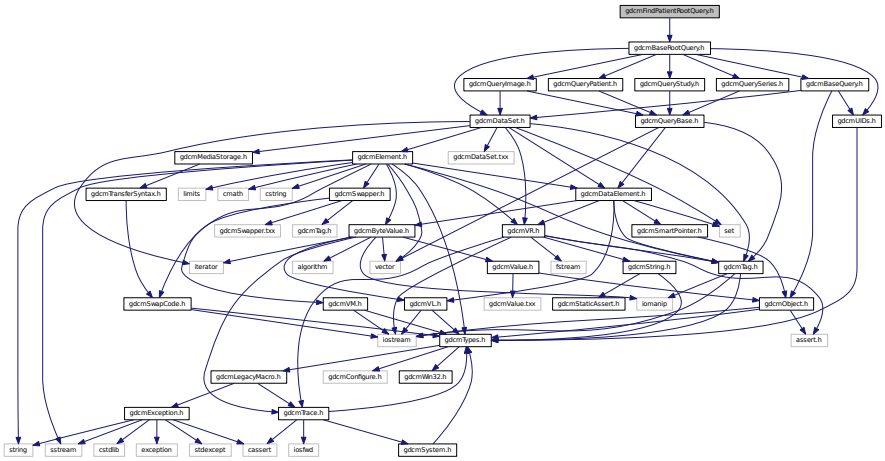
Namespaces

- gdcm

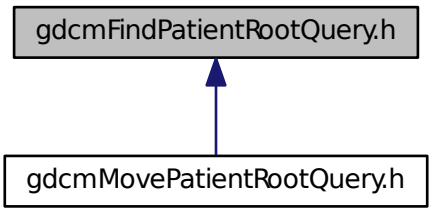
11.96 gdcmFindPatientRootQuery.h File Reference

```
#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.

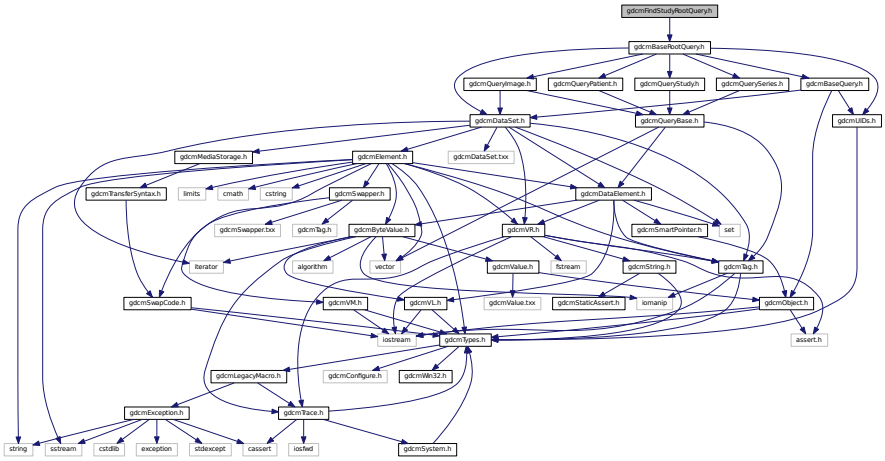
Namespaces

- gdcm

11.97 gdcmFindStudyRootQuery.h File Reference

```
#include "gdcmBaseRootQuery.h"
```

Include dependency graph for gdcmFindStudyRootQuery.h:



Classes

- class gdcm::FindStudyRootQuery

  FindStudyRootQuery.

Namespaces

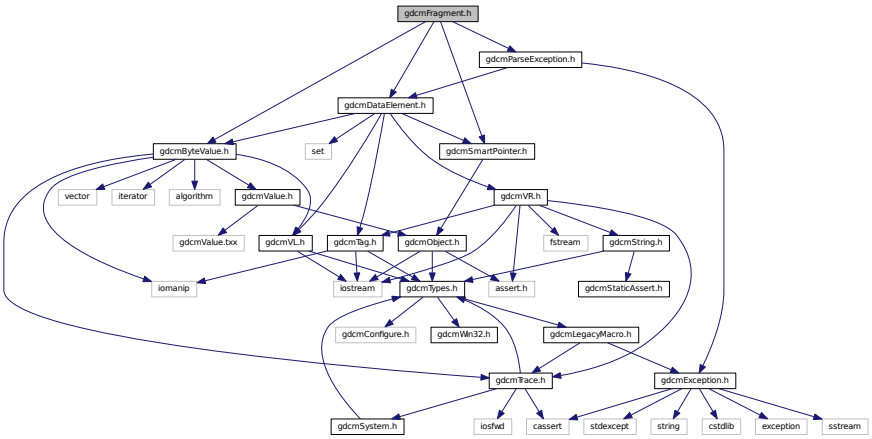
- gdcm

11.98 gdcmFragment.h File Reference

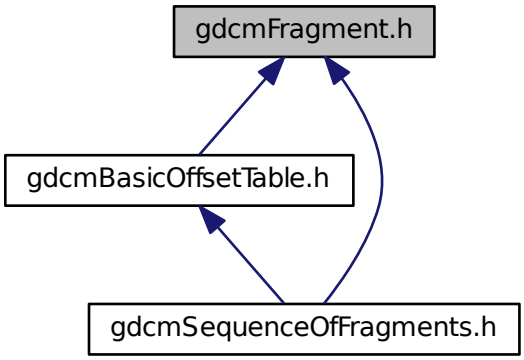
```
#include "gdcmDataElement.h"
#include "gdcmByteValue.h"
#include "gdcmSmartPointer.h"
```

```
#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

- gdcm

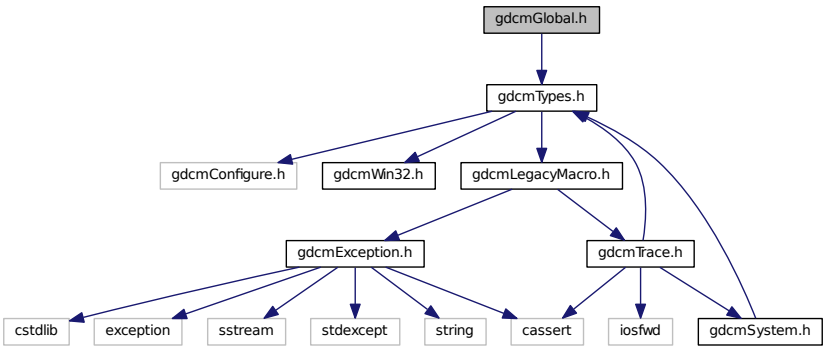
Functions

- std::ostream & gdcm::operator<< (std::ostream &os, const Fragment &val)

11.99 gdcmGlobal.h File Reference

```
#include "gdcmTypes.h"
```

Include dependency graph for gdcmGlobal.h:



Classes

- class gdcm::Global

  Global.

Namespaces

- gdcm

Functions

- std::ostream & gdcm::operator<< (std::ostream &os, const Global &g)

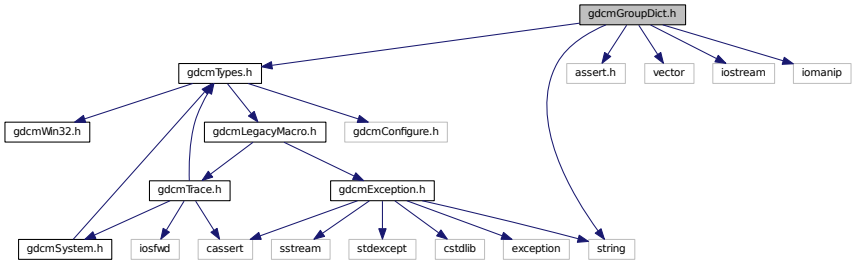
Variables

- static Global gdcm::GlobalInstance

11.100 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

- gdcm

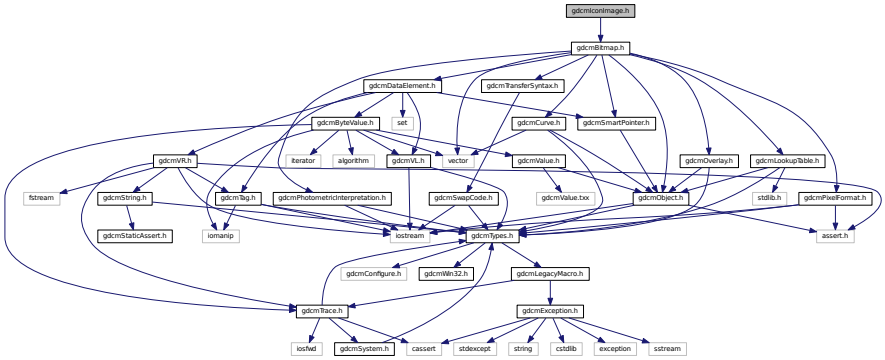
Functions

- std::ostream & gdcm::operator<< (std::ostream &_os, const GroupDict &_val)

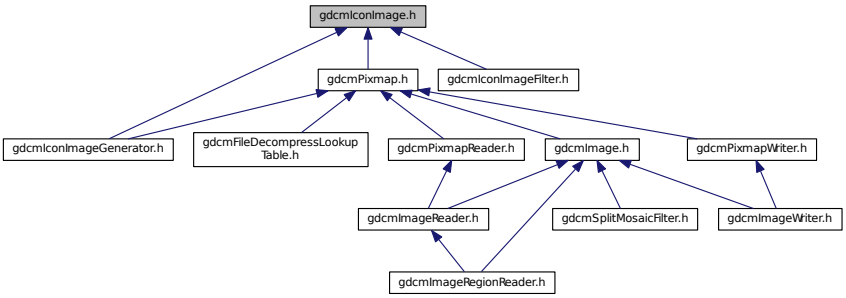
11.101 gdcmIconImage.h File Reference

```
#include "gdcmBitmap.h"
```

Include dependency graph for gdcmIconImage.h:



This graph shows which files directly or indirectly include this file:



Namespaces

- gdcm

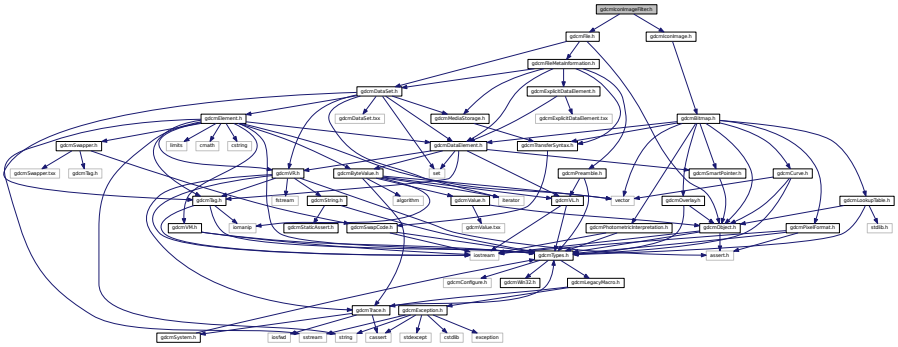
Typedefs

- typedef Bitmap gdcm::IconImage

11.102 gdcmIconImageFilter.h File Reference

```
#include "gdcmFile.h"
#include "gdcmIconImage.h"
```

Include dependency graph for gdcmIconImageFilter.h:



Classes

- class gdcm::IconImageFilter

  IconImageFilter.

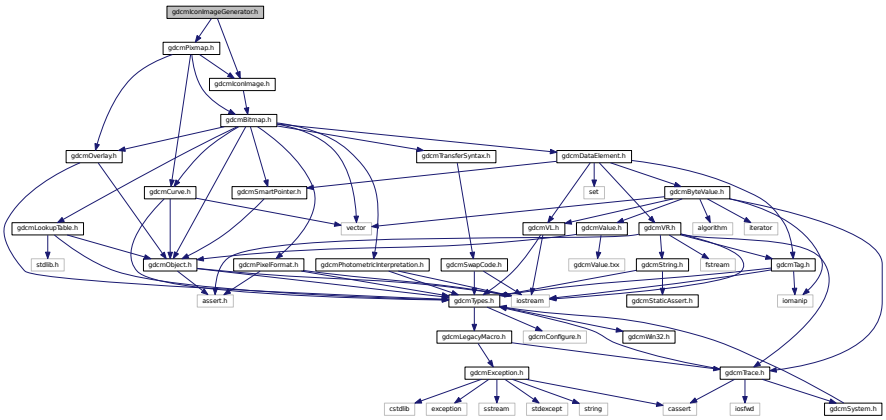
Namespaces

- gdcm

11.103 gdcmIconImageGenerator.h File Reference

```
#include "gdcmPixmap.h"
#include "gdcmIconImage.h"
```

Include dependency graph for gdcmIconImageGenerator.h:



Classes

- class gdcm::IconImageGenerator

  IconImageGenerator.

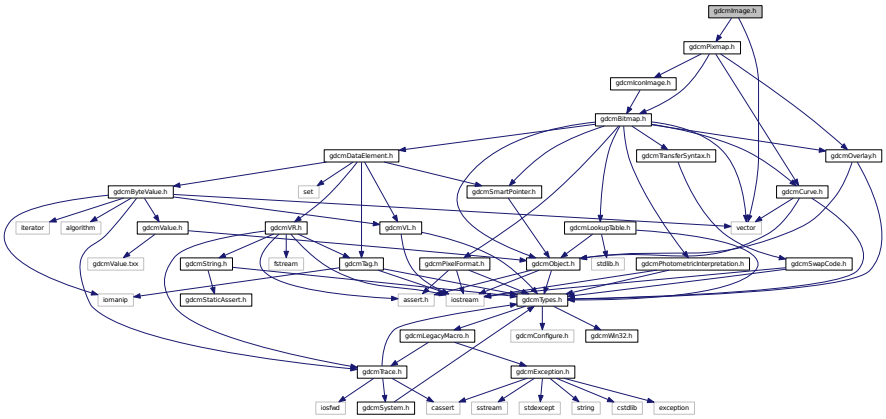
Namespaces

- gdcm

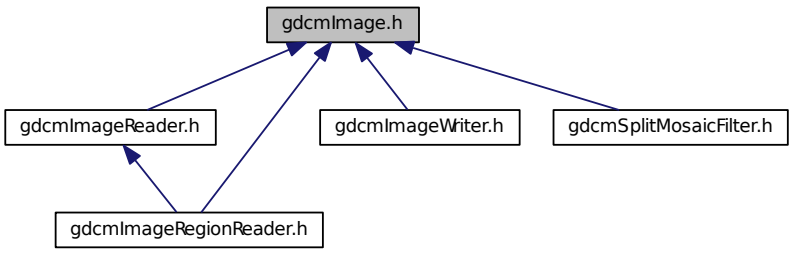
11.104 gdcmImage.h File Reference

```
#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.

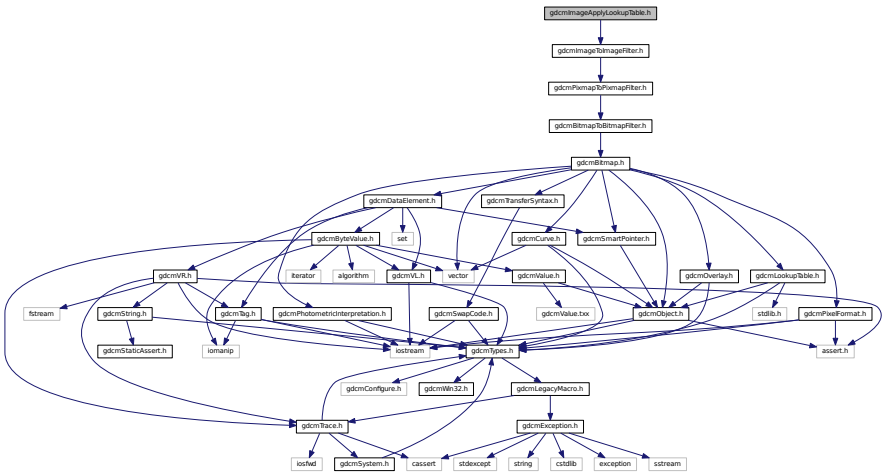
Namespaces

- gdcm

11.105 gdcmImageApplyLookupTable.h File Reference

```
#include "gdcmImageToImageFilter.h"
```

Include dependency graph for gdcmImageApplyLookupTable.h:



Classes

- class gdcm::ImageApplyLookupTable

  ImageApplyLookupTable class.

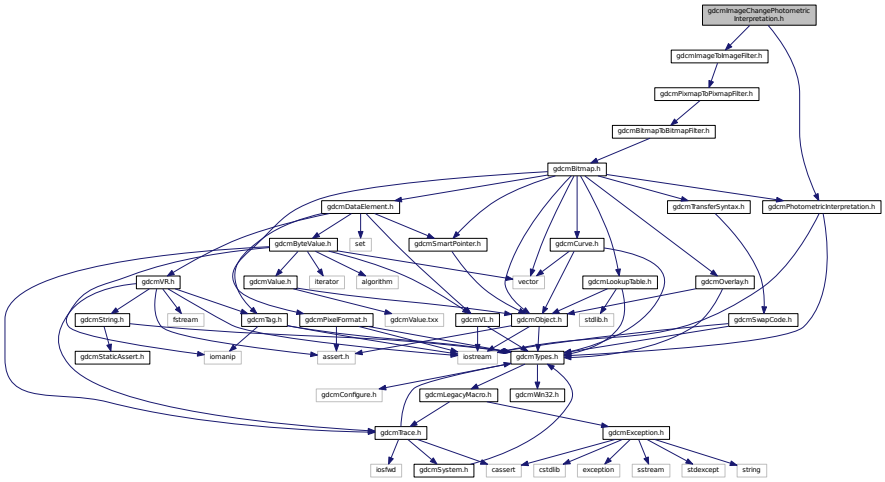
Namespaces

- gdcm

11.106 gdcmImageChangePhotometricInterpretation.h File Reference

```
#include "gdcmImageToImageFilter.h"
#include "gdcmPhotometricInterpretation.h"
```

Include dependency graph for gdcmImageChangePhotometricInterpretation.h:



Classes

- class gdcm::ImageChangePhotometricInterpretation

  ImageChangePhotometricInterpretation class.

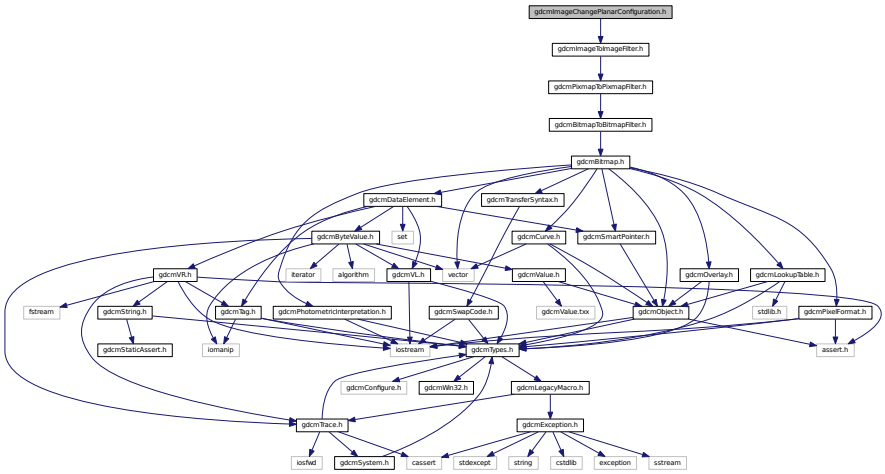
Namespaces

- gdcm

11.107 gdcmImageChangePlanarConfiguration.h File Reference

```
#include "gdcmImageToImageFilter.h"
```

Include dependency graph for gdcmImageChangePlanarConfiguration.h:



Classes

- class gdcm::ImageChangePlanarConfiguration

  ImageChangePlanarConfiguration class.

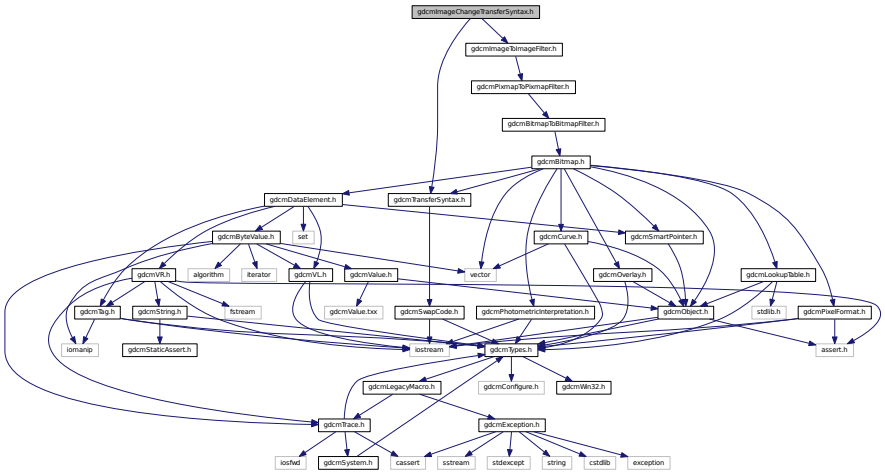
Namespaces

- gdcm

11.108 gdcmImageChangeTransferSyntax.h File Reference

```
#include "gdcmImageToImageFilter.h"
#include "gdcmTransferSyntax.h"
```

Include dependency graph for gdcmImageChangeTransferSyntax.h:



Classes

- class gdcm::ImageChangeTransferSyntax

  ImageChangeTransferSyntax class.

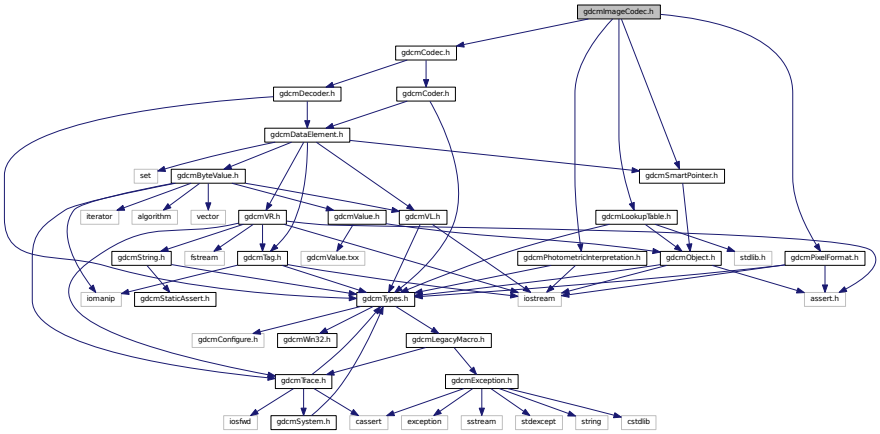
Namespaces

- gdcm

11.109 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.

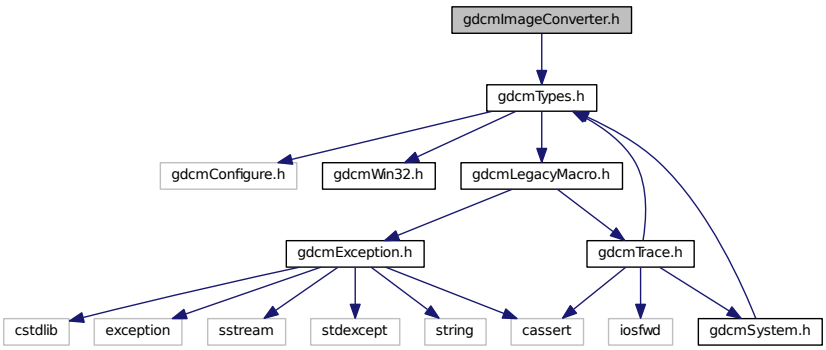
Namespaces

- gdcm

11.110 gdcmImageConverter.h File Reference

```
#include "gdcmTypes.h"
```

Include dependency graph for gdcmImageConverter.h:



Classes

- class gdcm::ImageConverter

  Image Converter.

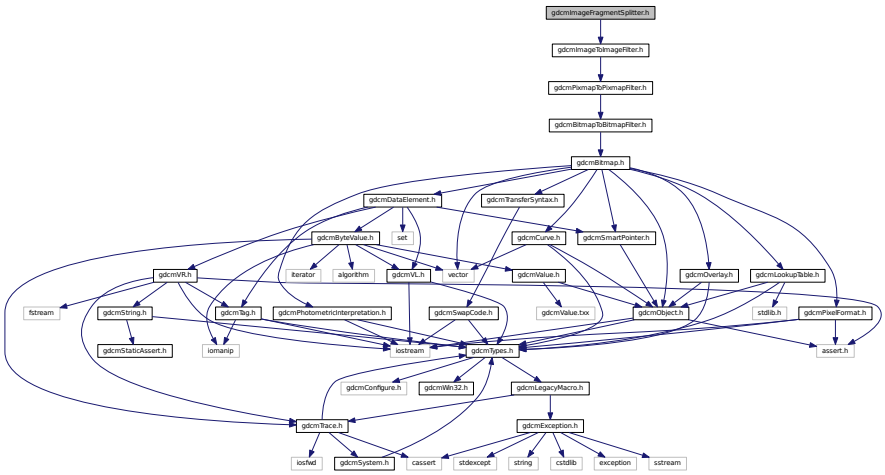
Namespaces

- gdcm

11.111 gdcmImageFragmentSplitter.h File Reference

```
#include "gdcmImageToImageFilter.h"
```

Include dependency graph for gdcmImageFragmentSplitter.h:



Classes

- class gdcm::ImageFragmentSplitter

  ImageFragmentSplitter class.

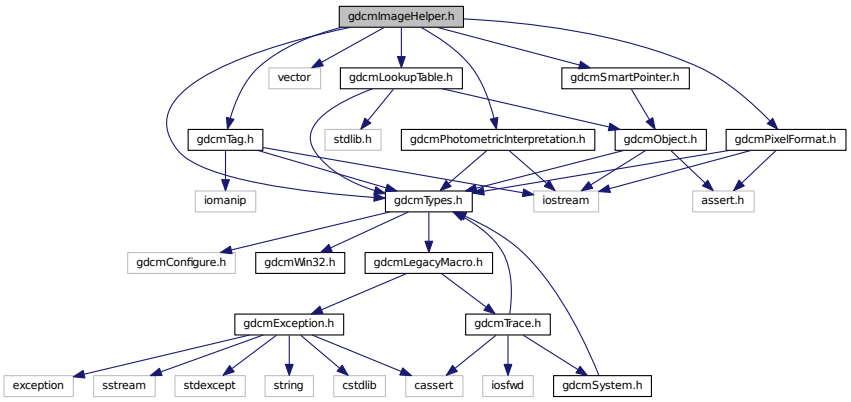
Namespaces

- gdcm

11.112 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

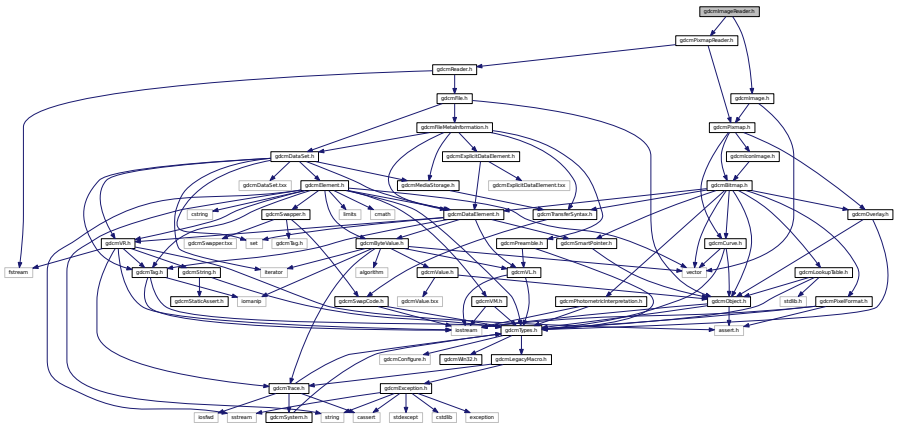
Namespaces

- gdcm

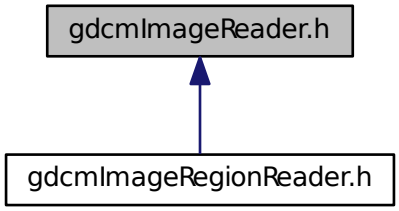
11.113 gdcmImageReader.h File Reference

```
#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.

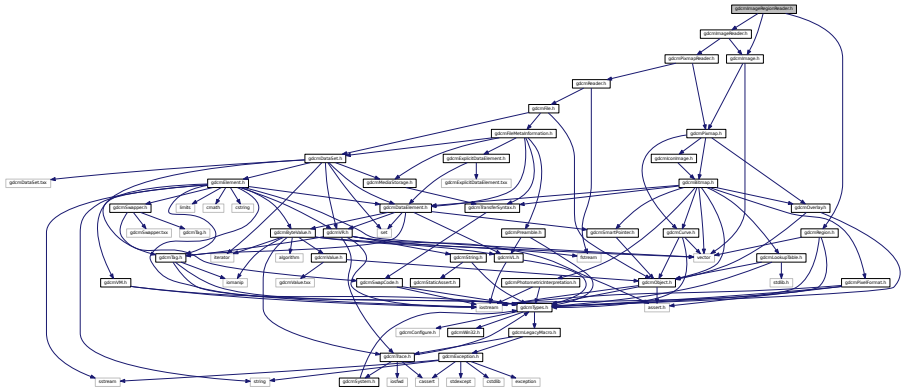
Namespaces

- gdcm

11.114 gdcmImageRegionReader.h File Reference

```
#include "gdcmImageReader.h"
#include "gdcmImage.h"
#include "gdcmRegion.h"
```

Include dependency graph for gdcmImageRegionReader.h:



Classes

- class gdcm::ImageRegionReader

  ImageRegionReader.

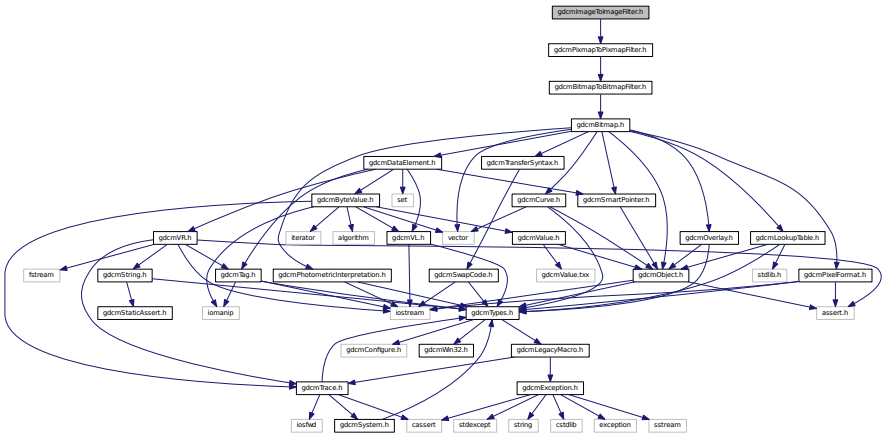
Namespaces

- gdcm

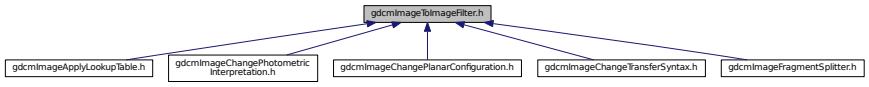
11.115 gdcmImageToImageFilter.h File Reference

```
#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.

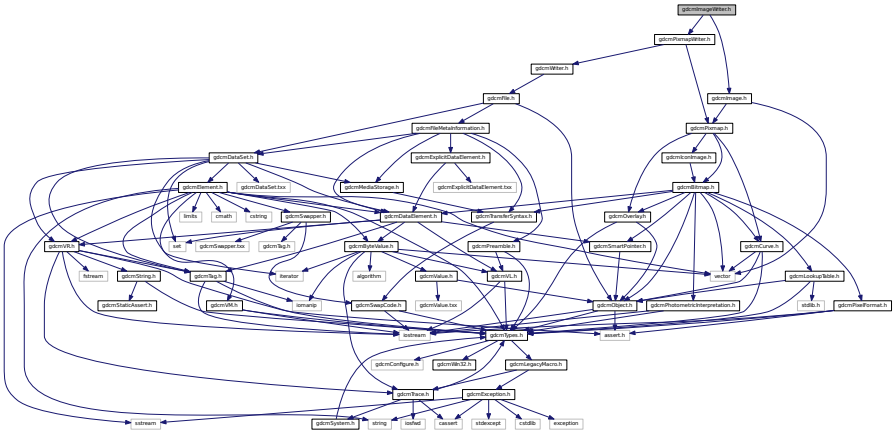
Namespaces

- gdcm

11.116 gdcmImageWriter.h File Reference

```
#include "gdcmPixmapWriter.h"
#include "gdcmImage.h"
```

Include dependency graph for gdcmImageWriter.h:



Classes

- class gdcm::ImageWriter

  ImageWriter.

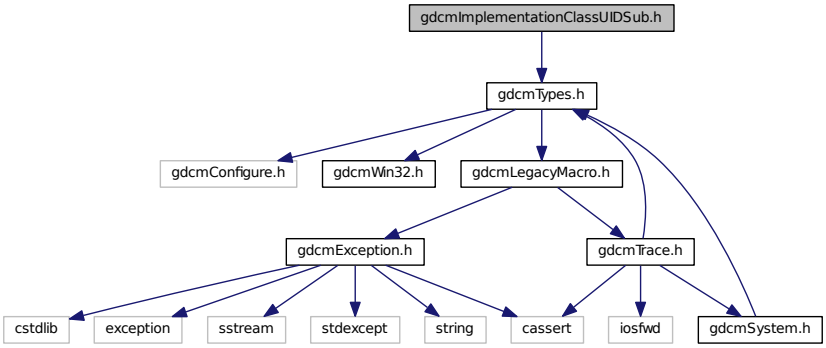
Namespaces

- gdcm

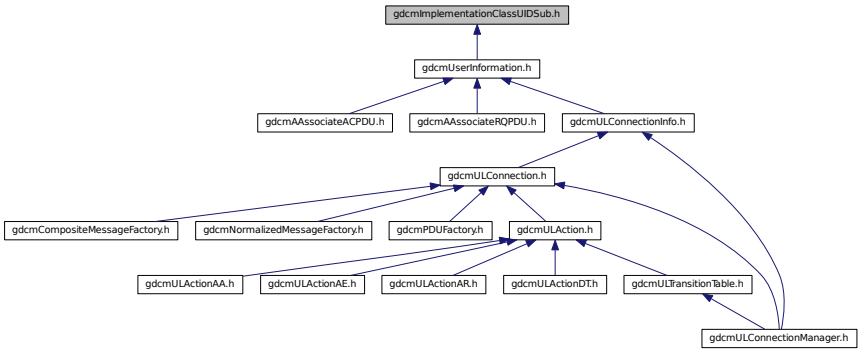
11.117 gdcmImplementationClassUIDSub.h File Reference

```
#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.

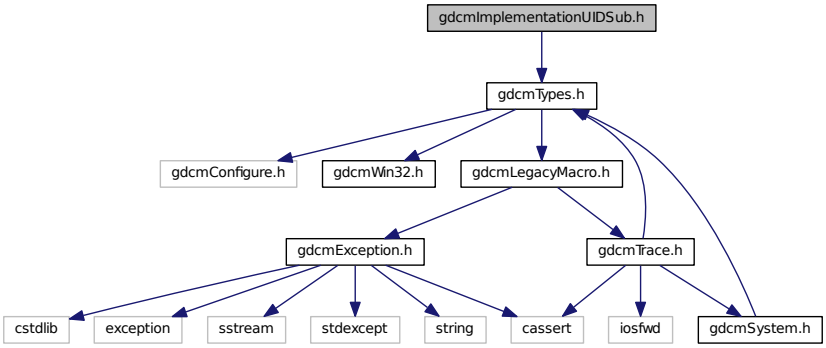
Namespaces

- gdcm
- gdcm::network

11.118 gdcmImplementationUIDSub.h File Reference

```
#include "gdcmTypes.h"
```

Include dependency graph for gdcmImplementationUIDSub.h:



Classes

- class gdcm::network::ImplementationUIDSub

  ImplementationUIDSub.

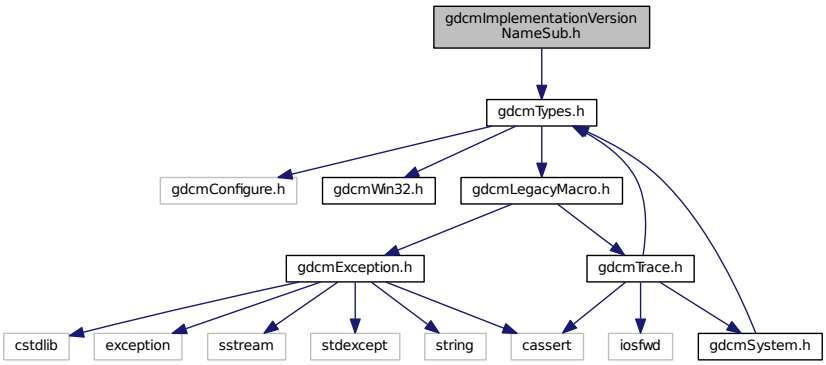
Namespaces

- gdcm
- gdcm::network

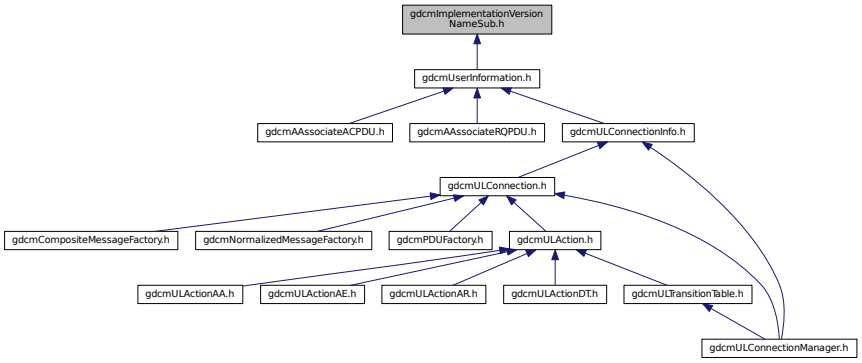
11.119 gdcmImplementationVersionNameSub.h File Reference

```
#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.

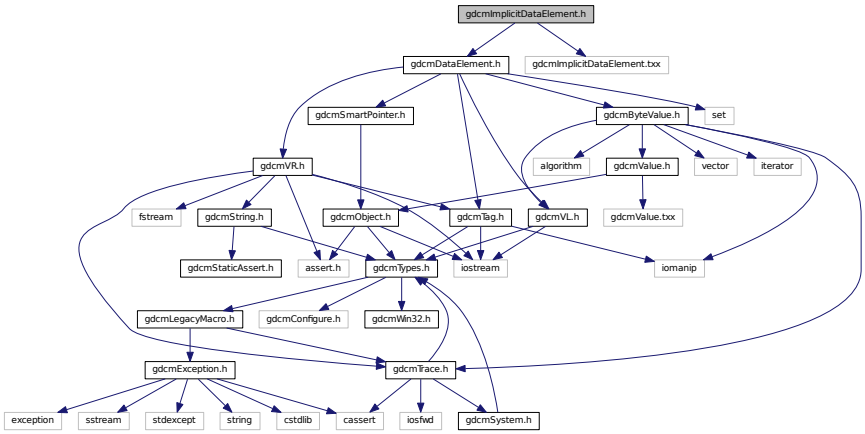
Namespaces

- gdcm
- gdcm::network

11.120 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.

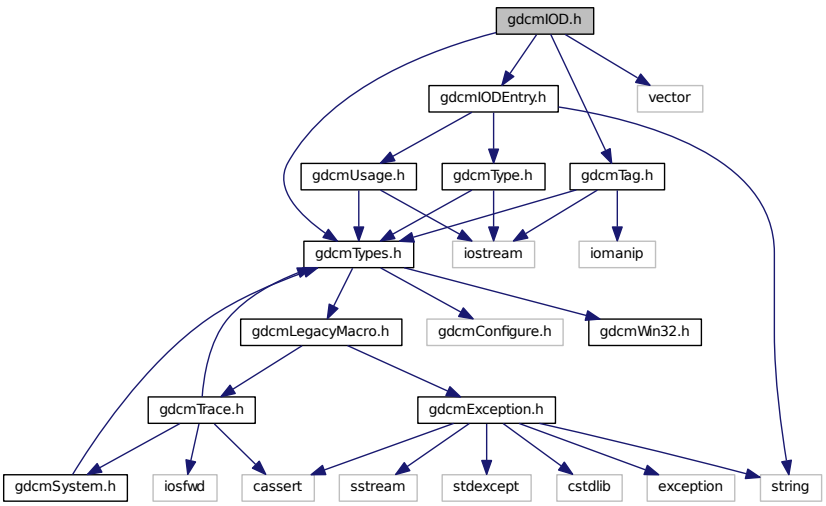
Namespaces

- gdcm

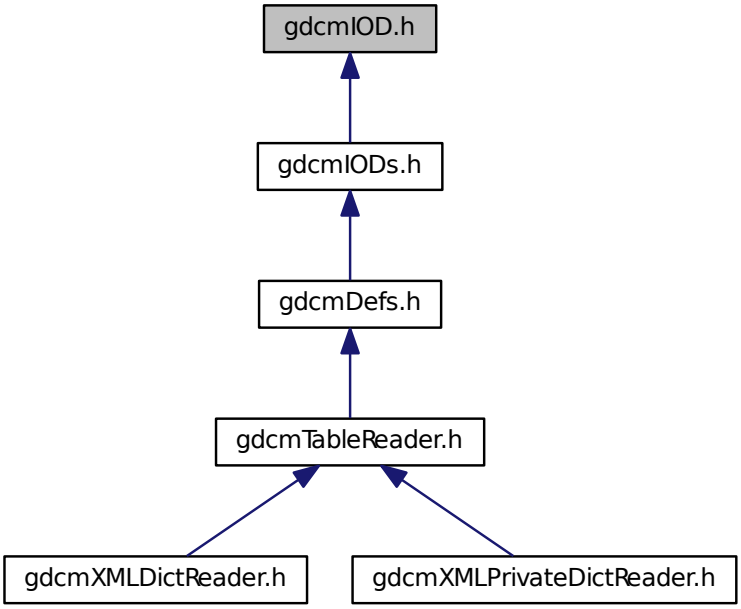
11.121 gdcmIOD.h File Reference

```
#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

- gdcm

Functions

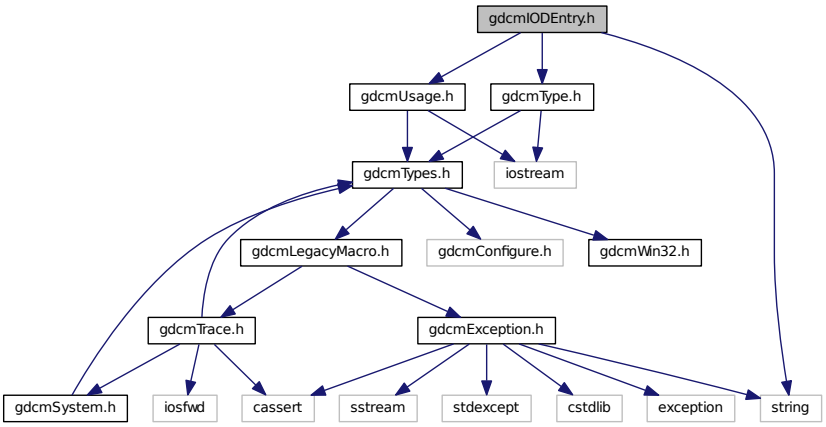
- std::ostream & gdcm::operator<< (std::ostream &_os, const IOD &_val)

11.122 gdcmIODEntry.h File Reference

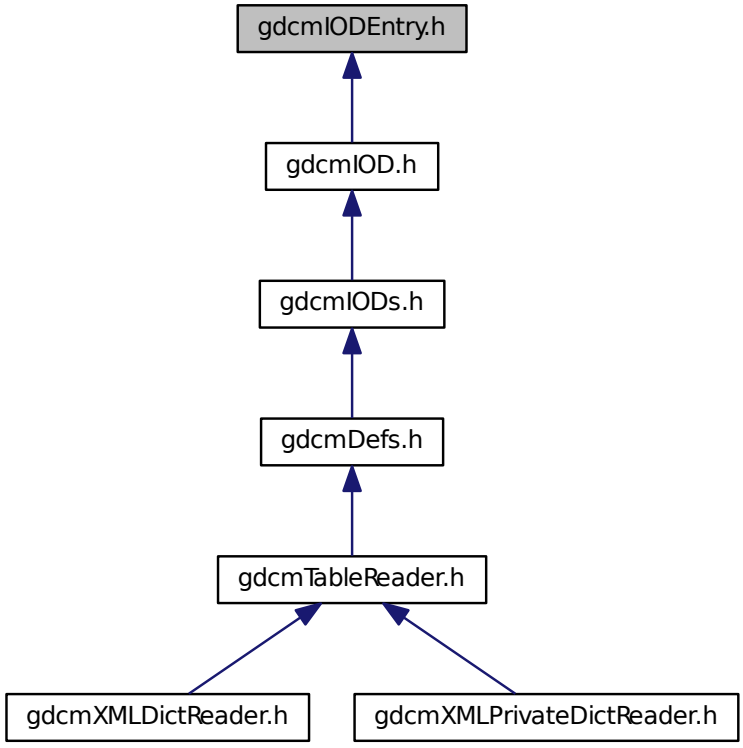
```
#include "gdcmUsage.h"
#include "gdcmType.h"
```

```
#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

- gdcm

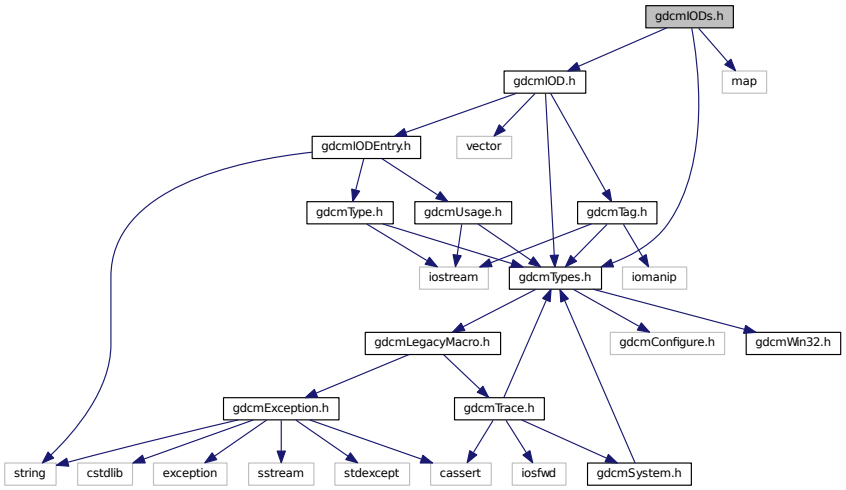
Functions

- std::ostream & gdcm::operator<< (std::ostream &_os, const IODEntry &_val)

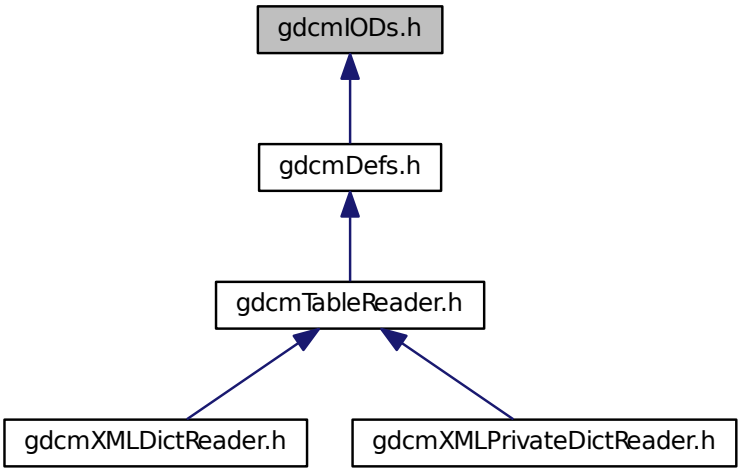
11.123 gdcmIODs.h File Reference

```
#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

- gdcm

Functions

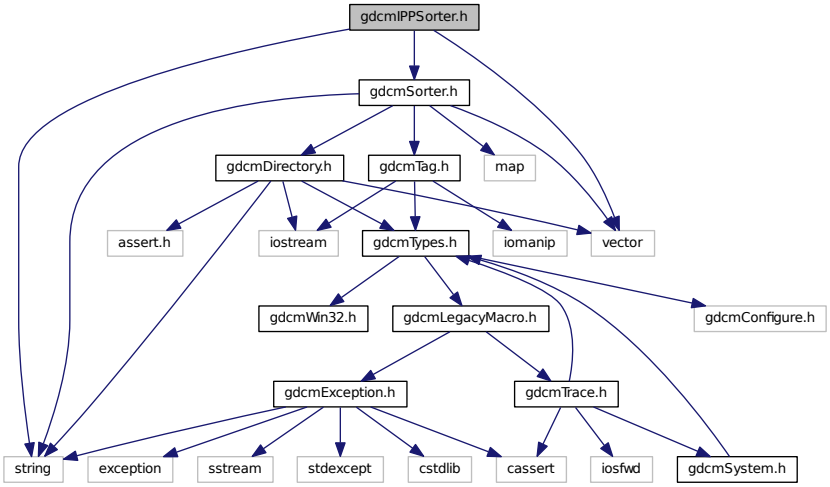
- std::ostream & gdcm::operator<< (std::ostream &_os, const IODs &_val)

11.124 gdcmIPPSorter.h File Reference

```
#include "gdcmSorter.h"
#include <vector>
```

```
#include <string>
```

Include dependency graph for gdcmIPPSorter.h:



Classes

- class gdcm::IPPSorter

  IPPSorter.

Namespaces

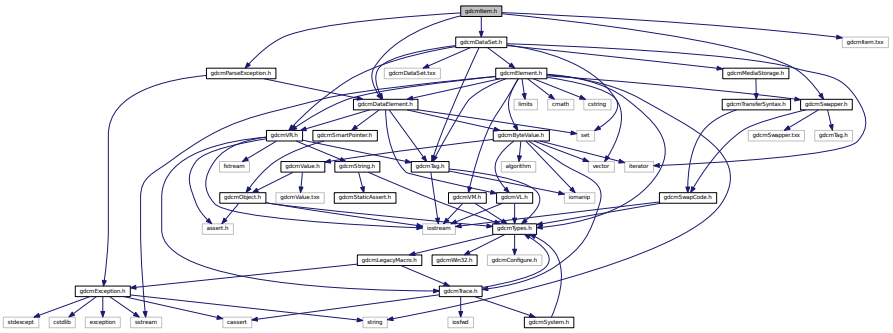
- gdcm

11.125 gdcmItem.h File Reference

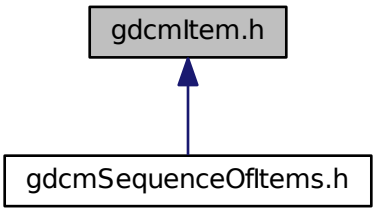
```
#include "gdcmDataElement.h"
#include "gdcmDataSet.h"
#include "gdcmParseException.h"
#include "gdcmSwapper.h"
```

```
#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

- gdcm

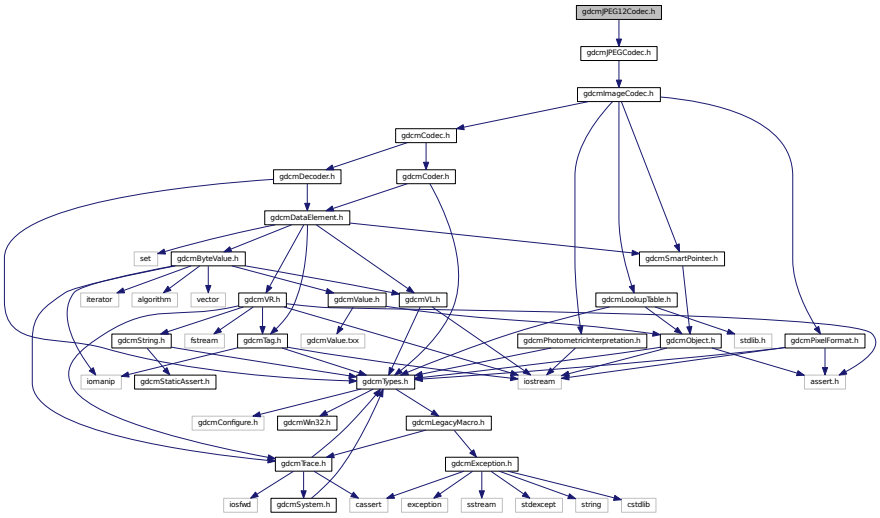
Functions

- std::ostream & gdcm::operator<< (std::ostream &os, const Item &val)

11.126 gdcmJPEG12Codec.h File Reference

```
#include "gdcmJPEGCodec.h"
```

Include dependency graph for gdcmJPEG12Codec.h:



Classes

- class gdcm::JPEG12Codec

  Class to do JPEG 12bits (lossy & lossless)

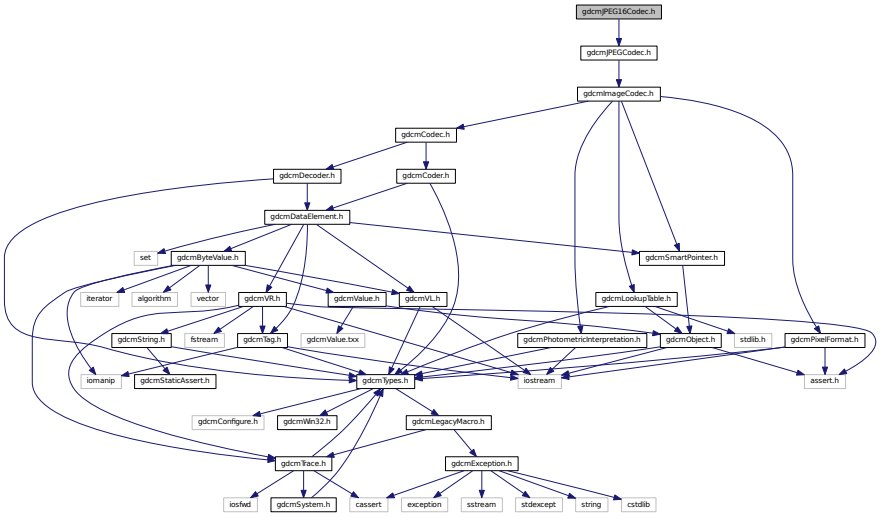
Namespaces

- gdcm

11.127 gdcmJPEG16Codec.h File Reference

```
#include "gdcmJPEGCodec.h"
```

Include dependency graph for gdcmJPEG16Codec.h:



Classes

- class gdcm::JPEG16Codec

  Class to do JPEG 16bits (lossless)

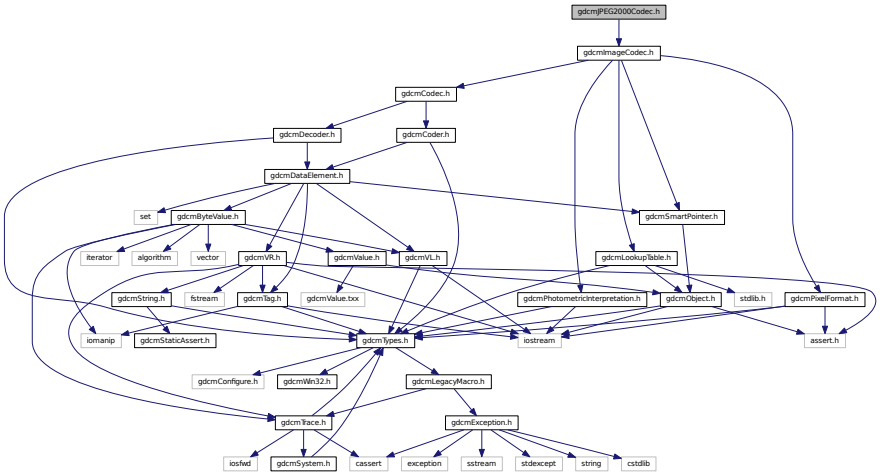
Namespaces

- gdcm

11.128 gdcmJPEG2000Codec.h File Reference

```
#include "gdcmImageCodec.h"
```

Include dependency graph for gdcmJPEG2000Codec.h:



Classes

- class gdcm::JPEG2000Codec

  Class to do JPEG 2000.

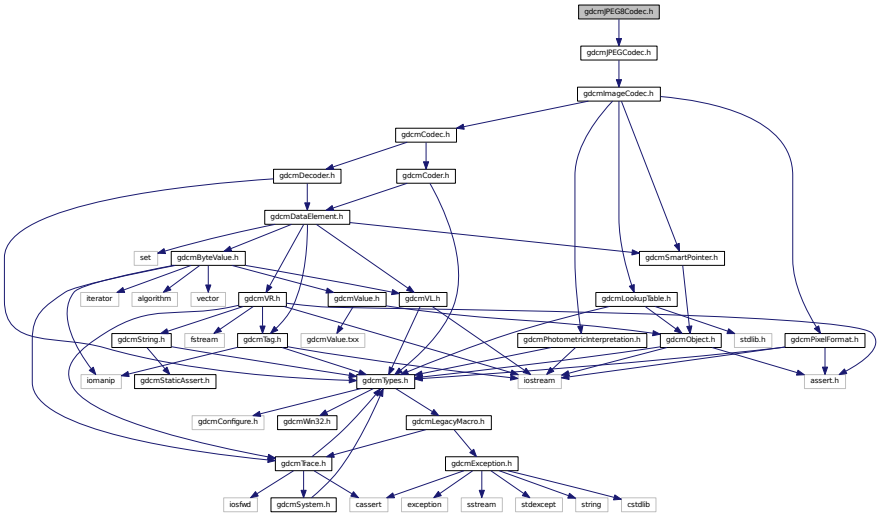
Namespaces

- gdcm

11.129 gdcmJPEG8Codec.h File Reference

```
#include "gdcmJPEGCodec.h"
```

Include dependency graph for gdcmJPEG8Codec.h:



Classes

- class gdcm::JPEG8Codec

  Class to do JPEG 8bits (lossy & lossless)

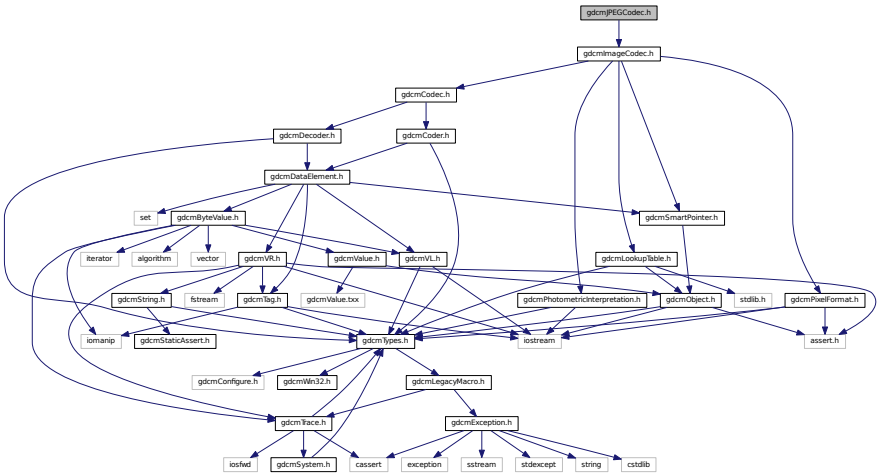
Namespaces

- gdcm

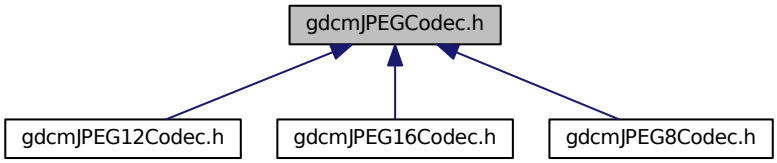
11.130 gdcmJPEGCodec.h File Reference

```
#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.

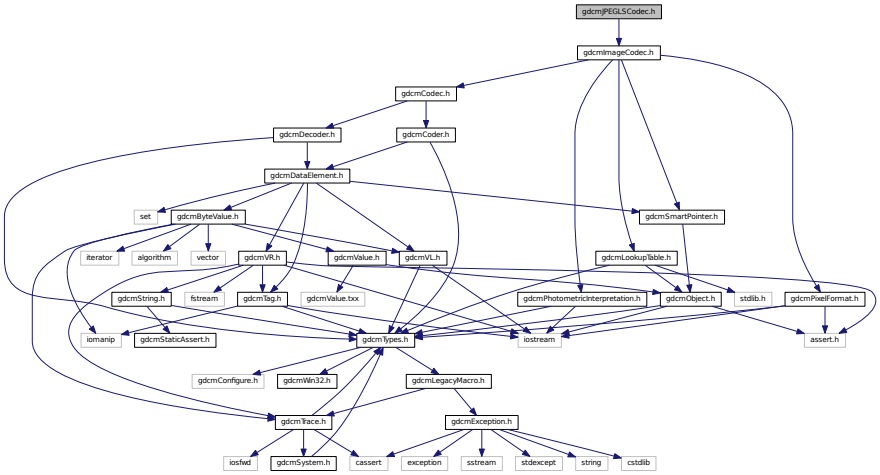
Namespaces

- gdcm

11.131 gdcmJPEGLSCodec.h File Reference

```
#include "gdcmImageCodec.h"
```

Include dependency graph for gdcmJPEGLSCodec.h:



Classes

- class gdcm::JPEGLSCodec

  JPEG-LS.

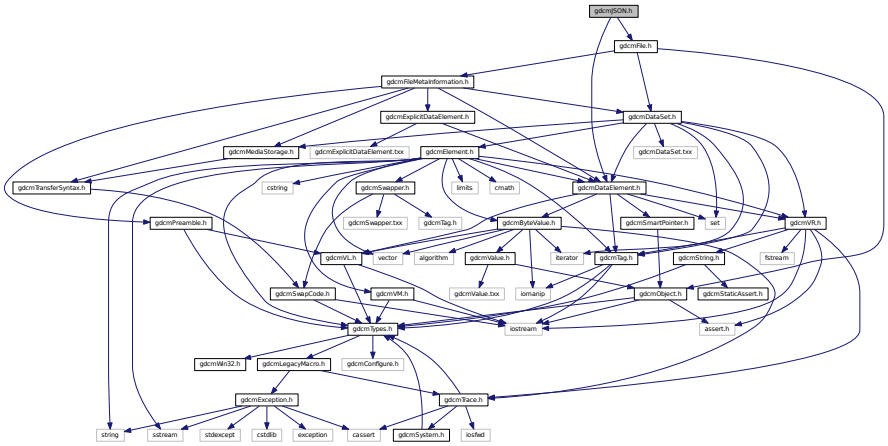
Namespaces

- gdcm

11.132 gdcmJSON.h File Reference

```
#include "gdcmFile.h"
#include "gdcmDataElement.h"
```

Include dependency graph for gdcmJSON.h:



Classes

- class gdcm::JSON

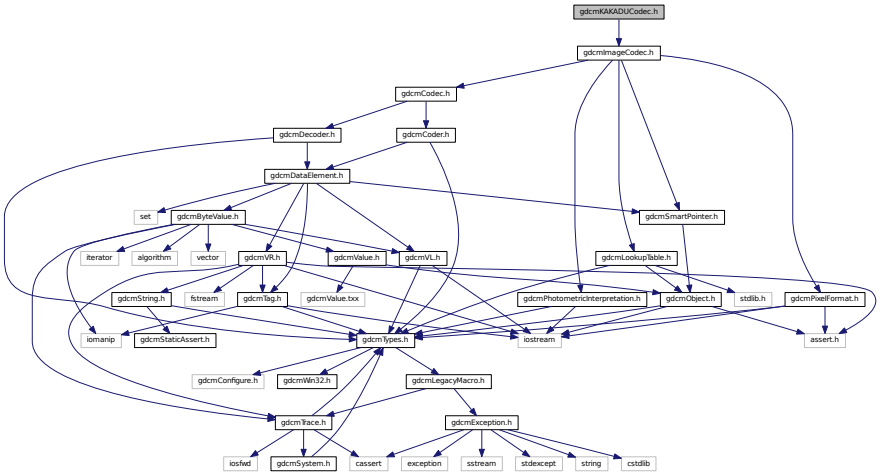
Namespaces

- gdcm

11.133 gdcmKAKADUCodec.h File Reference

```
#include "gdcmImageCodec.h"
```

Include dependency graph for gdcmKAKADUCodec.h:



Classes

- class gdcm::KAKADUCodec

  KAKADUCodec.

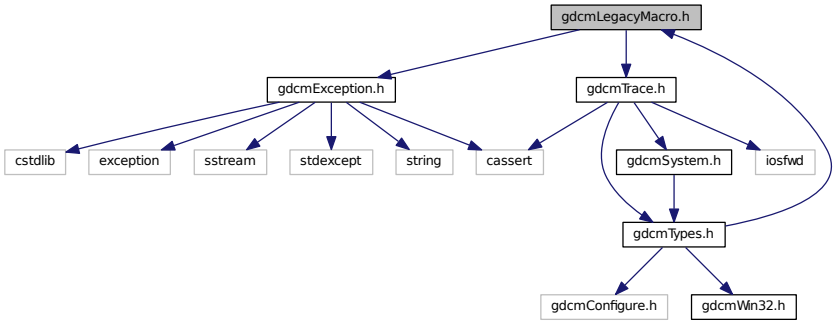
Namespaces

- gdcm

11.134 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.")

11.134.1 Macro Definition Documentation

11.134.1.1 GDCM_LEGACY

```
#define GDCM_LEGACY(
              method ) method;
```

Referenced by gdcm::StringFilter::UseDictAlways().

11.134.1.2 GDCM_LEGACY_BODY

```
#define GDCM_LEGACY_BODY(
              method,
              version ) gdcmWarningMacro(#method " was deprecated for " version " and will be
removed in a future version.")
```

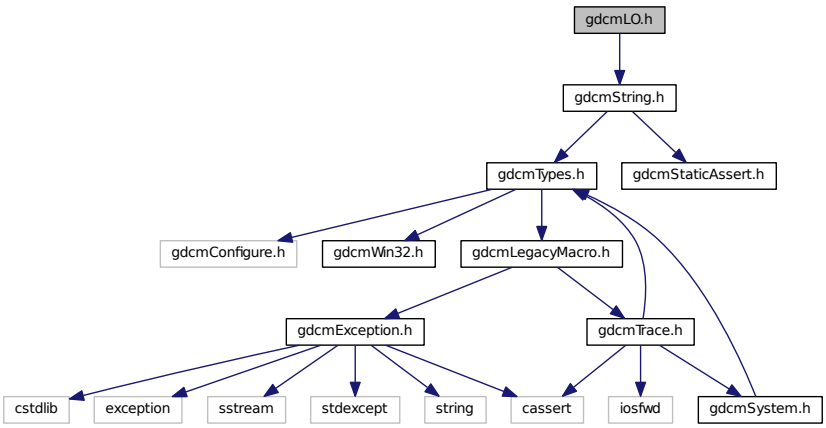
11.134.1.3 GDCM_LEGACY_REPLACED_BODY

```
#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.")
```

11.135 gdcmLO.h File Reference

```
#include "gdcmString.h"
```

Include dependency graph for gdcmLO.h:



Classes

- class gdcm::LO

  LO.

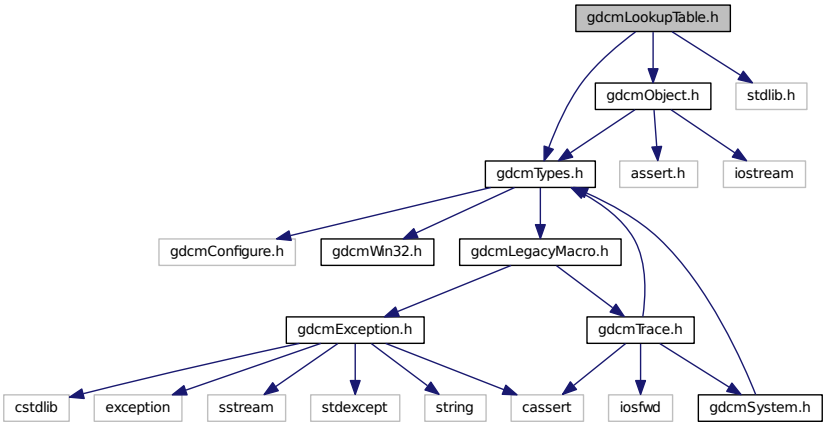
Namespaces

- gdcm

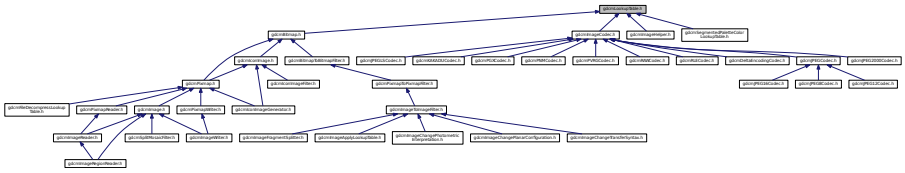
11.136 gdcmLookupTable.h File Reference

```
#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.

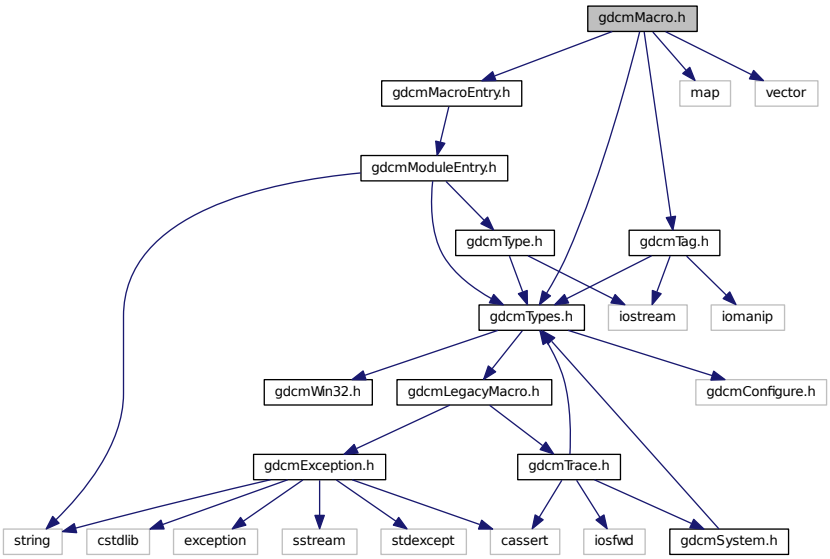
Namespaces

- gdcm

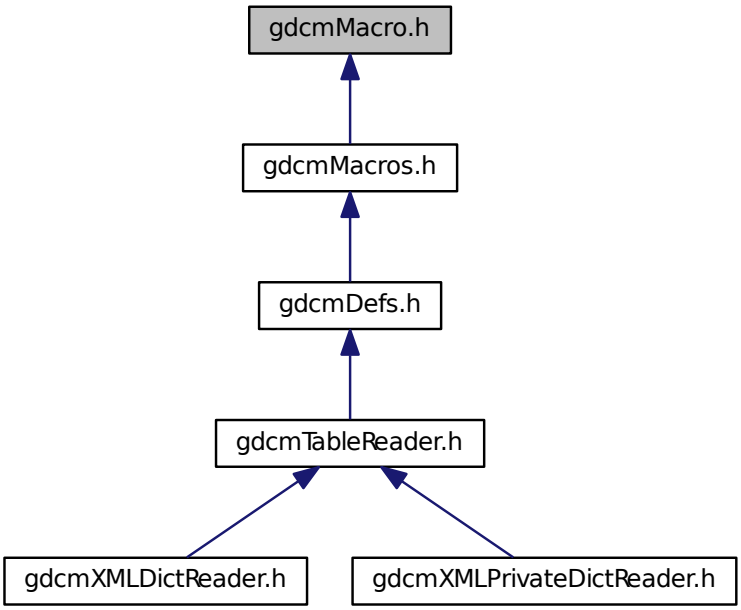
11.137 gdcmMacro.h File Reference

```
#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

- gdcm

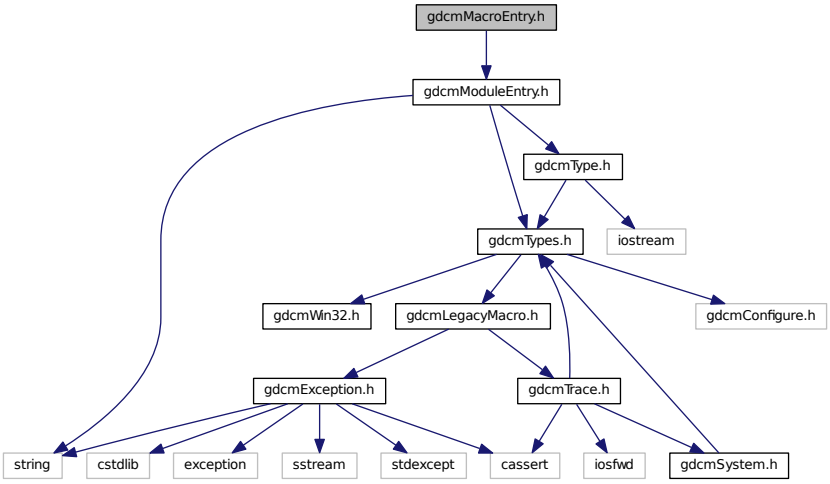
Functions

- std::ostream & gdcm::operator<< (std::ostream &_os, const Macro &_val)

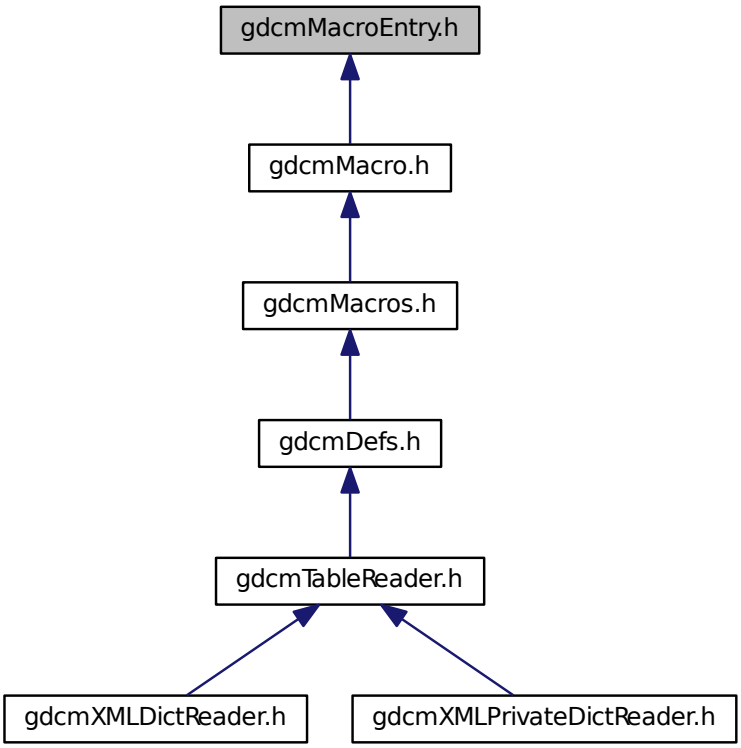
11.138 gdcmMacroEntry.h File Reference

```
#include "gdcmModuleEntry.h"
```

Include dependency graph for gdcmMacroEntry.h:



This graph shows which files directly or indirectly include this file:



Macros

- #define GDCMMACROENTRY_H

11.138.1 Macro Definition Documentation

11.138.1.1 GDCMMACROENTRY_H

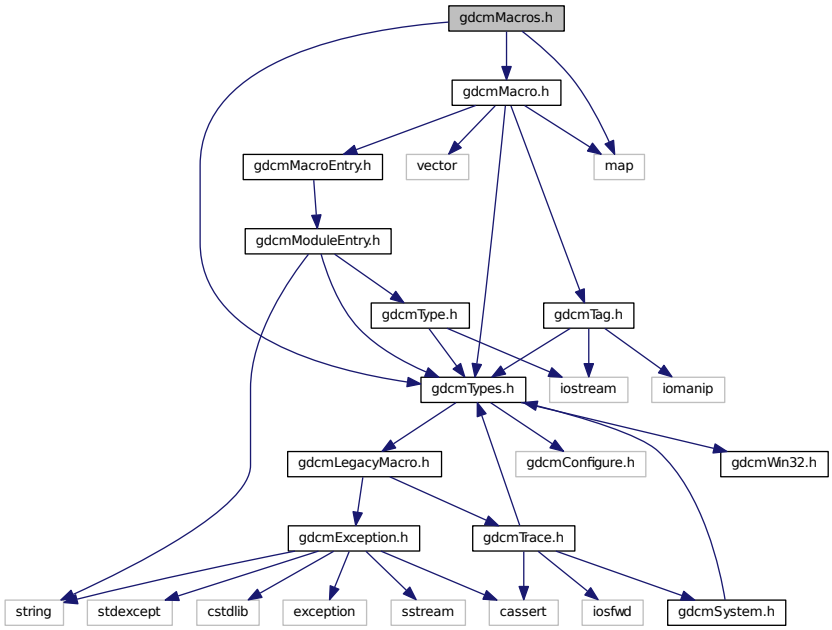
```
#define GDCMMACROENTRY_H
```

11.139 gdcmMacros.h File Reference

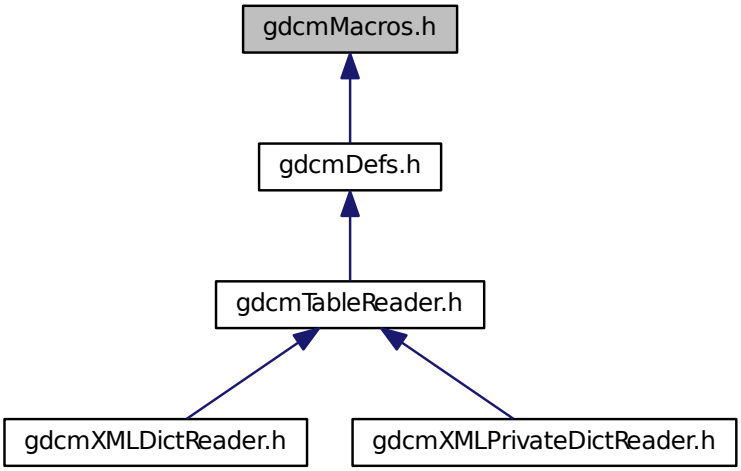
```
#include "gdcmTypes.h"
#include "gdcmMacro.h"
```

```
#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

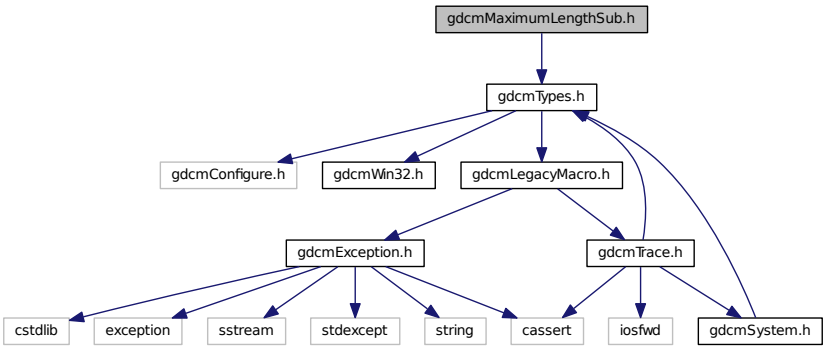
- gdcm

Functions

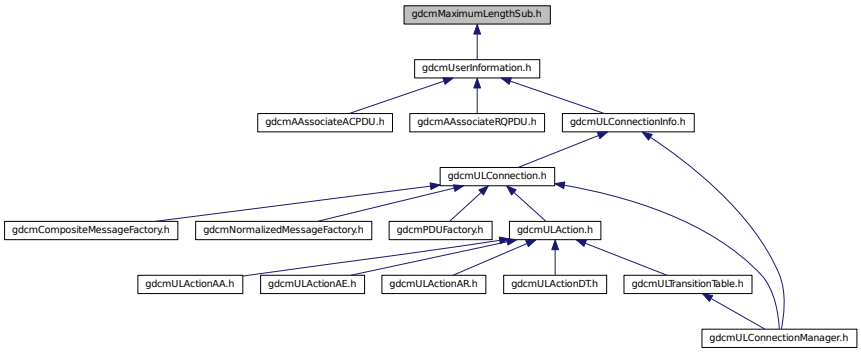
- std::ostream & gdcm::operator<< (std::ostream &_os, const Macros &_val)

11.140 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.

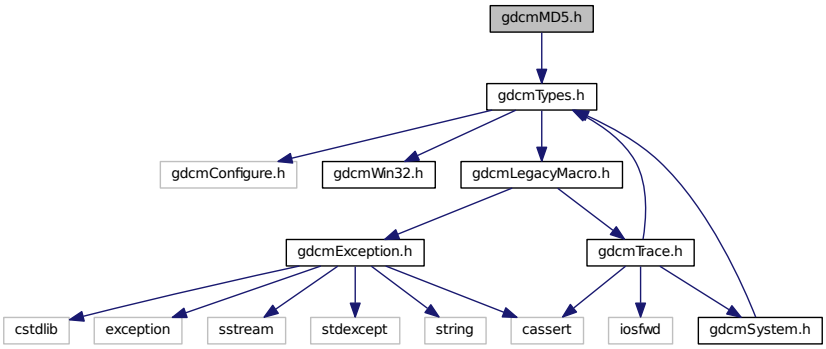
Namespaces

- gdcm
- gdcm::network

11.141 gdcmMD5.h File Reference

```
#include "gdcmTypes.h"
```

Include dependency graph for gdcmMD5.h:



Classes

- class gdcm::MD5

  Class for MD5.

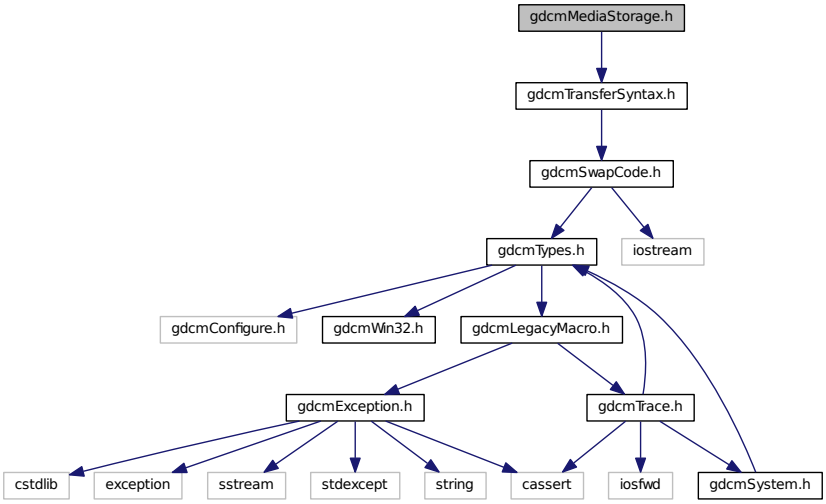
Namespaces

- gdcm

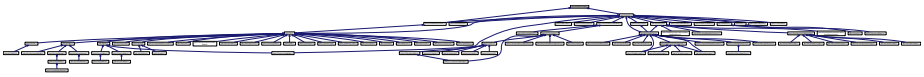
11.142 gdcmMediaStorage.h File Reference

```
#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

- gdcm

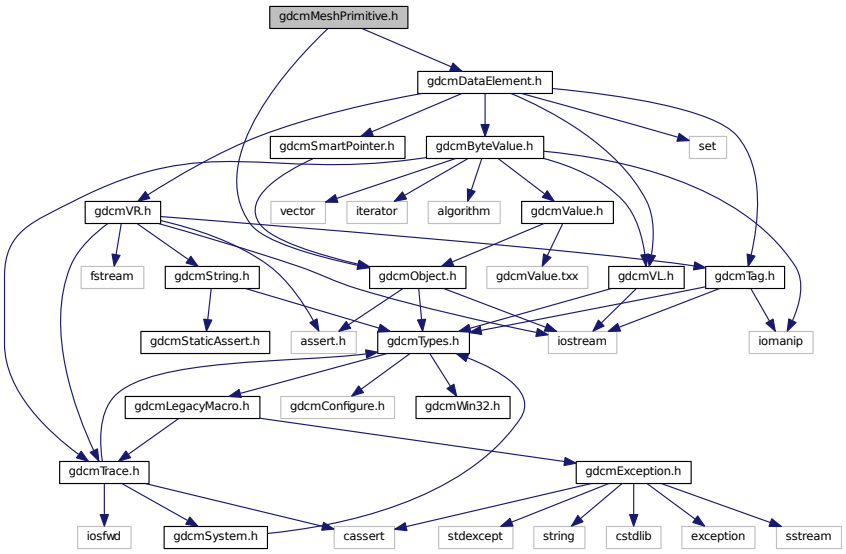
Functions

- std::ostream & gdcm::operator<< (std::ostream &_os, const MediaStorage &ms)

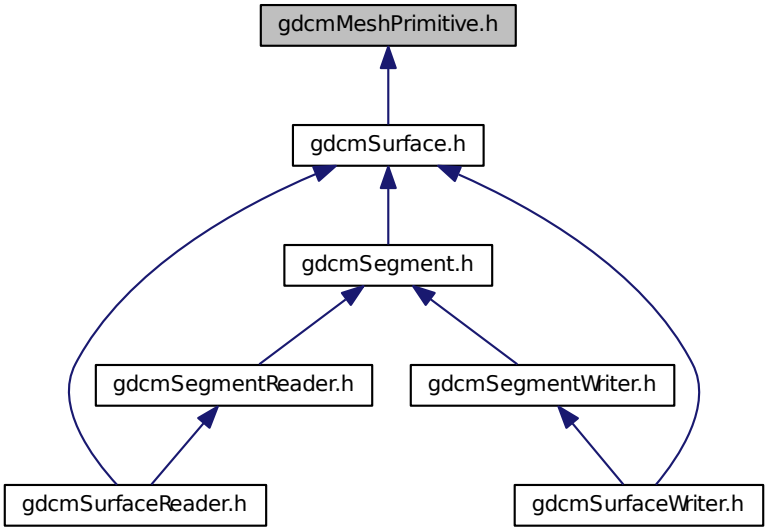
11.143 gdcmMeshPrimitive.h File Reference

```
#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.

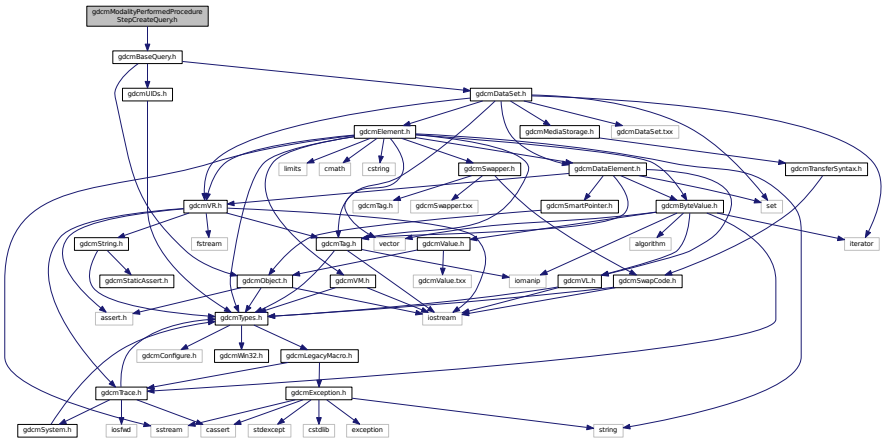
Namespaces

- gdcm

11.144 gdcmModalityPerformedProcedureStepCreateQuery.h File Reference

```
#include "gdcmBaseQuery.h"
```

Include dependency graph for gdcmModalityPerformedProcedureStepCreateQuery.h:



Classes

- class gdcm::ModalityPerformedProcedureStepCreateQuery

  ModalityPerformedProcedureStepCreateQuery.

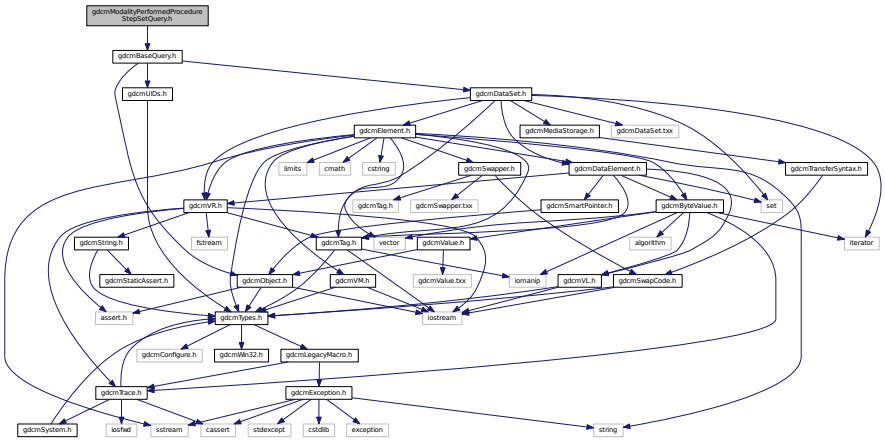
Namespaces

- gdcm

11.145 gdcmModalityPerformedProcedureStepSetQuery.h File Reference

```
#include "gdcmBaseQuery.h"
```

Include dependency graph for gdcmModalityPerformedProcedureStepSetQuery.h:



Classes

- class gdcm::ModalityPerformedProcedureStepSetQuery

  ModalityPerformedProcedureStepSetQuery.

Namespaces

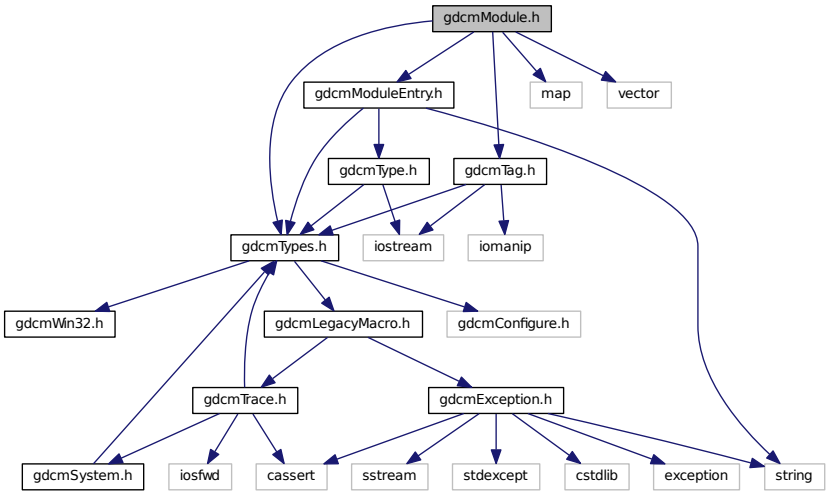
- gdcm

11.146 gdcmModule.h File Reference

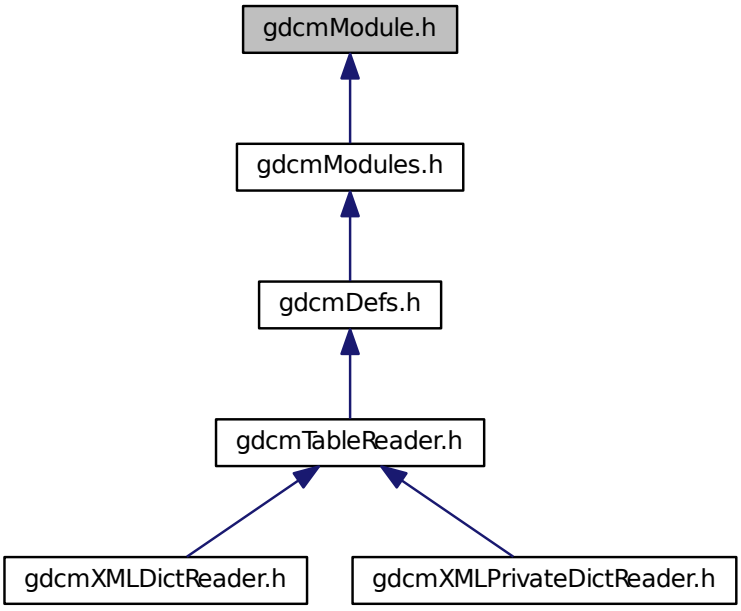
```
#include "gdcmTypes.h"
#include "gdcmTag.h"
#include "gdcmModuleEntry.h"
#include <map>
```

```
#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

- gdcm

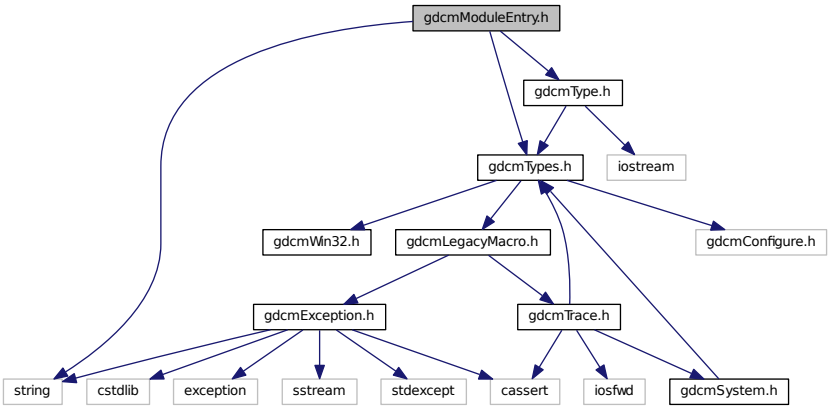
Functions

- std::ostream & gdcm::operator<< (std::ostream &_os, const Module &_val)

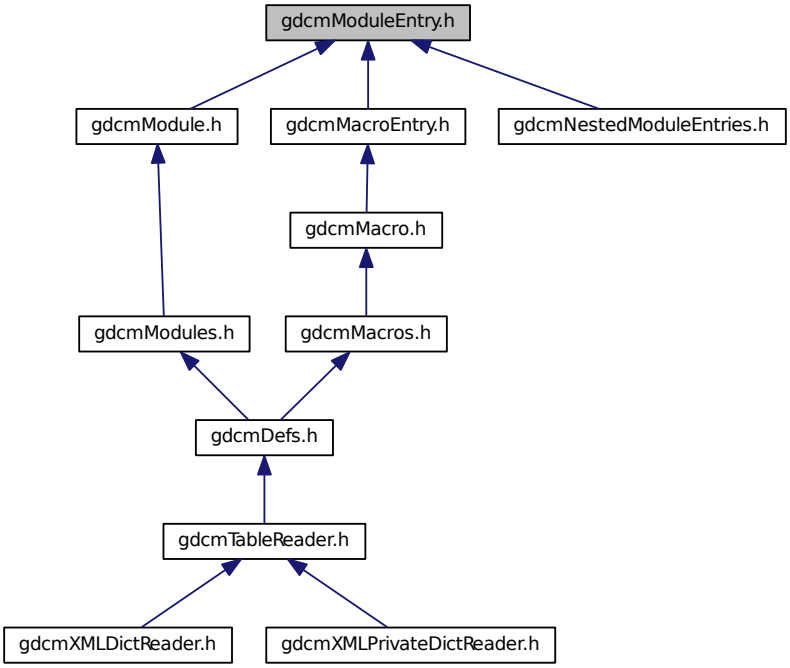
11.147 gdcmModuleEntry.h File Reference

```
#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

- gdcm

Typedefs

- typedef ModuleEntry gdcm::MacroEntry

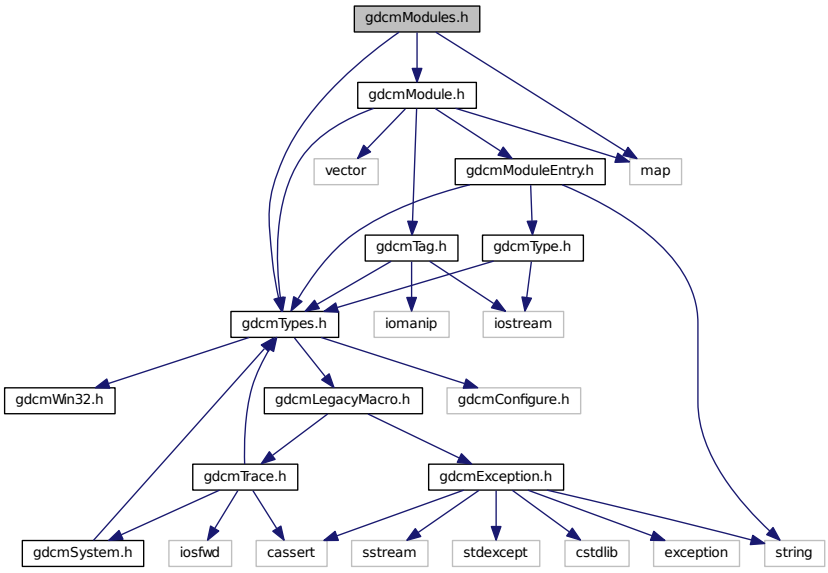
Functions

- std::ostream & gdcm::operator<< (std::ostream &_os, const ModuleEntry &_val)

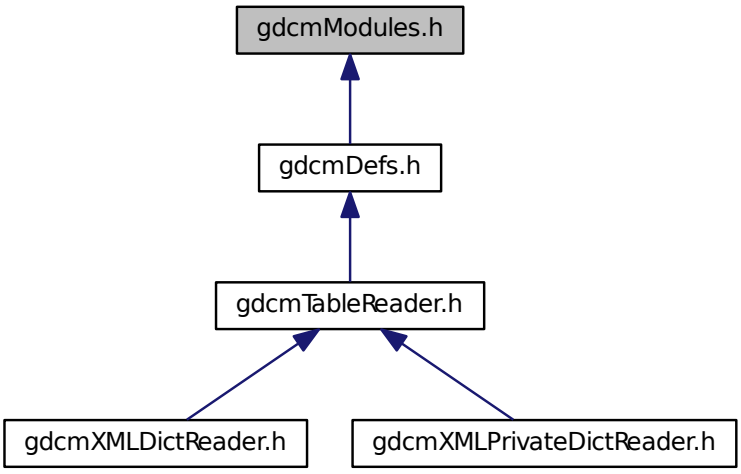
11.148 gdcmModules.h File Reference

```
#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

- gdcm

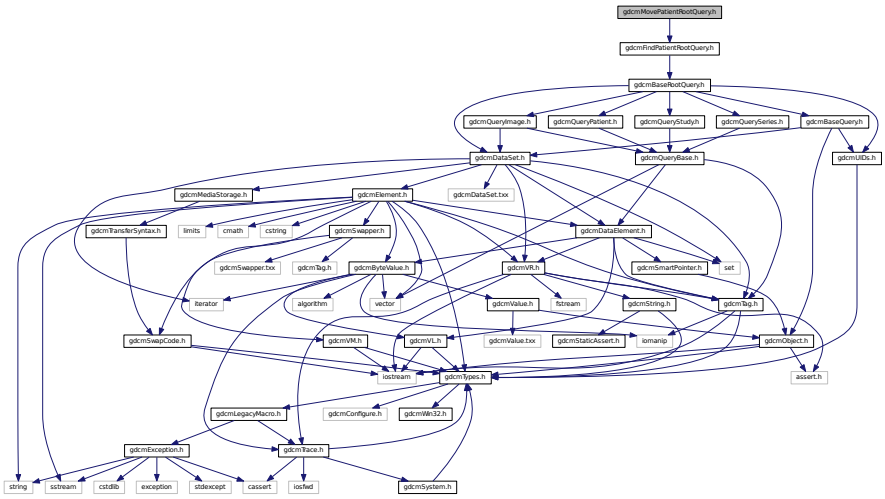
Functions

- std::ostream & gdcm::operator<< (std::ostream &_os, const Modules &_val)

11.149 gdcmMovePatientRootQuery.h File Reference

```
#include "gdcmFindPatientRootQuery.h"
```

Include dependency graph for gdcmMovePatientRootQuery.h:



Classes

- class gdcm::MovePatientRootQuery

  MovePatientRootQuery.

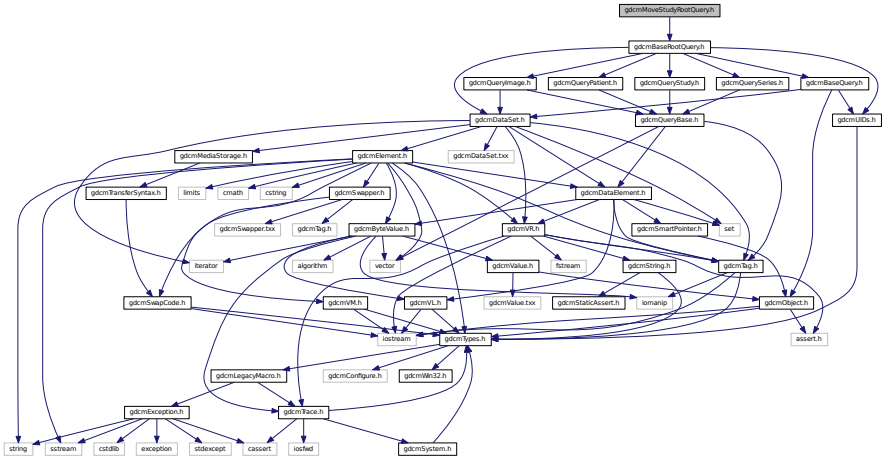
Namespaces

- gdcm

11.150 gdcmMoveStudyRootQuery.h File Reference

```
#include "gdcmBaseRootQuery.h"
```

Include dependency graph for gdcmMoveStudyRootQuery.h:



Classes

- class gdcm::MoveStudyRootQuery

  MoveStudyRootQuery.

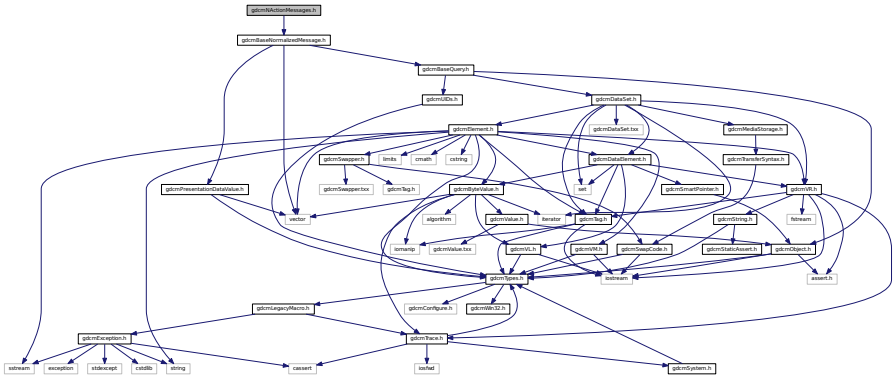
Namespaces

- gdcm

11.151 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.

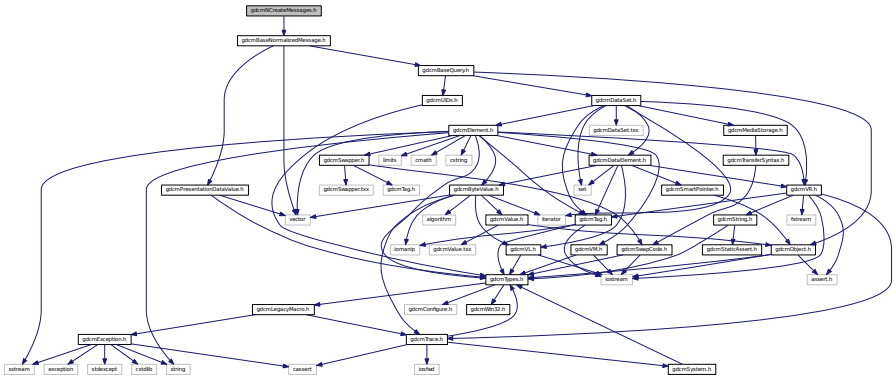
Namespaces

- gdcm
- gdcm::network

11.152 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.

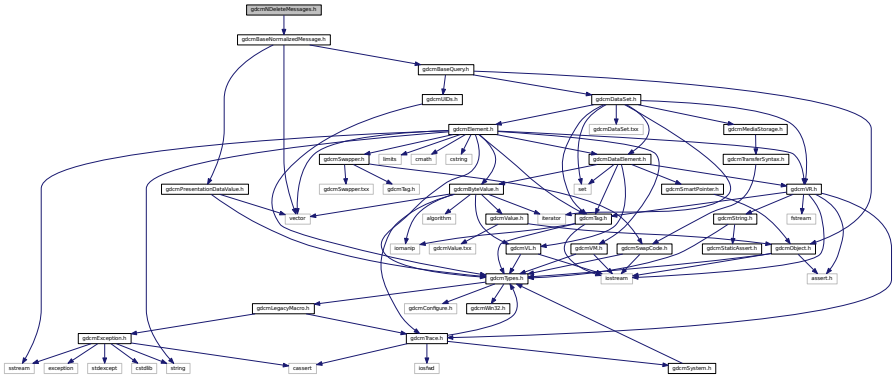
Namespaces

- gdcm
- gdcm::network

11.153 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.

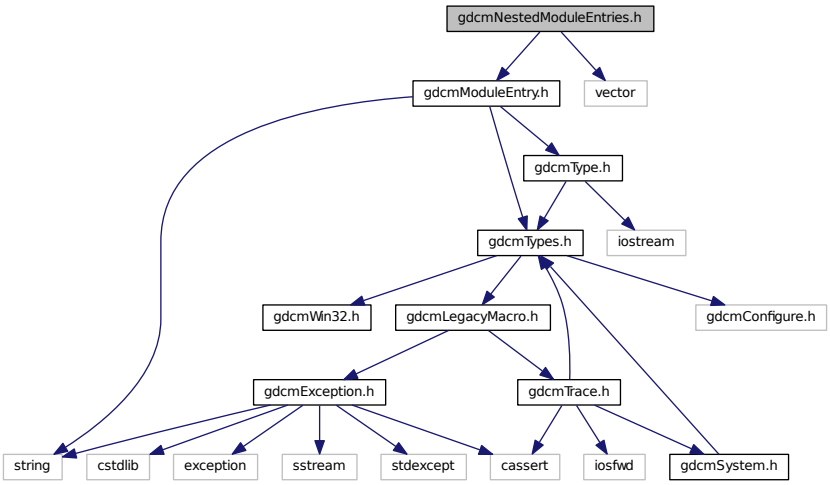
Namespaces

- gdcm
- gdcm::network

11.154 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

- gdcm

Typedefs

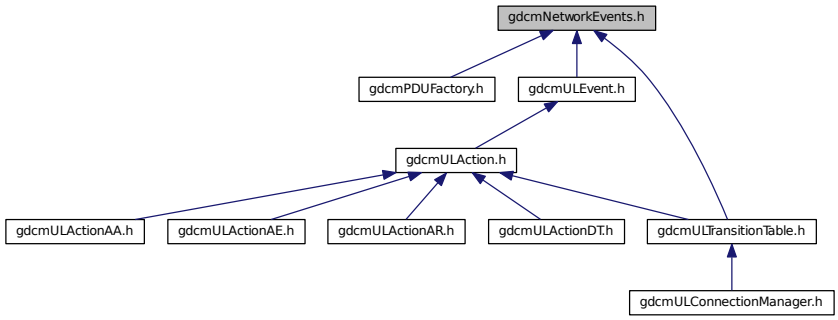
- typedef NestedModuleEntries gdcm::NestedMacroEntries

Functions

- std::ostream & gdcm::operator<< (std::ostream &_os, const NestedModuleEntries &_val)

11.155 gdcmNetworkEvents.h File Reference

This graph shows which files directly or indirectly include this file:



Namespaces

- gdcm
- gdcm::network

Enumerations

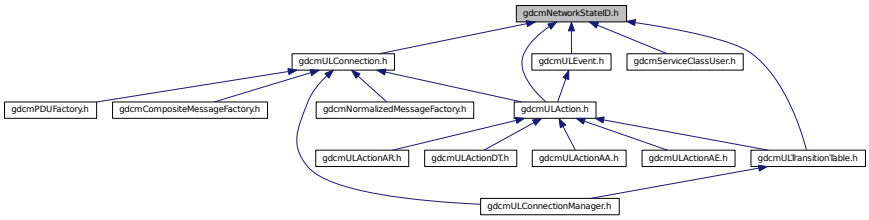
- enum gdcm::network::EEventID {
  gdcm::network::eAASSOCIATERequestLocalUser = 0,
  gdcm::network::eTransportConnConfirmLocal,
  gdcm::network::eASSOCIATE_ACPDUreceived,
  gdcm::network::eASSOCIATE_RJPDUreceived,
  gdcm::network::eTransportConnIndicLocal,
  gdcm::network::eAASSOCIATE_RQPDUreceived,
  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

11.156 gdcmNetworkStateID.h File Reference

This graph shows which files directly or indirectly include this file:



Namespaces

- gdcm
- 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)

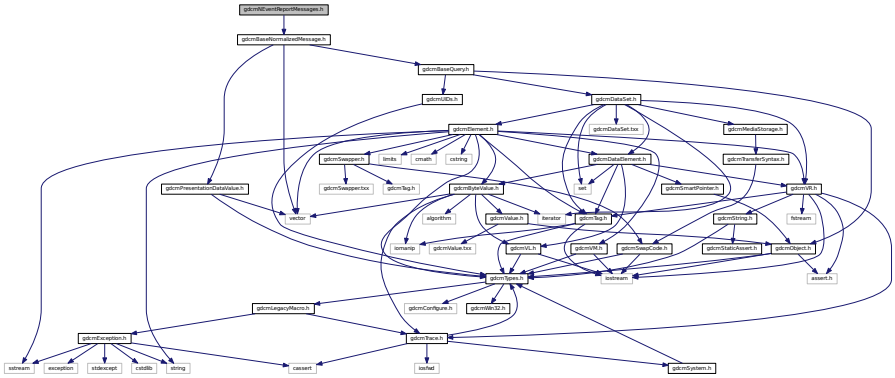
Variables

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

11.157 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.

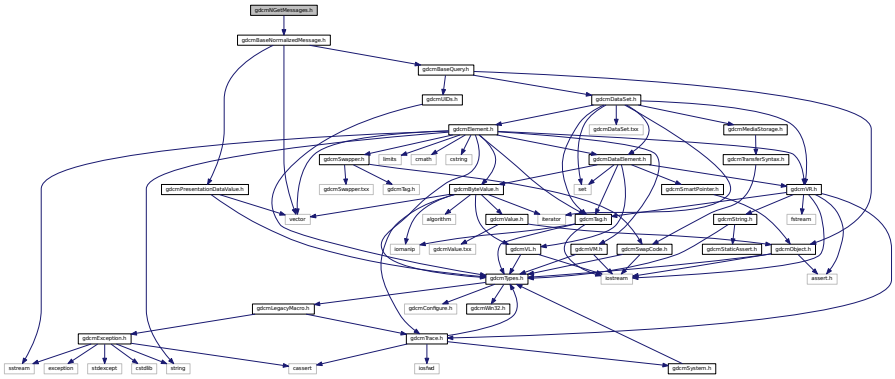
Namespaces

- gdcm
- gdcm::network

11.158 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.

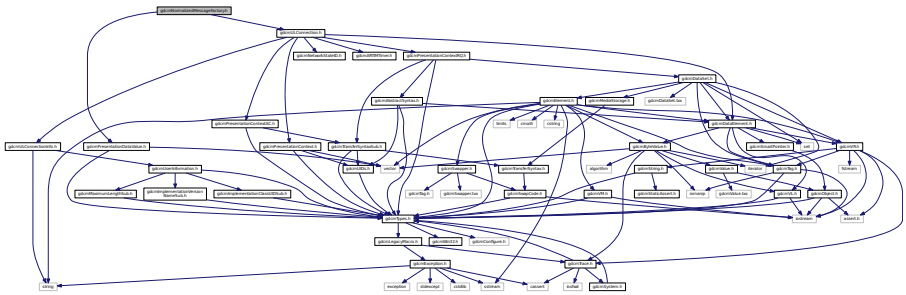
Namespaces

- gdcm
- gdcm::network

11.159 gdcmNormalizedMessageFactory.h File Reference

```
#include "gdcmPresentationDataValue.h"
#include "gdcmULConnection.h"
```

Include dependency graph for gdcmNormalizedMessageFactory.h:



Classes

- class gdcm::network::NormalizedMessageFactory

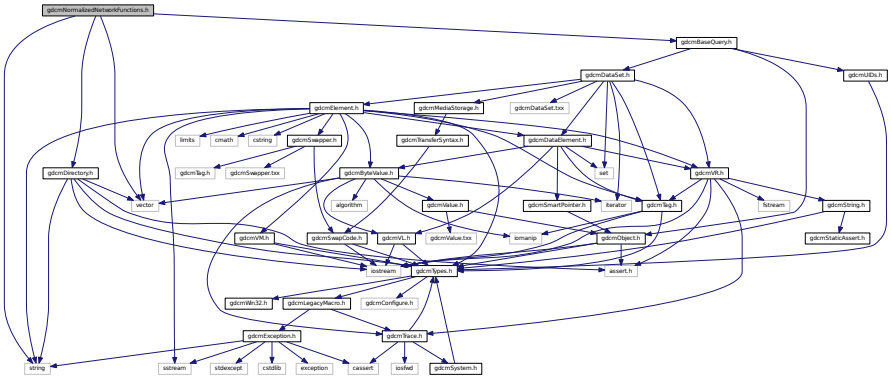
Namespaces

- gdcm
- gdcm::network

11.160 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.

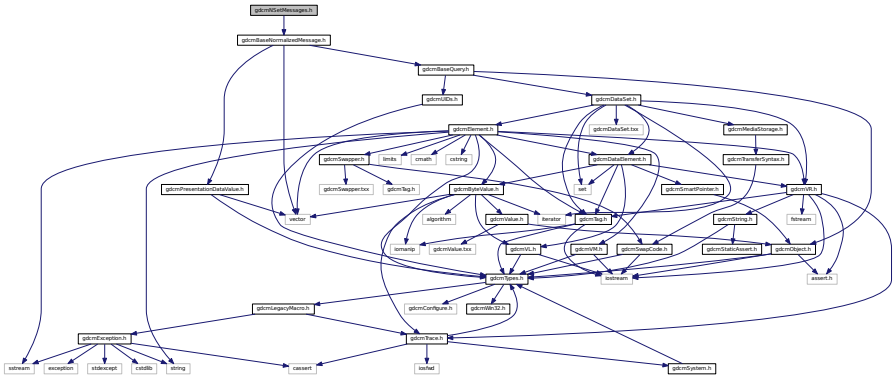
Namespaces

- gdcm

11.161 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.

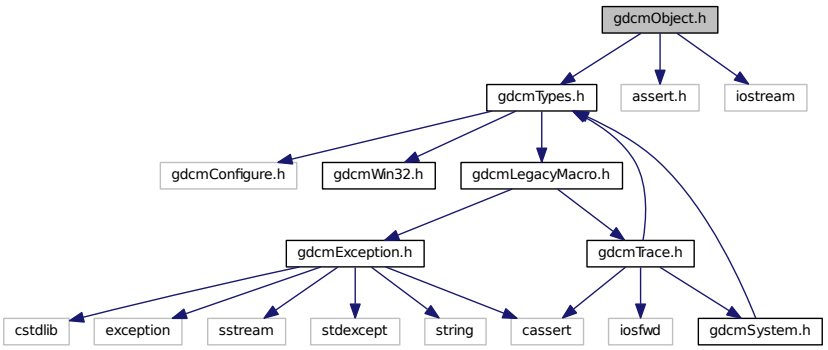
Namespaces

- gdcm
- gdcm::network

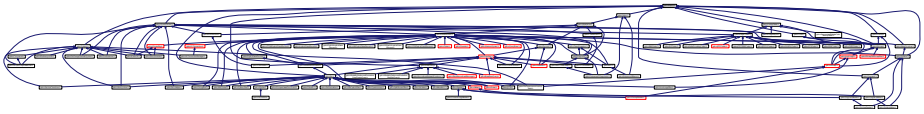
11.162 gdcmObject.h File Reference

```
#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.
- class gdcm::SmartPointer< ObjectType >

  Class for Smart Pointer.

Namespaces

- gdcm

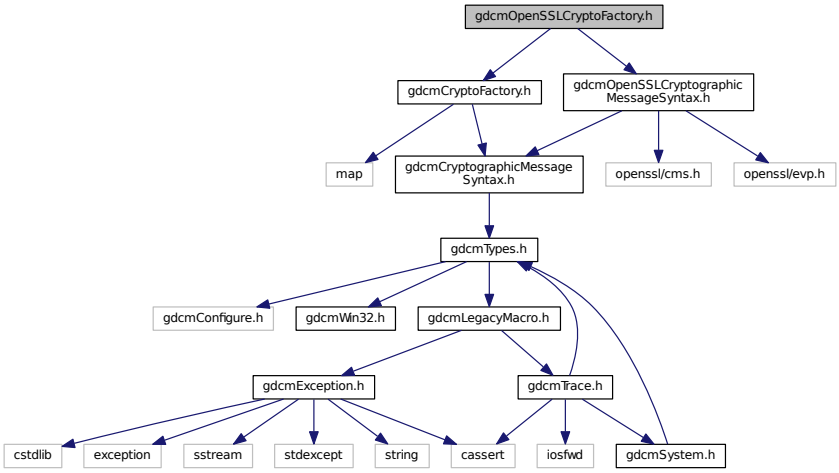
Functions

- std::ostream & gdcm::operator<< (std::ostream &os, const Object &obj)

11.163 gdcmOpenSSLCryptoFactory.h File Reference

```
#include "gdcmCryptoFactory.h"
#include "gdcmOpenSSLCryptographicMessageSyntax.h"
```

Include dependency graph for gdcmOpenSSLCryptoFactory.h:



Classes

- class gdcm::OpenSSLCryptoFactory

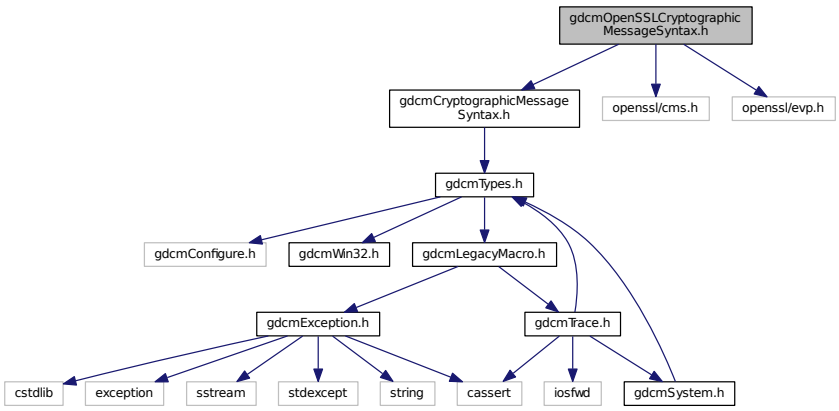
Namespaces

- gdcm

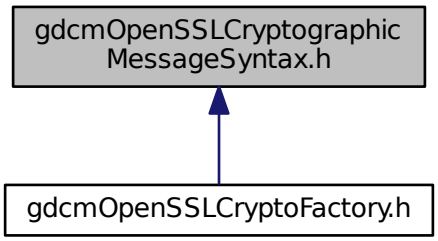
11.164 gdcmOpenSSLCryptographicMessageSyntax.h File Reference

```
#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

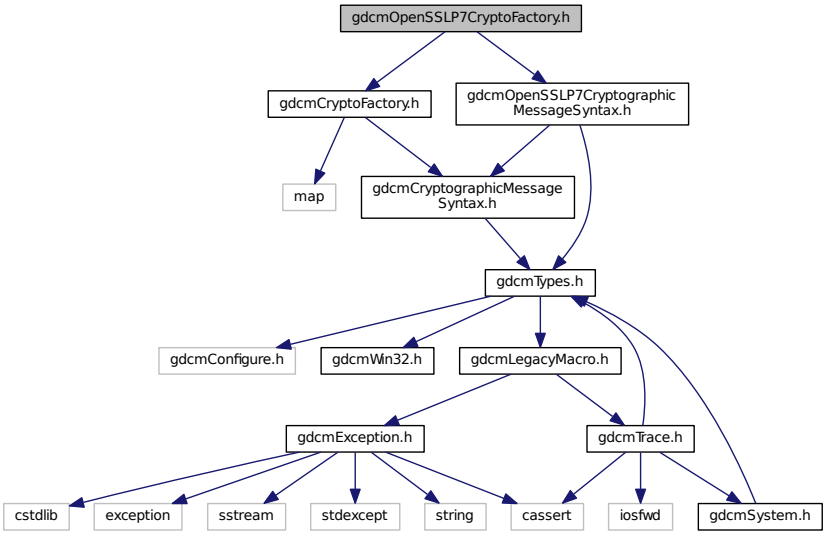
Namespaces

- gdcm

11.165 gdcmOpenSSLP7CryptoFactory.h File Reference

```
#include "gdcmCryptoFactory.h"
#include "gdcmOpenSSLP7CryptographicMessageSyntax.h"
```

Include dependency graph for gdcmOpenSSLP7CryptoFactory.h:



Classes

- class gdcm::OpenSSLP7CryptoFactory

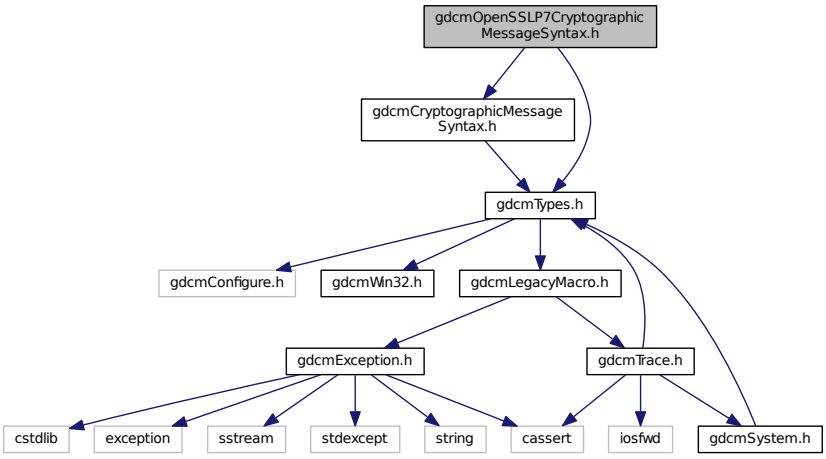
Namespaces

- gdcm

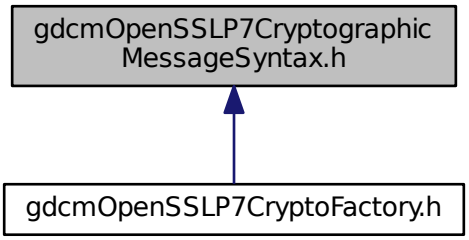
11.166 gdcmOpenSSLP7CryptographicMessageSyntax.h File Reference

```
#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

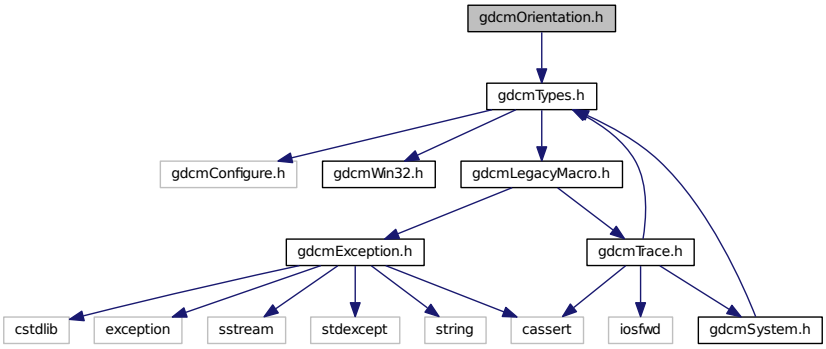
Namespaces

- gdcm

11.167 gdcmOrientation.h File Reference

```
#include "gdcmTypes.h"
```

Include dependency graph for gdcmOrientation.h:



Classes

- class gdcm::Orientation

  class to handle Orientation

Namespaces

- gdcm

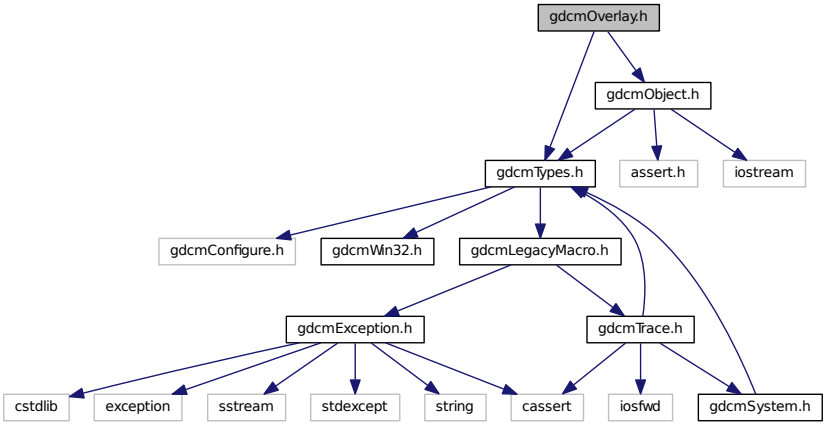
Functions

- std::ostream & gdcm::operator<< (std::ostream &os, const Orientation &o)

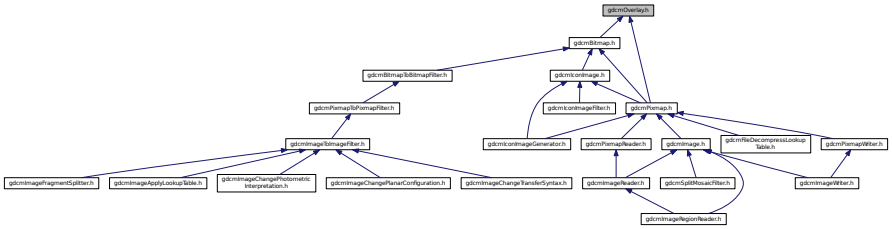
11.168 gdcmOverlay.h File Reference

```
#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.

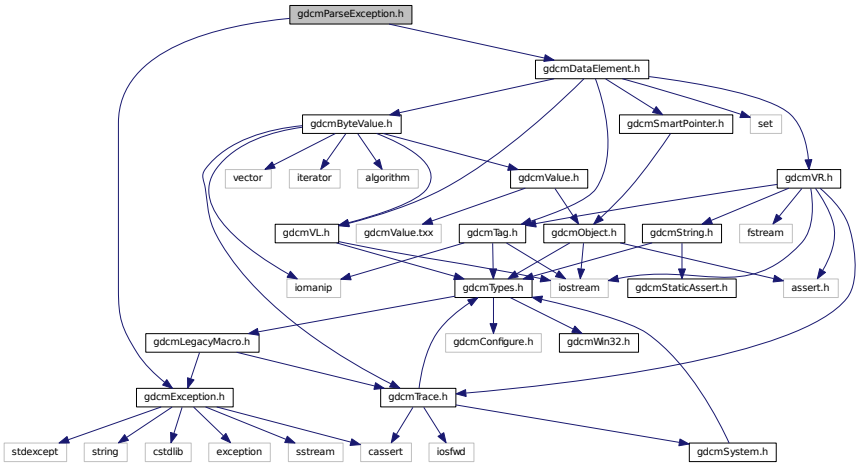
Namespaces

- gdcm

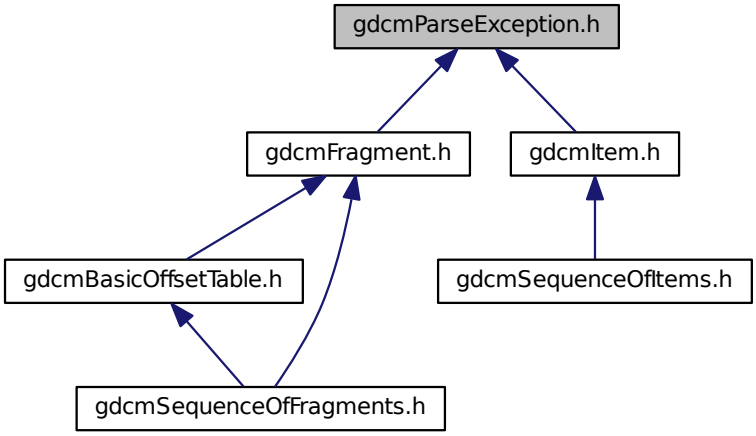
11.169 gdcmParseException.h File Reference

```
#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.*

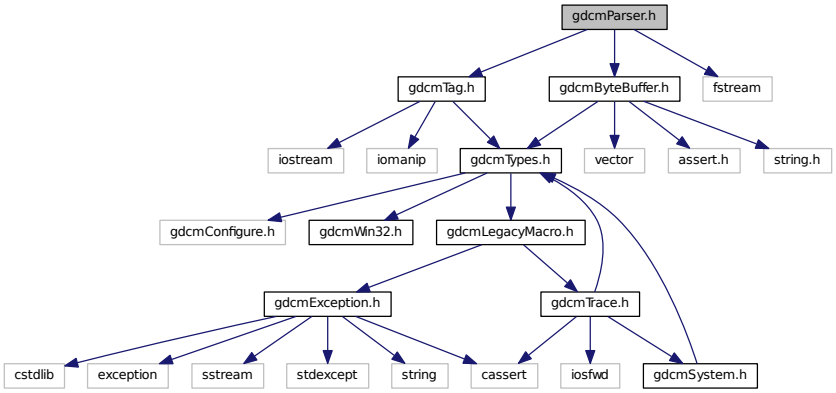
Namespaces

- gdcm

11.170 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)

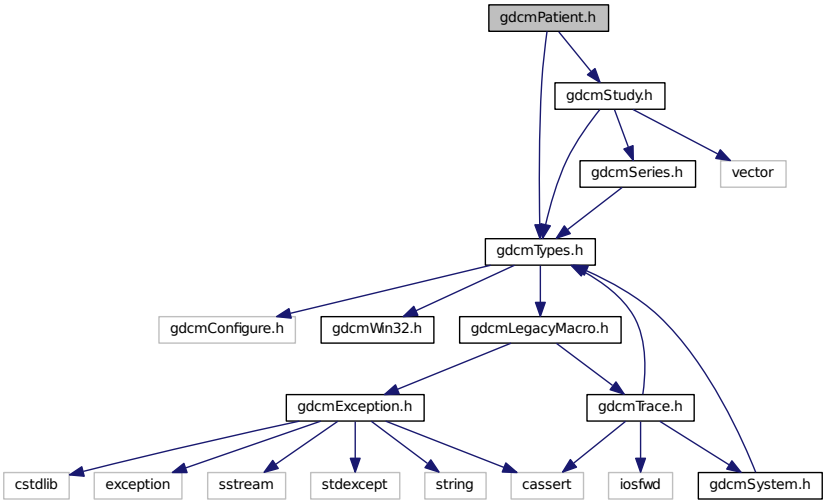
Namespaces

- gdcm

11.171 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.

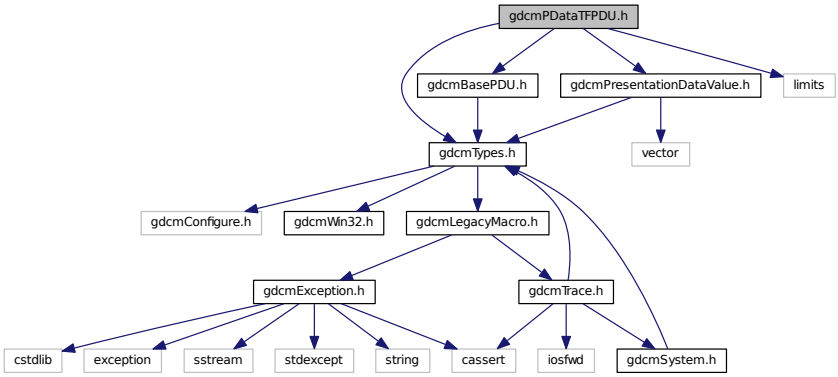
Namespaces

- gdcm

11.172 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.

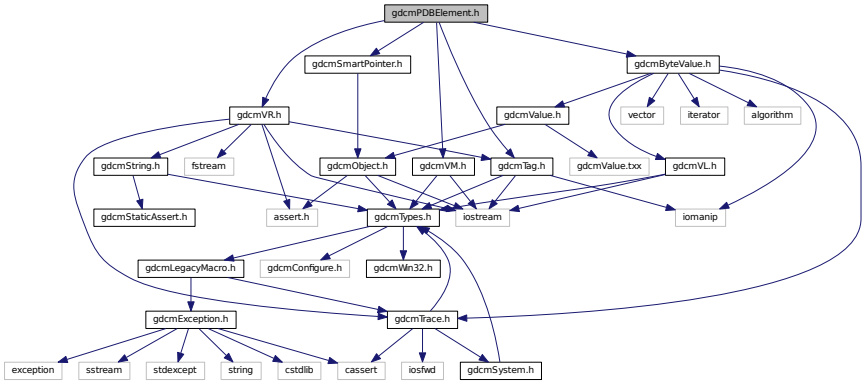
Namespaces

- gdcm
- gdcm::network

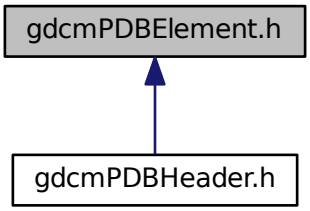
11.173 gdcmPDBElement.h File Reference

```
#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

- gdcm

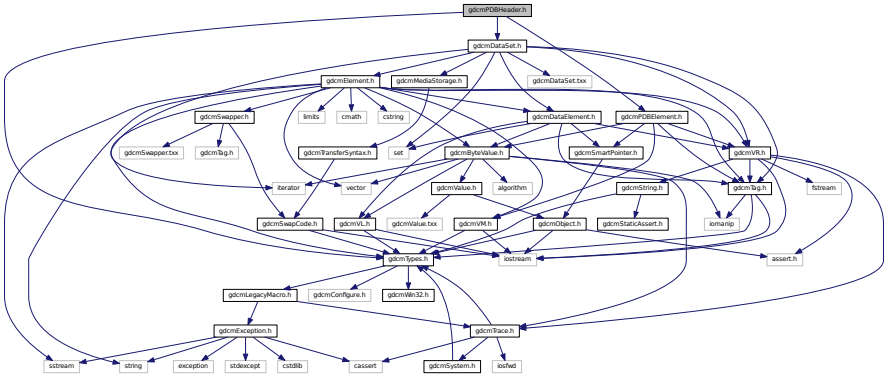
Functions

- std::ostream & gdcm::operator<< (std::ostream &os, const PDBElement &val)

11.174 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

- gdcm

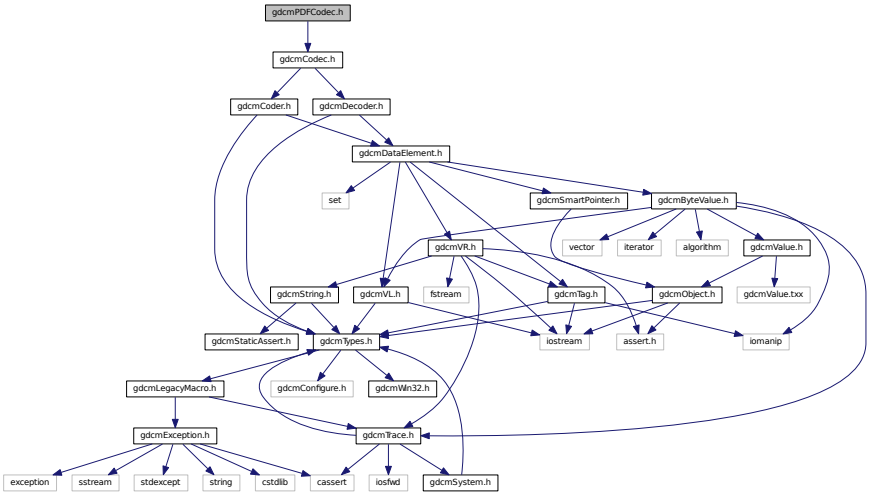
Functions

- std::ostream & gdcm::operator<< (std::ostream &os, const PDBHeader &d)

11.175 gdcmPDFCodec.h File Reference

```
#include "gdcmCodec.h"
```

Include dependency graph for gdcmPDFCodec.h:



Classes

- class gdcm::PDFCodec

  PDFCodec class.

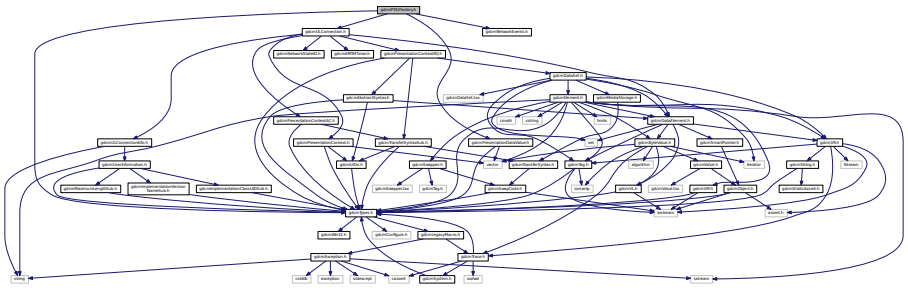
Namespaces

- gdcm

11.176 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.

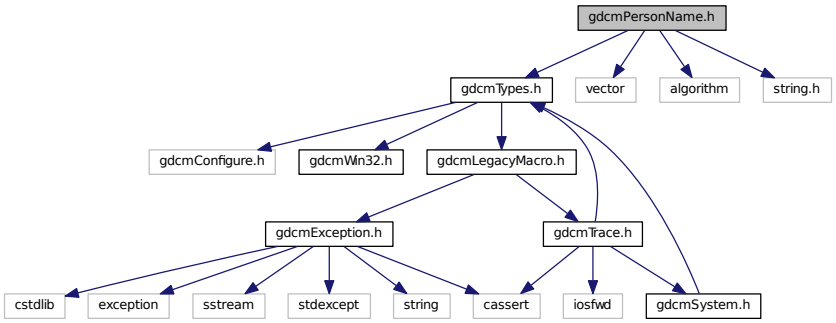
Namespaces

- gdcm
- gdcm::network

11.177 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.

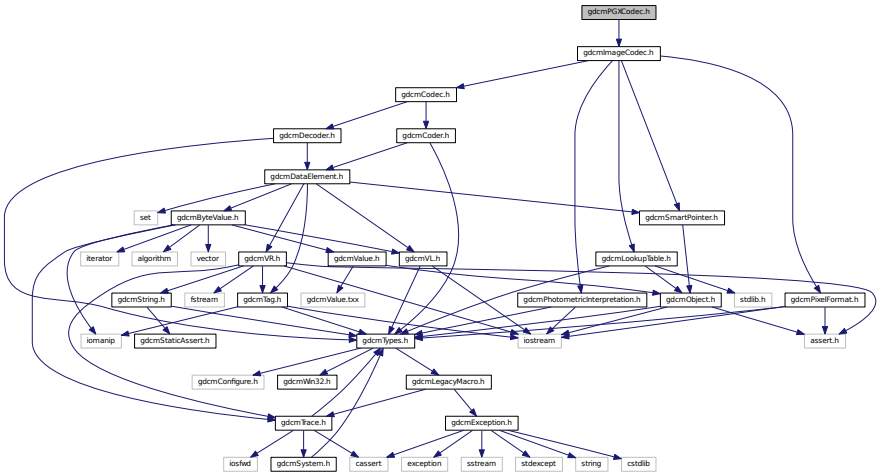
Namespaces

- gdcm

11.178 gdcmPGXCodec.h File Reference

```
#include "gdcmImageCodec.h"
```

Include dependency graph for gdcmPGXCodec.h:



Classes

- class gdcm::PGXCodec

  Class to do PGX.

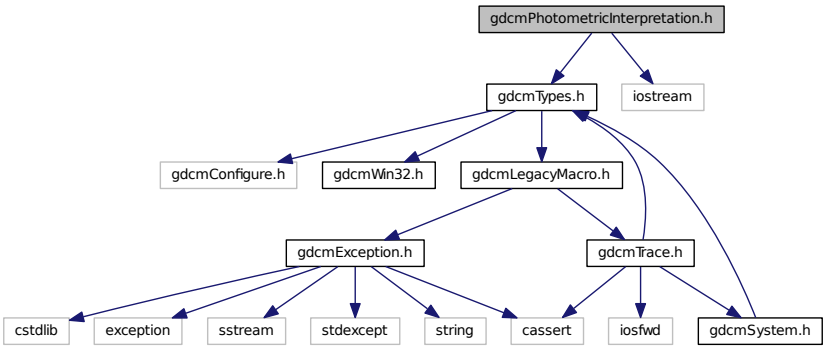
Namespaces

- gdcm

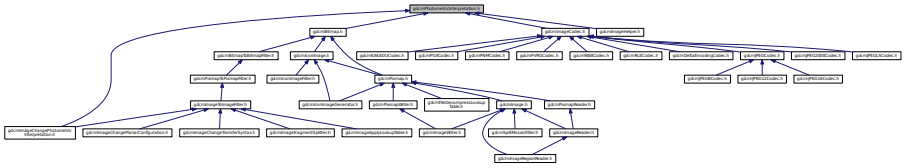
11.179 gdcmPhotometricInterpretation.h File Reference

```
#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

- gdcm

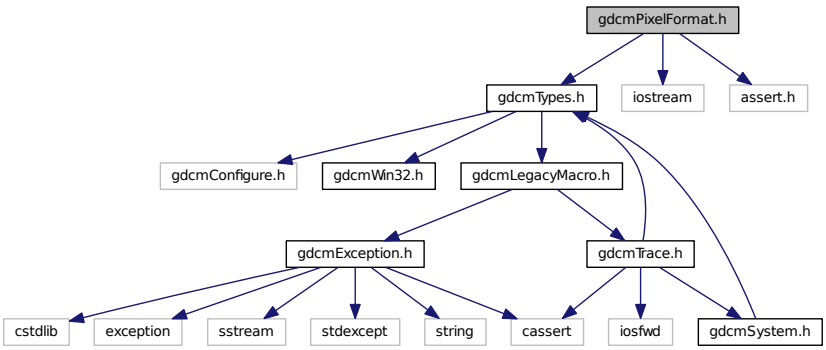
Functions

- std::ostream & gdcm::operator<< (std::ostream &os, const PhotometricInterpretation &val)

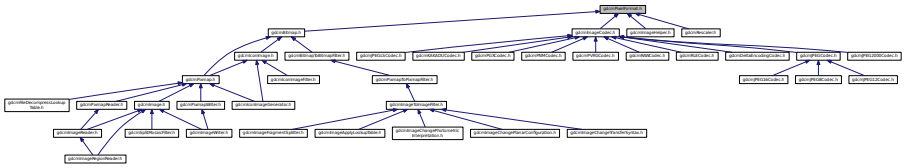
11.180 gdcmPixelFormat.h File Reference

```
#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

- gdcm

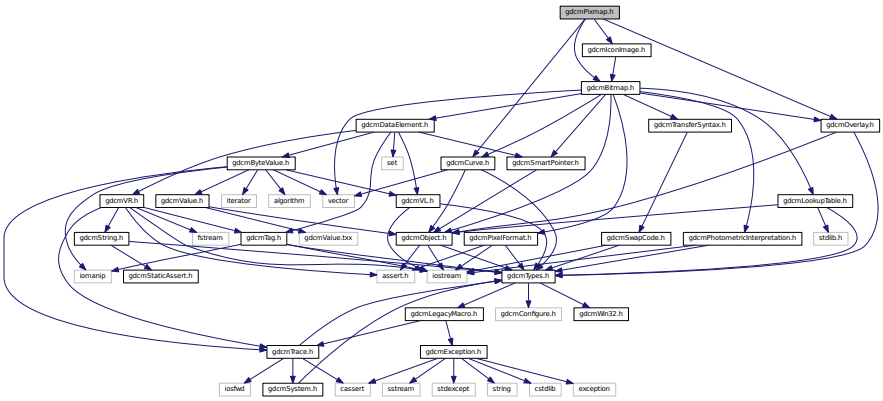
Functions

- std::ostream & gdcm::operator<< (std::ostream &os, const PixelFormat &pf)

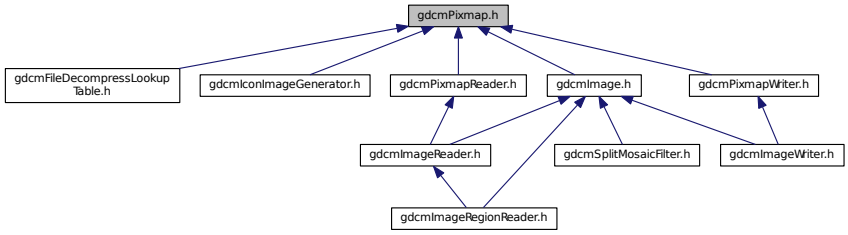
11.181 gdcmPixmap.h File Reference

```
#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.

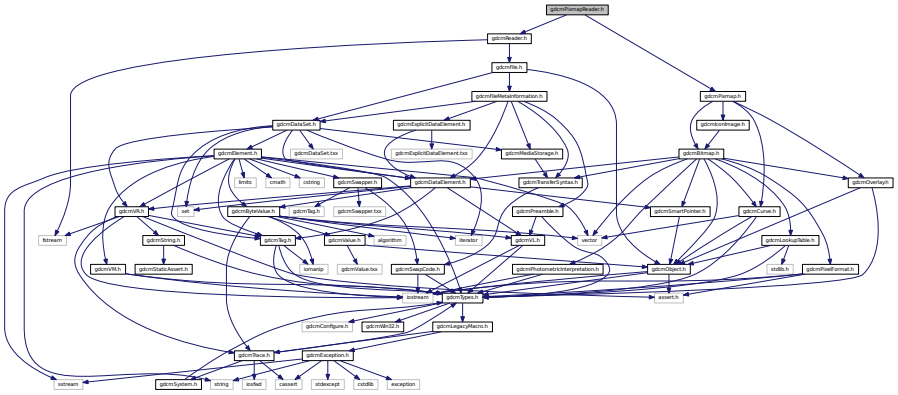
Namespaces

- gdcm

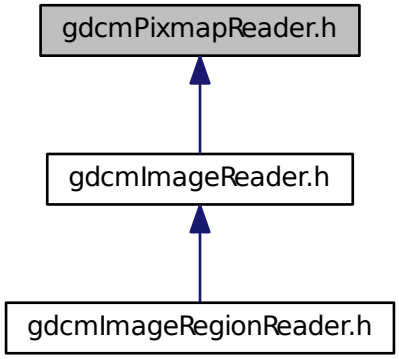
11.182 gdcmPixmapReader.h File Reference

```
#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.

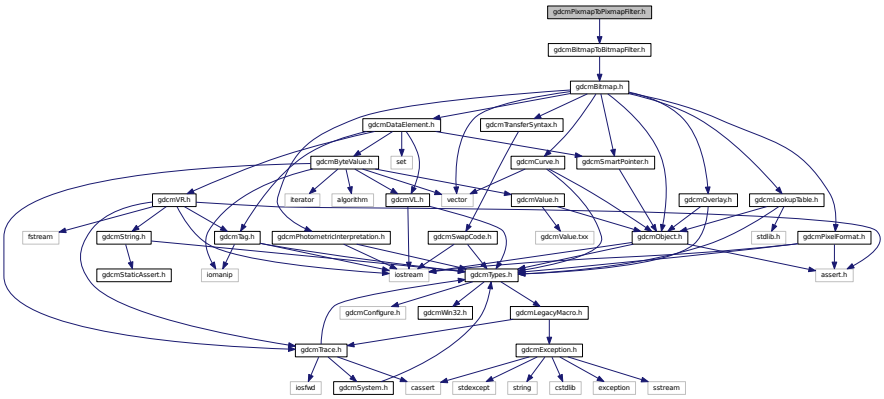
Namespaces

- gdcm

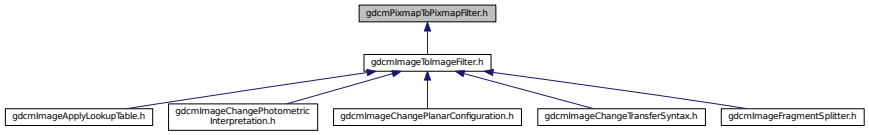
11.183 gdcmPixmapToPixmapFilter.h File Reference

```
#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.

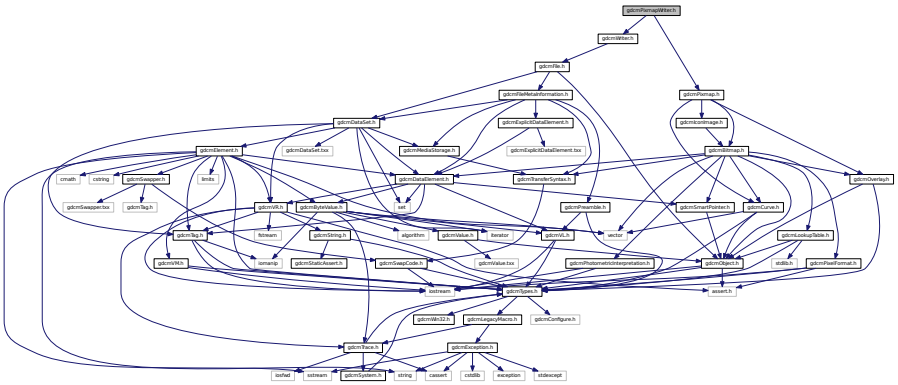
Namespaces

- gdcm

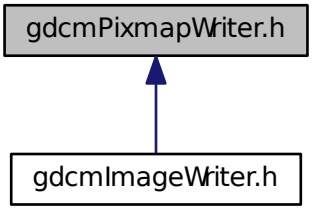
11.184 gdcmPixmapWriter.h File Reference

```
#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.

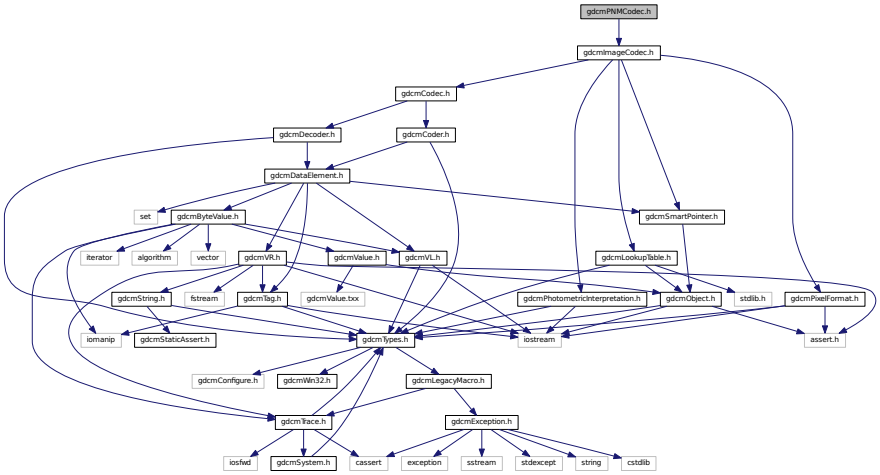
Namespaces

- gdcm

11.185 gdcmPNMCodec.h File Reference

```
#include "gdcmImageCodec.h"
```

Include dependency graph for gdcmPNMCodec.h:



Classes

- class gdcm::PNMCodec

  Class to do PNM.

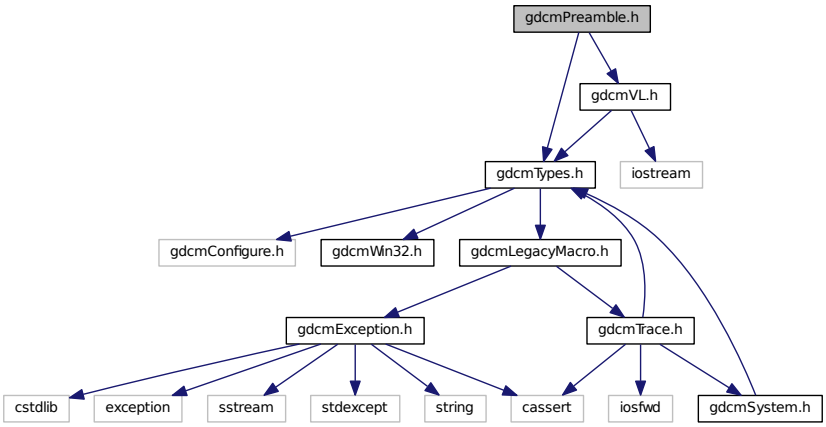
Namespaces

- gdcm

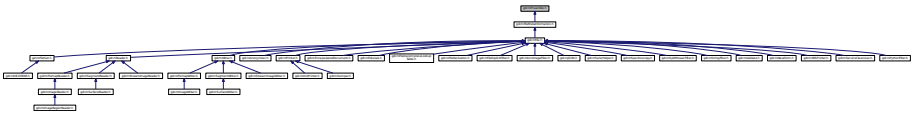
11.186 gdcmPreamble.h File Reference

```
#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

- gdcm

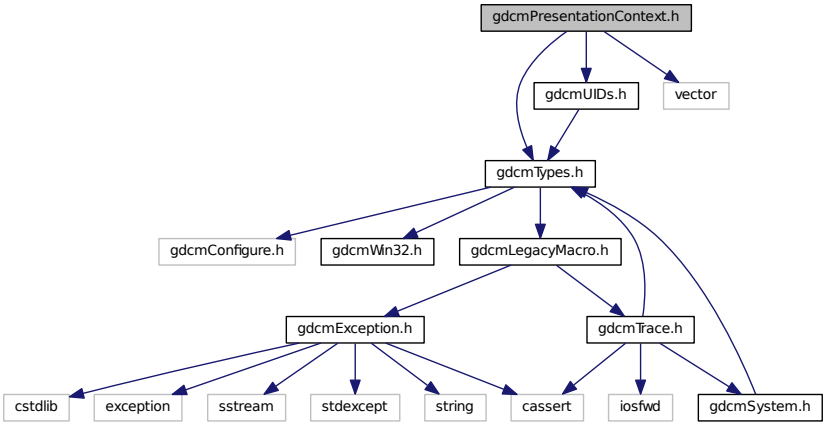
Functions

- std::ostream & gdcm::operator<< (std::ostream &os, const Preamble &val)

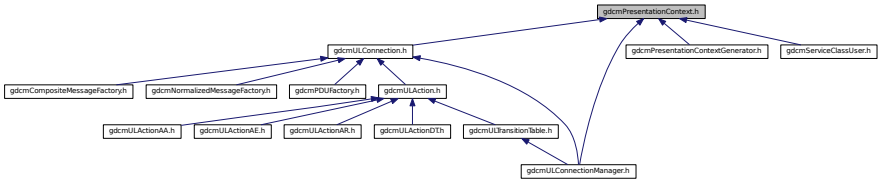
11.187 gdcmPresentationContext.h File Reference

```
#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.

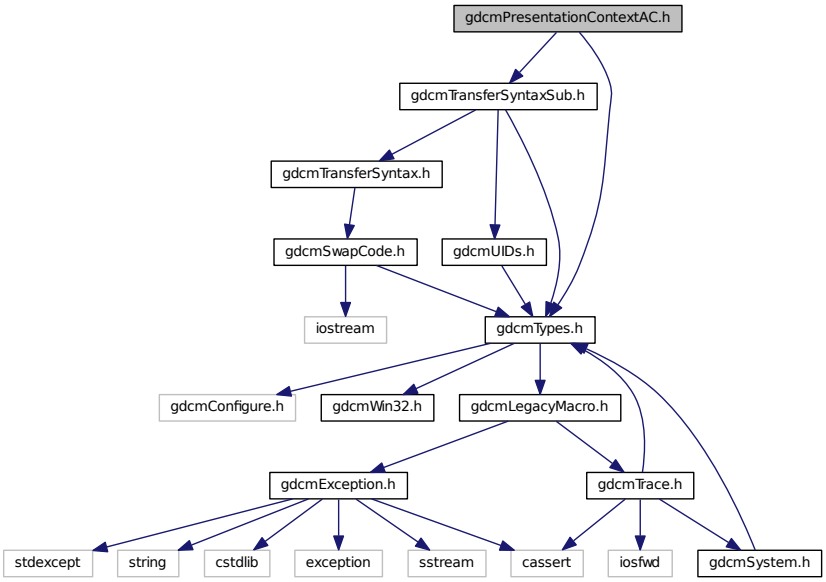
Namespaces

- gdcm

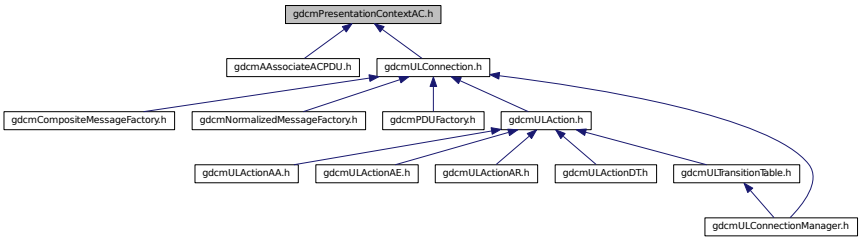
11.188 gdcmPresentationContextAC.h File Reference

```
#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.

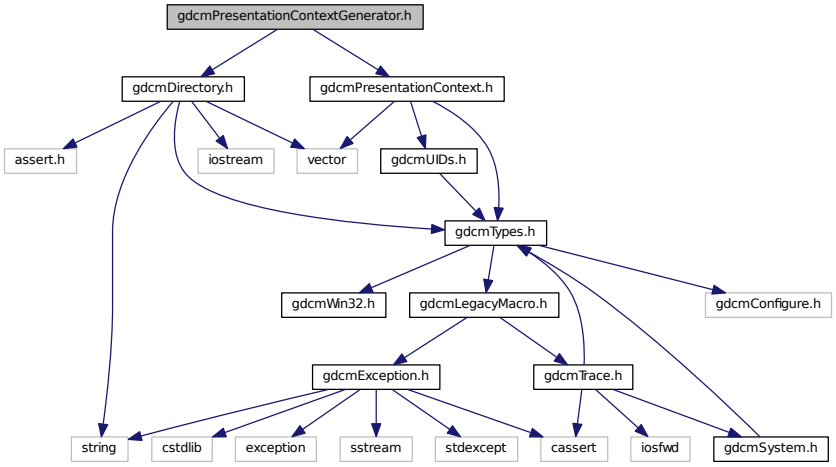
Namespaces

- gdcm
- gdcm::network

11.189 gdcmPresentationContextGenerator.h File Reference

```
#include "gdcmDirectory.h"
#include "gdcmPresentationContext.h"
```

Include dependency graph for gdcmPresentationContextGenerator.h:



Classes

- class gdcm::PresentationContextGenerator

  PresentationContextGenerator.

Namespaces

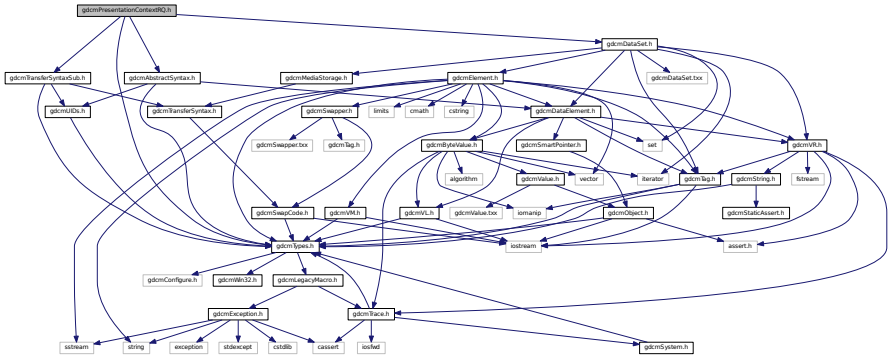
- gdcm

11.190 gdcmPresentationContextRQ.h File Reference

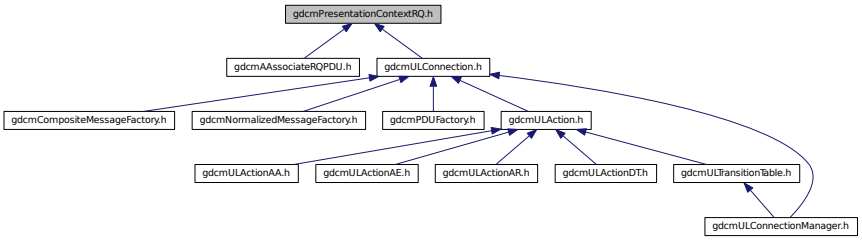
```
#include "gdcmTypes.h"
#include "gdcmAbstractSyntax.h"
#include "gdcmTransferSyntaxSub.h"
```

```
#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.

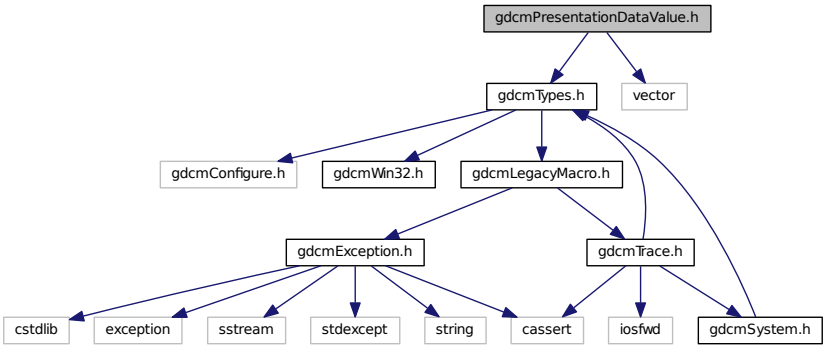
Namespaces

- gdcm
- gdcm::network

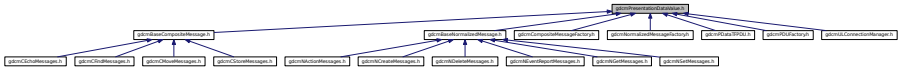
11.191 gdcmPresentationDataValue.h File Reference

```
#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.

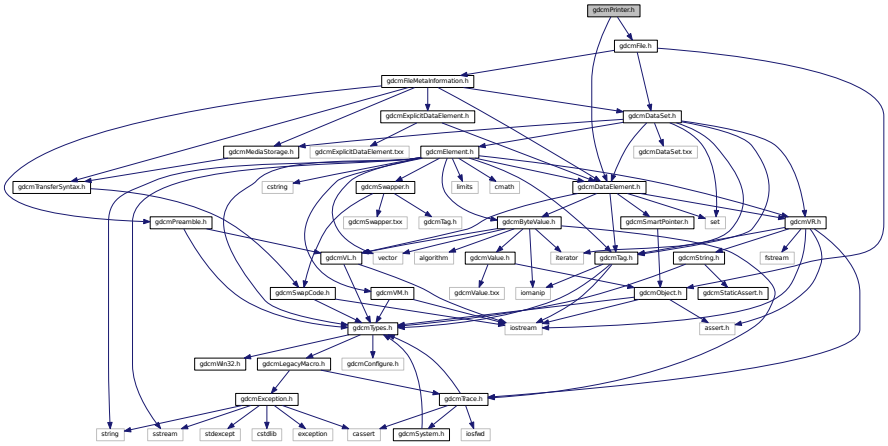
Namespaces

- gdcm
- gdcm::network

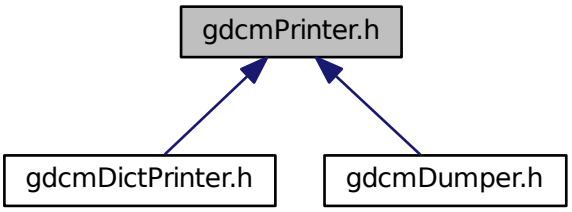
11.192 gdcmPrinter.h File Reference

```
#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.

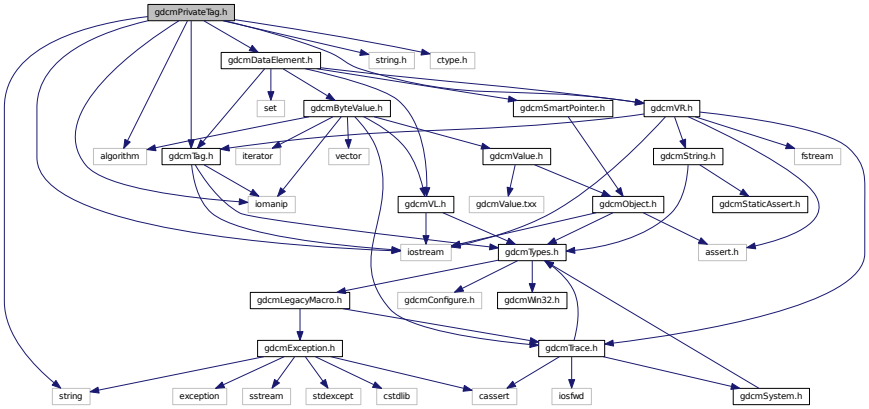
Namespaces

- gdcm

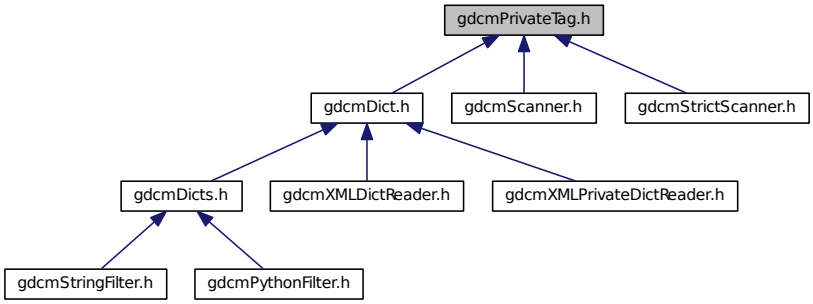
11.193 gdcmPrivateTag.h File Reference

```
#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

- gdcm

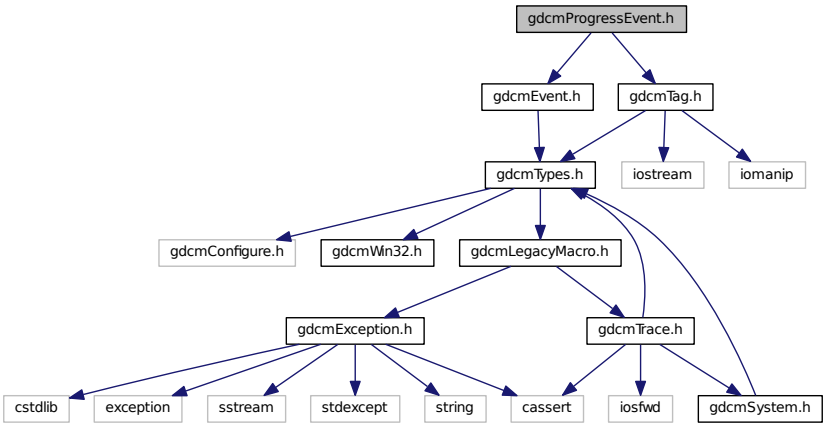
Functions

- std::ostream & gdcm::operator<< (std::ostream &os, const PrivateTag &val)

11.194 gdcmProgressEvent.h File Reference

```
#include "gdcmEvent.h"
#include "gdcmTag.h"
```

Include dependency graph for gdcmProgressEvent.h:



Classes

- class gdcm::ProgressEvent

  ProgressEvent.

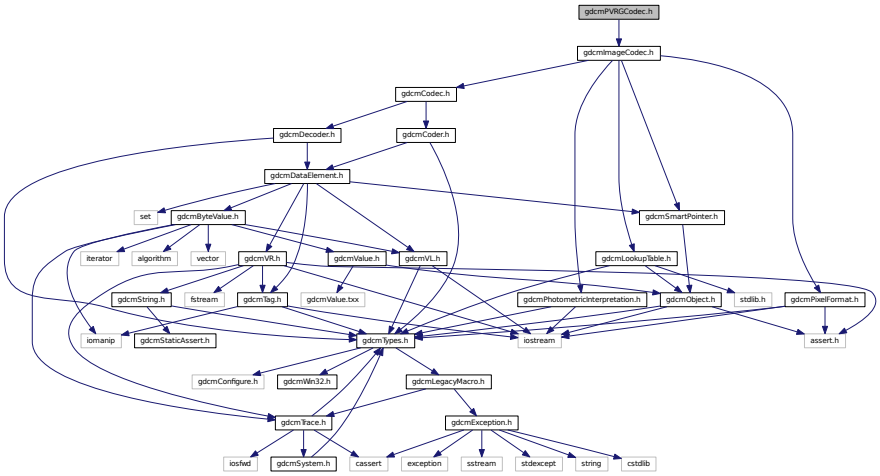
Namespaces

- gdcm

11.195 gdcmPVRGCodec.h File Reference

```
#include "gdcmImageCodec.h"
```

Include dependency graph for gdcmPVRGCodec.h:



Classes

- class gdcm::PVRGCodec

  PVRGCodec.

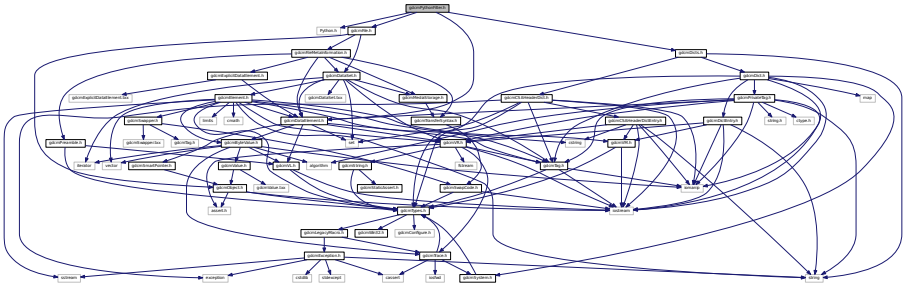
Namespaces

- gdcm

11.196 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.

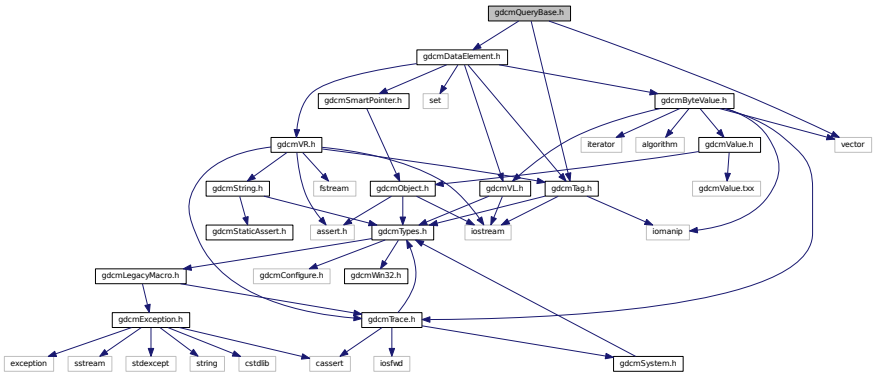
Namespaces

- gdcm

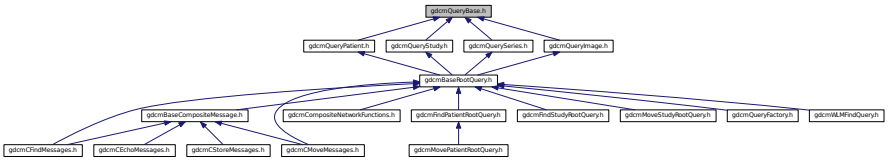
11.197 gdcmQueryBase.h File Reference

```
#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

- gdcm

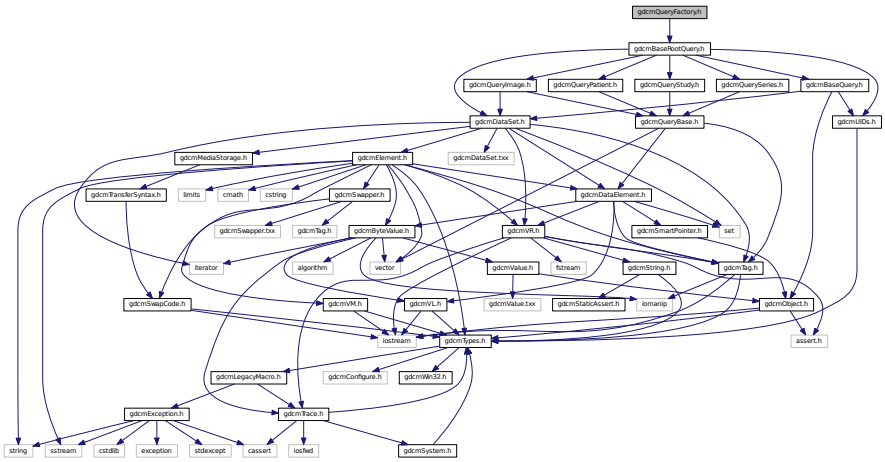
Enumerations

- enum gdcm::ERootType {
  gdcm::ePatientRootType,
  gdcm::eStudyRootType }

11.198 gdcmQueryFactory.h File Reference

```
#include "gdcmBaseRootQuery.h"
```

Include dependency graph for gdcmQueryFactory.h:



Classes

- class gdcm::QueryFactory

  QueryFactory.h.

Namespaces

- gdcm

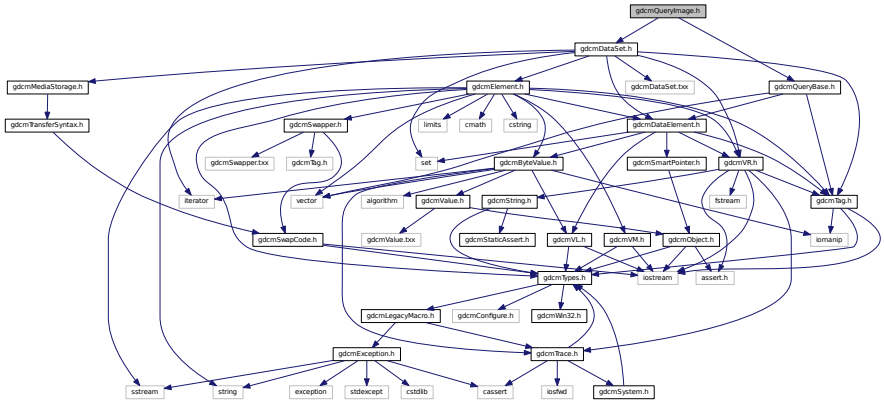
Enumerations

- 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 }

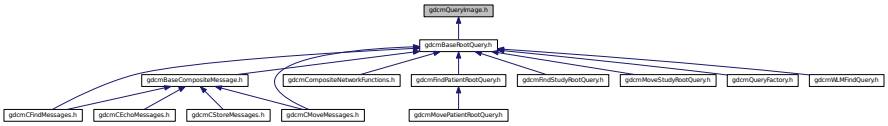
11.199 gdcmQueryImage.h File Reference

```
#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.

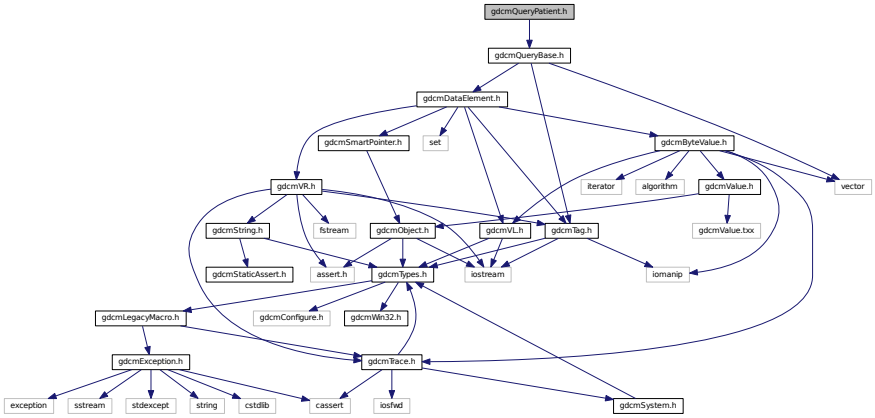
Namespaces

- gdcm

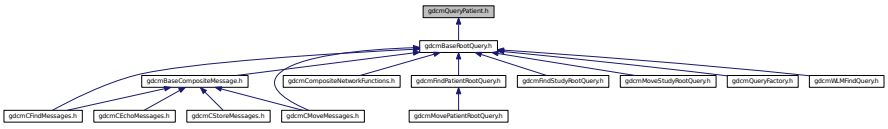
11.200 gdcmQueryPatient.h File Reference

```
#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.

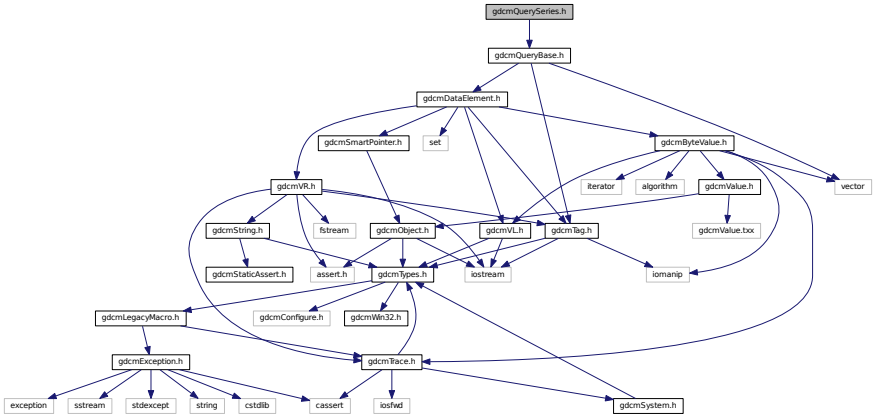
Namespaces

- gdcm

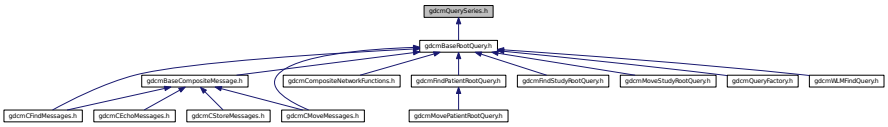
11.201 gdcmQuerySeries.h File Reference

```
#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.

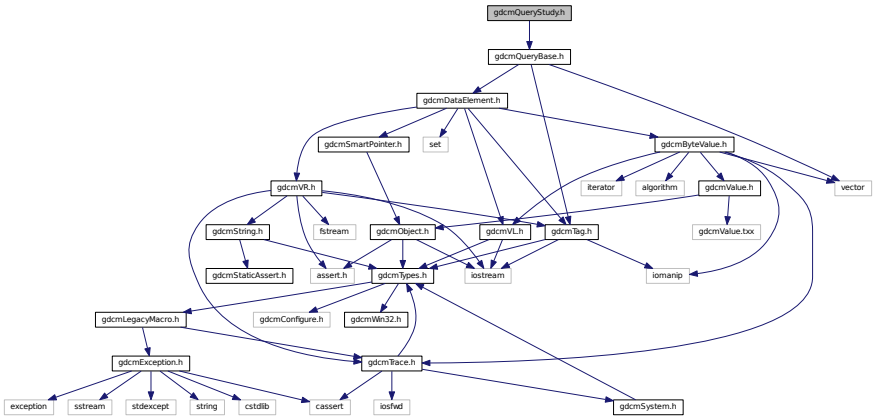
Namespaces

- gdcm

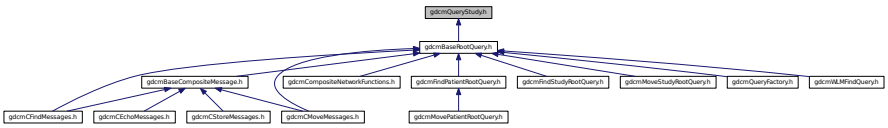
11.202 gdcmQueryStudy.h File Reference

```
#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.

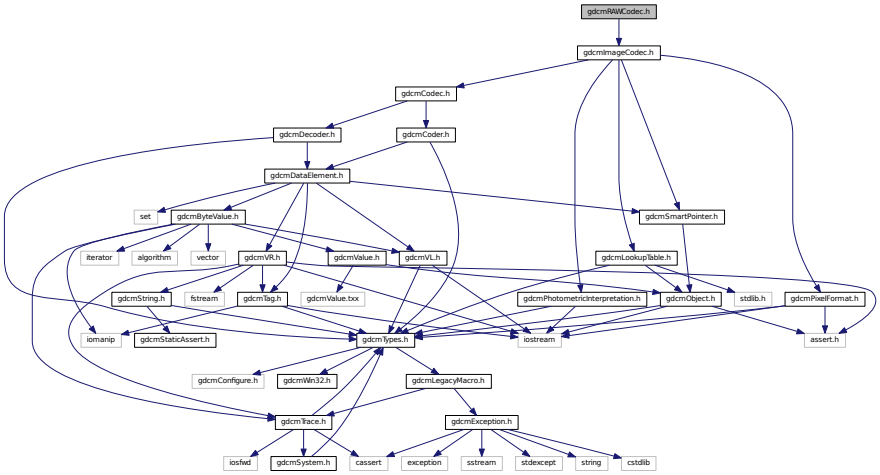
Namespaces

- gdcm

11.203 gdcmRAWCodec.h File Reference

```
#include "gdcmImageCodec.h"
```

Include dependency graph for gdcmRAWCodec.h:



Classes

- class gdcm::RAWCodec

  RAWCodec class.

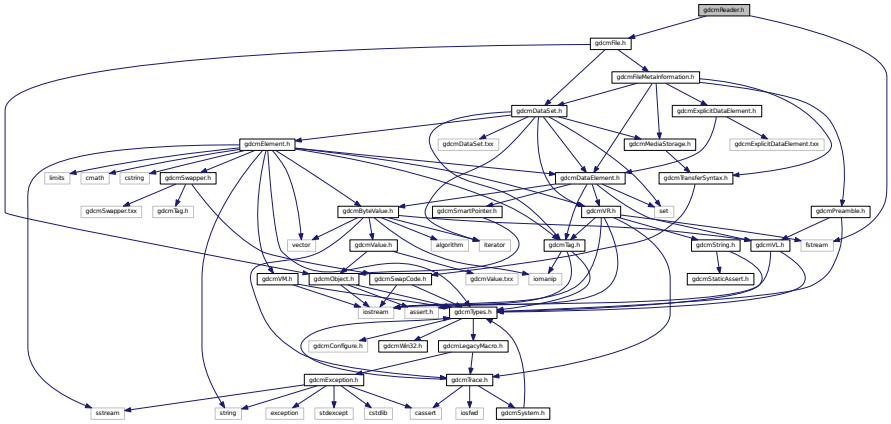
Namespaces

- gdcm

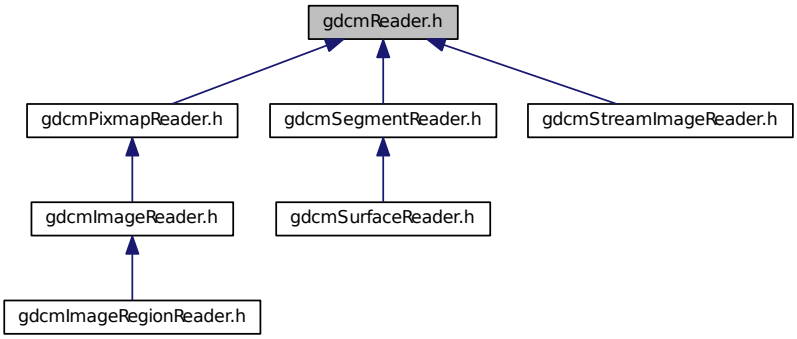
11.204 gdcmReader.h File Reference

```
#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)

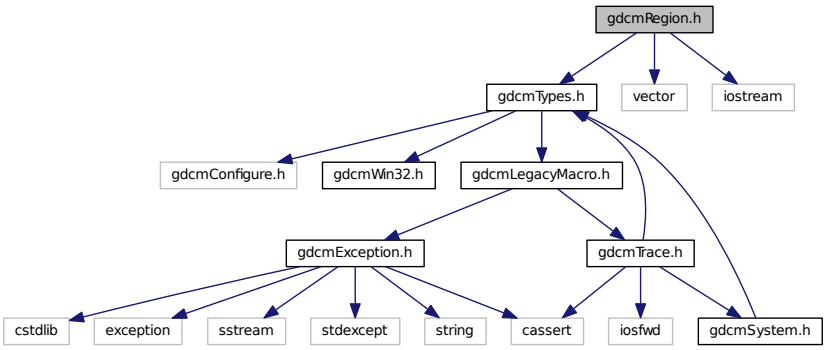
Namespaces

- gdcm

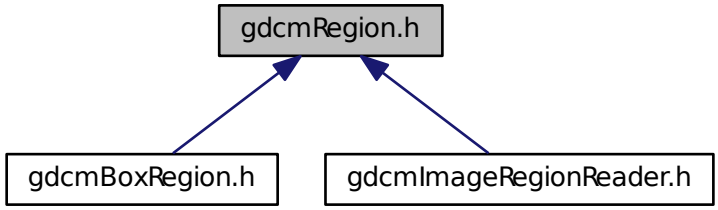
11.205 gdcmRegion.h File Reference

```
#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

- gdcm

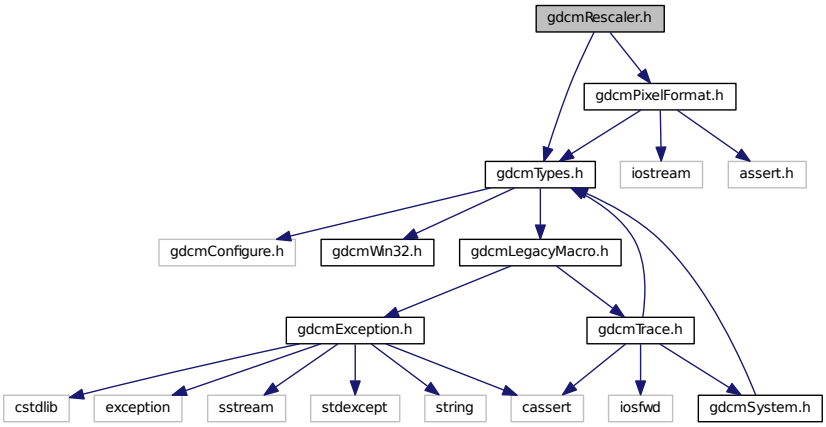
Functions

- std::ostream & gdcm::operator<< (std::ostream &os, const Region &r)

11.206 gdcmRescaler.h File Reference

```
#include "gdcmTypes.h"
#include "gdcmPixelFormat.h"
```

Include dependency graph for gdcmRescaler.h:



Classes

- class gdcm::Rescaler

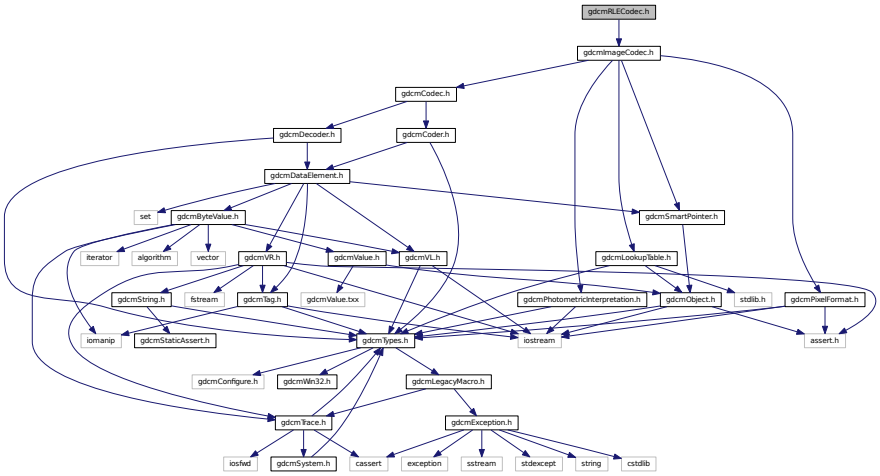
  Rescale class.

Namespaces

- gdcm

11.207 gdcmRLECodec.h File Reference

`#include "gdcmImageCodec.h"`
Include dependency graph for gdcmRLECodec.h:



Classes

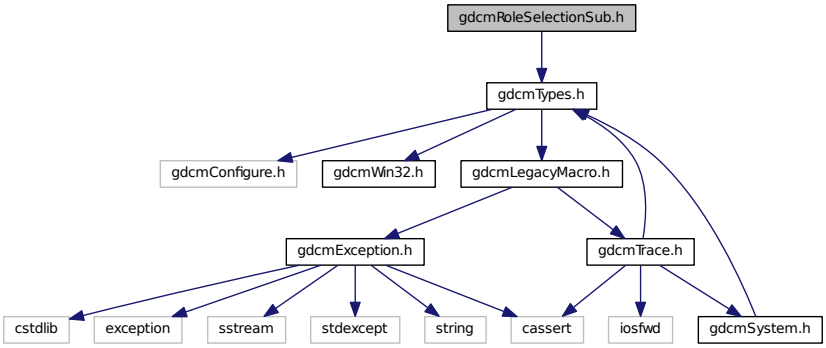
- class gdcm::RLECodec
  Class to do RLE.

Namespaces

- gdcm

11.208 gdcmRoleSelectionSub.h File Reference

`#include "gdcmTypes.h"`
Include dependency graph for gdcmRoleSelectionSub.h:



Classes

- class gdcm::network::RoleSelectionSub

  RoleSelectionSub.

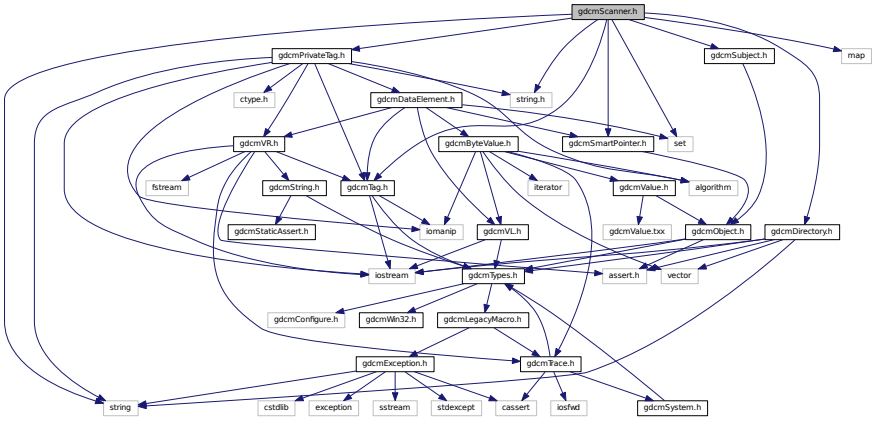
Namespaces

- gdcm
- gdcm::network

11.209 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

- gdcm

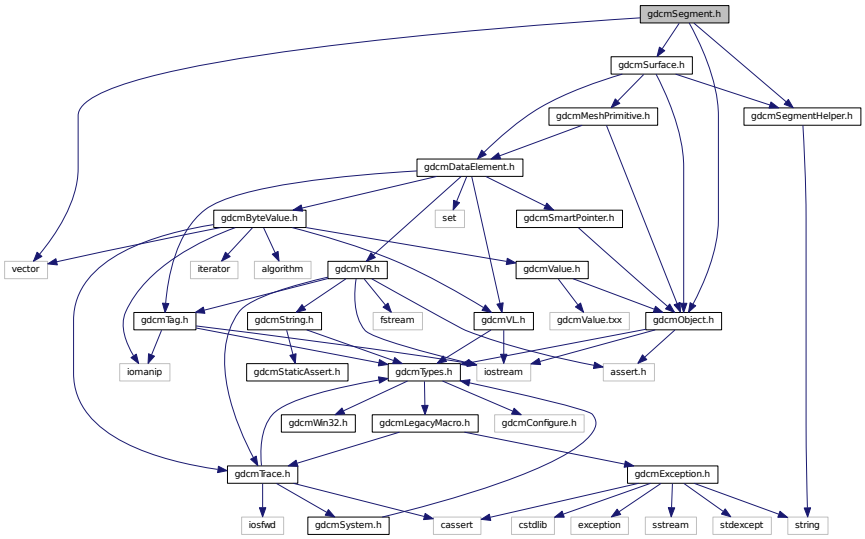
Functions

- std::ostream & gdcm::operator<< (std::ostream &os, const Scanner &s)

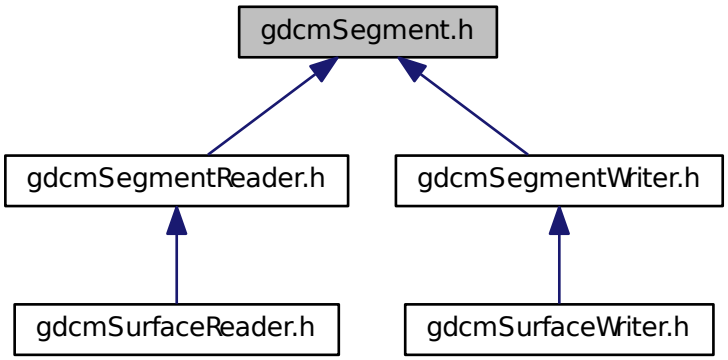
11.210 gdcmSegment.h File Reference

```
#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.

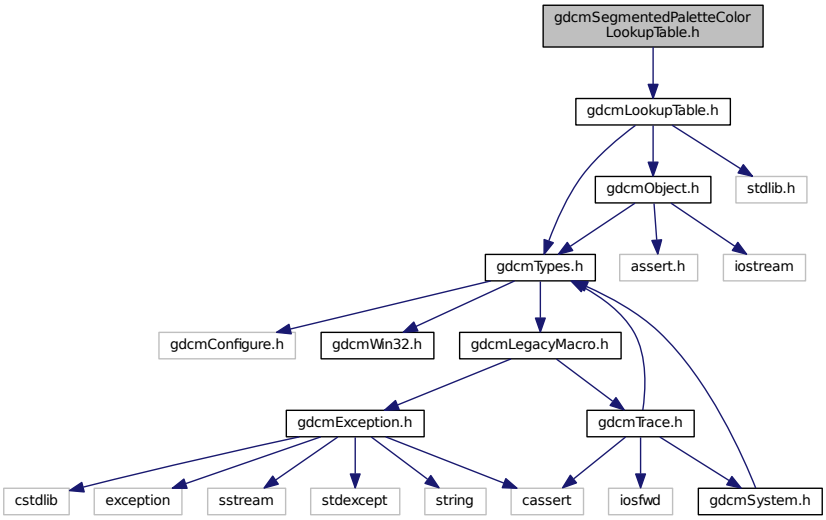
Namespaces

- gdcm

11.211 gdcmSegmentedPaletteColorLookupTable.h File Reference

```
#include "gdcmLookupTable.h"
```

Include dependency graph for gdcmSegmentedPaletteColorLookupTable.h:



Classes

- class gdcm::SegmentedPaletteColorLookupTable

  SegmentedPaletteColorLookupTable class.

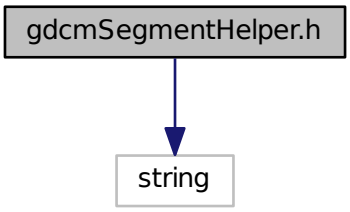
Namespaces

- gdcm

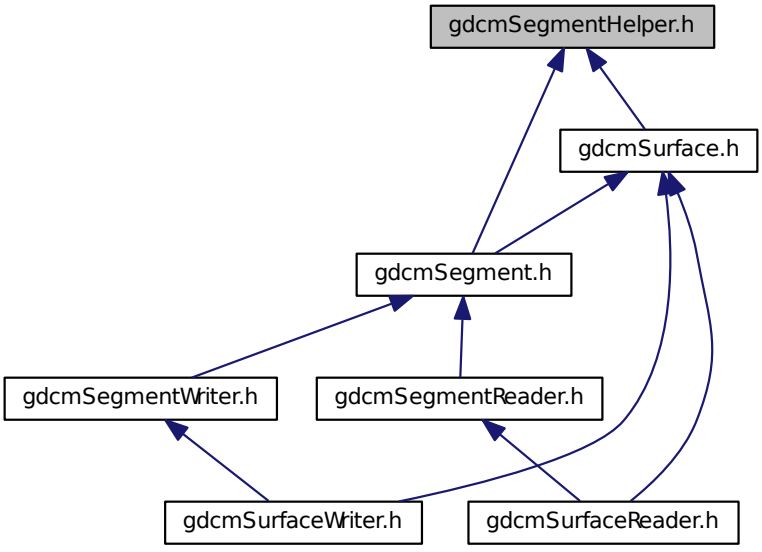
11.212 gdcmSegmentHelper.h File Reference

```
#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.

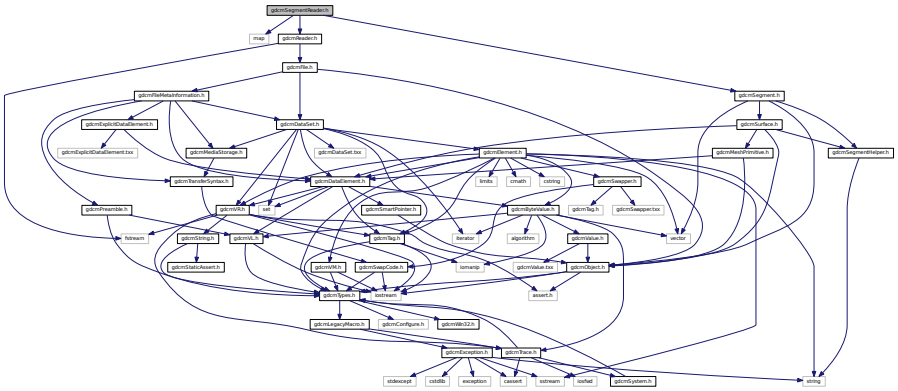
Namespaces

- gdcm
- gdcm::SegmentHelper

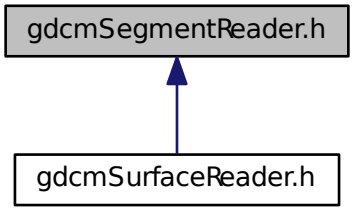
11.213 gdcmSegmentReader.h File Reference

```
#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.

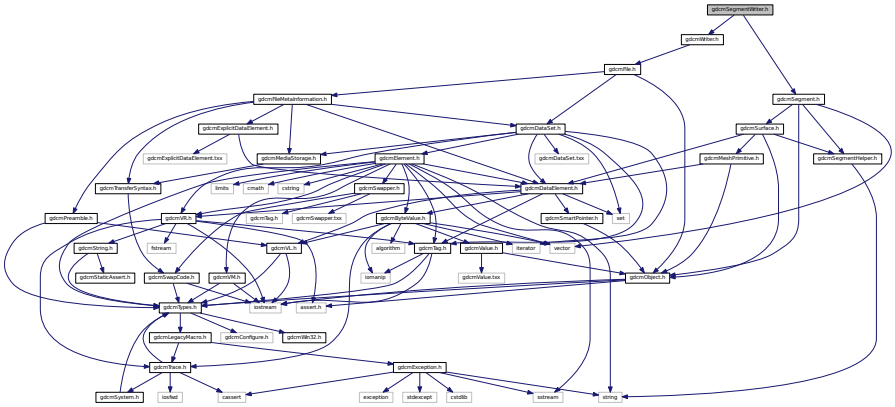
Namespaces

- gdcm

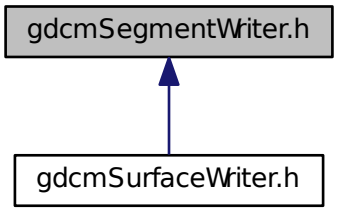
11.214 gdcmSegmentWriter.h File Reference

```
#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.

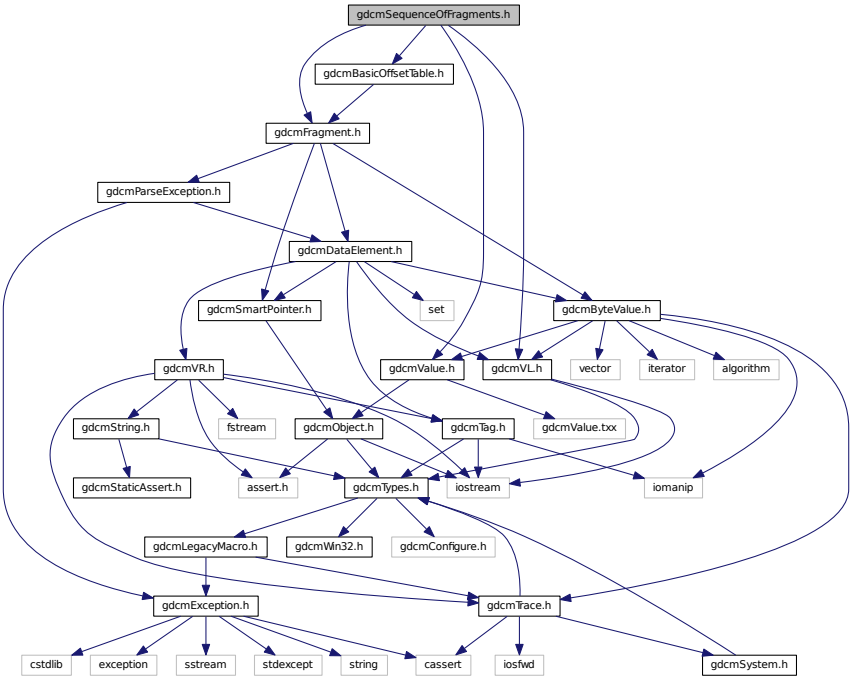
Namespaces

- gdcm

11.215 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.

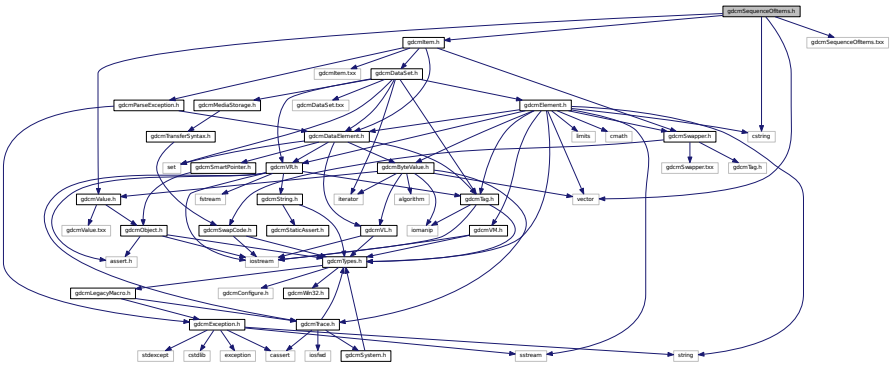
Namespaces

- gdcm

11.216 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

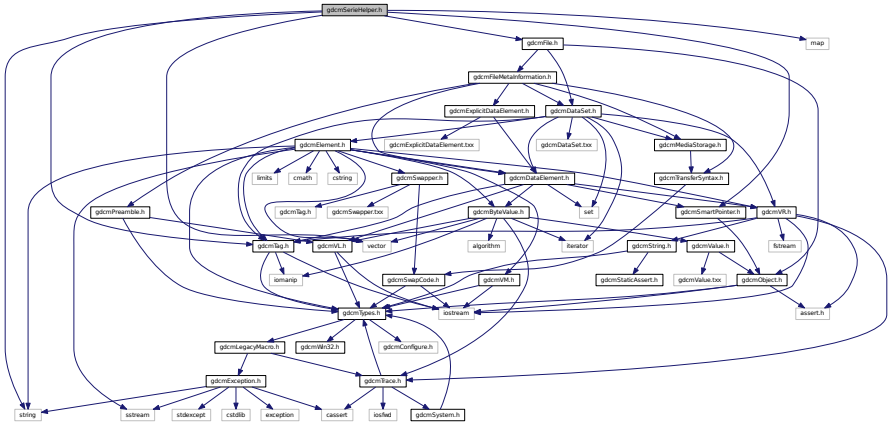
- gdcm

11.217 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.
- struct gdcm::SerieHelper::Rule
- 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 disapear soon, you've been warned.

Namespaces

- gdcm

Typedefs

- typedef bool(∗ gdcm::BOOL_FUNCTION_PFILE_PFILE_POINTER) (File ∗, File ∗)
- typedef std::vector< SmartPointer< FileWithName > > gdcm::FileList

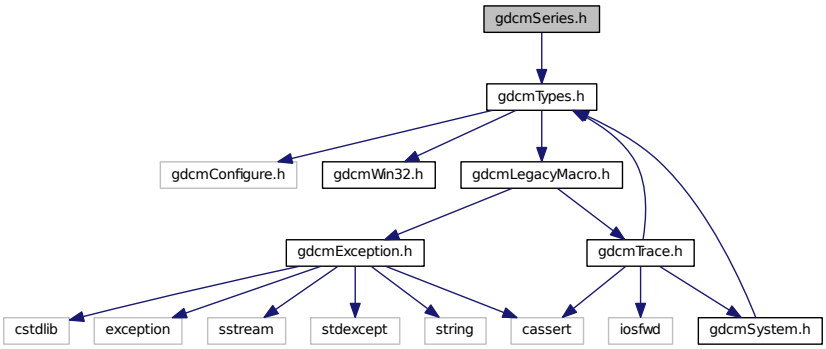
Enumerations

- 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 }

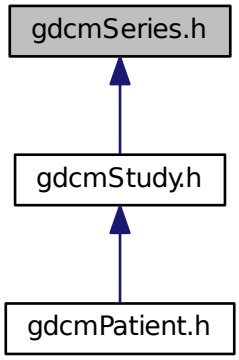
11.218 gdcmSeries.h File Reference

```
#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.

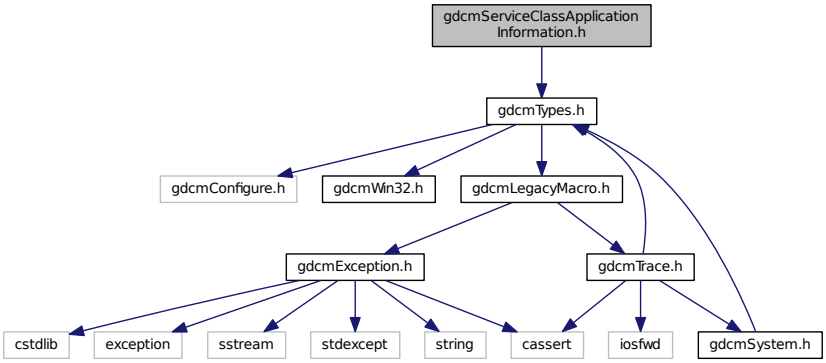
Namespaces

- gdcm

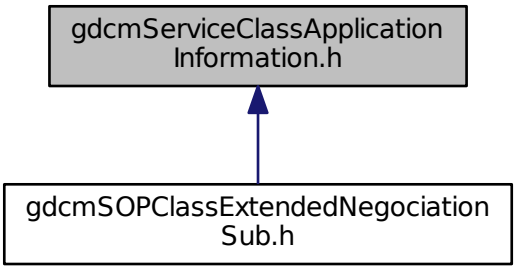
11.219 gdcmServiceClassApplicationInformation.h File Reference

```
#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

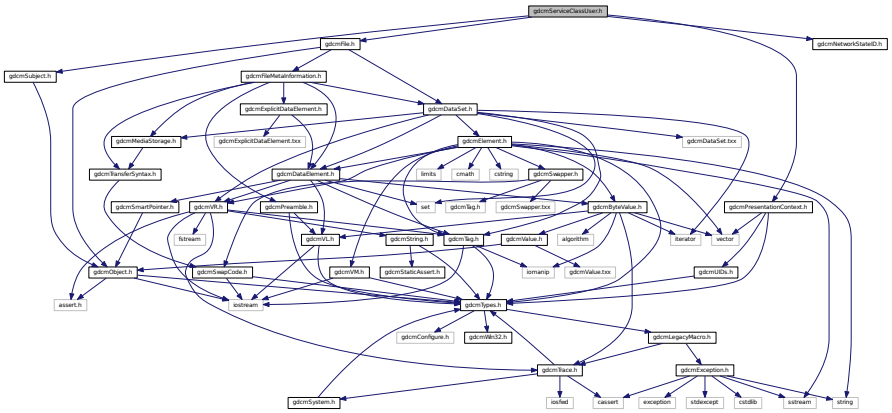
Namespaces

- gdcm
- gdcm::network

11.220 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.

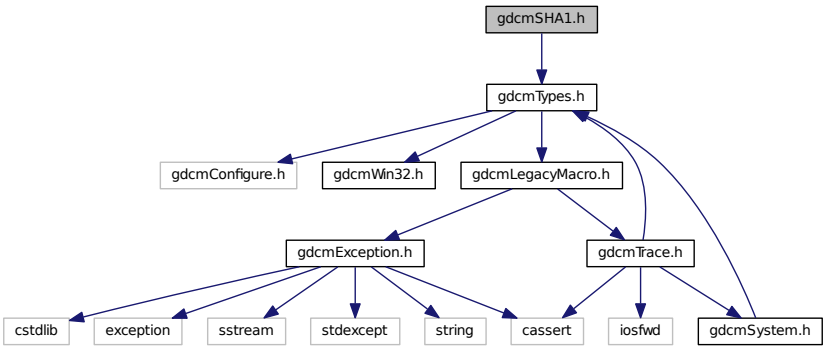
Namespaces

- gdcm
- gdcm::network

11.221 gdcmSHA1.h File Reference

```
#include "gdcmTypes.h"
```

Include dependency graph for gdcmSHA1.h:



Classes

- class gdcm::SHA1

  Class for SHA1.

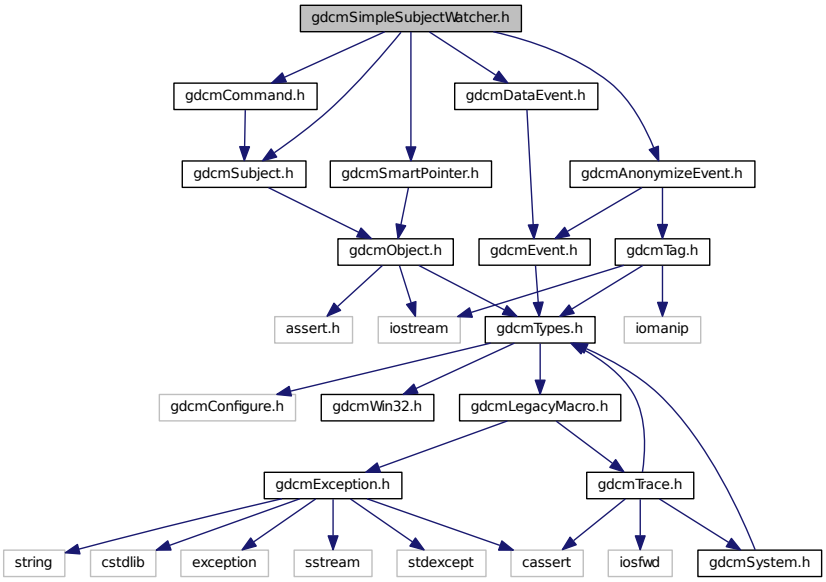
Namespaces

- gdcm

11.222 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.

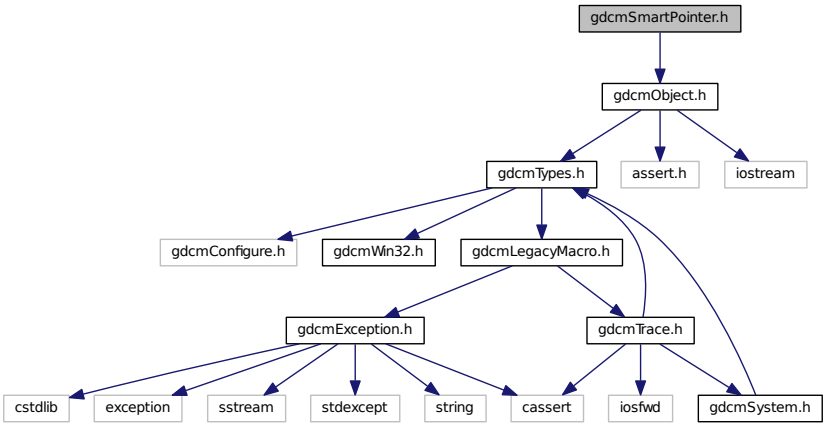
Namespaces

- gdcm

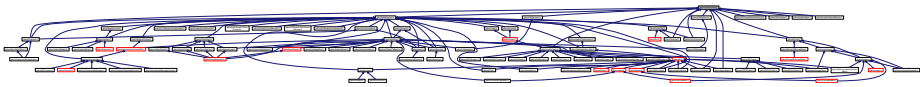
11.223 gdcmSmartPointer.h File Reference

```
#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.

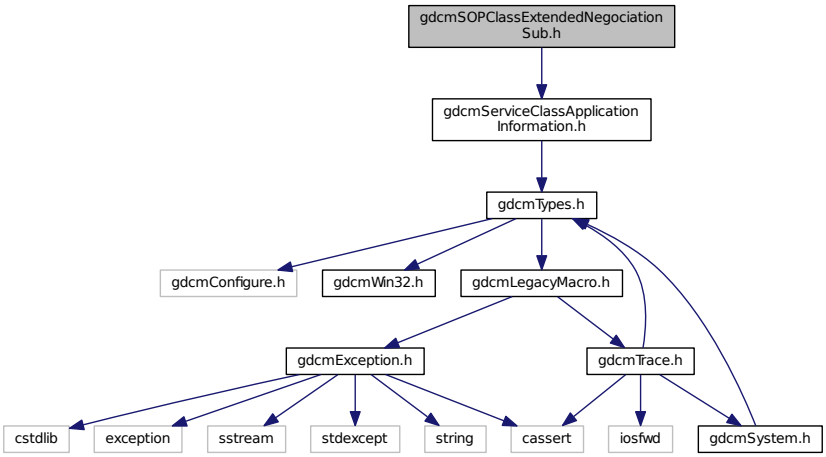
Namespaces

- gdcm

11.224 gdcmSOPClassExtendedNegociationSub.h File Reference

```
#include "gdcmServiceClassApplicationInformation.h"
```

Include dependency graph for gdcmSOPClassExtendedNegociationSub.h:



Classes

- class gdcm::network::SOPClassExtendedNegociationSub

  SOPClassExtendedNegociationSub.

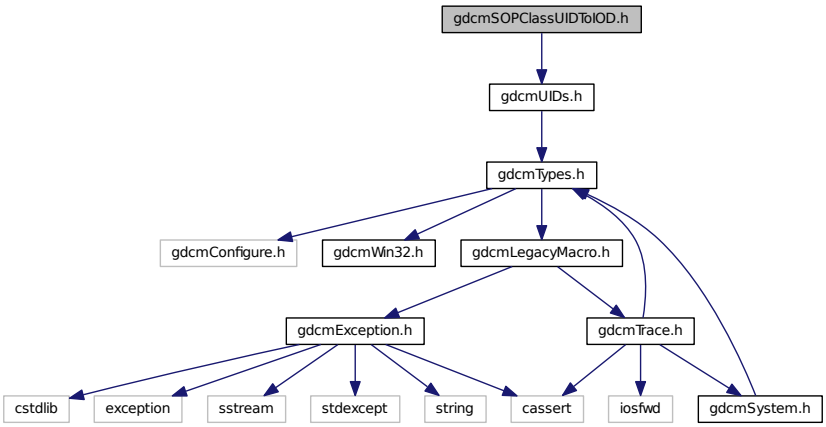
Namespaces

- gdcm
- gdcm::network

11.225 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.

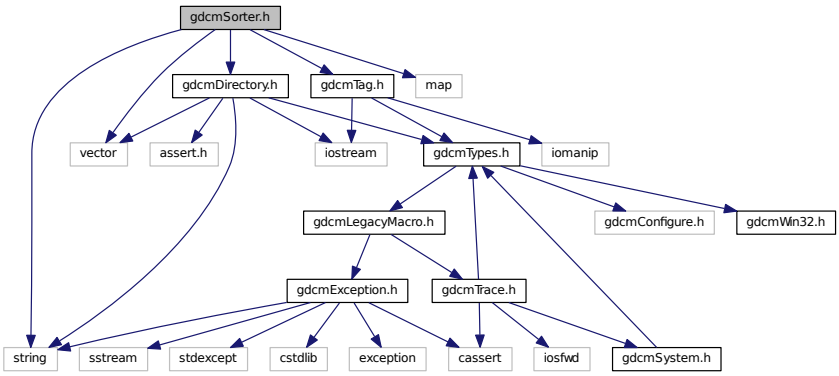
Namespaces

- gdcm

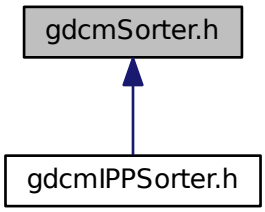
11.226 gdcmSorter.h File Reference

```
#include "gdcmDirectory.h"
#include "gdcmTag.h"
#include <vector>
#include <string>
#include <map>
```

Include dependency graph for gdcmSorter.h:



This graph shows which files directly or indirectly include this file:



Classes

- class gdcm::Sorter
  Sorter.

Namespaces

- gdcm

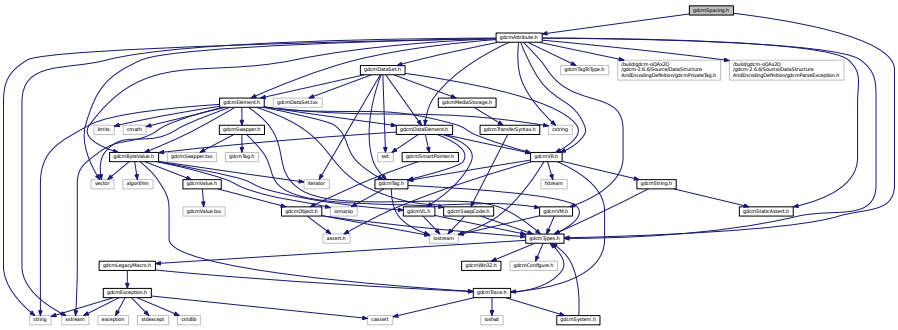
Functions

- std::ostream & gdcm::operator<< (std::ostream &os, const Sorter &s)

11.227 gdcmSpacing.h File Reference

```
#include "gdcmTypes.h"
#include "gdcmAttribute.h"
```

Include dependency graph for gdcmSpacing.h:



Classes

- class gdcm::Spacing

  Class for Spacing.

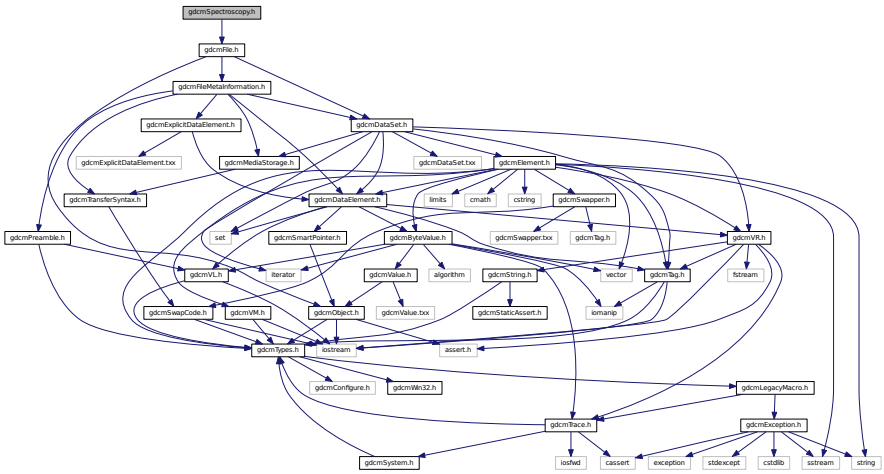
Namespaces

- gdcm

11.228 gdcmSpectroscopy.h File Reference

```
#include "gdcmFile.h"
```

Include dependency graph for gdcmSpectroscopy.h:



Classes

- class gdcm::Spectroscopy

  Spectroscopy class.

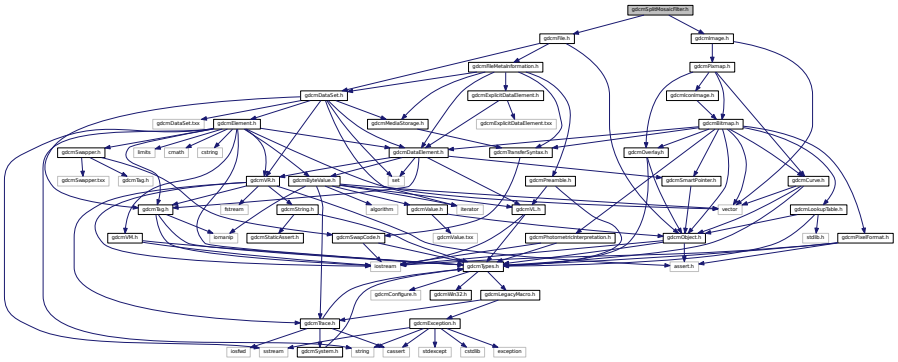
Namespaces

- gdcm

11.229 gdcmSplitMosaicFilter.h File Reference

```
#include "gdcmFile.h"
#include "gdcmImage.h"
```

Include dependency graph for gdcmSplitMosaicFilter.h:



Classes

- class gdcm::SplitMosaicFilter

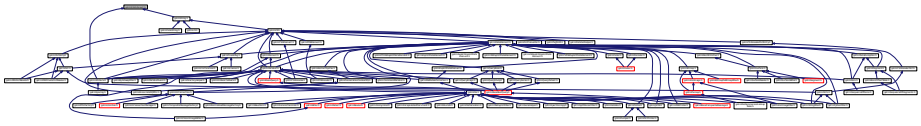
  SplitMosaicFilter class.

Namespaces

- gdcm

11.230 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< x >
- struct gdcm::STATIC_ASSERTION_FAILURE< true >

Namespaces

- 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.*

11.230.1 Macro Definition Documentation

11.230.1.1 GDCM_DO_JOIN

```
#define GDCM_DO_JOIN(
              X,
              Y ) GDCM_DO_JOIN2(X,Y)
```

11.230.1.2 GDCM_DO_JOIN2

```
#define GDCM_DO_JOIN2(
              X,
              Y ) X##Y
```

11.230.1.3 GDCM_JOIN

```
#define GDCM_JOIN(
              X,
              Y ) GDCM_DO_JOIN( X, Y )
```

11.230.1.4 GDCM_STATIC_ASSERT

```
#define GDCM_STATIC_ASSERT(
              B )
```

Value:

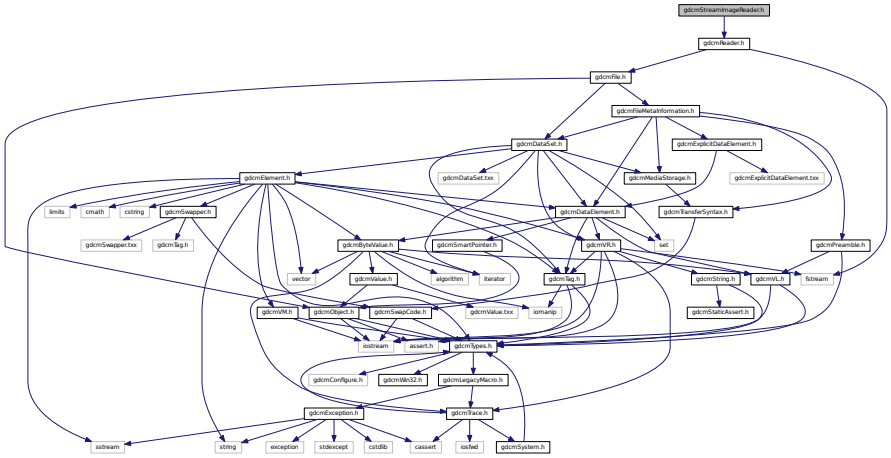
```
typedef ::gdcm::static_assert_test<\
    sizeof(::gdcm::STATIC_ASSERTION_FAILURE< (bool)( B ) >)>\
      GDCM_JOIN(gdcm_static_assert_typedef_, __LINE__)
```

The GDCM_JOIN + **LINE** is needed to create a uniq identifier.

11.231 gdcmStreamImageReader.h File Reference

```
#include "gdcmReader.h"
```

Include dependency graph for gdcmStreamImageReader.h:



Classes

- class gdcm::StreamImageReader

  StreamImageReader.

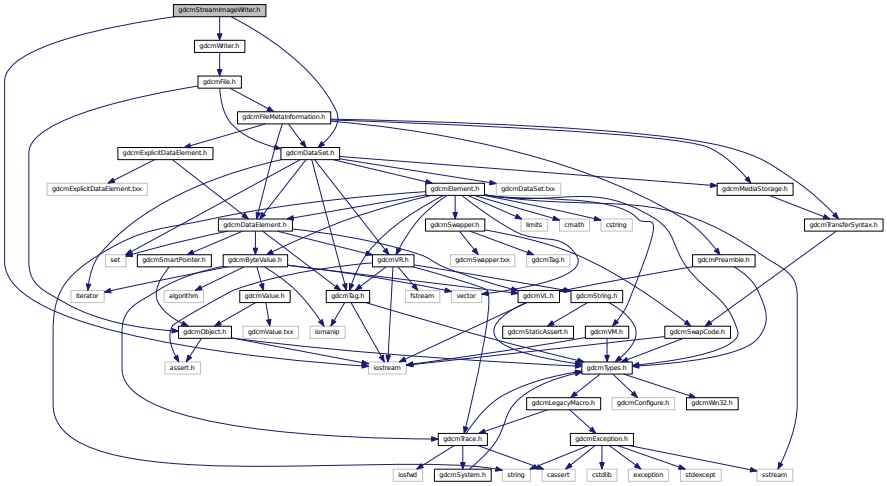
Namespaces

- gdcm

11.232 gdcmStreamImageWriter.h File Reference

```
#include "gdcmWriter.h"
#include <iostream>
#include "gdcmDataSet.h"
```

Include dependency graph for gdcmStreamImageWriter.h:



Classes

- class gdcm::StreamImageWriter

  StreamImageReader.

Namespaces

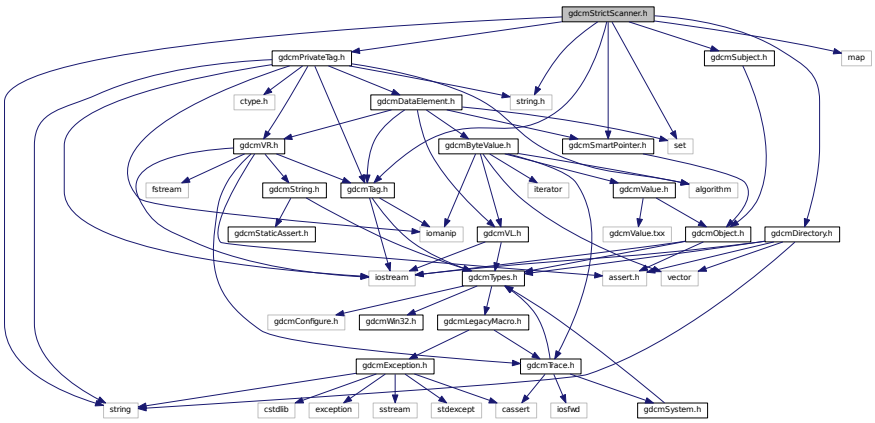
- gdcm

11.233 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

- gdcm

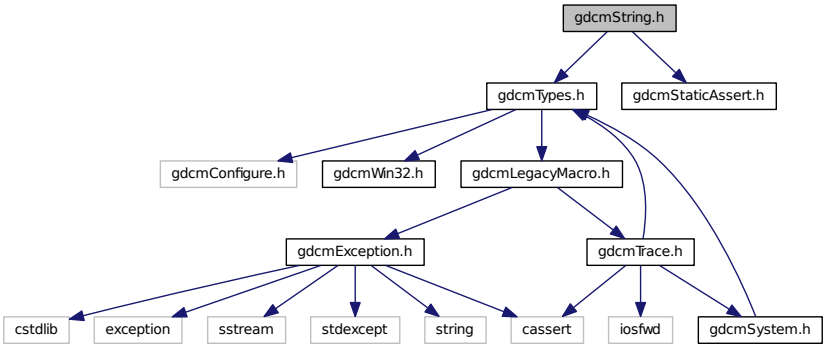
Functions

- std::ostream & gdcm::operator<< (std::ostream &os, const StrictScanner &s)

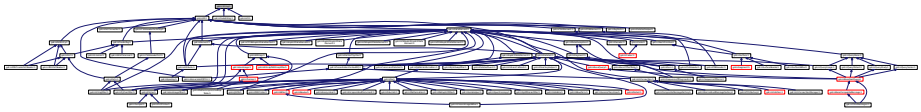
11.234 gdcmString.h File Reference

```
#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

- gdcm

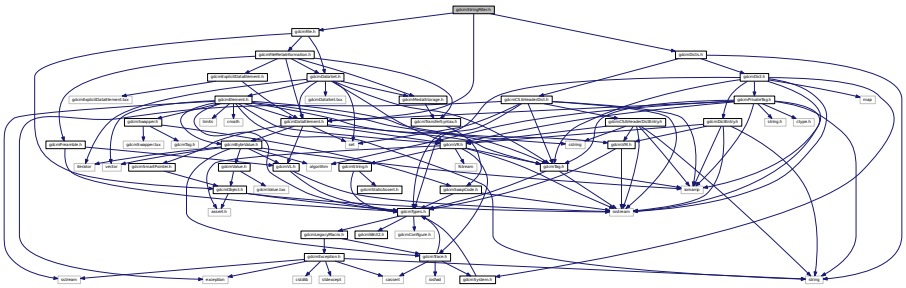
Functions

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

11.235 gdcmStringFilter.h File Reference

```
#include "gdcmDataElement.h"
#include "gdcmDicts.h"
#include "gdcmFile.h"
```

Include dependency graph for gdcmStringFilter.h:



Classes

- class gdcm::StringFilter

  StringFilter.

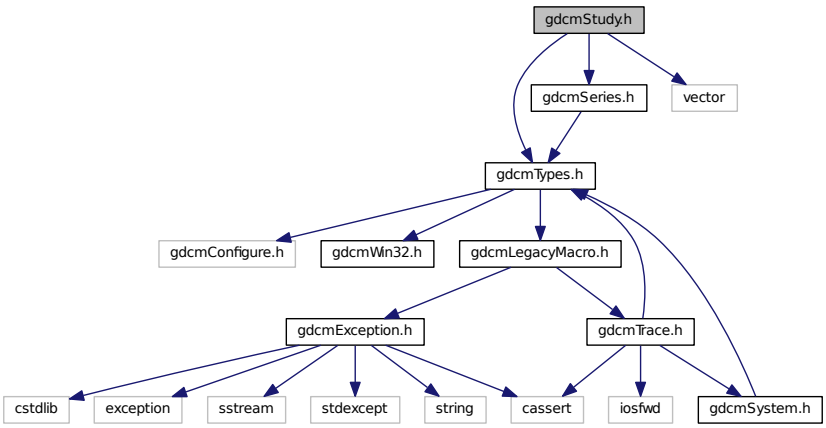
Namespaces

- gdcm

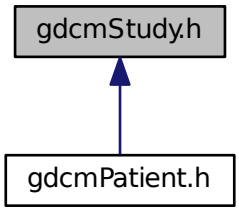
11.236 gdcmStudy.h File Reference

```
#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.

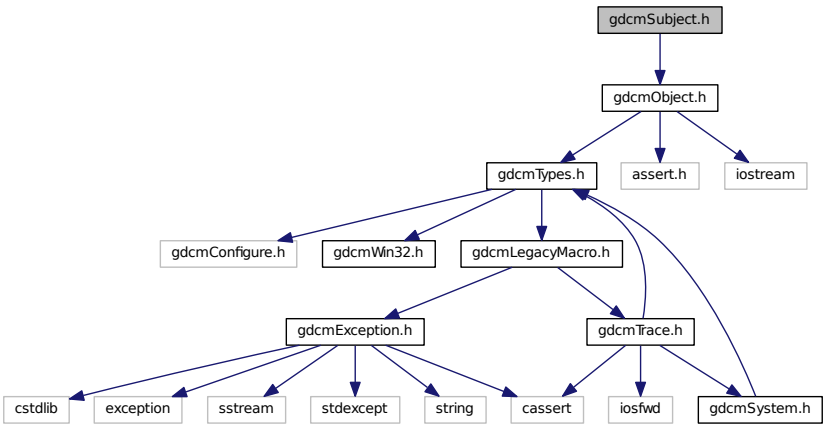
Namespaces

- gdcm

11.237 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.

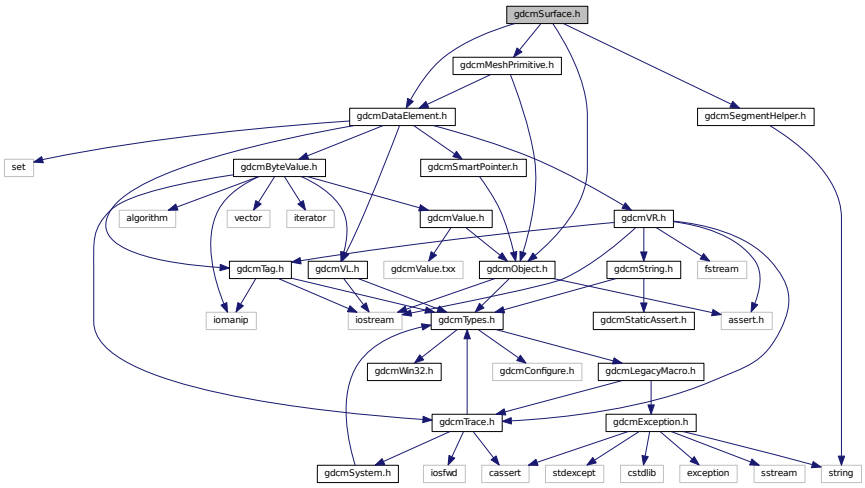
Namespaces

- gdcm

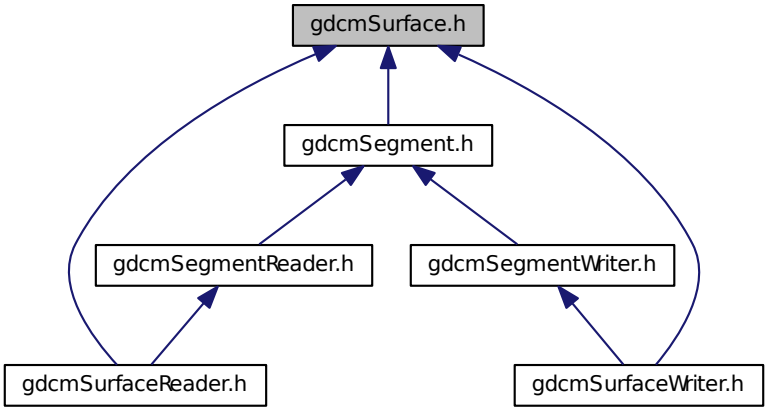
11.238 gdcmSurface.h File Reference

```
#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.

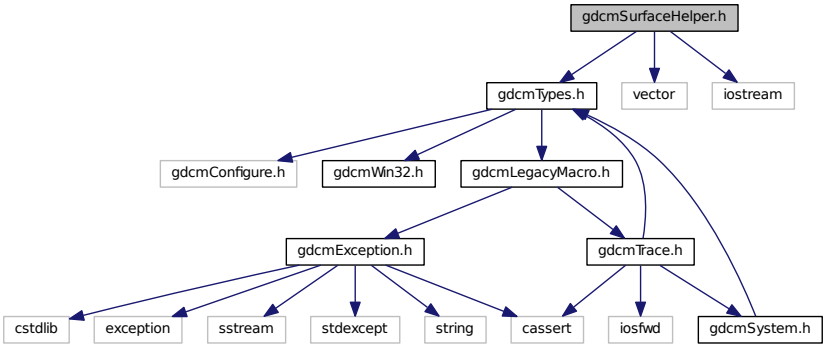
Namespaces

- gdcm

11.239 gdcmSurfaceHelper.h File Reference

```
#include "gdcmTypes.h"
#include <vector>
#include <iostream>
```

Include dependency graph for gdcmSurfaceHelper.h:



Classes

- class gdcm::SurfaceHelper
  SurfaceHelper.

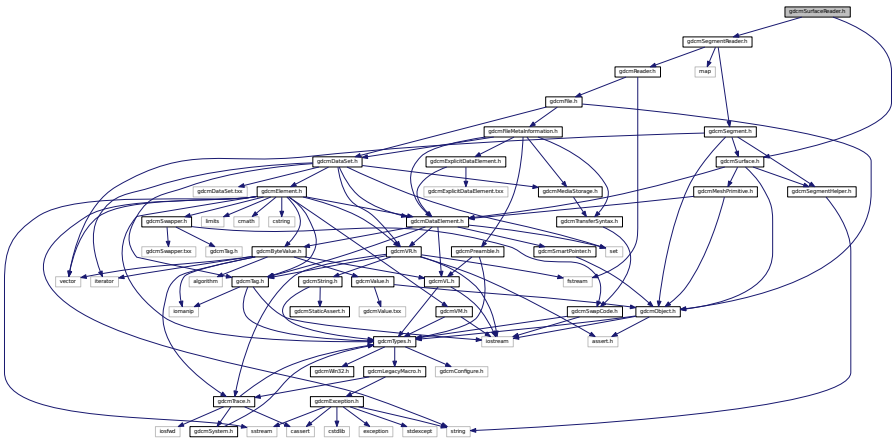
Namespaces

- gdcm

11.240 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.

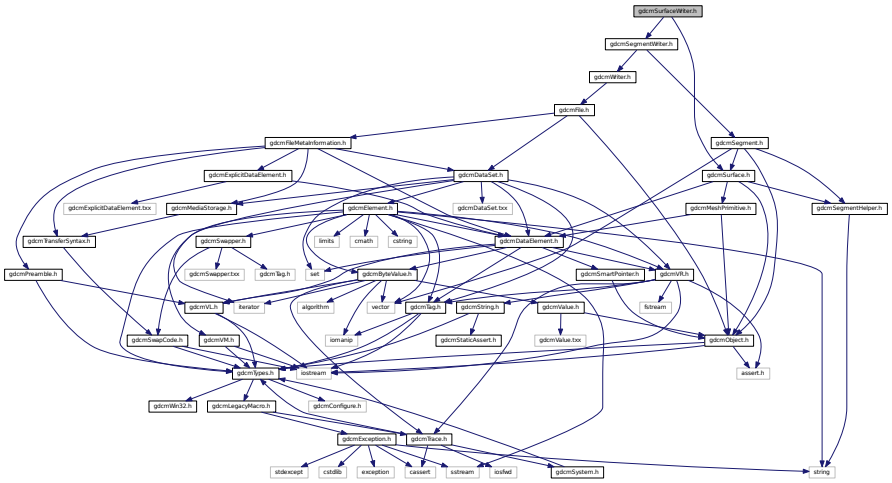
Namespaces

- gdcm

11.241 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.

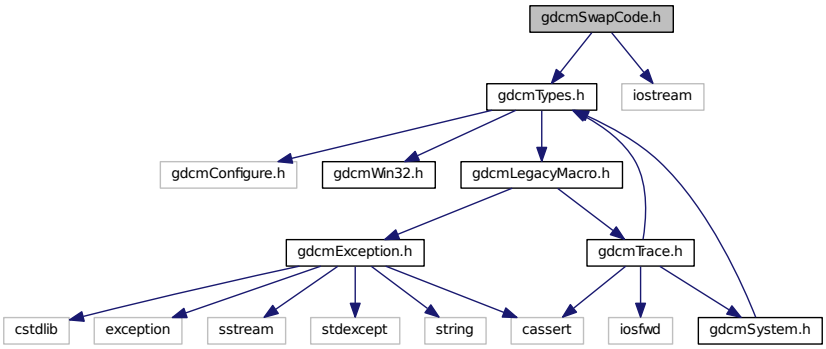
Namespaces

- gdcm

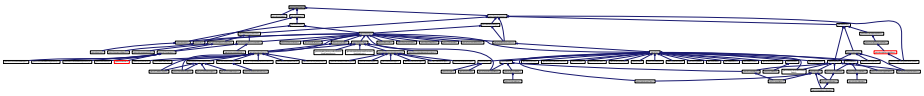
11.242 gdcmSwapCode.h File Reference

```
#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

- gdcm

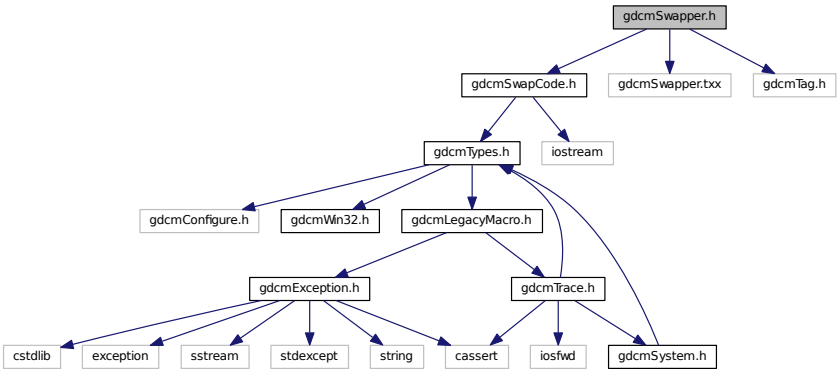
Functions

- std::ostream & gdcm::operator<< (std::ostream &os, const SwapCode &sc)

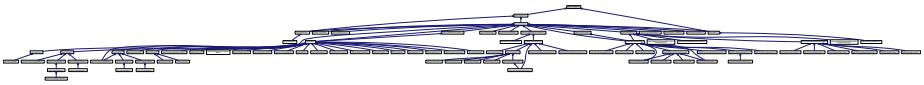
11.243 gdcmSwapper.h File Reference

```
#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

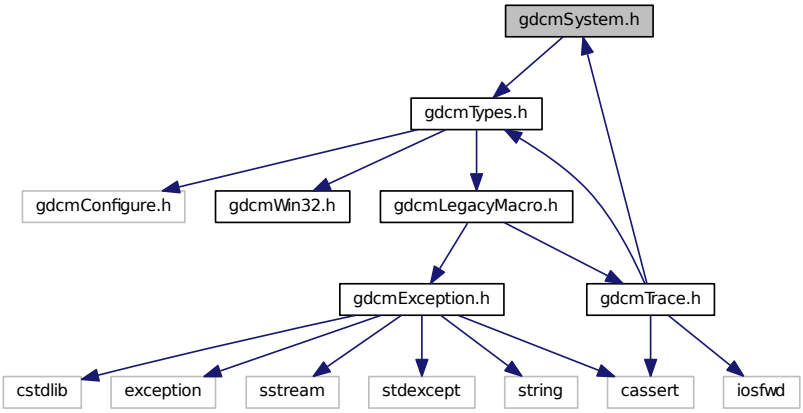
Namespaces

- gdcm

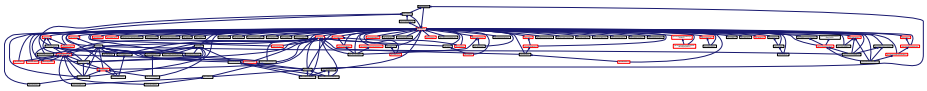
11.244 gdcmSystem.h File Reference

```
#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

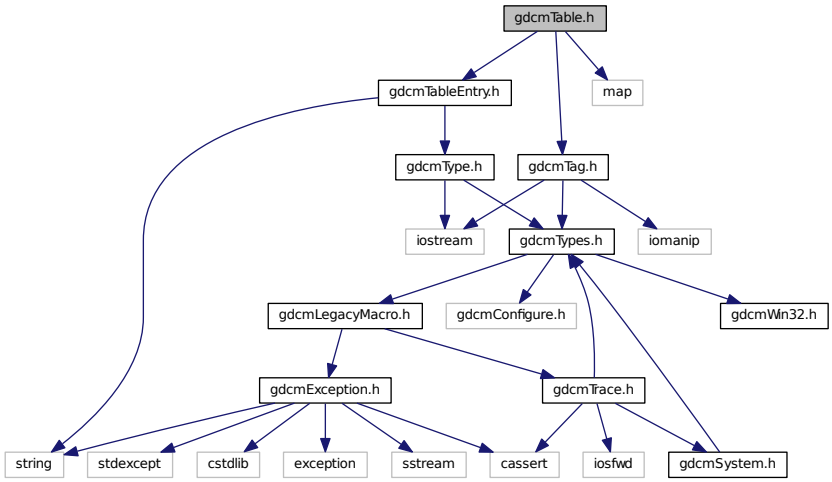
- gdcm

11.245 gdcmTable.h File Reference

```
#include "gdcmTableEntry.h"
#include "gdcmTag.h"
```

```
#include <map>
```

Include dependency graph for gdcmTable.h:



Classes

- class gdcm::Table

  Table.

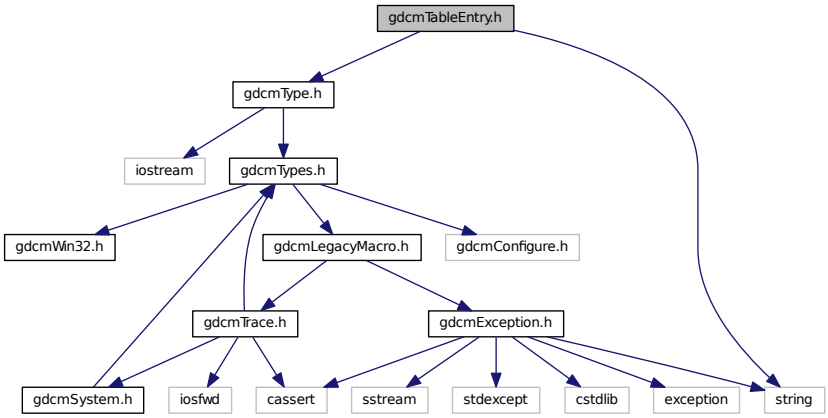
Namespaces

- gdcm

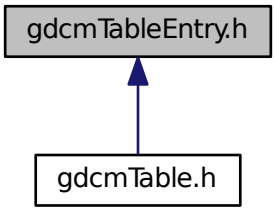
11.246 gdcmTableEntry.h File Reference

```
#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.

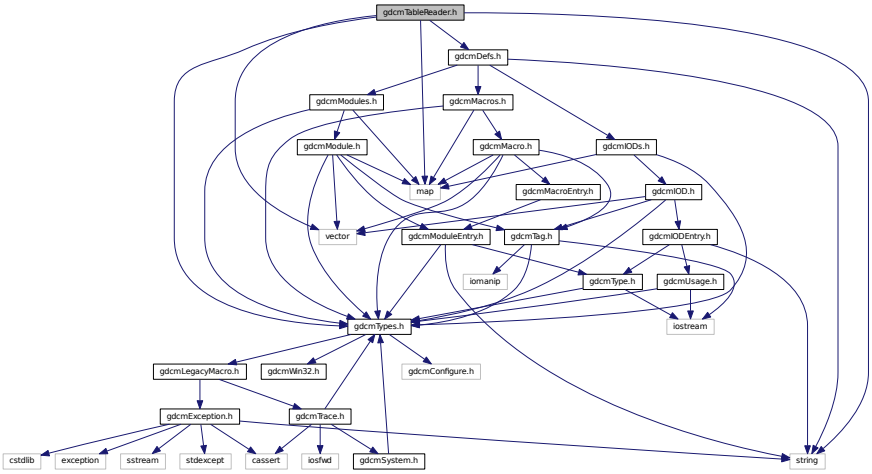
Namespaces

- gdcm

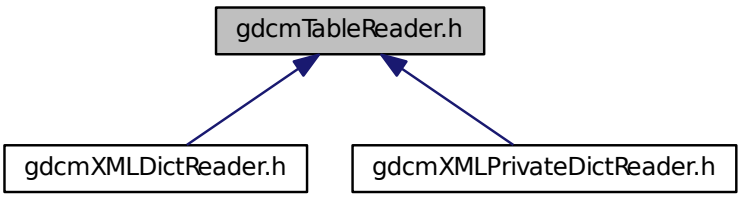
11.247 gdcmTableReader.h File Reference

```
#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.

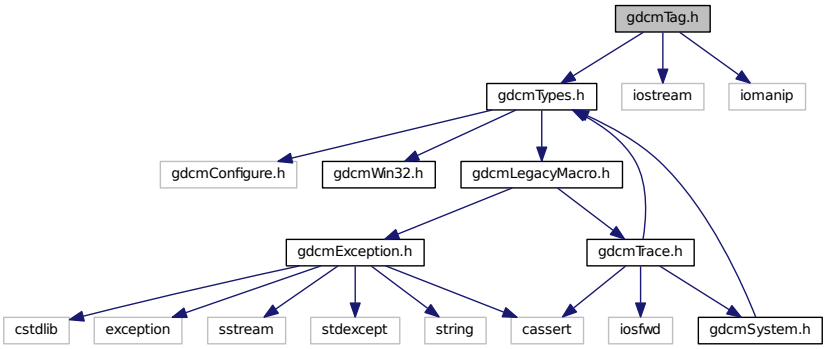
Namespaces

- gdcm

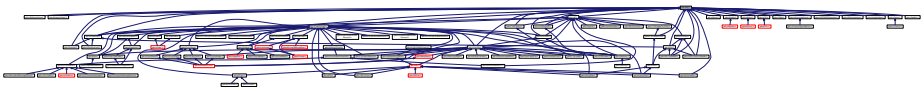
11.248 gdcmTag.h File Reference

```
#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

- gdcm

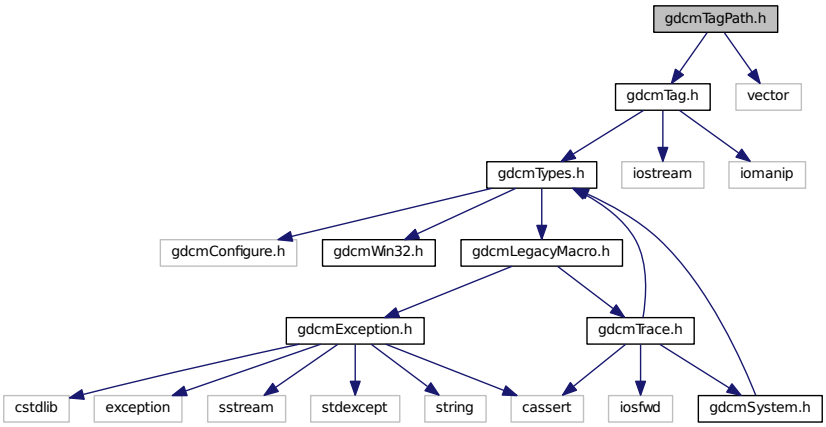
Functions

- std::ostream & gdcm::operator<< (std::ostream &_os, const Tag &_val)
- std::istream & gdcm::operator>> (std::istream &_is, Tag &_val)

11.249 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.

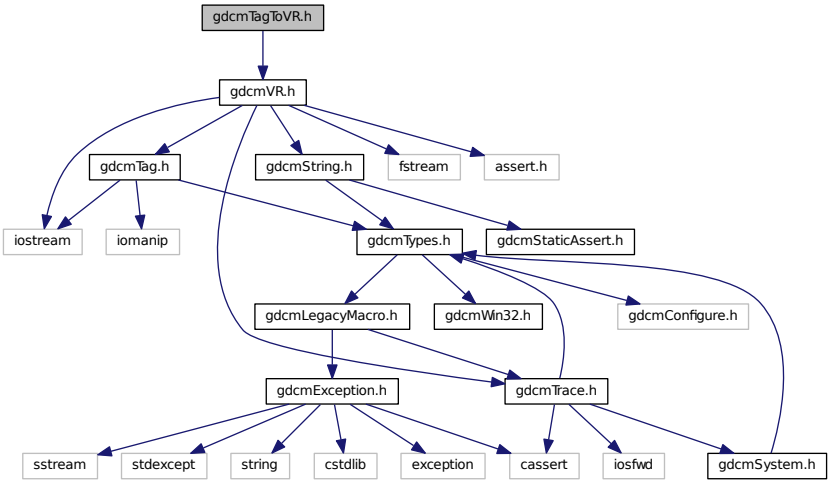
Namespaces

- gdcm

11.250 gdcmTagToVR.h File Reference

```
#include "gdcmVR.h"
```

Include dependency graph for gdcmTagToVR.h:



Namespaces

- gdcm

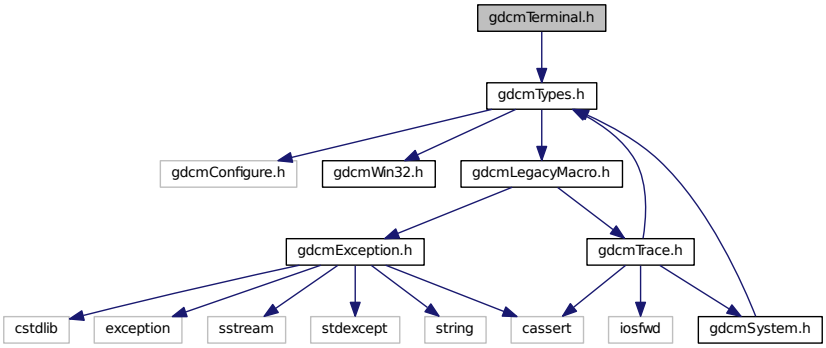
Functions

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

11.251 gdcmTerminal.h File Reference

```
#include "gdcmTypes.h"
```

Include dependency graph for gdcmTerminal.h:



Namespaces

- gdcm
- 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 }

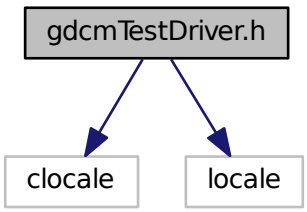
Functions

- 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)

11.252 gdcmTestDriver.h File Reference

```
#include <clocale>
#include <locale>
```

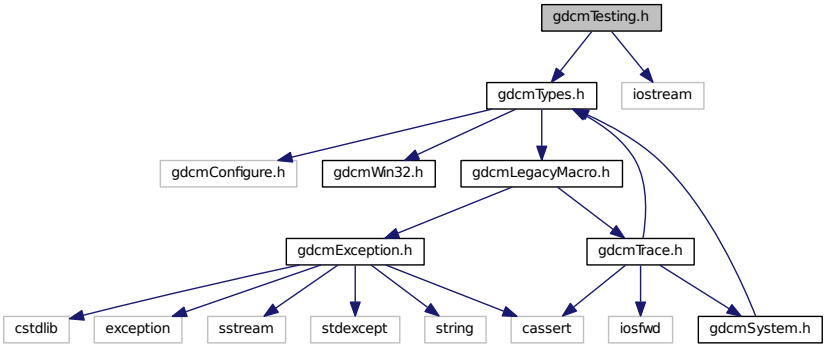
Include dependency graph for gdcmTestDriver.h:



11.253 gdcmTesting.h File Reference

```
#include "gdcmTypes.h"
#include <iostream>
```

Include dependency graph for gdcmTesting.h:



Classes

- class gdcm::Testing

  class for testing

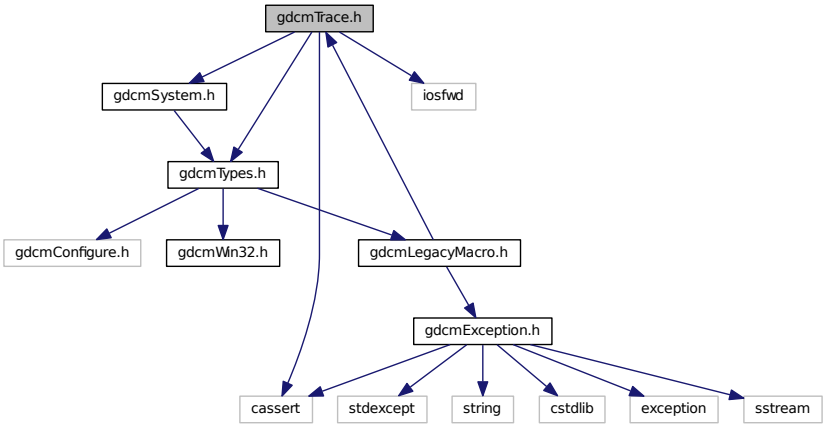
Namespaces

- gdcm

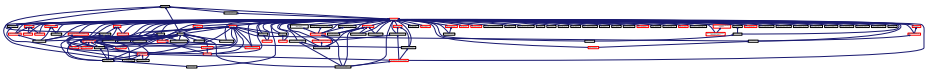
11.254 gdcmTrace.h File Reference

```
#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

- 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.

11.254.1 Macro Definition Documentation

11.254.1.1 GDCM_FUNCTION

```
#define GDCM_FUNCTION "<unknown>"
```

11.254.1.2 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)

Referenced by gdcm::DataElement::GetValue(), gdcm::SequenceOfFragments::ReadValue(), gdcm::DataSet::↩ Replace(), gdcm::DataSet::ReplaceEmpty(), and gdcm::VR::Write().

11.254.1.3 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 );                                                   \
   }                                                                 \
}
```

Assert.

Parameters

arg	argument to test An easy solution to pass also a message is to do: gdcmAssertMacro("my message" && 2 < 3)

Referenced by gdcm::PixelFormat::SetSamplesPerPixel().

11.254.1.4 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;                      \
   }                                                                 \
}
```

Debug.

Parameters

msg	message part

Referenced by gdcm::ByteValue::ByteValue(), gdcm::OpenSSLCryptoFactory::OpenSSLCryptoFactory(), gdcm::OpenSSLP7CryptoFactory::OpenSSLP7CryptoFactory(), gdcm::BasicOffsetTable::Read(), gdcm::Item::Read(), gdcm::SequenceOfItems::Read(), gdcm::VR::Read(), gdcm::SequenceOfFragments::ReadPreValue(), and gdcm::SequenceOfFragments::ReadValue().

11.254.1.5 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;                                \
   }                                                                 \
}
```

Error this is pretty bad, more than just warning It could mean lost of data, something not handle...

Parameters

msg	second message part

Referenced by gdcm::CryptoFactory::CryptoFactory(), gdcm::CommandDataSet::Insert(), gdcm::FileMetaInformation::Insert(), gdcm::DataSet::Insert(), gdcm::Item::Read(), and gdcm::Fragment::ReadBacktrack().

11.254.1.6 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;                                \
   }                                                                 \
}
```

Warning.

Parameters

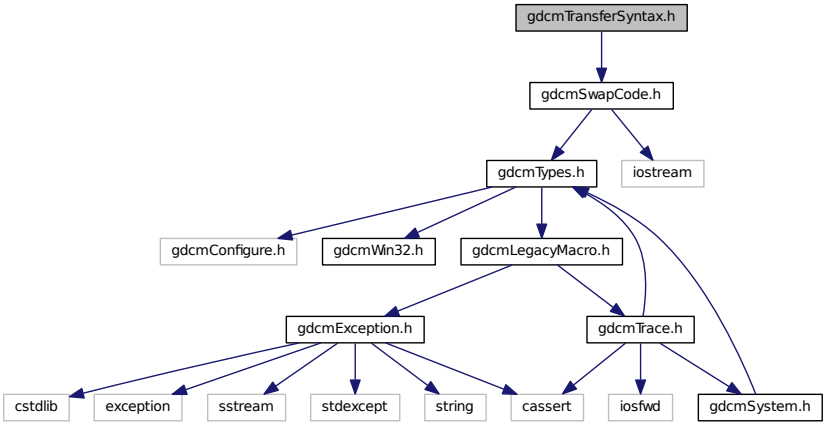
msg	message part

Referenced by gdcm::DataSet::InsertDataElement(), gdcm::Item::Read(), gdcm::SequenceOfItems::Read(), gdcm::Fragment::ReadBacktrack(), gdcm::Fragment::ReadValue(), gdcm::SequenceOfFragments::ReadValue(), gdcm::OpenSSLP7CryptographicMessageSyntax::SetPassword(), and gdcm::Item::Write().

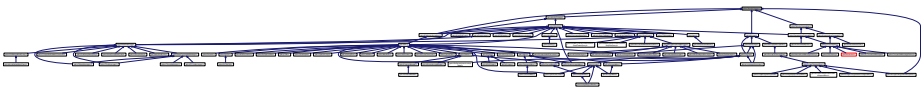
11.255 gdcmTransferSyntax.h File Reference

```
#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

- gdcm

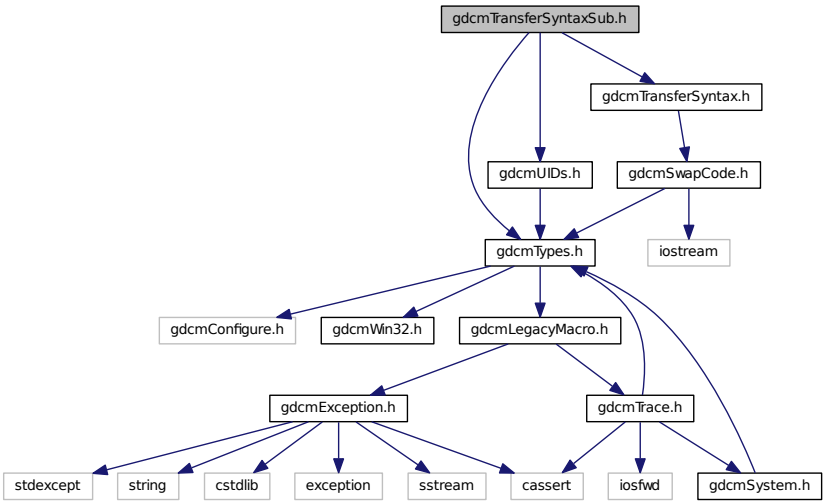
Functions

- std::ostream & gdcm::operator<< (std::ostream &_os, const TransferSyntax &ts)

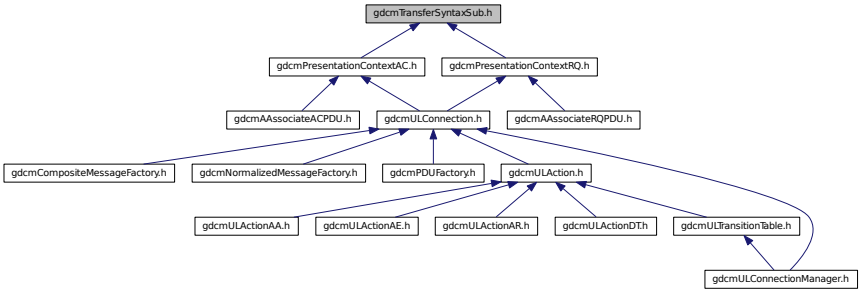
11.256 gdcmTransferSyntaxSub.h File Reference

```
#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.

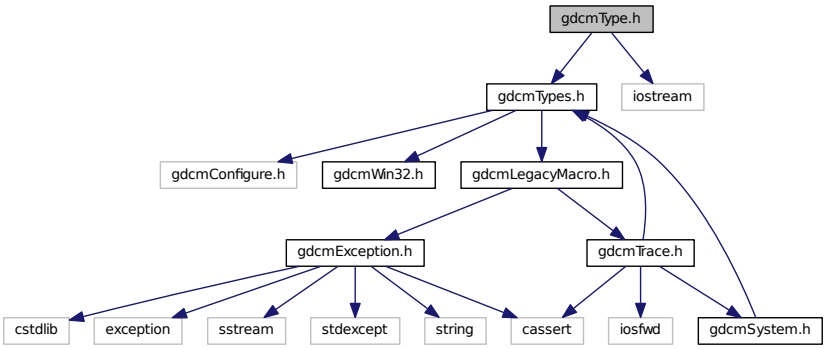
Namespaces

- gdcm
- gdcm::network

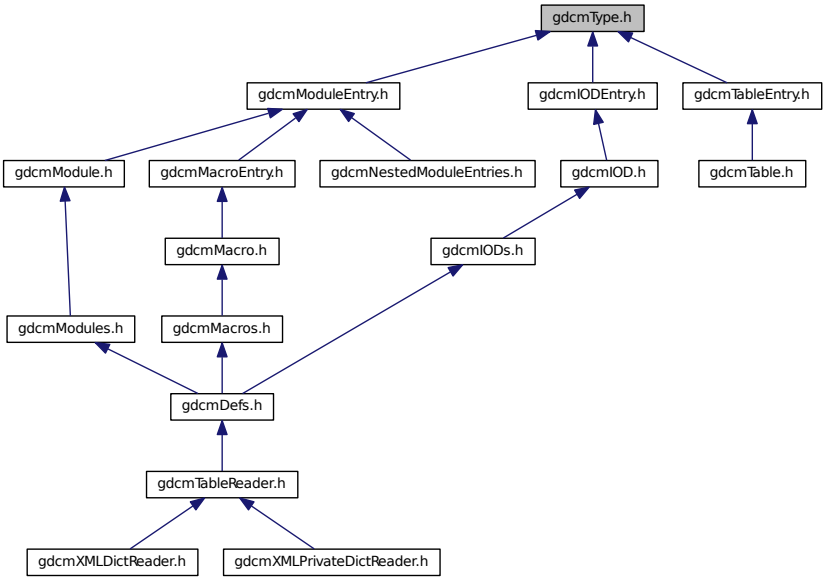
11.257 gdcmType.h File Reference

```
#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

- gdcm

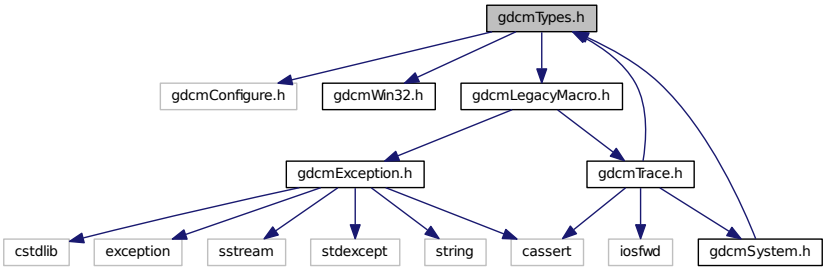
Functions

- std::ostream & gdcm::operator<< (std::ostream &_os, const Type &val)

11.258 gdcmTypes.h File Reference

```
#include "gdcmConfigure.h"
#include "gdcmWin32.h"
#include "gdcmLegacyMacro.h"
```

Include dependency graph for gdcmTypes.h:

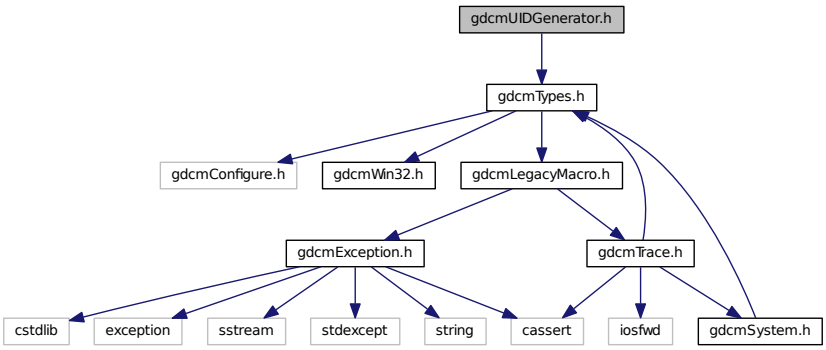


This graph shows which files directly or indirectly include this file:

11.259 gdcmUIDGenerator.h File Reference

```
#include "gdcmTypes.h"
```

Include dependency graph for gdcmUIDGenerator.h:



Classes

- class gdcm::UIDGenerator

  Class for generating unique UID.

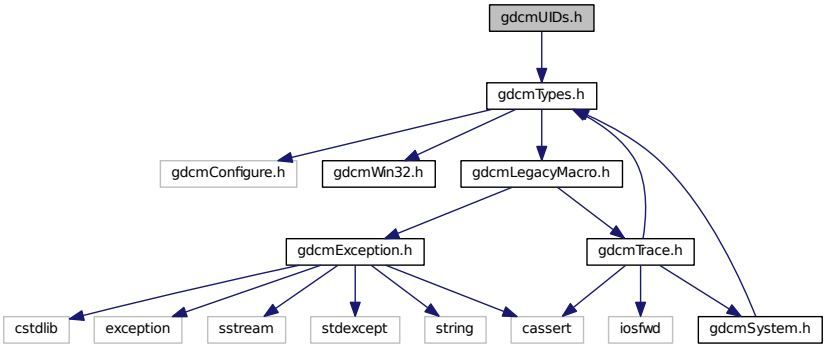
Namespaces

- gdcm

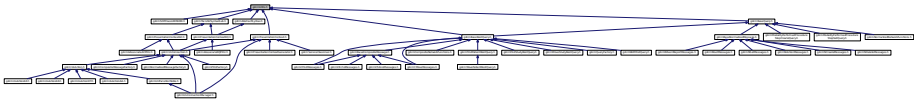
11.260 gdcmUIDs.h File Reference

```
#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

- gdcm

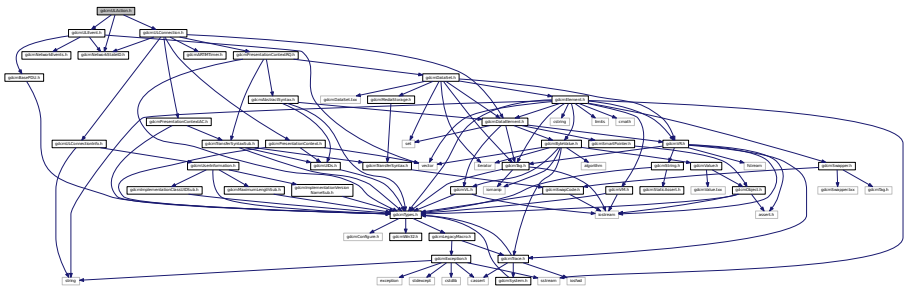
Functions

- std::ostream & gdcm::operator<< (std::ostream &_os, const UIDs &uid)

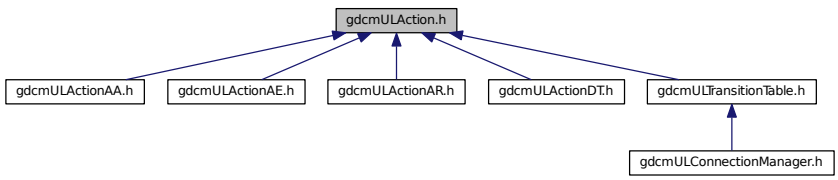
11.261 gdcmULAction.h File Reference

```
#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.

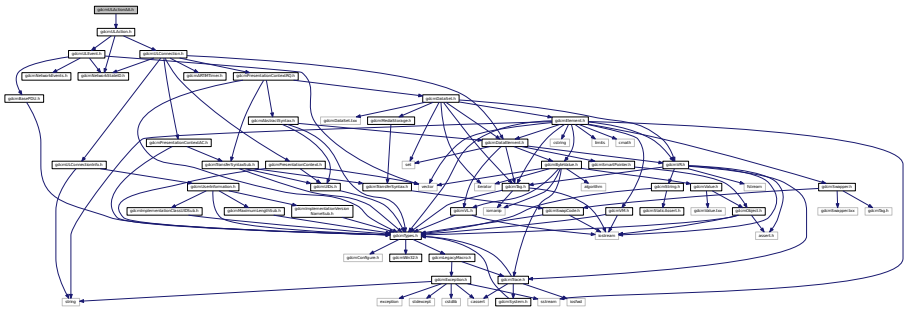
Namespaces

- gdcm
- gdcm::network

11.262 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

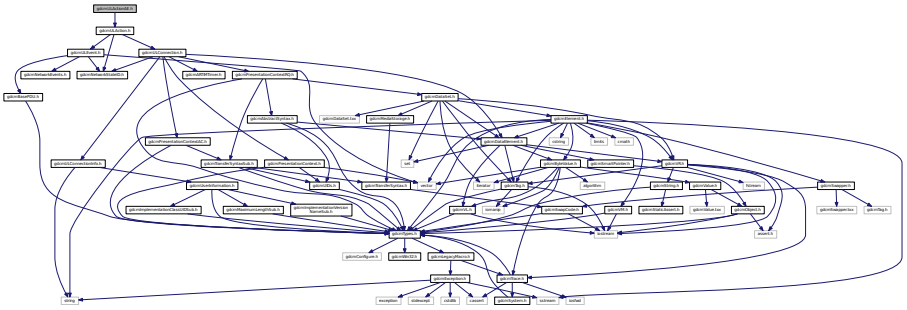
Namespaces

- gdcm
- gdcm::network

11.263 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

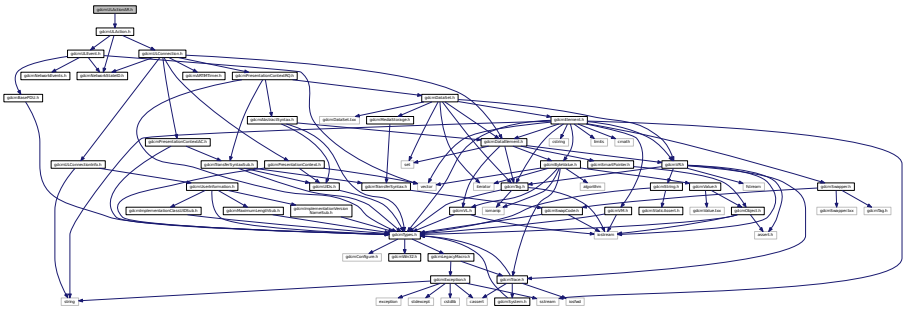
Namespaces

- gdcm
- gdcm::network

11.264 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

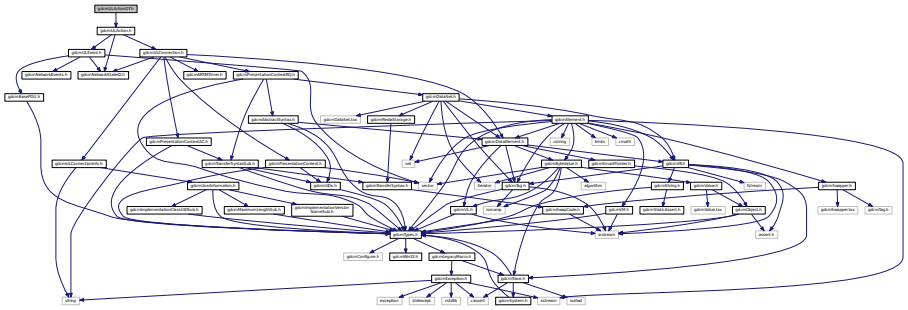
Namespaces

- gdcm
- gdcm::network

11.265 gdcmULActionDT.h File Reference

```
#include "gdcmULAction.h"
```

Include dependency graph for gdcmULActionDT.h:



Classes

- class gdcm::network::ULActionDT1
- class gdcm::network::ULActionDT2

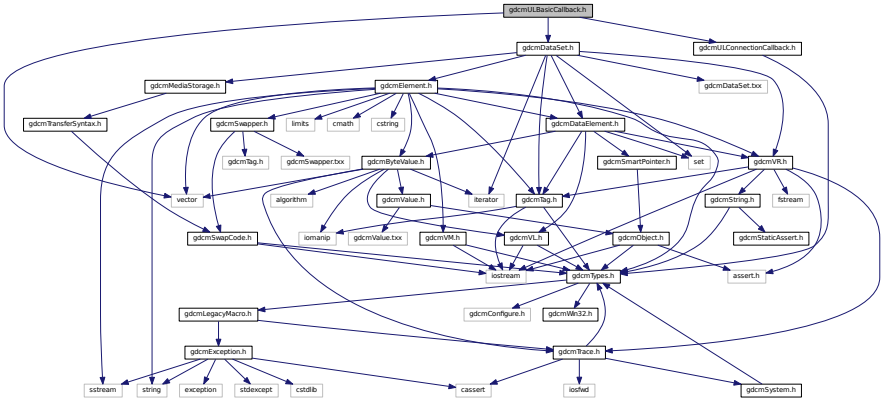
Namespaces

- gdcm
- gdcm::network

11.266 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

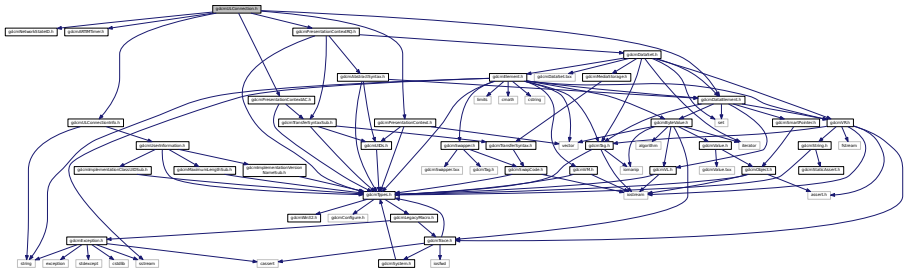
- gdcm
- gdcm::network

11.267 gdcmULConnection.h File Reference

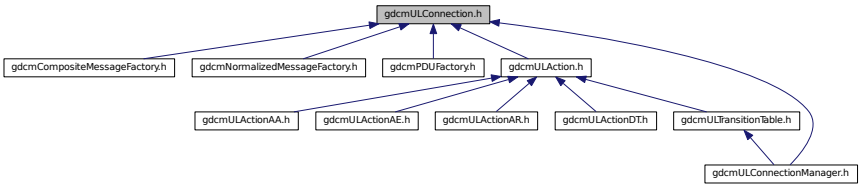
```
#include "gdcmNetworkStateID.h"
#include "gdcmARTIMTimer.h"
#include "gdcmULConnectionInfo.h"
#include "gdcmPresentationContextRQ.h"
#include "gdcmDataElement.h"
#include "gdcmPresentationContextAC.h"
```

```
#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.

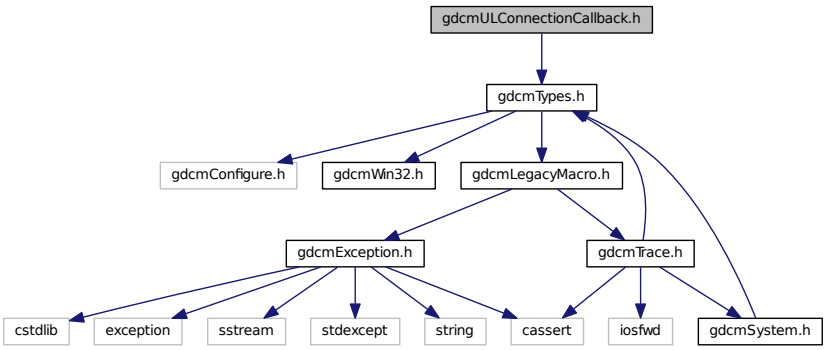
Namespaces

- gdcm
- gdcm::network

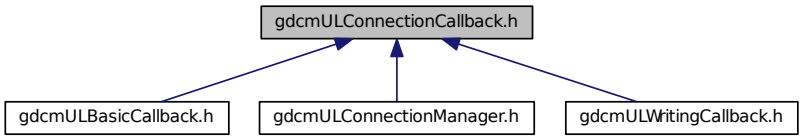
11.268 gdcmULConnectionCallback.h File Reference

```
#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

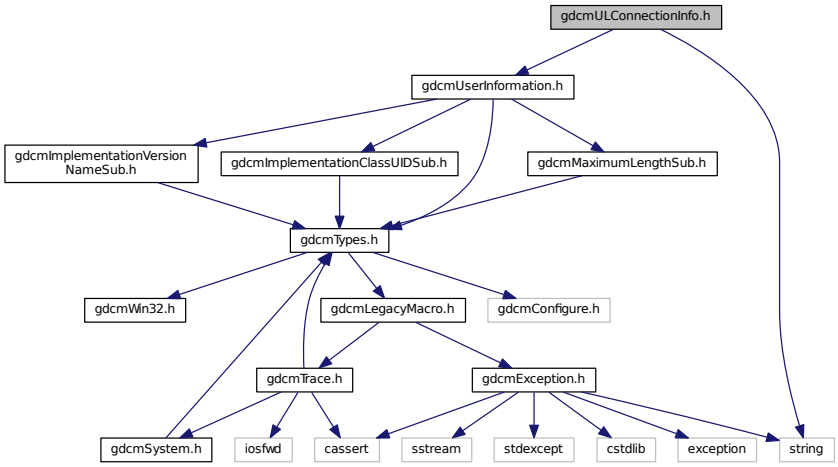
Namespaces

- gdcm
- gdcm::network

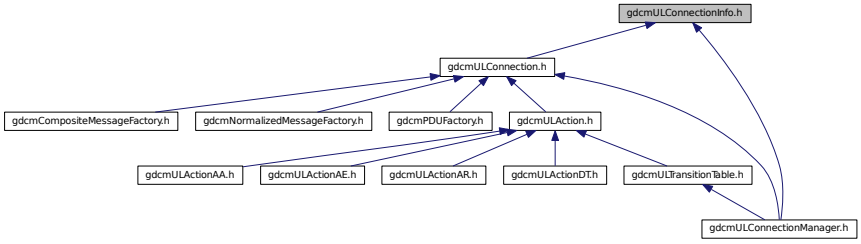
11.269 gdcmULConnectionInfo.h File Reference

```
#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.

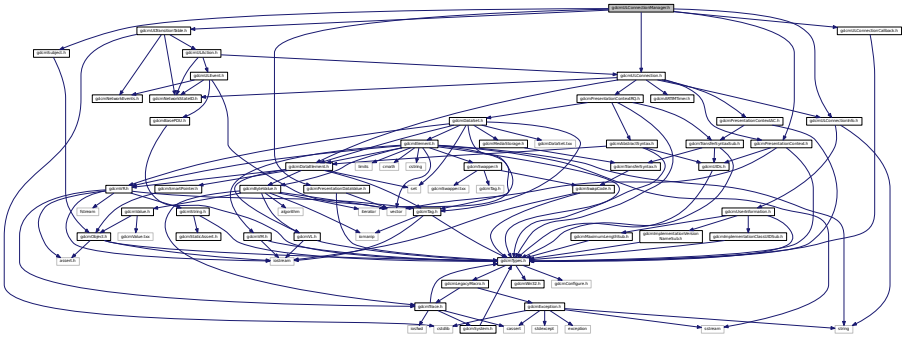
Namespaces

- gdcm
- gdcm::network

11.270 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

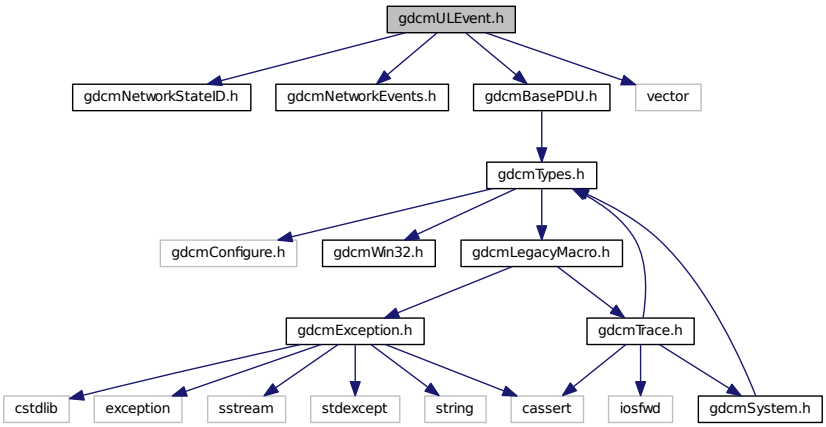
- gdcm
- gdcm::network

11.271 gdcmULEvent.h File Reference

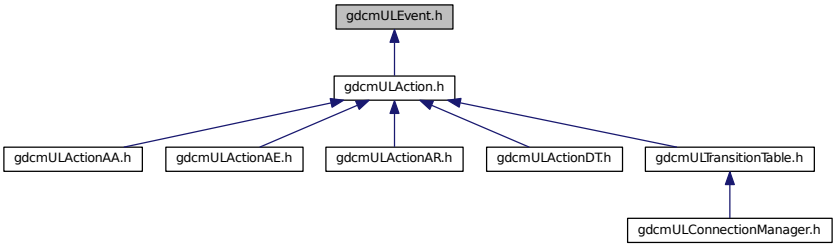
```
#include "gdcmNetworkStateID.h"
#include "gdcmNetworkEvents.h"
#include "gdcmBasePDU.h"
```

```
#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.

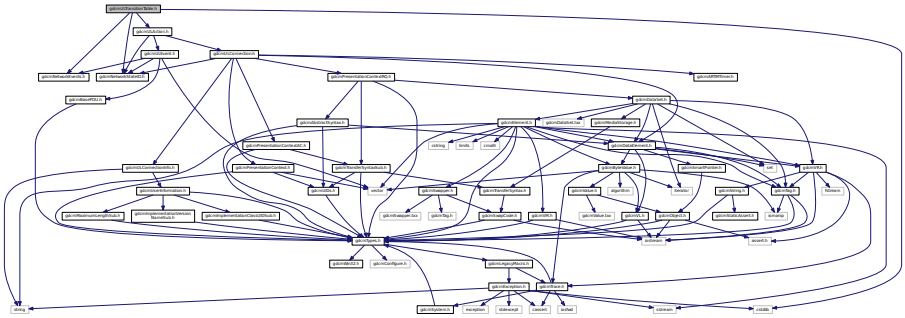
Namespaces

- gdcm
- gdcm::network

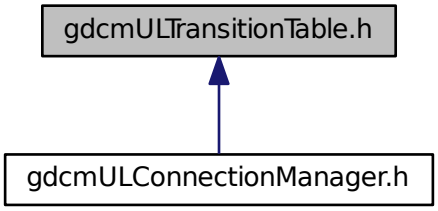
11.272 gdcmULTransitionTable.h File Reference

```
#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

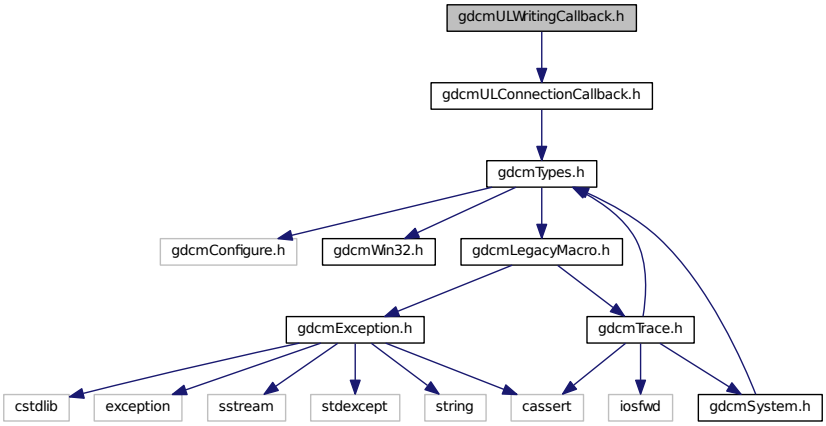
  ULTransitionTable The transition table of all the ULEvents, new ULActions, and ULStates.

Namespaces

- gdcm
- gdcm::network

11.273 gdcmULWritingCallback.h File Reference

```
#include "gdcmULConnectionCallback.h"
```
Include dependency graph for gdcmULWritingCallback.h:



Classes

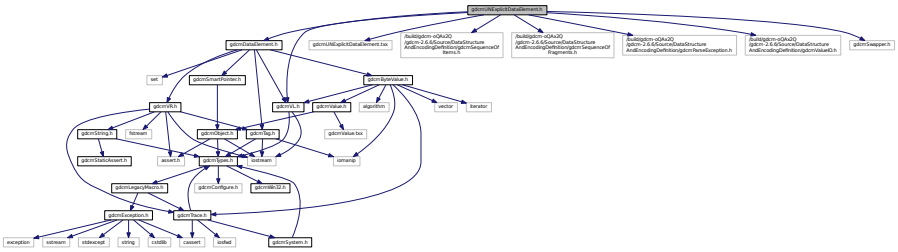
- class gdcm::network::ULWritingCallback

Namespaces

- gdcm
- gdcm::network

11.274 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.

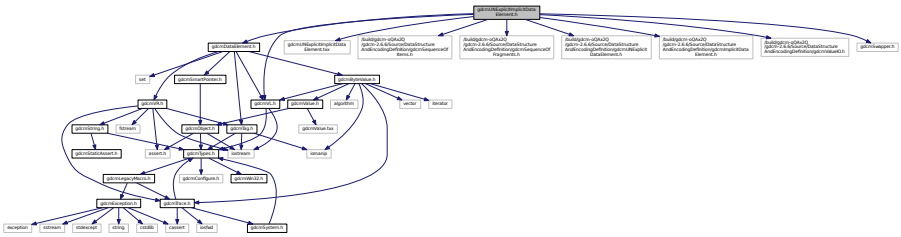
Namespaces

- gdcm

11.275 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.

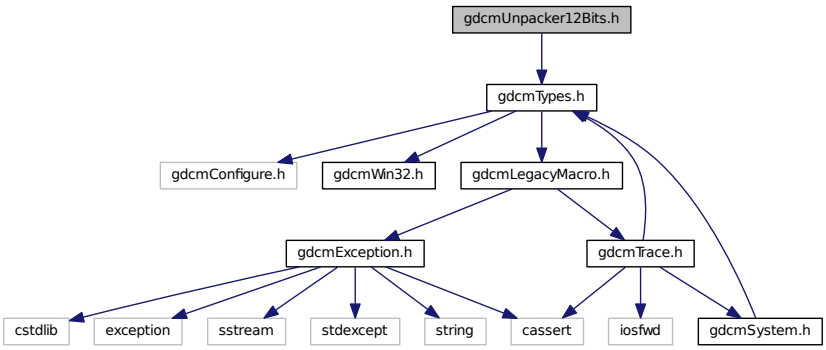
Namespaces

- gdcm

11.276 gdcmUnpacker12Bits.h File Reference

```
#include "gdcmTypes.h"
```

Include dependency graph for gdcmUnpacker12Bits.h:



Classes

- class gdcm::Unpacker12Bits

  Pack/Unpack 12 bits pixel into 16bits.

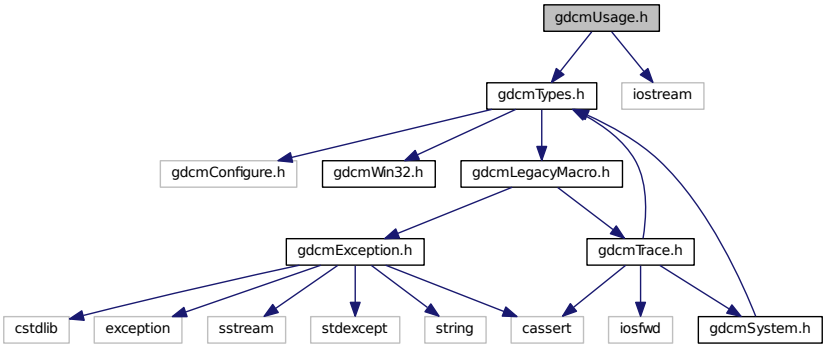
Namespaces

- gdcm

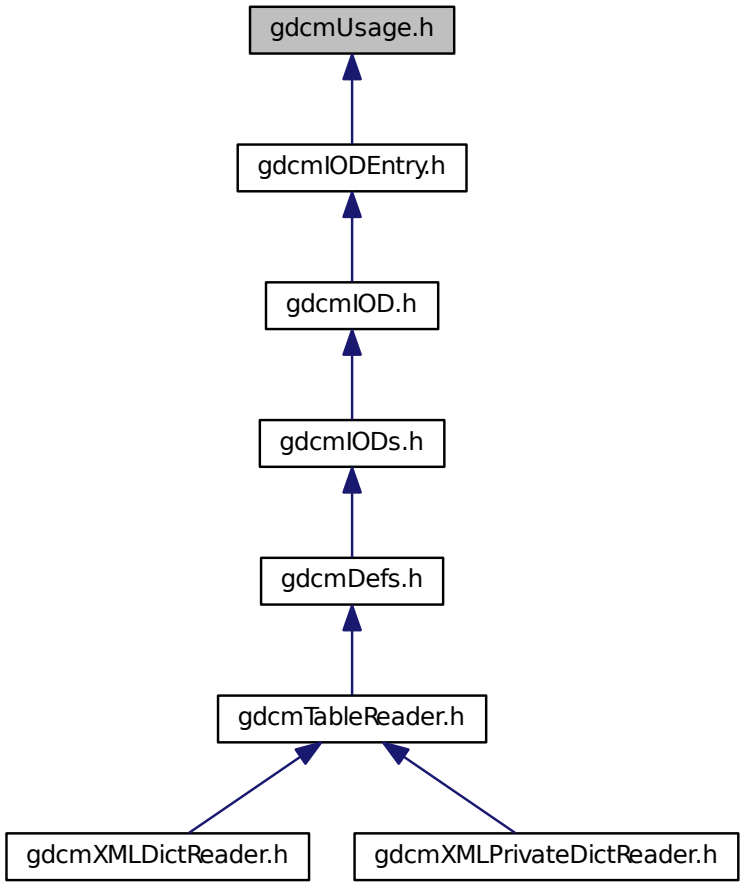
11.277 gdcmUsage.h File Reference

```
#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

- gdcm

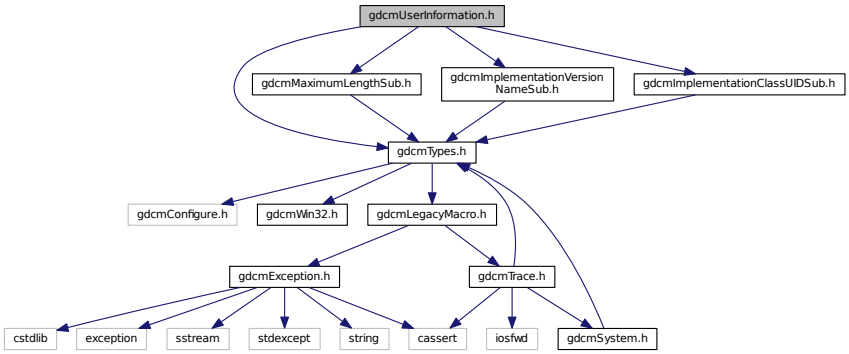
Functions

- std::ostream & gdcm::operator<< (std::ostream &_os, const Usage &val)

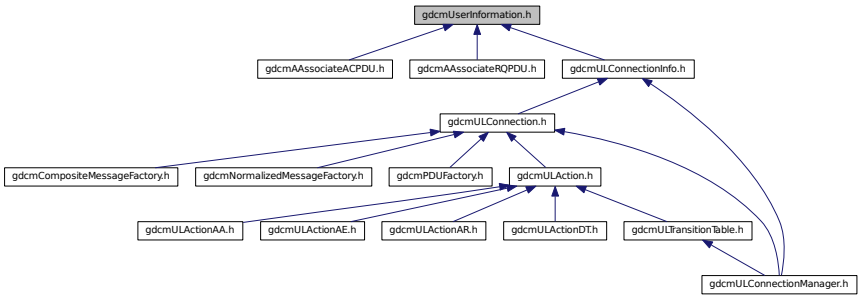
11.278 gdcmUserInformation.h File Reference

```
#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.

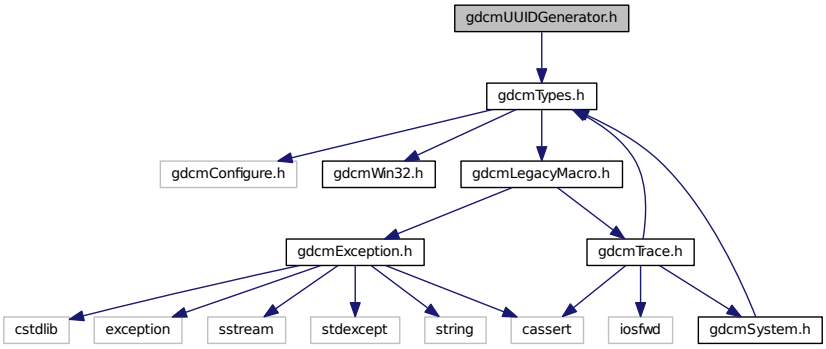
Namespaces

- gdcm
- gdcm::network

11.279 gdcmUUIDGenerator.h File Reference

```
#include "gdcmTypes.h"
```

Include dependency graph for gdcmUUIDGenerator.h:



Classes

- class gdcm::UUIDGenerator

  Class for generating unique UUID.

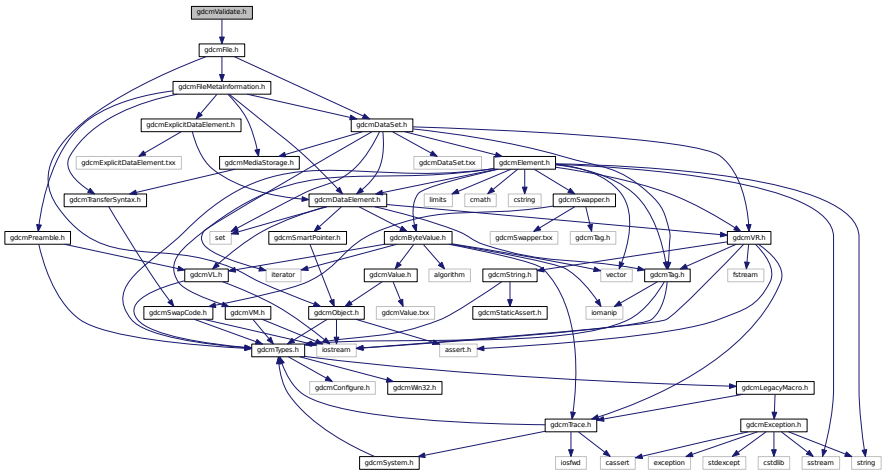
Namespaces

- gdcm

11.280 gdcmValidate.h File Reference

```
#include "gdcmFile.h"
```

Include dependency graph for gdcmValidate.h:



Classes

- class gdcm::Validate

  Validate class.

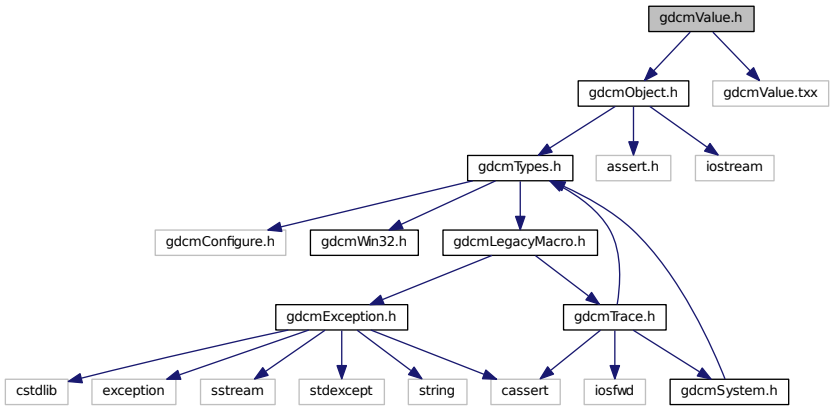
Namespaces

- gdcm

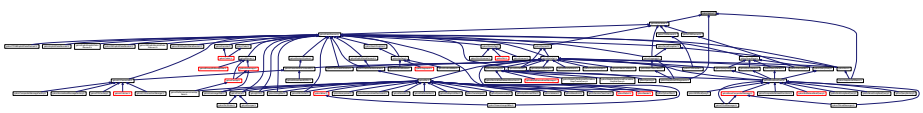
11.281 gdcmValue.h File Reference

```
#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.

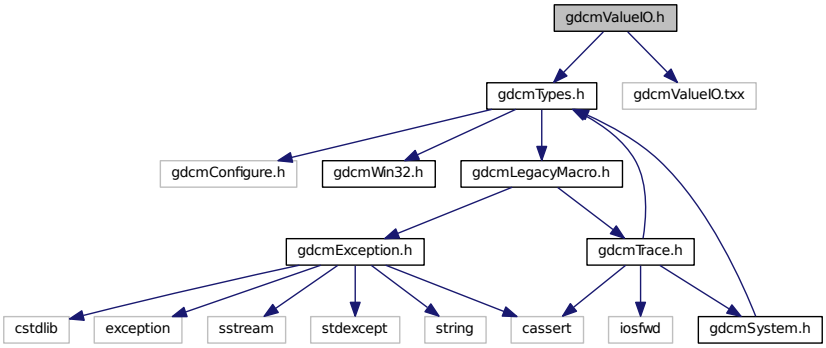
Namespaces

- gdcm

11.282 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.

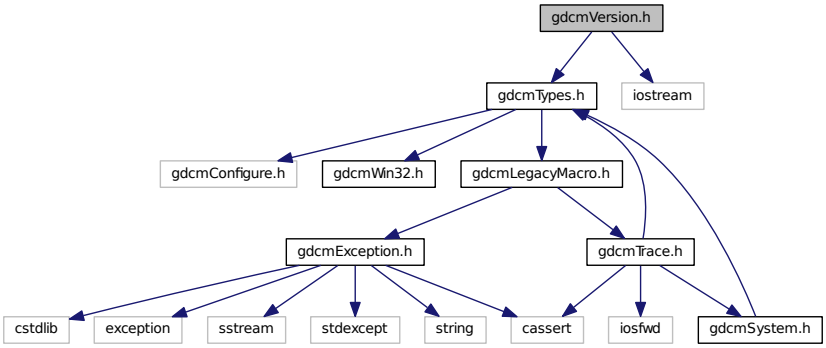
Namespaces

- gdcm

11.283 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

- gdcm

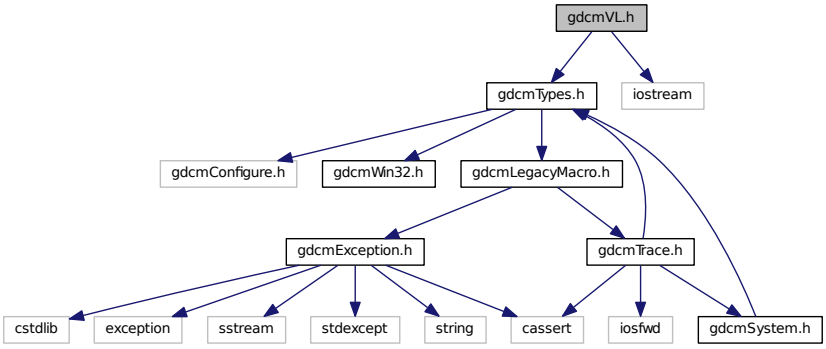
Functions

- std::ostream & gdcm::operator<< (std::ostream &os, const Version &v)

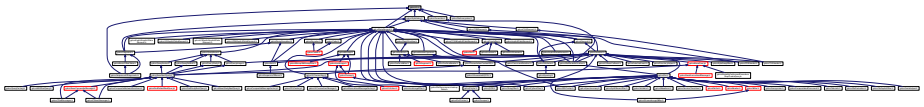
11.284 gdcmVL.h File Reference

```
#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

- gdcm

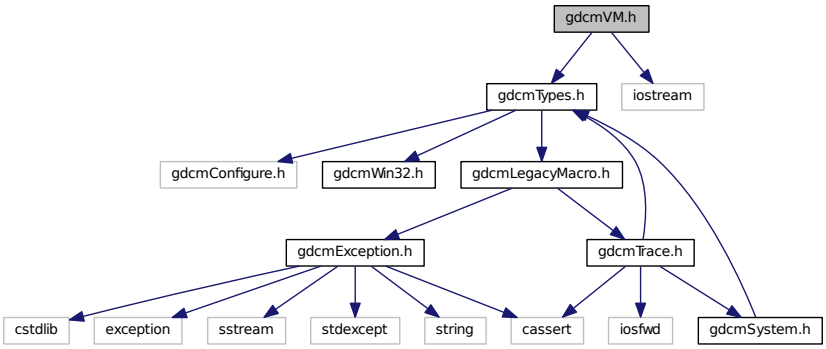
Functions

- std::ostream & gdcm::operator<< (std::ostream &os, const VL &val)

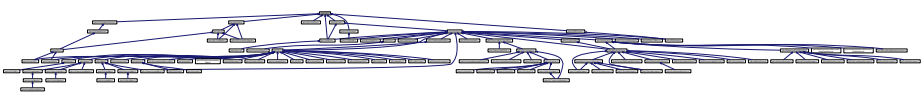
11.285 gdcmVM.h File Reference

```
#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.

- struct gdcm::VMToLength< T >

Namespaces

- gdcm

Macros

- #define TYPETOLENGTH(type, length)

Functions

- std::ostream & gdcm::operator<< (std::ostream &_os, const VM &_val)

11.285.1 Macro Definition Documentation

11.285.1.1 TYPETOLENGTH

```
#define TYPETOLENGTH(
              type,
              length )
```

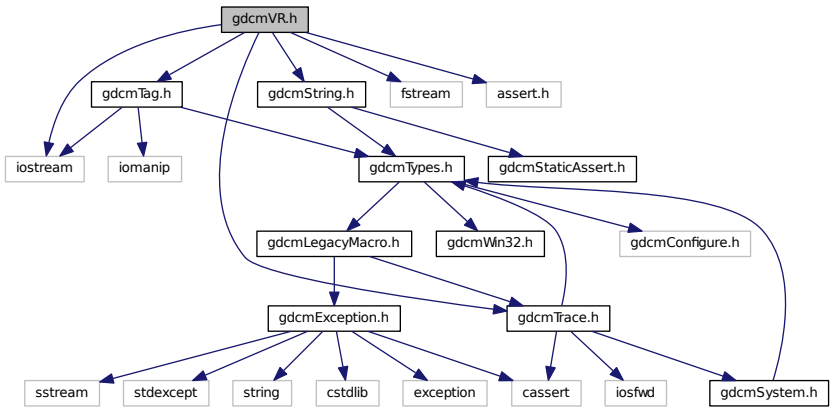
Value:

```
template<> struct VMToLength<VM::type> \
  { enum { Length = length }; };
```

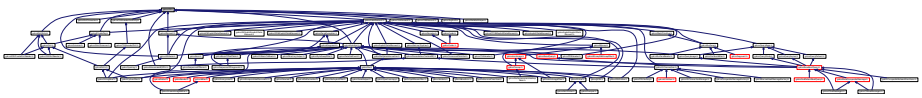
11.286 gdcmVR.h File Reference

```
#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.
- struct gdcm::VRToEncoding< T >
- struct gdcm::VRToType< T >

Namespaces

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

Functions

- std::ostream & gdcm::operator<< (std::ostream &_os, const VR &val)
- std::ostream & gdcm::operator<< (std::ostream &_os, const UI &_val)
- gdcm::TYPETOENCODING (SQ, VRBINARY, unsigned char) TYPETOENCODING(UN

Variables

- gdcm::VRBINARY

11.286.1 Macro Definition Documentation

11.286.1.1 TYPETOENCODING

```
#define TYPETOENCODING(
              type,
              rep,
              rtype )
```

Value:

```
template<> struct VRToEncoding<VR::type>    \
  { enum { Mode = VR::rep }; };                \
  template<> struct VRToType<VR::type>        \
  { typedef rtype Type; };
```

11.286.1.2 VRTypeTemplateCase

```
#define VRTypeTemplateCase(
              type )
```

Value:

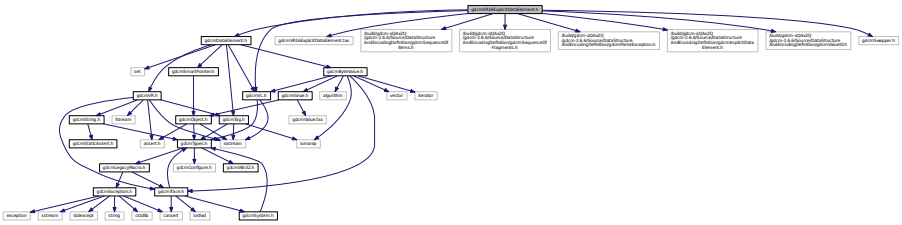
```
case VR::type: \
    return sizeof ( VRToType<VR::type>::Type );
```

Referenced by gdcm::VR::GetSize().

11.287 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.

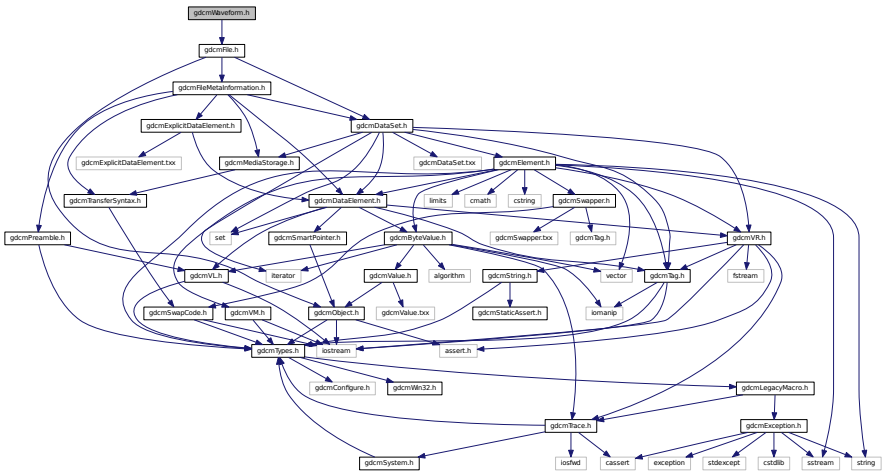
Namespaces

- gdcm

11.288 gdcmWaveform.h File Reference

```
#include "gdcmFile.h"
```

Include dependency graph for gdcmWaveform.h:



Classes

- class gdcm::Waveform

  Waveform class.

Namespaces

- gdcm

11.289 gdcmWin32.h File Reference

This graph shows which files directly or indirectly include this file:

Macros

- #define GDCM_EXPORT

11.289.1 Macro Definition Documentation

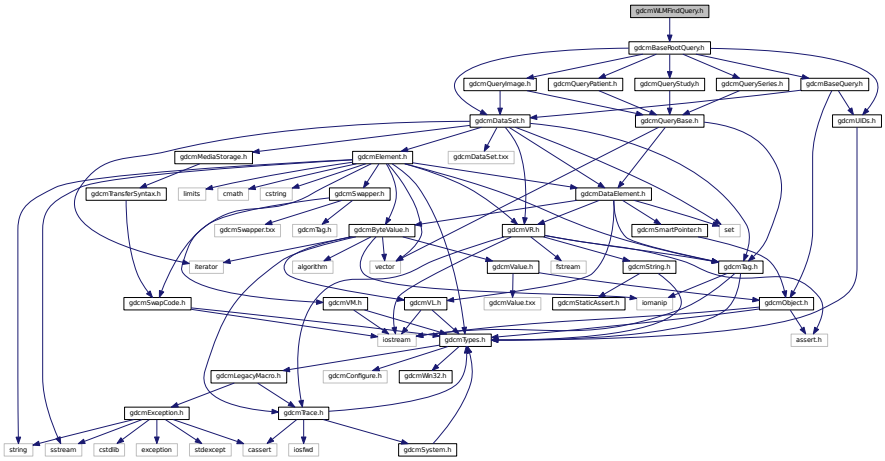
11.289.1.1 GDCM_EXPORT

```
#define GDCM_EXPORT
```

11.290 gdcmWLMFindQuery.h File Reference

```
#include "gdcmBaseRootQuery.h"
```

Include dependency graph for gdcmWLMFindQuery.h:



Classes

- class gdcm::WLMFindQuery

  PatientRootQuery.

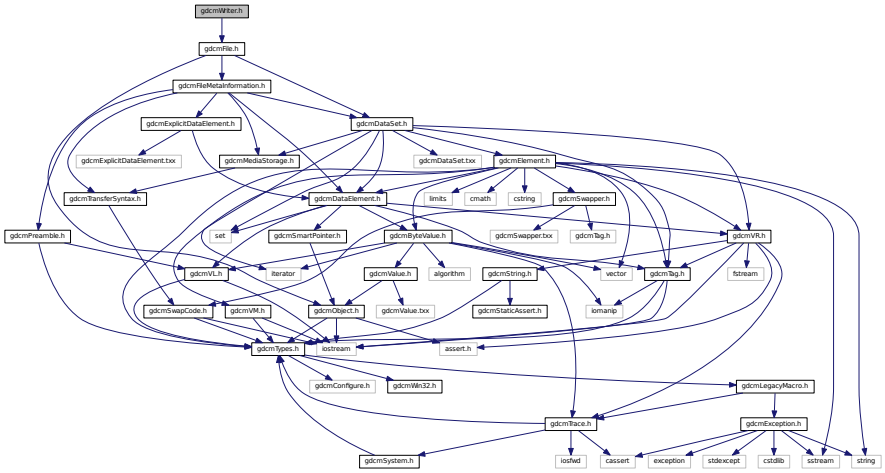
Namespaces

- gdcm

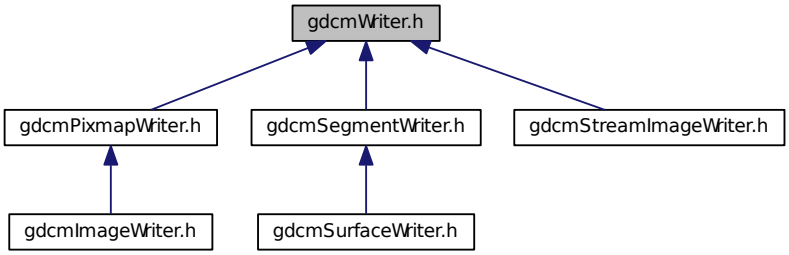
11.291 gdcmWriter.h File Reference

```
#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)

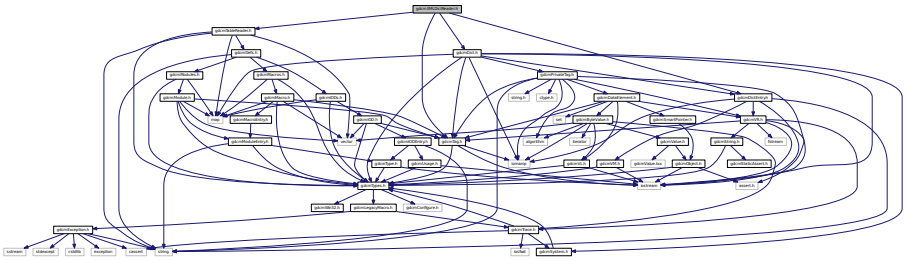
Namespaces

- gdcm

11.292 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.

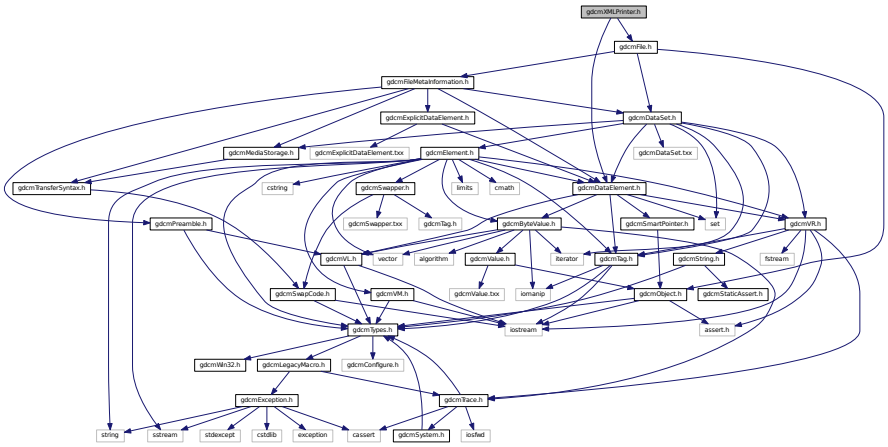
Namespaces

- gdcm

11.293 gdcmXMLPrinter.h File Reference

```
#include "gdcmFile.h"
#include "gdcmDataElement.h"
```

Include dependency graph for gdcmXMLPrinter.h:



Classes

- class gdcm::XMLPrinter

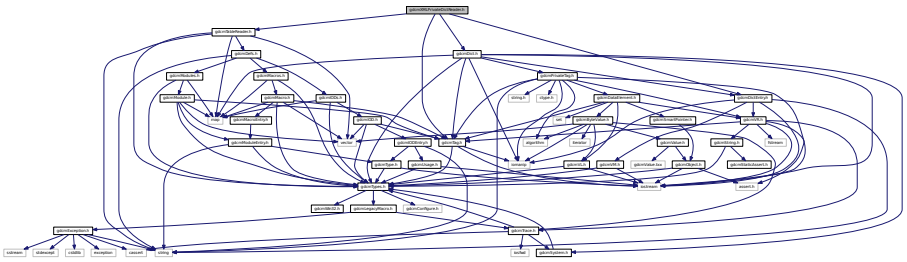
Namespaces

- gdcm

11.294 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

- gdcm

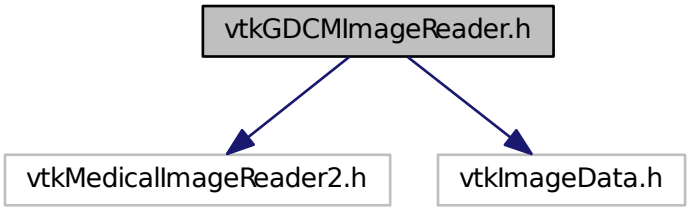
11.295 README.txt File Reference

11.296 TestsList.txt File Reference

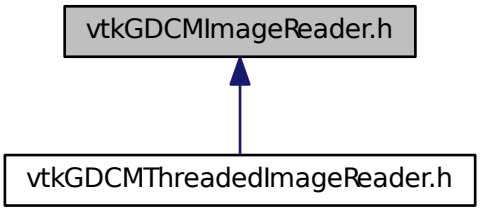
11.297 vtkGDCMImageReader.h File Reference

```
#include "vtkMedicalImageReader2.h"
#include "vtkImageData.h"
```

Include dependency graph for vtkGDCMImageReader.h:



This graph shows which files directly or indirectly include this file:



Classes

- class vtkGDCMImageReader

Namespaces

- gdcm

Macros

- #define VTK_CMYK 8
- #define VTK_INVERSE_LUMINANCE 5
- #define VTK_LOOKUP_TABLE 6
- #define VTK_YBR 7

11.297.1 Macro Definition Documentation

11.297.1.1 VTK_CMYK

```
#define VTK_CMYK 8
```

11.297.1.2 VTK_INVERSE_LUMINANCE

```
#define VTK_INVERSE_LUMINANCE 5
```

11.297.1.3 VTK_LOOKUP_TABLE

```
#define VTK_LOOKUP_TABLE 6
```

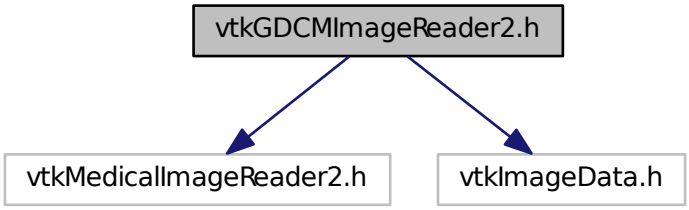
11.297.1.4 VTK_YBR

```
#define VTK_YBR 7
```

11.298 vtkGDCMImageReader2.h File Reference

```
#include "vtkMedicalImageReader2.h"
#include "vtkImageData.h"
```

Include dependency graph for vtkGDCMImageReader2.h:



Classes

- class vtkGDCMImageReader2

Namespaces

- gdcm

Macros

- #define VTK_CMYK 8
- #define VTK_INVERSE_LUMINANCE 5
- #define VTK_LOOKUP_TABLE 6
- #define VTK_YBR 7

11.298.1 Macro Definition Documentation

11.298.1.1 VTK_CMYK

```
#define VTK_CMYK 8
```

11.298.1.2 VTK_INVERSE_LUMINANCE

```
#define VTK_INVERSE_LUMINANCE 5
```

11.298.1.3 VTK_LOOKUP_TABLE

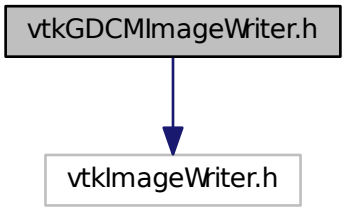
```
#define VTK_LOOKUP_TABLE 6
```

11.298.1.4 VTK_YBR

```
#define VTK_YBR 7
```

11.299 vtkGDCMImageWriter.h File Reference

```
#include "vtkImageWriter.h"
```
Include dependency graph for vtkGDCMImageWriter.h:

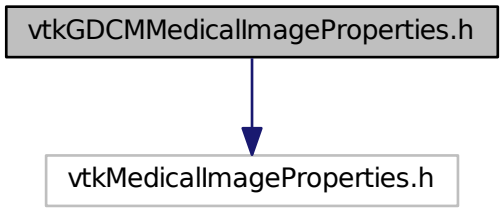


Classes

- class vtkGDCMImageWriter

11.300 vtkGDCMMedicalImageProperties.h File Reference

```
#include "vtkMedicalImageProperties.h"
```
Include dependency graph for vtkGDCMMedicalImageProperties.h:



Classes

- class vtkGDCMMedicalImageProperties

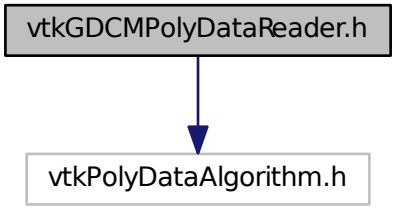
Namespaces

- gdcm

11.301 vtkGDCMPolyDataReader.h File Reference

```
#include "vtkPolyDataAlgorithm.h"
```

Include dependency graph for vtkGDCMPolyDataReader.h:



Classes

- class vtkGDCMPolyDataReader

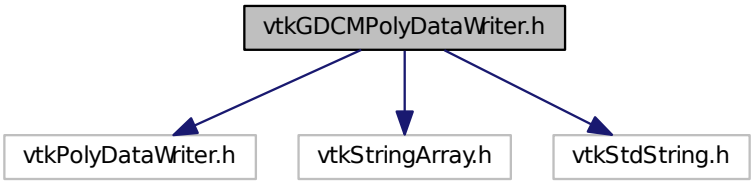
Namespaces

- gdcm

11.302 vtkGDCMPolyDataWriter.h File Reference

```
#include "vtkPolyDataWriter.h"
#include "vtkStringArray.h"
#include "vtkStdString.h"
```

Include dependency graph for vtkGDCMPolyDataWriter.h:



Classes

- class vtkGDCMPolyDataWriter

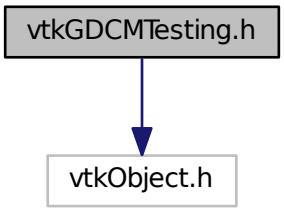
Namespaces

- gdcm

11.303 vtkGDCMTesting.h File Reference

```
#include "vtkObject.h"
```

Include dependency graph for vtkGDCMTesting.h:



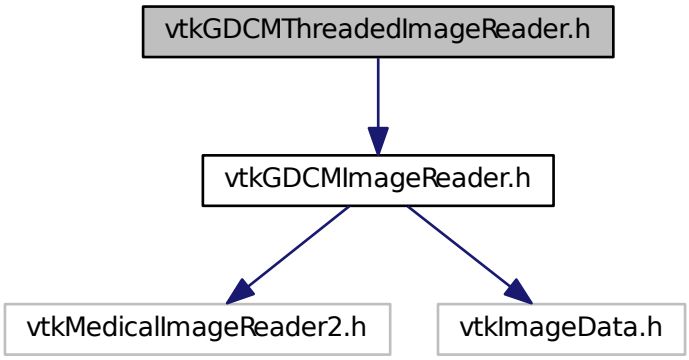
Classes

- class vtkGDCMTesting

11.304 vtkGDCMThreadedImageReader.h File Reference

```
#include "vtkGDCMImageReader.h"
```

Include dependency graph for vtkGDCMThreadedImageReader.h:



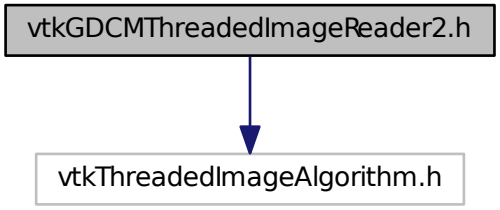
Classes

- class vtkGDCMThreadedImageReader

11.305 vtkGDCMThreadedImageReader2.h File Reference

```
#include "vtkThreadedImageAlgorithm.h"
```

Include dependency graph for vtkGDCMThreadedImageReader2.h:



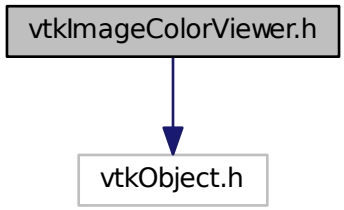
Classes

- class vtkGDCMThreadedImageReader2

11.306 vtkImageColorViewer.h File Reference

```
#include "vtkObject.h"
```

Include dependency graph for vtkImageColorViewer.h:



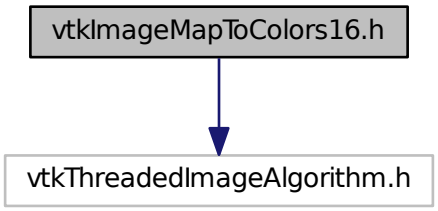
Classes

- class vtkImageColorViewer

11.307 vtkImageMapToColors16.h File Reference

```
#include "vtkThreadedImageAlgorithm.h"
```

Include dependency graph for vtkImageMapToColors16.h:



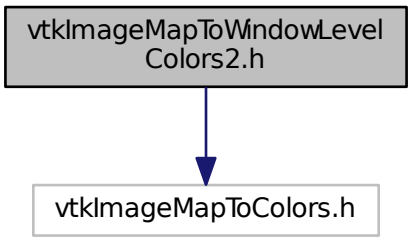
Classes

- class vtkImageMapToColors16

11.308 vtkImageMapToWindowLevelColors2.h File Reference

```
#include "vtkImageMapToColors.h"
```

Include dependency graph for vtkImageMapToWindowLevelColors2.h:



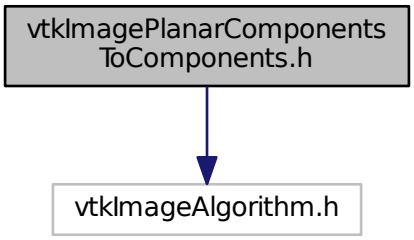
Classes

- class vtkImageMapToWindowLevelColors2

11.309 vtkImagePlanarComponentsToComponents.h File Reference

```
#include "vtkImageAlgorithm.h"
```

Include dependency graph for vtkImagePlanarComponentsToComponents.h:

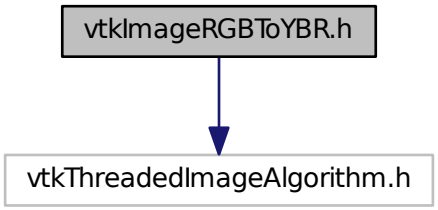


Classes

- class vtkImagePlanarComponentsToComponents

11.310 vtkImageRGBToYBR.h File Reference

```
#include "vtkThreadedImageAlgorithm.h"
```
Include dependency graph for vtkImageRGBToYBR.h:

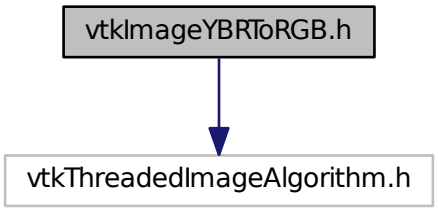


Classes

- class vtkImageRGBToYBR

11.311 vtkImageYBRToRGB.h File Reference

```
#include "vtkThreadedImageAlgorithm.h"
```
Include dependency graph for vtkImageYBRToRGB.h:



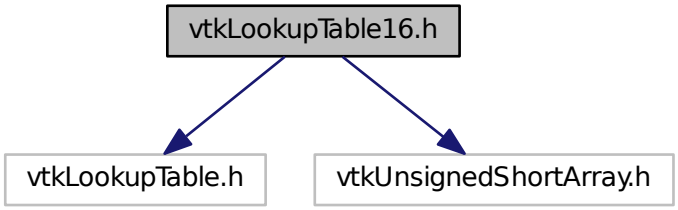
Classes

- class vtkImageYBRToRGB

11.312 vtkLookupTable16.h File Reference

```
#include "vtkLookupTable.h"
#include "vtkUnsignedShortArray.h"
```

Include dependency graph for vtkLookupTable16.h:



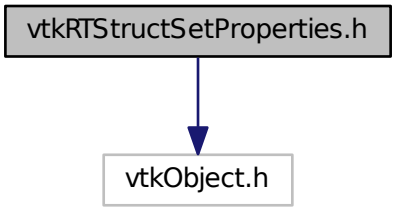
Classes

- class vtkLookupTable16

11.313 vtkRTStructSetProperties.h File Reference

```
#include "vtkObject.h"
```

Include dependency graph for vtkRTStructSetProperties.h:



Classes

- class vtkRTStructSetProperties

Chapter 12

Example Documentation

12.1 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);
  }

}
```

12.2 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;
    }
}
```

12.3 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;
    }
}
```

12.4 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"
```

12.5 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;
}
```

12.6 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;
}
```

12.7 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;
}
```

12.8 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;
}
```

12.9 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;
    }
}
```

12.10 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();

  const gdcm::Image &image = reader.GetImage();
  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;
}
```

12.11 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;
    }

}
```

12.12 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"

#if VTK_MAJOR_VERSION >= 6
#ifndef vtkFloatingPointType
#define vtkFloatingPointType double
#endif

#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;
}
```

12.13 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;
}
```

12.14 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
65     result.shape = d
66     return result
67
68 if __name__ == "__main__":
69   import sys
70   r = gdcm.ImageReader()
71   filename = sys.argv[1]
72   r.SetFileName( filename )
73   if not r.Read():  sys.exit(1)
74   numpy_array = gdcm_to_numpy( r.GetImage() )
75
76   subplot(111)# one plot, on left
77   title(filename)
78
79   imshow(numpy_array, interpolation='bilinear', cmap=cm.jet)
80
81   subplots_adjust(bottom=0.1, right=0.8, top=0.9)
82   cax = axes([0.85, 0.1, 0.075, 0.8])
83   colorbar(cax=cax)
84   title('values')
85   get_current_fig_manager().window.title('plot')
86   show()
```

12.15 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;
}
```

12.16 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
```

12.17 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')
```

12.18 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;
}
```

12.19 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;
}
```

12.20 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;
}
```

12.21 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;
}
```

12.22 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;
}
```

12.23 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;
}
```

12.24 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;
}
```

12.25 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;
}
```

12.26 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.SetByteValue( uid, gdcm.VL(len(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.SetByteValue( uid, gdcm.VL(len(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.SetByteValue( uid, gdcm.VL(len(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()
```

12.27 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;
}
```

12.28 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();
}
```

12.29 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;
    }
}
```

12.30 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 );
      }

    }
}
```

12.31 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   image = gdcm.Image()
35   ir = r.GetImage()
36
37   image.SetNumberOfDimensions( ir.GetNumberOfDimensions() );
38   dims = ir.GetDimensions();
39   print ir.GetDimension(0);
40   print ir.GetDimension(1);
41   print "Dims:",dims
42
43   #  Just for fun:
44   dircos =  ir.GetDirectionCosines()
45   t = gdcm.Orientation.GetType(dircos)
46   l = gdcm.Orientation.GetLabel(t)
47   print "Orientation label:",l
48
49   image.SetDimension(0, ir.GetDimension(0) );
50   image.SetDimension(1, ir.GetDimension(1) );
51
52   pixeltype = ir.GetPixelFormat();
53   image.SetPixelFormat( pixeltype );
54
55   pi = ir.GetPhotometricInterpretation();
56   image.SetPhotometricInterpretation( pi );
57
58   pixeldata = gdcm.DataElement( gdcm.Tag(0x7fe0,0x0010) )
59   str1 = ir.GetBuffer()
60   #print ir.GetBufferLength()
61   pixeldata.SetByteValue( str1, gdcm.VL( len(str1) ) )
62   image.SetDataElement( pixeldata )
63
64   w = gdcm.ImageWriter()
65   w.SetFileName( file2 )
66   w.SetFile( r.GetFile() )
67   w.SetImage( image )
68   if not w.Write():
69     sys.exit(1)
```

12.32 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;
    }
}
```

12.33 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;
    }
}
```

12.34 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 );
      }

    }
}
```

12.35 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;
}
```

12.36 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 = NULL;
#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;
}
```

12.37 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)
```

12.38 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, NULL }
};

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;
}
```

12.39 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;
        }
}
```

12.40 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
```

```
  "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_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,
  param_string,
  param_3, // ??
  param_enum,
} param_type;

static const char *gettypenamefromtype( int i)
{
  const char *ret = NULL;
  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));
    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];
  int8_t boolean;
  int32_t type;
  uint32_t dim;
  uint32_t v4;
  /*int32_t*/ std::streamoff offset;
  param_type gettype() const { return (param_type)type; }
  uint32_t getdim() const { return dim; }
  void read( std::istream & is )
    {
    is.read( name, 32 + 1);
    //assert( name[32] == 0 ); // fails sometimes...
    // 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 ); // sigh
    is.read( (char*)&boolean,1);
    assert( boolean == 0 || boolean == 1 ); // some kind of bool...
    is.read( (char*)&type, sizeof( type ) );
    assert( gettypenamefromtype( type ) );
    is.read( (char*)&dim, sizeof( dim ) );
    is.read( (char*)&v4, sizeof( v4 ) );
    //assert( v4 == 0 ); // always 0 ? sometimes not...
    const std::streamoff cur = is.tellg();
    is.read( (char*)&offset, sizeof( offset ) );
    offset += cur;
    }

  void print( std::ostream & os ) const
    {
    os << name << ",";
    os << (int)boolean << ",";
    os << type << ",";
    os << dim << ",";
    os << v4 << ",";
    os << offset << std::endl;
    }
  void printvalue( std::ostream & os, std::istream & is ) const
    {
    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:
        {
        for( uint32_t idx = 0; idx < dim; ++idx )
          {
          if( idx ) os << ",";
```

```
          int32_t v;
          is.read( (char*)&v, sizeof(v) );
          os << v;
          }
        }
      break;
    case param_string:
        {
        std::string v;
        v.resize( dim );
        is.read( &v[0], dim );
        os << v;
        }
      break;
    case param_enum:
        {
        for( uint32_t idx = 0; idx < dim; ++idx )
          {
          if( idx ) os << ",";
          int32_t v;
          is.read( (char*)&v, sizeof(v) );
          os << v;
          }
        }
      break;
      }

    }
  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 ?
  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 ?
  const gdcm::PrivateTag pt1(0x2005,0x39,"Philips MR Imaging DD 002");
```

```
  if( !ds.FindDataElement( pt1 ) ) return false;
  const gdcm::DataElement &de1 = ds.GetDataElement( pt1 );

  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();
    std::string s1( bv1->GetPointer() , bv1->GetLength() );

    if( s1 == "IEEE_PDF" )
      {
      //  std::cout << "Len= " << bv->GetLength() << std::endl;
#if 0
      std::string fn = gdcm::LOComp::Trim( s.c_str() ); // remove trailing space
      std::ofstream out( fn.c_str() );
      out.write( bv->GetPointer(), bv->GetLength() );
      out.close();
#endif

      std::istringstream is;
      std::string dup( bv->GetPointer(), bv->GetLength() );
      is.str( dup );

      header h;
      h.read( is );
#if 0
      std::cout << s0.c_str() << std::endl;
      h.print( std::cout );
#endif

      assert( is.tellg() == std::streampos(0x20) );
```

```
      is.seekg( 0x20 );

      std::vector< param > params;
      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 ) ) return 1;
    }

  return 0;
}
```

12.41 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 1;
  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 1;
    }
  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 1;
      }
    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;
}
```

12.42 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.size() )
    {
    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;
}
```

12.43 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 = (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;
}
```

12.44 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>

bool DumpToshibaDTI( const char * input, size_t len )
{
  if( len % 2 ) return false;

  std::vector<char> copy( input, input + len );
  std::reverse( copy.begin(), copy.end() );

  std::istringstream is;
  std::string dup( &copy[0], copy.size() );
  is.str( dup );

  gdcm::Reader reader;
  reader.SetStream( is );
  if( !reader.Read() )
    return false;

  //std::cout << reader.GetFile().GetDataSet() << std::endl;
  //gdcm::DictPrinter p;
  gdcm::Printer p;
  p.SetFile( reader.GetFile() );
  p.SetColor( true );
  p.Print( std::cout );

  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;
}
```

12.45 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(0);

  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(0);

  std::cout << "Finished loading data from : " << nfiles << " files" << std::endl;

//  MappingType const &mappings = s.GetMappings();


  sqlite3* db;
  sqlite3_open("./dicom.db", &db);

  if(db == 0)
    {
    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, 0, 0, &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,
      0  // 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(0);

  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;
}
```

12.46 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;
}
```

12.47 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 )
{
  short * buffer = (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;
}
```

12.48 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;
}
```

12.49 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;
    }
}
```

12.50 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;
}
```

12.51 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;
}
```

12.52 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;
    }
}
```

12.53 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");
        }
      }

    }
}
```

12.54 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;
    }
}
```

12.55 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 <openjpeg.h>
#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 "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;
}
```

12.56 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;
    }
}
```

12.57 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;
}
```

12.58 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;
    }
}
```

12.59 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" );
    }
}
```

12.60 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;
    }
}
```

12.61 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;
    }
}
```

12.62 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;
    }
}
```

12.63 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
19 Usage:
20
21  python FindAllPatientName.py
22
23 """
24
25 import gdcm
26
27 # Patient Name
28 tag = gdcm.Tag(0x10,0x10)
29 de = gdcm.DataElement(tag)
30
31 # Search all patient name where string match 'F*'
32 de.SetByteValue('F*',gdcm.VL(2))
33
34 ds = gdcm.DataSet()
```

```
35 ds.Insert(de)
36
37 cnf = gdcm.CompositeNetworkFunctions()
38 theQuery = cnf.ConstructQuery (gdcm.ePatientRootType,gdcm.ePatient,ds)
39
40 #print theQuery.ValidateQuery()
41
42 # prepare the variable for output
43 ret = gdcm.DataSetArrayType()
44
45 # Execute the C-FIND query
46 cnf.CFind('dicom.example.com',11112,theQuery,ret,'GDCM_PYTHON','ANY-SCP')
47
48 for i in range(0,ret.size()):
49   print "Patient #",i
50   print ret[i]
```

12.64 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);

  const gdcm::ByteValue *bv = frag0.GetByteValue();
  const char *ptr = bv->GetPointer();
  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;
  const 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;
}
```

12.65 FixCommaBug.py

```
1
14
15 """
16 Using LC_NUMERIC set to something not compatible with "C" it is possible to write out "," instead of
17 "." as required by the DICOM standard
18 Issue is still current (IMHO) with gdcm 2.0.9
19 """
20
21 import gdcm
22 import sys
23
24 filename = sys.argv[1]
25 outname = sys.argv[2]
26
27 # read
28 r = gdcm.Reader()
29 r.SetFileName( filename )
30 if not r.Read():
31   print "not valid"
32   sys.exit(1)
33
34 file = r.GetFile()
35 dataset = file.GetDataSet()
36
37 ano = gdcm.Anonymizer()
38 ano.SetFile( file )
39
40 tags = [
41 gdcm.Tag(0x0018,0x1164),
42 gdcm.Tag(0x0018,0x0088),
43 gdcm.Tag(0x0018,0x0050),
44 gdcm.Tag(0x0028,0x0030),
45 ]
46
47 for tag in tags:
48   print tag
49   if dataset.FindDataElement( tag ):
50     pixelspacing = dataset.GetDataElement( tag )
51     #print pixelspacing
52     bv = pixelspacing.GetByteValue()
53     str = bv.GetBuffer()
54     #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
```

12.66 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<BYTE> 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 BYTE* pbyteCompressed0 = (const BYTE*)buffer;
while( totalLen > 0 && pbyteCompressed0[totalLen-1] != 0xd9 )
  {
  totalLen--;
  }

JlsParameters metadata;
if (JpegLsReadHeader(buffer, totalLen, &metadata) != OK)
  {
  std::cerr << "Cant parse jpegls" << std::endl;
  return false;
  }

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 = NULL;
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);

    std::vector<BYTE> rgbyteOut;
    //rgbyteOut.resize( image.GetBufferLength() );
    rgbyteOut.resize(params.height *params.width * ((params.bitspersample + 7)
        / 8) * params.components);

    JLS_ERROR result =
      JpegLsDecode(&rgbyteOut[0], rgbyteOut.size(), pbyteCompressed, cbyteCompressed, &params );
    if (result != OK)
      {
      std::cerr << "Could not patch JAI-JPEGLS" << 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;
}
```

12.67 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;
}
```

12.68 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
#if VTK_MAJOR_VERSION < 6
#define vtkFloatingPointType float
#else
#define vtkFloatingPointType double
#endif
#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;
}
```

12.69 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;
}
```

12.70 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\36.10\38.62\44.88\44.88\44.88\45.00\45.00\45.00\45.66\45.66\46.42\39.77\39.77\39.77\39.7
       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( (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/l,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\44.6\28.9\44.2\29.7\43.9\31.5\43.5\33.0\42.8\33.7\42.4\35.2\41.3\38.2\40.4\39.6\39.7
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;
}
```

12.71 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\36.10\38.62\44.88\44.88\44.88\45.00\45.00\45.00\45.66\45.66\46.42\39.77\39.77\39.77\39.7
       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( (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\44.6\28.9\44.2\29.7\43.9\31.5\43.5\33.0\42.8\33.7\42.4\35.2\41.3\38.2\40.4\39.6\39.7
      (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;
}
```

12.72 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;
}
```

12.73 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;
}
```

12.74 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;
}
```

12.75 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)(1 << 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;
}
```

12.76 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;
    }
}
```

12.77 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;
}
```

12.78 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;
}
```

12.79 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;
}
```

12.80 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;
}
```

12.81 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;
```

```
}
```

12.82 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;
}
```

12.83 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;
    }
}
```

12.84 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;
}
```

12.85 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("\\")
```

12.86 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;
}
```

12.87 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;
}
```

12.88 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()
```

12.89 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;
    }
}
```

12.90 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;
    }
}
```

12.91 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;
    }
}
```

12.92 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;
    }
}
```

12.93 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;
    }
}
```

12.94 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 );
    }
}
```

12.95 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;
}
```

12.96 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;
    }
}
```

12.97 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 );
    }
}
```

12.98 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;
    }
}
```

12.99 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.
 */
```

12.100 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))
```

12.101 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;
}
```

12.102 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;
}
```

12.103 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;
}
```

12.104 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;
}
```

12.105 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;
    }
}
```

12.106 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)
```

12.107 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.SetByteValue( mystr, gdcm.VL( len(mystr) ) )
68
69   w = gdcm.Writer()
70   w.SetFile( f )
71   w.SetFileName( file2 )
72   if not w.Write():
73     sys.exit(1)
```

12.108 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)
```

12.109 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;
}
```

12.110 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;
    }
}
```

12.111 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();
    }
}
```

12.112 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;
    }
}
```

12.113 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();
    }
}
```

12.114 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 );
    }
}
```

12.115 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.""TlmapFunctor"">
      {
        <Class> ""TlmapFunctor@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;
    }

  return 0;
}
```

12.116 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;
    }
}
```

12.117 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.SetByteValue("Occupation", gdcm.VL(len("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)
```

12.118 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;
}
```

12.119 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;
}
```

12.120 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"
```

12.121 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()
```

12.122 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
  };

  byte* src = (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;
}
```

12.123 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") )
```

12.124 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 = 0;
  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;
}
```

12.125 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;
}
```

12.126 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;
}
```

12.127 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
```

12.128 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;
}
```

12.129 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;
}
```

12.130 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);
      }
    }
}
```

12.131 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;
}
```

12.132 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)
      {
      std::cerr << "BAD ALLOC Exception caught!" << std::endl;
      }
    catch (...)
      {
      std::cerr << "Exception caught!" << std::endl;
      }
    }

  return 0;
}
```

12.133 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() );
    }
}
```

12.134 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;
}
```

12.135 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;
    }
}
```

12.136 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;
    }
}
```

12.137 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)
```

12.138 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;
    }
}
```

12.139 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 <strstream>
#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();

  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::strstream 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());

        _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;
}
```

12.140 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)
```

12.141 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;

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<0x0028,0x0010> at1;
at1.SetFromDataSet( ds );
gdcm::Attribute<0x0028,0x0011> at2;
at2.SetFromDataSet( ds );

gdcm::DataElement pixeldata( gdcm::Tag(0x7fe0,0x0010) );
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
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 = "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;
}
```

12.142 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;
}
```

12.143 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;
    }
}
```

12.144 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");
      }

    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" );
    }
}
```

12.145 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)
```

12.146 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;
    }
}
```

12.147 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;
    }
}
```

12.148 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;
      }
    }
}
```

12.149 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)
    {
    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;
}
```

12.150 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;
}
```

12.151 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
```

12.152 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;
    }
}
```

12.153 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;
    }
}
```

12.154 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;
}
```

12.155 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;
}
```

12.156 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;
}
```

12.157 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)
```

12.158 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;
}
```

12.159 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;
}
```

12.160 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(1)
    {
    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 );
    }
}
```

12.161 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;
}
```

12.162 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() )
```

Index

~ASN1
gdcm::ASN1, 123
~AnonymizeEvent
gdcm::AnonymizeEvent, 102
~Anonymizer
gdcm::Anonymizer, 107
~Attribute
gdcm::Attribute< Group, Element, TVR, VM::VM1_n >, 143
~AudioCodec
gdcm::AudioCodec, 154
~BaseCompositeMessage
gdcm::network::BaseCompositeMessage, 158
~BaseNormalizedMessage
gdcm::network::BaseNormalizedMessage, 160
~BasePDU
gdcm::network::BasePDU, 162
~BaseQuery
gdcm::BaseQuery, 165
~BaseRootQuery
gdcm::BaseRootQuery, 170
~Bitmap
gdcm::Bitmap, 181
~BitmapToBitmapFilter
gdcm::BitmapToBitmapFilter, 192
~BoxRegion
gdcm::BoxRegion, 194
~ByteSwapFilter
gdcm::ByteSwapFilter, 201
~ByteValue
gdcm::ByteValue, 204
~CAPICryptographicMessageSyntax
gdcm::CAPICryptographicMessageSyntax, 211
~CSAHeader
gdcm::CSAHeader, 260
~Coder
gdcm::Coder, 227
~Command
gdcm::Command, 233
~CommandDataSet
gdcm::CommandDataSet, 235
~CryptoFactory
gdcm::CryptoFactory, 247
~CryptographicMessageSyntax
gdcm::CryptographicMessageSyntax, 249
~Curve
gdcm::Curve, 274
~DICOMDIRGenerator
gdcm::DICOMDIRGenerator, 319
~DataEvent
gdcm::DataEvent, 292
~DataSetEvent
gdcm::DataSetEvent, 306
~Decoder
gdcm::Decoder, 309
~Defs
gdcm::Defs, 312
~DeltaEncodingCodec
gdcm::DeltaEncodingCodec, 316
~DictConverter
gdcm::DictConverter, 326
~DictPrinter
gdcm::DictPrinter, 334
~Dicts
gdcm::Dicts, 336
~DirectionCosines
gdcm::DirectionCosines, 341
~Directory
gdcm::Directory, 344
~Dumper
gdcm::Dumper, 351
~Element
gdcm::Element< TVR, VM::VM1_n >, 358
~Event
gdcm::Event, 381
~Exception
gdcm::Exception, 383
~File
gdcm::File, 392
~FileAnonymizer
gdcm::FileAnonymizer, 397
~FileChangeTransferSyntax
gdcm::FileChangeTransferSyntax, 400
~FileDecompressLookupTable
gdcm::FileDecompressLookupTable, 403
~FileDerivation
gdcm::FileDerivation, 405
~FileExplicitFilter
gdcm::FileExplicitFilter, 409
~FileMetaInformation

gdcm::FileMetaInformation, 413
~FileNameEvent
gdcm::FileNameEvent, 424
~FileStreamer
gdcm::FileStreamer, 433
~FilenameGenerator
gdcm::FilenameGenerator, 427
~Global
gdcm::Global, 448
~GroupDict
gdcm::GroupDict, 451
~IconImageFilter
gdcm::IconImageFilter, 454
~IconImageGenerator
gdcm::IconImageGenerator, 456
~Image
gdcm::Image, 462
~ImageApplyLookupTable
gdcm::ImageApplyLookupTable, 468
~ImageChangePhotometricInterpretation
gdcm::ImageChangePhotometricInterpretation, 471
~ImageChangePlanarConfiguration
gdcm::ImageChangePlanarConfiguration, 475
~ImageChangeTransferSyntax
gdcm::ImageChangeTransferSyntax, 479
~ImageCodec
gdcm::ImageCodec, 484
~ImageConverter
gdcm::ImageConverter, 493
~ImageFragmentSplitter
gdcm::ImageFragmentSplitter, 496
~ImageReader
gdcm::ImageReader, 505
~ImageRegionReader
gdcm::ImageRegionReader, 509
~ImageToImageFilter
gdcm::ImageToImageFilter, 512
~ImageWriter
gdcm::ImageWriter, 515
~JPEG12Codec
gdcm::JPEG12Codec, 543
~JPEG16Codec
gdcm::JPEG16Codec, 546
~JPEG2000Codec
gdcm::JPEG2000Codec, 550
~JPEG8Codec
gdcm::JPEG8Codec, 555
~JPEGCodec
gdcm::JPEGCodec, 559
~JPEGLSCodec
gdcm::JPEGLSCodec, 566
~JSON
gdcm::JSON, 571
~KAKADUCodec
gdcm::KAKADUCodec, 573
~LookupTable
gdcm::LookupTable, 581
~MD5
gdcm::MD5, 593
~MemberCommand
gdcm::MemberCommand, 605
~MeshPrimitive
gdcm::MeshPrimitive, 610
~ModuleEntry
gdcm::ModuleEntry, 623
~Object
gdcm::Object, 659
~OpenSSLCryptographicMessageSyntax
gdcm::OpenSSLCryptographicMessageSyntax, 663
~OpenSSLP7CryptographicMessageSyntax
gdcm::OpenSSLP7CryptographicMessageSyntax, 668
~Orientation
gdcm::Orientation, 671
~Overlay
gdcm::Overlay, 676
~PDBHeader
gdcm::PDBHeader, 695
~PDFCodec
gdcm::PDFCodec, 698
~PGXCodec
gdcm::PGXCodec, 706
~PNMCodec
gdcm::PNMCodec, 736
~PVRGCodec
gdcm::PVRGCodec, 771
~ParseException
gdcm::ParseException, 683
~Parser
gdcm::Parser, 686
~Pixmap
gdcm::Pixmap, 721
~PixmapReader
gdcm::PixmapReader, 726
~PixmapToPixmapFilter
gdcm::PixmapToPixmapFilter, 729
~PixmapWriter
gdcm::PixmapWriter, 732
~Preamble
gdcm::Preamble, 738
~Printer
gdcm::Printer, 759
~PrivateDict
gdcm::PrivateDict, 762
~ProgressEvent
gdcm::ProgressEvent, 768
~PythonFilter
gdcm::PythonFilter, 773

~QueryBase
gdcm::QueryBase, 775
~RAWCodec
gdcm::RAWCodec, 789
~RLECodec
gdcm::RLECodec, 807
~Reader
gdcm::Reader, 794
~Region
gdcm::Region, 800
~Rescaler
gdcm::Rescaler, 803
~SHA1
gdcm::SHA1, 868
~Scanner
gdcm::Scanner, 818
~Segment
gdcm::Segment, 825
~SegmentReader
gdcm::SegmentReader, 834
~SegmentWriter
gdcm::SegmentWriter, 837
~SegmentedPaletteColorLookupTable
gdcm::SegmentedPaletteColorLookupTable, 831
~SerieHelper
gdcm::SerieHelper, 855
~ServiceClassUser
gdcm::ServiceClassUser, 862
~SimpleMemberCommand
gdcm::SimpleMemberCommand, 871
~SimpleSubjectWatcher
gdcm::SimpleSubjectWatcher, 874
~SmartPointer
gdcm::SmartPointer, 878
~Sorter
gdcm::Sorter, 885
~Spacing
gdcm::Spacing, 890
~SplitMosaicFilter
gdcm::SplitMosaicFilter, 892
~StreamImageReader
gdcm::StreamImageReader, 896
~StreamImageWriter
gdcm::StreamImageWriter, 901
~StrictScanner
gdcm::StrictScanner, 909
~StringFilter
gdcm::StringFilter, 920
~Subject
gdcm::Subject, 924
~Surface
gdcm::Surface, 930
~SurfaceReader
gdcm::SurfaceReader, 943
~SurfaceWriter
gdcm::SurfaceWriter, 946
~Table
gdcm::Table, 958
~TableEntry
gdcm::TableEntry, 960
~TableReader
gdcm::TableReader, 961
~TableRow
gdcm::network::TableRow, 964
~TagPath
gdcm::TagPath, 975
~Testing
gdcm::Testing, 978
~Trace
gdcm::Trace, 985
~Transition
gdcm::network::Transition, 997
~ULAction
gdcm::network::ULAction, 1031
~ULBasicCallback
gdcm::network::ULBasicCallback, 1068
~ULConnection
gdcm::network::ULConnection, 1071
~ULConnectionCallback
gdcm::network::ULConnectionCallback, 1075
~ULConnectionManager
gdcm::network::ULConnectionManager, 1080
~ULEvent
gdcm::network::ULEvent, 1085
~ULWritingCallback
gdcm::network::ULWritingCallback, 1089
~UserInformation
gdcm::network::UserInformation, 1099
~Validate
gdcm::Validate, 1103
~Value
gdcm::Value, 1105
~Version
gdcm::Version, 1108
~Writer
gdcm::Writer, 1234
~XMLDictReader
gdcm::XMLDictReader, 1238
~XMLPrinter
gdcm::XMLPrinter, 1241
~XMLPrivateDictReader
gdcm::XMLPrivateDictReader, 1244
~vtkGDCMImageReader
vtkGDCMImageReader, 1132
~vtkGDCMImageReader2
vtkGDCMImageReader2, 1144
~vtkGDCMImageWriter
vtkGDCMImageWriter, 1156

~vtkGDCMMedicalImageProperties
vtkGDCMMedicalImageProperties, 1163
~vtkGDCMPolyDataReader
vtkGDCMPolyDataReader, 1166
~vtkGDCMPolyDataWriter
vtkGDCMPolyDataWriter, 1171
~vtkGDCMTesting
vtkGDCMTesting, 1175
~vtkGDCMThreadedImageReader
vtkGDCMThreadedImageReader, 1178
~vtkGDCMThreadedImageReader2
vtkGDCMThreadedImageReader2, 1182
~vtkImageColorViewer
vtkImageColorViewer, 1190
~vtkImageMapToColors16
vtkImageMapToColors16, 1201
~vtkImageMapToWindowLevelColors2
vtkImageMapToWindowLevelColors2, 1206
~vtkImagePlanarComponentsToComponents
vtkImagePlanarComponentsToComponents, 1210
~vtkImageRGBToYBR
vtkImageRGBToYBR, 1212
~vtkImageYBRToRGB
vtkImageYBRToRGB, 1214
~vtkLookupTable16
vtkLookupTable16, 1216
~vtkRTStructSetProperties
vtkRTStructSetProperties, 1220

AAbortPDU
gdcm::network::AAbortPDU, 84
AAssociateACPDU
gdcm::network::AAssociateACPDU, 87
gdcm::network::AAssociateRQPDU, 97
AAssociateRJPDU
gdcm::network::AAssociateRJPDU, 90
AAssociateRQPDU
gdcm::network::AAssociateACPDU, 89
gdcm::network::AAssociateRQPDU, 94
AEComp
gdcm, 57
ALGOType
gdcm::Segment, 824
ARTIMTimer
gdcm::network::ARTIMTimer, 121
AReleaseRPPDU
gdcm::network::AReleaseRPPDU, 117
AReleaseRQPDU
gdcm::network::AReleaseRQPDU, 120
ASComp
gdcm, 57
ASN1
gdcm::ASN1, 123
AbstractSyntax
gdcm::PresentationContext, 744
gdcm::network::AbstractSyntax, 99
ActiveComponent
vtkImageMapToColors16, 1204
Add
gdcm::GroupDict, 451
AddAcceptedPresentationContext
gdcm::network::ULConnection, 1071
AddCSAHeaderDictEntry
gdcm::CSAHeaderDict, 264
AddContourReferencedFrameOfReference
vtkRTStructSetProperties, 1220
AddDerivationDescription
gdcm::FileDerivation, 405
AddDictEntry
gdcm::Dict, 322
gdcm::PrivateDict, 762
AddFile
gdcm::FileSet, 430
gdcm::SerieHelper, 856
AddFileName
gdcm::SerieHelper, 856
AddFragment
gdcm::SequenceOfFragments, 841
AddFromFile
gdcm::PresentationContextGenerator, 749
AddGroupLength
gdcm::DictConverter, 326
AddIODEntry
gdcm::IOD, 525
AddIOD
gdcm::IODs, 530
AddImageDirectoryRecord
gdcm::DICOMDIRGenerator, 319
AddInput
vtkImageColorViewer, 1191
AddInputConnection
vtkImageColorViewer, 1191
AddItem
gdcm::SequenceOfItems, 849
AddMacro
gdcm::Macros, 590
gdcm::Module, 619
AddMacroEntry
gdcm::Macro, 587
AddModule
gdcm::Modules, 626
AddModuleEntry
gdcm::Module, 619
gdcm::NestedModuleEntries, 643
AddNewUndefinedLengthItem
gdcm::SequenceOfItems, 849
AddObserver
gdcm::Subject, 925

AddPatientDirectoryRecord
gdcm::DICOMDIRGenerator, 319
AddPresentationContext
gdcm::PresentationContextGenerator, 749
gdcm::network::AAssociateRQPDU, 94
AddPresentationContextAC
gdcm::network::AAssociateACPDU, 87
AddPresentationDataValue
gdcm::network::PDataTFPDU, 690
AddPrimitiveData
gdcm::MeshPrimitive, 610
AddPrivateTag
gdcm::Scanner, 818
gdcm::StrictScanner, 909
AddPurposeOfReferenceCodeSequence
gdcm::FileDerivation, 405
AddQueryDataSet
gdcm::BaseQuery, 166
AddReference
gdcm::FileDerivation, 405
AddReferencedFrameOfReference
vtkRTStructSetProperties, 1220
AddRestriction
gdcm::SerieHelper, 856
AddRoleSelectionSub
gdcm::network::UserInformation, 1100
AddSOPClassExtendedNegociationSub
gdcm::network::UserInformation, 1100
AddSegment
gdcm::SegmentWriter, 837
AddSelect
gdcm::Sorter, 885
AddSeriesDirectoryRecord
gdcm::DICOMDIRGenerator, 319
AddSkipTag
gdcm::Scanner, 818
gdcm::StrictScanner, 909
AddSourceImageSequence
gdcm::FileDerivation, 406
AddStructureSetROIObservation
vtkRTStructSetProperties, 1221
AddStructureSetROI
vtkRTStructSetProperties, 1220
AddStudyDirectoryRecord
gdcm::DICOMDIRGenerator, 319
AddSurface
gdcm::Segment, 825
AddTag
gdcm::Scanner, 818
gdcm::StrictScanner, 910
AddTransferSyntax
gdcm::PresentationContext, 743
gdcm::network::PresentationContextRQ, 751
AffectedSOPClassUID
gdcm::network::CEchoRQ, 215
Allocate
gdcm::LookupTable, 581
AnatomicRegion
gdcm::Segment, 828
AnonymizeEvent
gdcm::AnonymizeEvent, 102
Anonymizer
gdcm::Anonymizer, 107
Append
gdcm::ByteValue, 204
gdcm::Global, 448
AppendFrameEncode
gdcm::ImageCodec, 485
gdcm::JPEG2000Codec, 550
gdcm::JPEGCodec, 559
gdcm::JPEGLSCodec, 567
gdcm::RLECodec, 808
AppendImplementationClassUID
gdcm::FileMetaInformation, 414
AppendRowEncode
gdcm::ImageCodec, 485
gdcm::JPEG2000Codec, 550
gdcm::JPEGCodec, 559
gdcm::JPEGLSCodec, 567
gdcm::RLECodec, 808
AppendToDataElement
gdcm::FileStreamer, 433
AppendToGroupDataElement
gdcm::FileStreamer, 433
ApplicationContext
gdcm::network::ApplicationContext, 113
Apply
gdcm::ImageApplyLookupTable, 468
ApplyInverseVideo
vtkGDCMImageReader, 1139
vtkGDCMImageReader2, 1151
ApplyLookupTable
vtkGDCMImageReader, 1139
vtkGDCMImageReader2, 1151
ApplyPlanarConfiguration
vtkGDCMImageReader, 1139
vtkGDCMImageReader2, 1151
ApplyShiftScale
vtkGDCMImageReader, 1139
vtkGDCMImageReader2, 1151
ApplyYBRToRGB
vtkGDCMImageReader, 1139
vtkGDCMImageReader2, 1151
AreOverlaysInPixelData
gdcm::Bitmap, 181
gdcm::Pixmap, 722
Area
gdcm::BoxRegion, 195

gdcm::Region, 800
ArrayIncludeMacrosType
gdcm::Macro, 587
gdcm::Module, 619
ArrayType
gdcm::Attribute, 127
gdcm::Attribute< Group, Element, TVR, VM::VM1 >, 134
gdcm::Attribute< Group, Element, TVR, VM::VM1_n >, 142
AsynchronousOperationsWindowSub
gdcm::network::AsynchronousOperationsWindowSub, 124
Attribute
gdcm::Attribute< Group, Element, TVR, VM::VM1_n >, 143
gdcm::terminal, 80
AudioCodec
gdcm::AudioCodec, 154
AutoPixelMinMax
gdcm::IconImageGenerator, 457

BALCPProtect
gdcm::Anonymizer, 107
BOOL_FUNCTION_PFILE_PFILE_POINTER
gdcm, 58
backslash
gdcm, 61
BaseQuery
gdcm::BaseQuery, 165
BaseRootQuery
gdcm::BaseRootQuery, 170
BasicApplicationLevelConfidentialityProfile
gdcm::Anonymizer, 107
BasicCodedEntry
gdcm::SegmentHelper::BasicCodedEntry, 174
BasicOffsetTable
gdcm::BasicOffsetTable, 177
Begin
gdcm::CSAHeaderDict, 264
gdcm::DataSet, 297
gdcm::Dict, 322
gdcm::IODs, 530
gdcm::Scanner, 818
gdcm::SequenceOfFragments, 841, 842
gdcm::SequenceOfItems, 849
gdcm::StrictScanner, 910
BitSample
gdcm::JPEGCodec, 564
gdcm::LookupTable, 584
Bitmap
gdcm::Bitmap, 181
gdcm::JPEG2000Codec, 553
gdcm::PixelFormat, 719
BitmapToBitmapFilter
gdcm::BitmapToBitmapFilter, 192
BoundingBox
gdcm::BoxRegion, 195
BoxRegion
gdcm::BoxRegion, 194, 195
BreakConnection
gdcm::network::ULConnectionManager, 1080
BreakConnectionNow
gdcm::network::ULConnectionManager, 1080
Build
vtkLookupTable16, 1216
ByteBuffer
gdcm::ByteBuffer, 198
ByteSwap
gdcm::ByteSwapFilter, 201
ByteSwapFilter
gdcm::ByteSwapFilter, 201
ByteValue
gdcm::ByteValue, 204
bytes
gdcm::Tag, 974

CAPICryptoFactory
gdcm::CAPICryptoFactory, 210
CAPICryptographicMessageSyntax
gdcm::CAPICryptographicMessageSyntax, 211
CEcho
gdcm::CompositeNetworkFunctions, 239
CFind
gdcm::CompositeNetworkFunctions, 240
cMaxEventID
gdcm::network, 79
cMaxStateID
gdcm::network, 79
CMove
gdcm::CompositeNetworkFunctions, 240
CSAElement
gdcm::CSAElement, 253
CSAHeader
gdcm::CSAHeader, 260
gdcm::DataSet, 304
CSAHeaderDict
gdcm::CSAHeaderDict, 264
CSAHeaderDictEntry
gdcm::CSAHeaderDictEntry, 267
CSAHeaderType
gdcm::CSAHeader, 259
CSComp
gdcm, 58
CSD
gdcm::SegmentHelper::BasicCodedEntry, 175
CStore
gdcm::CompositeNetworkFunctions, 241

CSV
gdcm::SegmentHelper::BasicCodedEntry, 175
CanCode
gdcm::AudioCodec, 154
gdcm::Coder, 227
gdcm::ImageCodec, 485
gdcm::JPEG2000Codec, 550
gdcm::JPEGCodec, 560
gdcm::JPEGLSCodec, 567
gdcm::KAKADUCodec, 574
gdcm::PDFCodec, 698
gdcm::PGXCodec, 706
gdcm::PNMCodec, 736
gdcm::PVRGCodec, 771
gdcm::RAWCodec, 789
gdcm::RLECodec, 808
CanDecode
gdcm::AudioCodec, 154
gdcm::Decoder, 309
gdcm::DeltaEncodingCodec, 316
gdcm::ImageCodec, 485
gdcm::JPEG2000Codec, 550
gdcm::JPEGCodec, 560
gdcm::JPEGLSCodec, 567
gdcm::KAKADUCodec, 574
gdcm::PDFCodec, 698
gdcm::PGXCodec, 706
gdcm::PNMCodec, 736
gdcm::PVRGCodec, 771
gdcm::RAWCodec, 789
gdcm::RLECodec, 808
CanDisplay
gdcm::VR, 1121
CanEmptyTag
gdcm::Anonymizer, 107
CanRead
gdcm::Reader, 794
CanReadFile
vtkGDCMImageReader, 1132
vtkGDCMImageReader2, 1144
CanReadImage
gdcm::StreamImageReader, 896
CanStoreLossy
gdcm::TransferSyntax, 991
CanWriteFile
gdcm::StreamImageWriter, 902
Change
gdcm::FileChangeTransferSyntax, 401
gdcm::FileDecompressLookupTable, 403
gdcm::FileExplicitFilter, 409
gdcm::ImageChangePhotometricInterpretation, 471
gdcm::ImageChangePlanarConfiguration, 475
gdcm::ImageChangeTransferSyntax, 480
ChangeFMI
gdcm::FileExplicitFilter, 409
ChangeMonochrome
gdcm::ImageChangePhotometricInterpretation, 471
CharacterDataHandler
gdcm::TableReader, 961
gdcm::XMLDictReader, 1239
gdcm::XMLPrivateDictReader, 1244
CheckDataElement
gdcm::FileStreamer, 434
CheckEvent
gdcm::AnonymizeEvent, 103
gdcm::DataEvent, 293
gdcm::DataSetEvent, 307
gdcm::Event, 381
gdcm::FileNameEvent, 424
gdcm::ProgressEvent, 769
CheckFileMetaInformationOff
gdcm::Writer, 1234
CheckFileMetaInformationOn
gdcm::Writer, 1234
CheckTemplateFileName
gdcm::FileStreamer, 434
CipherTypes
gdcm::CryptographicMessageSyntax, 248
Clear
gdcm::Bitmap, 181
gdcm::ByteValue, 204
gdcm::DataElement, 281
gdcm::DataSet, 297
gdcm::IODs, 531
gdcm::IOD, 525
gdcm::Item, 538
gdcm::LookupTable, 581
gdcm::Macro, 587
gdcm::Macros, 590
gdcm::Module, 620
gdcm::Modules, 626
gdcm::Preamble, 739
gdcm::SequenceOfFragments, 842
gdcm::SequenceOfItems, 849
gdcm::SerieHelper, 856
gdcm::Value, 1105
vtkGDCMMedicalImageProperties, 1163
vtkRTStructSetProperties, 1221
ClearInternalUIDs
gdcm::Anonymizer, 107
ClearSkipTags
gdcm::Scanner, 818
gdcm::StrictScanner, 910
ClearTags
gdcm::Scanner, 819
gdcm::StrictScanner, 910
Clone
gdcm::BoxRegion, 195

gdcm::ImageCodec, 485
gdcm::JPEG2000Codec, 550
gdcm::JPEGCodec, 560
gdcm::JPEGLSCodec, 567
gdcm::KAKADUCodec, 574
gdcm::PGXCodec, 706
gdcm::PNMCodec, 736
gdcm::PVRGCodec, 772
gdcm::RAWCodec, 790
gdcm::RLECodec, 808
gdcm::Region, 800
CM
gdcm::SegmentHelper::BasicCodedEntry, 175
Code
gdcm::Coder, 227
gdcm::JPEG2000Codec, 551
gdcm::JPEGCodec, 560
gdcm::JPEGLSCodec, 568
gdcm::JSON, 571
gdcm::KAKADUCodec, 574
gdcm::PVRGCodec, 772
gdcm::RAWCodec, 790
gdcm::RLECodec, 809
CodeMeaning
gdcm::RealWorldValueMappingContent, 798
CodeString
gdcm::CodeString, 230
CodeValue
gdcm::RealWorldValueMappingContent, 798
Color
gdcm::terminal, 81
ColorArray
gdcm::SurfaceHelper, 939
Command
gdcm::Command, 233
CommandDataSet
gdcm::CommandDataSet, 235
CommandTypes
gdcm::network::DIMSE, 339
CompOperators
gdcm, 59
Compatible
gdcm::VM, 1116
gdcm::VR, 1121
Component
gdcm::PersonName, 704
CompressionTypes
vtkGDCMImageWriter, 1156
Compute
gdcm::MD5, 593
gdcm::SHA1, 868
ComputeBoundingBox
gdcm::BoxRegion, 195
gdcm::Region, 800
ComputeBufferLength
gdcm::ImageRegionReader, 509
ComputeByteLength
gdcm::SequenceOfFragments, 842
ComputeDataElement
gdcm::DataSet, 297
ComputeDataSetMediaStorageSOPClass
gdcm::FileMetaInformation, 414
ComputeDataSetTransferSyntax
gdcm::FileMetaInformation, 414
ComputeDistAlongNormal
gdcm::DirectionCosines, 341
ComputeFile
gdcm::MD5, 593
gdcm::SHA1, 868
ComputeFileMD5
gdcm::Testing, 978
ComputeGroupLength
gdcm::DataSet, 297
ComputeInterceptSlopePixelType
gdcm::Rescaler, 803
ComputeLength
gdcm::ByteValue, 204
gdcm::Fragment, 445
gdcm::SequenceOfFragments, 842
gdcm::SequenceOfItems, 850
ComputeLossyFlag
gdcm::Bitmap, 181
ComputeMD5
gdcm::Testing, 978
ComputeMOSAICDimensions
gdcm::SplitMosaicFilter, 892
ComputeMediaStorageFromModality
gdcm::ImageHelper, 498
ComputeNumberOfSurfaces
gdcm::SurfaceWriter, 946
ComputeOffsetTable
gdcm::JPEGCodec, 560
ComputePixelAspectRatioFromPixelSpacing
gdcm::Spacing, 890
ComputePixelTypeFromMinMax
gdcm::Rescaler, 803
ComputeSpacingFromImagePositionPatient
gdcm::ImageHelper, 498
ComputeTargetMediaStorage
gdcm::ImageWriter, 515
ComputeVR
gdcm::DataSetHelper, 308
ComputeZSpacing
gdcm::IPPSorter, 536
ConcatenatePDVBlobs
gdcm::network::PresentationDataValue, 754
ConcatenatePDVBlobsAsExplicit
gdcm::network::PresentationDataValue, 754

const
gdcm::SOPClassUIDToIOD, 882
const_iterator
gdcm::CodeString, 229
gdcm::LO, 576
gdcm::String, 916
const_reference
gdcm::CodeString, 229
gdcm::LO, 576
gdcm::String, 916
const_reverse_iterator
gdcm::CodeString, 229
gdcm::LO, 576
gdcm::String, 916
ConstCharWrapper
gdcm::ConstCharWrapper, 243
ConstIterator
gdcm::CSAHeaderDict, 264
gdcm::DataSet, 296
gdcm::Dict, 322
gdcm::Scanner, 817
gdcm::SequenceOfFragments, 841
gdcm::SequenceOfItems, 848
gdcm::StrictScanner, 908
Construct
gdcm::BaseRootQuery, 170
ConstructAbortPDU
gdcm::network::PDUFactory, 700
ConstructCEchoRQ
gdcm::network::CompositeMessageFactory, 237
ConstructCFindRQ
gdcm::network::CompositeMessageFactory, 237
ConstructCMoveRQ
gdcm::network::CompositeMessageFactory, 237
ConstructCStoreRSP
gdcm::network::CompositeMessageFactory, 238
ConstructCStoreRQ
gdcm::network::CompositeMessageFactory, 238
ConstructFromString
gdcm::TagPath, 975
ConstructFromTagList
gdcm::TagPath, 975
ConstructNAction
gdcm::network::NormalizedMessageFactory, 651
ConstructNCreate
gdcm::network::NormalizedMessageFactory, 651
ConstructNDelete
gdcm::network::NormalizedMessageFactory, 651
ConstructNEventReport
gdcm::network::NormalizedMessageFactory, 651
ConstructNGet
gdcm::network::NormalizedMessageFactory, 652
ConstructNSet
gdcm::network::NormalizedMessageFactory, 652
ConstructPDVByDataSet
gdcm::network::CEchoRSP, 216
gdcm::network::CFindCancelRQ, 218
gdcm::network::CFindRSP, 221
gdcm::network::CMoveCancelRq, 222
gdcm::network::CMoveRSP, 225
gdcm::network::NActionRSP, 636
gdcm::network::NCreateRSP, 638
gdcm::network::NDeleteRSP, 641
gdcm::network::NEventReportRSP, 647
gdcm::network::NGetRSP, 649
gdcm::network::NSetRSP, 657
ConstructPDU
gdcm::network::PDUFactory, 700
ConstructPDV
gdcm::network::BaseCompositeMessage, 158
gdcm::network::BaseNormalizedMessage, 160
gdcm::network::CEchoRQ, 214
gdcm::network::CFindRQ, 219
gdcm::network::CMoveRQ, 223
gdcm::network::CStoreRSP, 272
gdcm::network::CStoreRQ, 271
gdcm::network::NActionRQ, 634
gdcm::network::NCreateRQ, 637
gdcm::network::NDeleteRQ, 640
gdcm::network::NEventReportRQ, 645
gdcm::network::NGetRQ, 648
gdcm::network::NSetRQ, 656
ConstructQuery
gdcm::CompositeNetworkFunctions, 241
gdcm::NormalizedNetworkFunctions, 653
ConstructReleasePDU
gdcm::network::PDUFactory, 700
ConstructorType
gdcm::Dicts, 336
Convert
gdcm::DictConverter, 326
gdcm::ImageConverter, 494
ConvertRGBToPaletteColor
gdcm::IconImageGenerator, 457
ConvertToCXX
gdcm::DictConverter, 326
ConvertToXML
gdcm::DictConverter, 326
Create
gdcm::Preamble, 739
CreateCEchoPDU
gdcm::network::PDUFactory, 700
CreateCFindPDU
gdcm::network::PDUFactory, 700
CreateCMSProvider
gdcm::CAPICryptoFactory, 210
gdcm::CryptoFactory, 247
gdcm::OpenSSLCryptoFactory, 662

gdcm::OpenSSLP7CryptoFactory, 666
CreateCMovePDU
gdcm::network::PDUFactory, 700
CreateCStoreRQPDU
gdcm::network::PDUFactory, 700
CreateCStoreRSPPDU
gdcm::network::PDUFactory, 701
CreateDefaultUniqueSeriesIdentifier
gdcm::SerieHelper, 856
CreateNActionPDU
gdcm::network::PDUFactory, 701
CreateNCreatePDU
gdcm::network::PDUFactory, 701
CreateNDeletePDU
gdcm::network::PDUFactory, 701
CreateNEventReportPDU
gdcm::network::PDUFactory, 701
CreateNGetPDU
gdcm::network::PDUFactory, 701
CreateNSetPDU
gdcm::network::PDUFactory, 701
CreateUniqueSeriesIdentifier
gdcm::SerieHelper, 856
Cross
gdcm::DirectionCosines, 341
CrossDot
gdcm::DirectionCosines, 342
CryptoFactory
gdcm::CryptoFactory, 247
CryptoLib
gdcm::CryptoFactory, 246
CryptographicMessageSyntax
gdcm::CryptographicMessageSyntax, 249
Curve
gdcm::Curve, 274
vtkGDCMImageReader, 1139
vtkGDCMImageReader2, 1151
Curves
gdcm::Pixmap, 724
CV
gdcm::SegmentHelper::BasicCodedEntry, 175

DAComp
gdcm, 58
DICOMDIRGenerator
gdcm::DICOMDIRGenerator, 319
DICOMDIR
gdcm::DICOMDIR, 317
DTComp
gdcm, 58
DataElement
gdcm::DataElement, 281
gdcm::Value, 1106
DataElementSet
gdcm::DataSet, 296
DataElementType
gdcm::ModuleEntry, 624
DataEvent
gdcm::DataEvent, 292
DataField
gdcm::CSAElement, 257
DataPtr
gdcm::CSAElement, 252
DataSetEvent
gdcm::DataSetEvent, 306
DataSetHandled
gdcm::network::ULConnectionCallback, 1075
DataSetHandles
gdcm::network::ULConnectionCallback, 1075
DataSetMS
gdcm::FileMetaInformation, 418
DataSetTS
gdcm::FileMetaInformation, 418
DataWasPassed
vtkImageMapToColors16, 1204
DebugOff
gdcm::Trace, 985
DebugOn
gdcm::Trace, 985
Decode
gdcm::AudioCodec, 154
gdcm::Base64, 155
gdcm::Curve, 274
gdcm::Decoder, 309
gdcm::DeltaEncodingCodec, 316
gdcm::ImageCodec, 486
gdcm::JPEG2000Codec, 551
gdcm::JPEGCodec, 561
gdcm::JPEGLSCodec, 568
gdcm::JSON, 571
gdcm::KAKADUCodec, 574
gdcm::LookupTable, 581
gdcm::PDFCodec, 698
gdcm::PVRGCodec, 772
gdcm::RAWCodec, 790
gdcm::RLECodec, 809
DecodeByStreams
gdcm::Decoder, 309
gdcm::ImageCodec, 486
gdcm::JPEG12Codec, 543
gdcm::JPEG16Codec, 546
gdcm::JPEG2000Codec, 551
gdcm::JPEG8Codec, 555
gdcm::JPEGCodec, 561
gdcm::RAWCodec, 790
gdcm::RLECodec, 809
DecodeBytes
gdcm::RAWCodec, 790

DecodeExtent
gdcm::JPEG2000Codec, 551
gdcm::JPEGCodec, 561
gdcm::JPEGLSCodec, 568
gdcm::RLECodec, 809
Decompress
gdcm::Overlay, 677
Decrypt
gdcm::CAPICryptographicMessageSyntax, 212
gdcm::CryptographicMessageSyntax, 249
gdcm::OpenSSLCryptographicMessageSyntax, 664
gdcm::OpenSSLP7CryptographicMessageSyntax, 668
DeepCopy
vtkRTStructSetProperties, 1221
Default
gdcm::FileMetaInformation, 414
DefinePixelExtent
gdcm::StreamImageReader, 896
gdcm::StreamImageWriter, 902
DefineProperBufferLength
gdcm::StreamImageReader, 897
gdcm::StreamImageWriter, 902
DefinedTerms
gdcm::DefinedTerms, 310
Defs
gdcm::Defs, 312
DeleteDirectory
gdcm::System, 952
DeltaEncodingCodec
gdcm::DeltaEncodingCodec, 316
Derive
gdcm::FileDerivation, 406
Description
gdcm::ModuleEntry, 623
DescriptionField
gdcm::ModuleEntry, 624
DetermineEventByPDU
gdcm::network::PDUFactory, 702
Dict
gdcm::Dict, 322
gdcm::DictEntry, 332
DictConverter
gdcm::DictConverter, 326
DictEntry
gdcm::DictEntry, 330
DictPrinter
gdcm::DictPrinter, 334
Dicts
gdcm::CSAHeaderDict, 265
gdcm::Dict, 324
gdcm::Dicts, 336
gdcm::PrivateDict, 763
difference_type
gdcm::CodeString, 229
gdcm::LO, 576
gdcm::String, 916
Dimensions
gdcm::Bitmap, 189
gdcm::ImageCodec, 491
DirCosTolerance
gdcm::IPPSorter, 536
DirectionCosines
gdcm::DirectionCosines, 341
vtkGDCMImageReader, 1140
vtkGDCMImageReader2, 1152
Directory
gdcm::Directory, 344
DoByteSwap
gdcm::ImageCodec, 486
DoIconImage
gdcm::PixmapWriter, 732
DoInvertMonochrome
gdcm::ImageCodec, 486
DoOverlayCleanup
gdcm::ImageCodec, 486
DoPaddedCompositePixelCode
gdcm::ImageCodec, 487
DoPlanarConfiguration
gdcm::ImageCodec, 487
DoSimpleCopy
gdcm::ImageCodec, 487
DoYBR
gdcm::ImageCodec, 487
Dot
gdcm::DirectionCosines, 342
DropDuplicatePositions
gdcm::IPPSorter, 536
Dumper
gdcm::Dumper, 351

ECharSet
gdcm, 60
EEventID
gdcm::network, 77
ENQueryType
gdcm, 60
EQueryLevel
gdcm, 60
EQueryType
gdcm, 61
ERootType
gdcm, 61
EStateID
gdcm::network, 78
elem
gdcm::SerieHelper::Rule, 813
Element

gdcm::Element< TVR, VM::VM1_n >, 358, 359
Empty
gdcm::Anonymizer, 107
gdcm::BoxRegion, 195
gdcm::DataElement, 281
gdcm::FileAnonymizer, 397
gdcm::Region, 800
EncapsulatedDocument
gdcm::EncapsulatedDocument, 374
Encode
gdcm::Base64, 155
EncodeBuffer
gdcm::JPEG12Codec, 543
gdcm::JPEG16Codec, 546
gdcm::JPEG8Codec, 555
gdcm::JPEGCodec, 561
EncodeBytes
gdcm::System, 952
Encrypt
gdcm::CAPICryptographicMessageSyntax, 212
gdcm::CryptographicMessageSyntax, 249
gdcm::OpenSSLCryptographicMessageSyntax, 664
gdcm::OpenSSLP7CryptographicMessageSyntax, 668
End
gdcm::CSAHeaderDict, 265
gdcm::DataSet, 297, 298
gdcm::Dict, 323
gdcm::IODs, 531
gdcm::Scanner, 819
gdcm::SequenceOfFragments, 842
gdcm::SequenceOfItems, 850
gdcm::StrictScanner, 910
EndElement
gdcm::TableReader, 961
gdcm::XMLDictReader, 1239
gdcm::XMLPrivateDictReader, 1244
EndElementHandler
gdcm::Parser, 685
EndFilter
gdcm::SimpleSubjectWatcher, 874
EndWith
gdcm::Filename, 420
EnumeratedValues
gdcm::EnumeratedValues, 379
ErrorOff
gdcm::Trace, 985
ErrorOn
gdcm::Trace, 985
ErrorType
gdcm::Parser, 685
EstablishConnection
gdcm::network::ULConnectionManager, 1080
EstablishConnectionMove
gdcm::network::ULConnectionManager, 1081
Event
gdcm::Event, 381
Exception
gdcm::Exception, 383
Execute
gdcm::Command, 234
gdcm::MemberCommand, 606
gdcm::SimpleMemberCommand, 872
ExecuteData
vtkGDCMImageReader, 1132
vtkGDCMThreadedImageReader, 1178
ExecuteInformation
vtkGDCMImageReader, 1132
vtkGDCMThreadedImageReader, 1178
ExecuteQuery
gdcm::StringFilter, 920
Explore
gdcm::Directory, 345
Extract
gdcm::IconImageFilter, 454
ExtractIconImages
gdcm::IconImageFilter, 454
ExtractVeproIconImages
gdcm::IconImageFilter, 454

F
gdcm::Printer, 761
gdcm::Reader, 797
gdcm::Validate, 1103
gdcm::XMLPrinter, 1242
Fiducials
gdcm::Fiducials, 390
File
gdcm::File, 392
FileAnonymizer
gdcm::FileAnonymizer, 397
FileChangeTransferSyntax
gdcm::FileChangeTransferSyntax, 400
gdcm::ImageCodec, 491
FileDecompressLookupTable
gdcm::FileDecompressLookupTable, 403
FileDerivation
gdcm::FileDerivation, 405
FileExists
gdcm::System, 952
FileExplicitFilter
gdcm::FileExplicitFilter, 409
FileIsDirectory
gdcm::System, 953
FileIsSymlink
gdcm::System, 953
FileList
gdcm, 58

FileMetaInformation
gdcm::FileMetaInformation, 413
FileName
vtkGDCMPolyDataReader, 1168
FileNameEvent
gdcm::FileNameEvent, 424
FileNameOrdering
gdcm::SerieHelper, 857
FileNames
vtkGDCMImageReader, 1140
FileSet
gdcm::FileSet, 430
FileSize
gdcm::System, 953
FileStreamer
gdcm::FileStreamer, 433
FileTime
gdcm::System, 953
FileType
gdcm::FileSet, 429
FileWithName
gdcm::FileWithName, 437
Filename
gdcm::Filename, 420
filename
gdcm::FileWithName, 438
FilenameGenerator
gdcm::FilenameGenerator, 427
FilenameType
gdcm::DICOMDIRGenerator, 319
gdcm::Directory, 344
gdcm::FilenameGenerator, 426
Filenames
gdcm::Sorter, 887
FilenamesType
gdcm::DICOMDIRGenerator, 319
gdcm::Directory, 344
gdcm::FilenameGenerator, 426
FilesType
gdcm::FileSet, 429
Fill
gdcm::ByteValue, 204
FillFromDataSet
gdcm::FileMetaInformation, 414
FillMedicalImageInformation
vtkGDCMImageReader, 1132
vtkGDCMImageReader2, 1144
vtkGDCMPolyDataReader, 1166
FindCSAElementByName
gdcm::CSAHeader, 260
FindContext
gdcm::network::ULConnection, 1071
FindDataElement
gdcm::DataSet, 298
gdcm::Item, 538
gdcm::SequenceOfItems, 850
FindDictEntry
gdcm::PrivateDict, 762
FindMacroEntry
gdcm::Macro, 588
FindModuleEntryInMacros
gdcm::Module, 620
FindNextDataElement
gdcm::DataSet, 298
FindPDBElementByName
gdcm::PDBHeader, 695
FindPatientRootQuery
gdcm::FindPatientRootQuery, 439
FindStudyRootQuery
gdcm::FindStudyRootQuery, 442
FirstRender
vtkImageColorViewer, 1198
ForceRescale
vtkGDCMImageReader, 1140
vtkGDCMImageReader2, 1152
FormatDateTime
gdcm::System, 954
Fragment
gdcm::Fragment, 445
FragmentVector
gdcm::SequenceOfFragments, 841
FromString
gdcm::StringFilter, 920

GDCM_DO_JOIN2
gdcmStaticAssert.h, 1457
GDCM_DO_JOIN
gdcmStaticAssert.h, 1457
GDCM_EXPORT
gdcmWin32.h, 1515
GDCM_FUNCTION
gdcmTrace.h, 1480
GDCM_JOIN
gdcmStaticAssert.h, 1457
GDCM_LEGACY_BODY
gdcmLegacyMacro.h, 1367
GDCM_LEGACY_REPLACED_BODY
gdcmLegacyMacro.h, 1367
GDCM_LEGACY
gdcmLegacyMacro.h, 1367
GDCM_STATIC_ASSERT
gdcm::Attribute, 128
gdcm::Attribute< Group, Element, TVR, VM::VM1 >, 134, 135
gdcm::Attribute< Group, Element, TVR, VM::VM1_n >, 143
gdcmStaticAssert.h, 1457
GDCMMACROENTRY_H

gdcmMacroEntry.h, 1372
gdcm, 43
AEComp, 57
ASComp, 57
BOOL_FUNCTION_PFILE_PFILE_POINTER, 58
backslash, 61
CSComp, 58
CompOperators, 59
DAComp, 58
DTComp, 58
ECharSet, 60
ENQueryType, 60
EQueryLevel, 60
EQueryType, 61
ERootType, 61
FileList, 58
GetVRFromTag, 61
GlobalInstance, 73
IconImage, 58
LOComp, 58
LTComp, 58
LodModeType, 61
MacroEntry, 58
NestedMacroEntries, 58
operator!=, 61, 62
operator<<, 62–71
operator>>, 72
operator==, 71
PNComp, 59
SHComp, 59
STComp, 59
TMComp, 59
TYPETOENCODING, 72
to_string, 72
UIComp, 59
UTComp, 59
VRBINARY, 73
gdcm::ASN1, 122
∼ASN1, 123
ASN1, 123
ParseDump, 123
ParseDumpFile, 123
TestPBKDF2, 123
gdcm::AbortEvent, 97
gdcm::AnonymizeEvent, 101
∼AnonymizeEvent, 102
AnonymizeEvent, 102
CheckEvent, 103
GetEventName, 103
GetTag, 103
MakeObject, 103
Self, 102
SetTag, 103
Superclass, 102
gdcm::Anonymizer, 104
∼Anonymizer, 107
Anonymizer, 107
BALCPProtect, 107
BasicApplicationLevelConfidentialityProfile, 107
CanEmptyTag, 107
ClearInternalUIDs, 107
Empty, 107
GetBasicApplicationLevelConfidentialityProfile↩
Attributes, 108
GetCryptographicMessageSyntax, 108
GetFile, 108
New, 108
RecurseDataSet, 108
Remove, 108
RemoveGroupLength, 109
RemovePrivateTags, 109
RemoveRetired, 109
Replace, 109
SetCryptographicMessageSyntax, 110
SetFile, 110
gdcm::AnyEvent, 110
gdcm::ApplicationEntity, 114
Internal, 115
IsValid, 115
MaxLength, 115
MaxNumberOfComponents, 115
Padding, 116
Print, 115
Separator, 116
SetBlob, 115
Squeeze, 115
gdcm::Attribute
ArrayType, 127
GDCM_STATIC_ASSERT, 128
GetAsDataElement, 128
GetDictVM, 128
GetDictVR, 128
GetNumberOfValues, 128
GetTag, 129
GetValue, 129
GetValues, 129
GetVM, 129
GetVR, 129
Internal, 132
operator!=, 130
operator<, 130
operator==, 130
operator[], 130
Print, 130
Set, 131
SetByteValue, 131
SetByteValueNoSwap, 131
SetFromDataElement, 131

SetFromDataSet, 131
SetValue, 132
SetValues, 132
gdcm::Attribute< Group, Element, TVR, TVM >, 125
gdcm::Attribute< Group, Element, TVR, VM::VM1 >, 133
ArrayType, 134
GDCM_STATIC_ASSERT, 134, 135
GetAsDataElement, 135
GetDictVM, 135
GetDictVR, 135
GetNumberOfValues, 135
GetTag, 135
GetValue, 136
GetValues, 136
GetVM, 136
GetVR, 136
Internal, 138
operator!=, 136
operator<, 136
operator==, 137
Print, 137
Set, 137
SetByteValue, 137
SetByteValueNoSwap, 137
SetFromDataElement, 137
SetFromDataSet, 138
SetValue, 138
gdcm::Attribute< Group, Element, TVR, VM::VM1_3 >, 139
GetVM, 140
gdcm::Attribute< Group, Element, TVR, VM::VM1_8 >, 140
GetVM, 141
gdcm::Attribute< Group, Element, TVR, VM::VM1_n >, 141
~Attribute, 143
ArrayType, 142
Attribute, 143
GDCM_STATIC_ASSERT, 143
GetAsDataElement, 143
GetDictVM, 143
GetDictVR, 144
GetNumberOfValues, 144
GetTag, 144
GetValue, 144
GetValues, 144
GetVM, 144
GetVR, 144
operator[], 145
Print, 145
Set, 145
SetByteValue, 145
SetFromDataElement, 145
SetFromDataSet, 145
SetNumberOfValues, 146
SetValue, 146
SetValues, 146
gdcm::Attribute< Group, Element, TVR, VM::VM2_2n >, 147
GetVM, 148
gdcm::Attribute< Group, Element, TVR, VM::VM2_n >, 148
GetVM, 149
gdcm::Attribute< Group, Element, TVR, VM::VM3_3n >, 149
GetVM, 151
gdcm::Attribute< Group, Element, TVR, VM::VM3_n >, 151
GetVM, 152
gdcm::AudioCodec, 152
~AudioCodec, 154
AudioCodec, 154
CanCode, 154
CanDecode, 154
Decode, 154
gdcm::Base64, 155
Decode, 155
Encode, 155
GetDecodeLength, 156
GetEncodeLength, 156
gdcm::BaseQuery, 164
~BaseQuery, 165
AddQueryDataSet, 166
BaseQuery, 165
GetAbstractSyntaxUID, 166
GetQueryDataSet, 166
GetSOPInstanceUID, 166
mDataSet, 168
mHelpDescription, 168
mSopInstanceUID, 168
Print, 166
QueryFactory, 168
SetSOPInstanceUID, 167
SetSearchParameter, 166, 167
ValidDataSet, 167
ValidateQuery, 167
WriteHelpFile, 167
WriteQuery, 167
gdcm::BaseRootQuery, 168
~BaseRootQuery, 170
BaseRootQuery, 170
Construct, 170
GetQueryLevelFromQueryRoot, 170
GetQueryLevelFromString, 170
GetQueryLevelString, 170
GetTagListByLevel, 171
InitializeDataSet, 171
mHelpDescription, 172

mImage, 172
mPatient, 172
mRootType, 172
mSeries, 172
mStudy, 172
QueryFactory, 172
ValidateQuery, 171
gdcm::BasicOffsetTable, 176
BasicOffsetTable, 177
operator<<, 177
Read, 177
gdcm::Bitmap, 178
~Bitmap, 181
AreOverlaysInPixelData, 181
Bitmap, 181
Clear, 181
ComputeLossyFlag, 181
Dimensions, 189
GetBuffer, 181
GetBuffer2, 181
GetBufferLength, 182
GetColumns, 182
GetDataElement, 182
GetDimension, 182
GetDimensions, 182
GetLUT, 183
GetNeedByteSwap, 183
GetNumberOfDimensions, 183
GetPhotometricInterpretation, 183
GetPixelFormat, 183, 184
GetPlanarConfiguration, 184
GetRows, 184
GetTransferSyntax, 184
ImageChangeTransferSyntax, 189
IsEmpty, 184
IsLossy, 184
IsTransferSyntaxCompatible, 184
LUTPtr, 181
LUT, 189
LossyFlag, 189
NeedByteSwap, 189
NumberOfDimensions, 189
PF, 189
PI, 190
PixelData, 190
PixmapReader, 189
PlanarConfiguration, 190
Print, 185
SetColumns, 185
SetDataElement, 185
SetDimension, 185
SetDimensions, 185
SetLUT, 186
SetLossyFlag, 186
SetNeedByteSwap, 186
SetNumberOfDimensions, 186
SetPhotometricInterpretation, 186
SetPixelFormat, 186
SetPlanarConfiguration, 187
SetRows, 187
SetTransferSyntax, 187
TryJPEG2000Codec, 187
TryJPEG2000Codec2, 187
TryJPEGCodec, 188
TryJPEGCodec2, 188
TryJPEGLSCodec, 188
TryKAKADUCodec, 188
TryPVRGCodec, 188
TryRAWCodec, 188
TryRLECodec, 188
TS, 190
gdcm::BitmapToBitmapFilter, 190
~BitmapToBitmapFilter, 192
BitmapToBitmapFilter, 192
GetOutput, 192
GetOutputAsBitmap, 192
Input, 192
Output, 192
SetInput, 192
gdcm::BoxRegion, 193
~BoxRegion, 194
Area, 195
BoundingBox, 195
BoxRegion, 194, 195
Clone, 195
ComputeBoundingBox, 195
Empty, 195
GetXMax, 196
GetXMin, 196
GetYMax, 196
GetYMin, 196
GetZMax, 196
GetZMin, 196
IsValid, 196
operator=, 196
Print, 197
SetDomain, 197
gdcm::ByteBuffer, 197
ByteBuffer, 198
Get, 198
GetStart, 198
ShiftEnd, 198
UpdatePosition, 198
gdcm::ByteSwap
Swap, 199
SwapFromSwapCodeIntoSystem, 199
SwapRange, 199
SwapRangeFromSwapCodeIntoSystem, 200

SystemIsBigEndian, 200
SystemIsLittleEndian, 200
gdcm::ByteSwap< T >, 199
gdcm::ByteSwapFilter, 200
~ByteSwapFilter, 201
ByteSwap, 201
ByteSwapFilter, 201
SetByteSwapTag, 201
gdcm::ByteValue, 202
~ByteValue, 204
Append, 204
ByteValue, 204
Clear, 204
ComputeLength, 204
Fill, 204
GetBuffer, 205
GetLength, 205
GetPointer, 205
IsEmpty, 206
IsPrintable, 206
operator const std::vector< char > &, 206
operator=, 206
operator==, 206
Print, 207
PrintASCIIXML, 207
PrintASCII, 207
PrintGroupLength, 207
PrintHex, 207
PrintHexXML, 207
PrintPNXML, 207
Read, 208
SetLength, 208
SetLengthOnly, 208
Write, 208
WriteBuffer, 209
gdcm::CAPICryptoFactory, 209
CAPICryptoFactory, 210
CreateCMSProvider, 210
gdcm::CAPICryptographicMessageSyntax, 210
~CAPICryptographicMessageSyntax, 211
CAPICryptographicMessageSyntax, 211
Decrypt, 212
Encrypt, 212
GetCipherType, 212
GetInitialized, 212
ParseCertificateFile, 212
ParseKeyFile, 212
SetCipherType, 213
SetPassword, 213
gdcm::CP246ExplicitDataElement, 243
GetLength, 244
Read, 244
ReadPreValue, 245
ReadValue, 245
ReadWithLength, 245
gdcm::CSAElement, 251
CSAElement, 253
DataField, 257
DataPtr, 252
GetByteValue, 253
GetKey, 253
GetName, 253
GetNoOfItems, 253
GetSyngoDT, 254
GetValue, 254
GetVM, 254
GetVR, 254
IsEmpty, 254
KeyField, 257
NameField, 257
NoOfItemsField, 257
operator<, 255
operator<<, 256
operator=, 255
operator==, 255
SetByteValue, 255
SetKey, 255
SetName, 255
SetNoOfItems, 256
SetSyngoDT, 256
SetValue, 256
SetVM, 256
SetVR, 256
SyngoDTField, 257
VRField, 257
ValueMultiplicityField, 257
gdcm::CSAHeader, 258
~CSAHeader, 260
CSAHeader, 260
CSAHeaderType, 259
FindCSAElementByName, 260
GetCSADataInfo, 260
GetCSAEEnd, 260
GetCSAElementByName, 261
GetCSAImageHeaderInfoTag, 261
GetCSASeriesHeaderInfoTag, 261
GetDataSet, 261
GetFormat, 261
GetInterfile, 262
LoadFromDataElement, 262
operator<<, 263
Print, 262
Read, 262
Write, 262
gdcm::CSAHeaderDict, 263
AddCSAHeaderDictEntry, 264
Begin, 264
CSAHeaderDict, 264

ConstIterator, 264
Dicts, 265
End, 265
GetCSAHeaderDictEntry, 265
IsEmpty, 265
Iterator, 264
LoadDefault, 265
MapCSAHeaderDictEntry, 264
operator<<, 265
gdcm::CSAHeaderDictEntry, 266
CSAHeaderDictEntry, 267
GetDescription, 267
GetName, 267
GetVM, 267
GetVR, 267
operator<, 267
operator<<, 268
SetDescription, 267
SetName, 268
SetVM, 268
SetVR, 268
gdcm::CSAHeaderDictException, 269
gdcm::CodeString, 228
CodeString, 230
const_iterator, 229
const_reference, 229
const_reverse_iterator, 229
difference_type, 229
GetAsString, 231
IsValid, 231
iterator, 229
operator!=, 231
operator<<, 231
operator==, 232
pointer, 229
reference, 230
reverse_iterator, 230
Size, 231
size_type, 230
TrimInternal, 231
value_type, 230
gdcm::Codec, 225
gdcm::Coder, 226
~Coder, 227
CanCode, 227
Code, 227
InternalCode, 227
gdcm::Command, 232
~Command, 233
Command, 233
Execute, 234
gdcm::CommandDataSet, 234
~CommandDataSet, 235
CommandDataSet, 235
Insert, 236
operator<<, 236
Read, 236
Replace, 236
Write, 236
gdcm::CompositeNetworkFunctions, 238
CEcho, 239
CFind, 240
CMove, 240
CStore, 241
ConstructQuery, 241
KeyValuePairArrayType, 239
KeyValuePairType, 239
gdcm::ConstCharWrapper, 242
ConstCharWrapper, 243
operator const char *, 243
gdcm::CryptoFactory, 245
~CryptoFactory, 247
CreateCMSProvider, 247
CryptoFactory, 247
CryptoLib, 246
GetFactoryInstance, 247
gdcm::CryptographicMessageSyntax, 248
~CryptographicMessageSyntax, 249
CipherTypes, 248
CryptographicMessageSyntax, 249
Decrypt, 249
Encrypt, 249
GetCipherType, 249
ParseCertificateFile, 250
ParseKeyFile, 250
SetCipherType, 250
SetPassword, 250
gdcm::Curve, 272
~Curve, 274
Curve, 274
Decode, 274
GetAsPoints, 274
GetCurveDataDescriptor, 275
GetDataValueRepresentation, 275
GetDimensions, 275
GetGroup, 275
GetNumberOfCurves, 275
GetNumberOfPoints, 275
GetTypeOfData, 275
GetTypeOfDataDescription, 275
IsEmpty, 275
Print, 276
SetCoordinateStartValue, 276
SetCoordinateStepValue, 276
SetCurve, 276
SetCurveDataDescriptor, 276
SetCurveDescription, 276
SetDataValueRepresentation, 276

SetDimensions, 276
SetGroup, 277
SetNumberOfPoints, 277
SetTypeOfData, 277
Update, 277
gdcm::DICOMDIRGenerator, 317
~DICOMDIRGenerator, 319
AddImageDirectoryRecord, 319
AddPatientDirectoryRecord, 319
AddSeriesDirectoryRecord, 319
AddStudyDirectoryRecord, 319
DICOMDIRGenerator, 319
FilenameType, 319
FilenamesType, 319
Generate, 319
GetFile, 320
GetScanner, 320
SetDescriptor, 320
SetFile, 320
SetFilenames, 320
SetRootDirectory, 320
gdcm::DICOMDIR, 317
DICOMDIR, 317
gdcm::DataElement, 277
Clear, 281
DataElement, 281
Empty, 281
GetByteValue, 281
GetLength, 282
GetSequenceOfFragments, 282
GetTag, 282, 283
GetValue, 283
GetValueAsSQ, 283
GetVL, 284
GetVR, 284
IsEmpty, 284
IsUndefinedLength, 285
operator<, 285
operator<<, 289
operator=, 285
operator==, 285
Read, 285
ReadOrSkip, 286
ReadPreValue, 286
ReadValue, 286
ReadValueWithLength, 286
ReadWithLength, 286
SetByteValue, 286
SetTag, 287
SetVLToUndefined, 288
SetValue, 287
SetValueFieldLength, 287
SetVL, 288
SetVR, 288
TagField, 289
VRField, 289
ValueField, 289
ValueLengthField, 289
ValuePtr, 281
Write, 288
gdcm::DataElementException, 290
gdcm::DataEvent, 290
~DataEvent, 292
CheckEvent, 293
DataEvent, 292
GetData, 293
GetDataLength, 293
GetEventName, 293
MakeObject, 293
Self, 292
SetData, 293
Superclass, 292
gdcm::DataSet, 294
Begin, 297
CSAHeader, 304
Clear, 297
ComputeDataElement, 297
ComputeGroupLength, 297
ConstIterator, 296
DataElementSet, 296
End, 297, 298
FindDataElement, 298
FindNextDataElement, 298
GetDEEnd, 299
GetDES, 299
GetDataElement, 298, 299
GetLength, 300
GetMediaStorage, 300
GetPrivateCreator, 300
Insert, 300
InsertDataElement, 300
IsEmpty, 301
Iterator, 296
operator<<, 304
operator(), 301
operator=, 301
operator[], 301
Print, 301
Read, 301
ReadNested, 301
ReadSelectedPrivateTags, 302
ReadSelectedPrivateTagsWithLength, 302
ReadSelectedTags, 302
ReadSelectedTagsWithLength, 302
ReadUpToTag, 302
ReadUpToTagWithLength, 302
ReadWithLength, 303
Remove, 303

Replace, 303
ReplaceEmpty, 303
Size, 304
SizeType, 297
Write, 304
gdcm::DataSetEvent, 305
~DataSetEvent, 306
CheckEvent, 307
DataSetEvent, 306
GetDataSet, 307
GetEventName, 307
MakeObject, 307
Self, 306
Superclass, 306
gdcm::DataSetHelper, 307
ComputeVR, 308
gdcm::Decoder, 308
~Decoder, 309
CanDecode, 309
Decode, 309
DecodeByStreams, 309
gdcm::DefinedTerms, 310
DefinedTerms, 310
gdcm::Defs, 311
~Defs, 312
Defs, 312
GetIODFromFile, 312
GetIODNameFromMediaStorage, 312
GetIODs, 312
GetMacros, 312, 313
GetModules, 313
GetTypeFromTag, 313
Global, 314
IsEmpty, 313
LoadDefaults, 313
LoadFromFile, 313
Verify, 314
gdcm::DeltaEncodingCodec, 314
~DeltaEncodingCodec, 316
CanDecode, 316
Decode, 316
DeltaEncodingCodec, 316
gdcm::Dict, 321
AddDictEntry, 322
Begin, 322
ConstIterator, 322
Dict, 322
Dicts, 324
End, 323
GetDictEntry, 323
GetDictEntryByKeyword, 323
GetDictEntryByName, 323
GetKeywordFromTag, 323
IsEmpty, 324
Iterator, 322
LoadDefault, 324
MapDictEntry, 322
operator<<, 324
gdcm::DictConverter, 324
~DictConverter, 326
AddGroupLength, 326
Convert, 326
ConvertToCXX, 326
ConvertToXML, 326
DictConverter, 326
GetDictName, 326
GetInputFilename, 327
GetOutputFilename, 327
GetOutputType, 327
OutputTypes, 325
ReadVM, 327
ReadVR, 327
Readuint16, 327
SetDictName, 327
SetInputFileName, 327
SetOutputFileName, 328
SetOutputType, 328
WriteFooter, 328
WriteHeader, 328
gdcm::DictEntry, 328
Dict, 332
DictEntry, 330
GetKeyword, 330
GetName, 330
GetRetired, 330
GetVM, 330
GetVR, 330
IsUnique, 331
operator<<, 332
SetElementXX, 331
SetGroupXX, 331
SetKeyword, 331
SetName, 331
SetRetired, 331
SetVM, 332
SetVR, 332
gdcm::DictPrinter, 333
~DictPrinter, 334
DictPrinter, 334
Print, 334
PrintDataElement2, 334
PrintDataSet2, 334
gdcm::Dicts, 335
~Dicts, 336
ConstructorType, 336
Dicts, 336
GetCSAHeaderDict, 336
GetConstructorString, 336

GetDictEntry, 337
GetPrivateDict, 337
GetPublicDict, 337
Global, 338
IsEmpty, 338
LoadDefaults, 338
operator<<, 338
gdcm::DirectionCosines, 340
~DirectionCosines, 341
ComputeDistAlongNormal, 341
Cross, 341
CrossDot, 342
DirectionCosines, 341
Dot, 342
IsValid, 342
Normalize, 342
operator const double *, 342
Print, 342
SetFromString, 343
gdcm::Directory, 343
~Directory, 344
Directory, 344
Explore, 345
FilenameType, 344
FilenamesType, 344
GetDirectories, 345
GetFilenames, 345
GetToplevel, 345
Load, 345
operator<<, 346
Print, 346
gdcm::DirectoryHelper, 347
GetCTImageSeriesUIDs, 347
GetFilenamesFromSeriesUIDs, 347
GetFrameOfReference, 347
GetMRImageSeriesUIDs, 348
GetRTStructSeriesUIDs, 348
GetSOPClassUID, 348
GetSeriesUIDsBySOPClassUID, 348
GetStringValueFromTag, 348
LoadImageFromFiles, 348
RetrieveSOPInstanceUIDFromIndex, 348
RetrieveSOPInstanceUIDFromZPosition, 349
gdcm::DummyValueGenerator, 349
Generate, 349
gdcm::Dumper, 350
~Dumper, 351
Dumper, 351
gdcm::Element
GetAsDataElement, 353
GetLength, 353
GetValue, 353, 354
GetValues, 354
GetVM, 354
GetVR, 354
Internal, 355
operator[], 354
Print, 354
Read, 354
Set, 355
SetFromDataElement, 355
SetNoSwap, 355
SetValue, 355
Type, 353
Write, 355
gdcm::Element< TVR, TVM >, 351
gdcm::Element< TVR, VM::VM1_2 >, 356
Parent, 357
SetLength, 357
gdcm::Element< TVR, VM::VM1_n >, 357
~Element, 358
Element, 358, 359
GetAsDataElement, 359
GetLength, 359
GetValue, 359
GetVM, 359
GetVR, 359
operator=, 360
operator[], 360
Print, 360
Read, 360
Set, 360
SetArray, 360
SetFromDataElement, 360
SetLength, 361
SetNoSwap, 361
SetValue, 361
Type, 358
Write, 361
WriteASCII, 361
gdcm::Element< TVR, VM::VM2_2n >, 362
Parent, 363
SetLength, 363
gdcm::Element< TVR, VM::VM2_n >, 363
Parent, 365
SetLength, 365
gdcm::Element< TVR, VM::VM3_3n >, 365
Parent, 366
SetLength, 366
gdcm::Element< TVR, VM::VM3_n >, 367
Parent, 368
SetLength, 368
gdcm::Element< VR::AS, VM::VM5 >, 368
GetLength, 369
Internal, 369
Print, 369
gdcm::Element< VR::OB, VM::VM1 >, 369
gdcm::Element< VR::OW, VM::VM1 >, 370

gdcm::ElementDisableCombinations< TVR, TVM >, 372
gdcm::ElementDisableCombinations< VR::OB, VM::V↩
M1_n >, 373
gdcm::ElementDisableCombinations< VR::OW, VM::V↩
M1_n >, 373
gdcm::EncapsulatedDocument, 373
EncapsulatedDocument, 374
gdcm::EncodingImplementation< T >, 374
gdcm::EncodingImplementation< VR::VRASCII >, 374
Read, 375
ReadComputeLength, 375
ReadNoSwap, 375
Write, 375, 376
gdcm::EncodingImplementation< VR::VRBINARY >, 376
Read, 377
ReadComputeLength, 377
ReadNoSwap, 377
Write, 377
gdcm::EndEvent, 378
gdcm::EnumeratedValues, 379
EnumeratedValues, 379
gdcm::Event, 380
∼Event, 381
CheckEvent, 381
Event, 381
GetEventName, 381
MakeObject, 381
Print, 382
gdcm::Exception, 382
∼Exception, 383
Exception, 383
GetDescription, 384
what, 384
gdcm::ExitEvent, 384
gdcm::ExplicitDataElement, 385
GetLength, 387
Read, 387
ReadPreValue, 387
ReadValue, 387
ReadWithLength, 387
Write, 387
gdcm::ExplicitImplicitDataElement, 388
GetLength, 389
Read, 389
ReadPreValue, 389
ReadValue, 389
ReadWithLength, 389
gdcm::Fiducials, 390
Fiducials, 390
gdcm::File, 391
∼File, 392
File, 392
GetDataSet, 393
GetHeader, 393
operator<<, 394
Read, 394
SetDataSet, 394
SetHeader, 394
Write, 394
gdcm::FileAnonymizer, 395
∼FileAnonymizer, 397
Empty, 397
FileAnonymizer, 397
Remove, 397
Replace, 397
SetInputFileName, 398
SetOutputFileName, 398
Write, 398
gdcm::FileChangeTransferSyntax, 399
∼FileChangeTransferSyntax, 400
Change, 401
FileChangeTransferSyntax, 400
GetCodec, 401
New, 401
SetInputFileName, 401
SetOutputFileName, 401
SetTransferSyntax, 401
gdcm::FileDecompressLookupTable, 402
∼FileDecompressLookupTable, 403
Change, 403
FileDecompressLookupTable, 403
GetFile, 403
GetPixmap, 403, 404
SetFile, 404
SetPixmap, 404
gdcm::FileDerivation, 404
∼FileDerivation, 405
AddDerivationDescription, 405
AddPurposeOfReferenceCodeSequence, 405
AddReference, 405
AddSourceImageSequence, 406
Derive, 406
FileDerivation, 405
GetFile, 406
SetDerivationCodeSequenceCodeValue, 406
SetDerivationDescription, 407
SetFile, 407
SetPurposeOfReferenceCodeSequenceCodeValue,
407
gdcm::FileExplicitFilter, 408
∼FileExplicitFilter, 409
Change, 409
ChangeFMI, 409
FileExplicitFilter, 409
GetFile, 409
ProcessDataSet, 409
SetChangePrivateTags, 409
SetFile, 409

SetRecomputeItemLength, 410
SetRecomputeSequenceLength, 410
SetUseVRUN, 410
gdcm::FileMetaInformation, 411
~FileMetaInformation, 413
AppendImplementationClassUID, 414
ComputeDataSetMediaStorageSOPClass, 414
ComputeDataSetTransferSyntax, 414
DataSetMS, 418
DataSetTS, 418
Default, 414
FileMetaInformation, 413
FillFromDataSet, 414
GetDataSetTransferSyntax, 414
GetFileMetaInformationVersion, 414
GetFullLength, 414
GetGDCMImplementationClassUID, 415
GetGDCMImplementationVersionName, 415
GetGDCMSourceApplicationEntityTitle, 415
GetImplementationClassUID, 415
GetImplementationVersionName, 415
GetMediaStorage, 415
GetMediaStorageAsString, 415
GetMetaInformationTS, 415
GetPreamble, 415, 416
GetSourceApplicationEntityTitle, 416
Insert, 416
IsValid, 416
MetaInformationTS, 418
operator<<, 418
Read, 416
ReadCompat, 416
ReadCompatInternal, 416
Replace, 417
SetDataSetTransferSyntax, 417
SetImplementationClassUID, 417
SetImplementationVersionName, 417
SetPreamble, 417
SetSourceApplicationEntityTitle, 417
Write, 418
gdcm::FileNameEvent, 422
~FileNameEvent, 424
CheckEvent, 424
FileNameEvent, 424
GetEventName, 424
GetFileName, 424
MakeObject, 425
Self, 424
SetFileName, 425
Superclass, 424
gdcm::FileSet, 429
AddFile, 430
FileSet, 430
FileType, 429
FilesType, 429
GetFiles, 430
operator<<, 431
SetFiles, 430
gdcm::FileStreamer, 431
~FileStreamer, 433
AppendToDataElement, 433
AppendToGroupDataElement, 433
CheckDataElement, 434
CheckTemplateFileName, 434
FileStreamer, 433
New, 434
ReserveDataElement, 434
ReserveGroupDataElement, 434
SetOutputFileName, 435
SetTemplateFileName, 435
StartDataElement, 435
StartGroupDataElement, 435
StopDataElement, 436
StopGroupDataElement, 436
gdcm::FileWithName, 436
FileWithName, 437
filename, 438
gdcm::Filename, 419
EndWith, 420
Filename, 420
GetExtension, 420
GetFileName, 420
GetName, 420
GetPath, 420
IsEmpty, 421
IsIdentical, 421
Join, 421
operator const char *, 421
ToUnixSlashes, 421
ToWindowsSlashes, 421
gdcm::FilenameGenerator, 425
~FilenameGenerator, 427
FilenameGenerator, 427
FilenameType, 426
FilenamesType, 426
Generate, 427
GetFilename, 427
GetFilenames, 427
GetNumberOfFilenames, 427
GetPattern, 428
GetPrefix, 428
SetNumberOfFilenames, 428
SetPattern, 428
SetPrefix, 428
SizeType, 426
gdcm::FindPatientRootQuery, 438
FindPatientRootQuery, 439
GetAbstractSyntaxUID, 440

GetTagListByLevel, 440
InitializeDataSet, 440
QueryFactory, 441
ValidateQuery, 440
gdcm::FindStudyRootQuery, 441
FindStudyRootQuery, 442
GetAbstractSyntaxUID, 443
GetTagListByLevel, 443
InitializeDataSet, 443
QueryFactory, 443
ValidateQuery, 443
gdcm::Fragment, 444
ComputeLength, 445
Fragment, 445
GetLength, 445
operator<<, 447
Read, 446
ReadBacktrack, 446
ReadPreValue, 446
ReadValue, 446
Write, 446
gdcm::Global, 447
~Global, 448
Append, 448
GetDefs, 448
GetDicts, 448, 449
GetInstance, 449
Global, 448
LoadResourcesFiles, 449
Locate, 449
operator<<, 450
Prepend, 449
gdcm::GroupDict, 450
~GroupDict, 451
Add, 451
GetAbbreviation, 451
GetName, 451
GroupDict, 451
GroupStringVector, 451
Insert, 452
operator<<, 452
Size, 452
gdcm::IODEntry, 526
GetIE, 527
GetName, 527
GetRef, 528
GetUsage, 528
GetUsageType, 528
IODEntry, 527
operator<<, 529
SetIE, 528
SetName, 528
SetRef, 528
SetUsage, 528
gdcm::IODs, 529
AddIOD, 530
Begin, 530
Clear, 531
End, 531
GetIOD, 531
IODMapType, 530
IODMapTypeConstIterator, 530
IODName, 530
IODs, 530
operator<<, 531
gdcm::IOD, 524
AddIODEntry, 525
Clear, 525
GetIODEntry, 525
GetNumberOfIODs, 525
GetTypeFromTag, 525
IOD, 525
MapIODEntry, 524
operator<<, 526
SizeType, 524
gdcm::IPPSorter, 532
ComputeZSpacing, 536
DirCosTolerance, 536
DropDuplicatePositions, 536
GetDirectionCosinesTolerance, 534
GetZSpacing, 534
GetZSpacingTolerance, 534
IPPSorter, 534
SetComputeZSpacing, 534
SetDirectionCosinesTolerance, 534
SetDropDuplicatePositions, 535
SetZSpacingTolerance, 535
Sort, 535
ZSpacing, 536
ZTolerance, 536
gdcm::IconImageFilter, 452
~IconImageFilter, 454
Extract, 454
ExtractIconImages, 454
ExtractVeproIconImages, 454
GetFile, 454
GetIconImage, 454
GetNumberOfIconImages, 455
IconImageFilter, 454
SetFile, 455
gdcm::IconImageGenerator, 455
~IconImageGenerator, 456
AutoPixelMinMax, 457
ConvertRGBToPaletteColor, 457
Generate, 457
GetIconImage, 457
GetPixmap, 457, 458
IconImageGenerator, 456

SetOutputDimensions, 458
SetOutsideValuePixel, 458
SetPixelMinMax, 458
SetPixmap, 458
gdcm::Image, 460
~Image, 462
GetDirectionCosines, 462
GetIntercept, 462
GetOrigin, 462, 463
GetSlope, 463
GetSpacing, 463
Image, 462
Print, 463
SetDirectionCosines, 463, 464
SetIntercept, 464
SetOrigin, 464
SetSlope, 464
SetSpacing, 465
gdcm::ImageApplyLookupTable, 465
~ImageApplyLookupTable, 468
Apply, 468
ImageApplyLookupTable, 468
gdcm::ImageChangePhotometricInterpretation, 468
~ImageChangePhotometricInterpretation, 471
Change, 471
ChangeMonochrome, 471
GetPhotometricInterpretation, 471
ImageChangePhotometricInterpretation, 471
RGB2YBR, 472
SetPhotometricInterpretation, 472
YBR2RGB, 472
gdcm::ImageChangePlanarConfiguration, 472
~ImageChangePlanarConfiguration, 475
Change, 475
GetPlanarConfiguration, 475
ImageChangePlanarConfiguration, 475
RGBPixelsToRGBPlanes, 475
RGBPlanesToRGBPixels, 476
SetPlanarConfiguration, 476
gdcm::ImageChangeTransferSyntax, 477
~ImageChangeTransferSyntax, 479
Change, 480
GetTransferSyntax, 480
ImageChangeTransferSyntax, 479
SetCompressIconImage, 480
SetForce, 480
SetTransferSyntax, 480
SetUserCodec, 480
TryJPEG2000Codec, 481
TryJPEGCodec, 481
TryJPEGLSCodec, 481
TryRAWCodec, 481
TryRLECodec, 481
gdcm::ImageCodec, 482
~ImageCodec, 484
AppendFrameEncode, 485
AppendRowEncode, 485
CanCode, 485
CanDecode, 485
Clone, 485
Decode, 486
DecodeByStreams, 486
Dimensions, 491
DoByteSwap, 486
DoInvertMonochrome, 486
DoOverlayCleanup, 486
DoPaddedCompositePixelCode, 487
DoPlanarConfiguration, 487
DoSimpleCopy, 487
DoYBR, 487
FileChangeTransferSyntax, 491
GetDimensions, 487
GetHeaderInfo, 487
GetLUT, 488
GetLossyFlag, 487
GetNeedByteSwap, 488
GetNumberOfDimensions, 488
GetPhotometricInterpretation, 488
GetPixelFormat, 488
GetPlanarConfiguration, 488
ImageChangePhotometricInterpretation, 491
ImageCodec, 484
IsFrameEncoder, 488
IsLossy, 489
IsRowEncoder, 489
IsValid, 489
LUTPtr, 484
LUT, 492
LossyFlag, 491
NeedByteSwap, 492
NeedOverlayCleanup, 492
NumberOfDimensions, 492
PF, 492
PI, 492
PlanarConfiguration, 492
RequestPaddedCompositePixelCode, 492
RequestPlanarConfiguration, 492
SetDimensions, 489
SetLUT, 489
SetLossyFlag, 489
SetNeedByteSwap, 490
SetNeedOverlayCleanup, 490
SetNumberOfDimensions, 490
SetPhotometricInterpretation, 490
SetPixelFormat, 490
SetPlanarConfiguration, 490
StartEncode, 491
StopEncode, 491

gdcm::ImageConverter, 493
~ImageConverter, 493
Convert, 494
GetOuput, 494
ImageConverter, 493
SetInput, 494
gdcm::ImageFragmentSplitter, 494
~ImageFragmentSplitter, 496
GetFragmentSizeMax, 496
ImageFragmentSplitter, 496
SetForce, 496
SetFragmentSizeMax, 496
Split, 496
gdcm::ImageHelper, 497
ComputeMediaStorageFromModality, 498
ComputeSpacingFromImagePositionPatient, 498
GetDimensionsValue, 498
GetDirectionCosinesFromDataSet, 499
GetDirectionCosinesValue, 499
GetForcePixelSpacing, 499
GetForceRescaleInterceptSlope, 499
GetLUT, 499
GetOriginValue, 499
GetPMSRescaleInterceptSlope, 500
GetPhotometricInterpretationValue, 500
GetPixelFormatValue, 500
GetPlanarConfigurationValue, 500
GetPointerFromElement, 500
GetRealWorldValueMappingContent, 500
GetRescaleInterceptSlopeValue, 500
GetSpacingTagFromMediaStorage, 501
GetSpacingValue, 501
GetZSpacingTagFromMediaStorage, 501
SetDimensionsValue, 501
SetDirectionCosinesValue, 501
SetForcePixelSpacing, 501
SetForceRescaleInterceptSlope, 502
SetOriginValue, 502
SetPMSRescaleInterceptSlope, 502
SetRescaleInterceptSlopeValue, 502
SetSpacingValue, 502
gdcm::ImageReader, 503
~ImageReader, 505
GetImage, 505
ImageReader, 505
Read, 506
ReadACRNEMAImage, 506
ReadImage, 506
gdcm::ImageRegionReader, 507
~ImageRegionReader, 509
ComputeBufferLength, 509
GetRegion, 509
ImageRegionReader, 509
Read, 510
ReadInformation, 510
ReadIntoBuffer, 510
SetRegion, 510
gdcm::ImageToImageFilter, 511
~ImageToImageFilter, 512
GetInput, 512
GetOutput, 512
ImageToImageFilter, 512
gdcm::ImageWriter, 513
~ImageWriter, 515
ComputeTargetMediaStorage, 515
GetImage, 515
ImageWriter, 515
Write, 516
gdcm::ImplicitDataElement, 520
GetLength, 521
Read, 521
ReadPreValue, 521
ReadValue, 521
ReadValueWithLength, 522
ReadWithLength, 522
Write, 522
gdcm::InitializeEvent, 522
gdcm::Item, 536
Clear, 538
FindDataElement, 538
GetDataElement, 539
GetLength, 539
GetNestedDataSet, 539
InsertDataElement, 539
Item, 538
operator<<, 540
Read, 539
SetNestedDataSet, 540
Write, 540
gdcm::IterationEvent, 541
gdcm::JPEG12Codec, 542
~JPEG12Codec, 543
DecodeByStreams, 543
EncodeBuffer, 543
GetHeaderInfo, 544
InternalCode, 544
IsStateSuspension, 544
JPEG12Codec, 543
gdcm::JPEG16Codec, 545
~JPEG16Codec, 546
DecodeByStreams, 546
EncodeBuffer, 546
GetHeaderInfo, 547
InternalCode, 547
IsStateSuspension, 547
JPEG16Codec, 546
gdcm::JPEG2000Codec, 548
~JPEG2000Codec, 550

AppendFrameEncode, 550
AppendRowEncode, 550
Bitmap, 553
CanCode, 550
CanDecode, 550
Clone, 550
Code, 551
Decode, 551
DecodeByStreams, 551
DecodeExtent, 551
GetHeaderInfo, 551
GetQuality, 552
GetRate, 552
ImageRegionReader, 553
IsFrameEncoder, 552
IsRowEncoder, 552
JPEG2000Codec, 550
SetNumberOfResolutions, 552
SetQuality, 552
SetRate, 552
SetReversible, 553
SetTileSize, 553
StartEncode, 553
StopEncode, 553
gdcm::JPEG8Codec, 554
~JPEG8Codec, 555
DecodeByStreams, 555
EncodeBuffer, 555
GetHeaderInfo, 556
InternalCode, 556
IsStateSuspension, 556
JPEG8Codec, 555
gdcm::JPEGCodec, 557
~JPEGCodec, 559
AppendFrameEncode, 559
AppendRowEncode, 559
BitSample, 564
CanCode, 560
CanDecode, 560
Clone, 560
Code, 560
ComputeOffsetTable, 560
Decode, 561
DecodeByStreams, 561
DecodeExtent, 561
EncodeBuffer, 561
GetHeaderInfo, 561
GetLossless, 562
GetQuality, 562
ImageRegionReader, 564
IsFrameEncoder, 562
IsRowEncoder, 562
IsStateSuspension, 562
IsValid, 562
JPEGCodec, 559
Quality, 564
SetBitSample, 563
SetLossless, 563
SetPixelFormat, 563
SetQuality, 563
StartEncode, 563
StopEncode, 563
gdcm::JPEGLSCodec, 564
~JPEGLSCodec, 566
AppendFrameEncode, 567
AppendRowEncode, 567
CanCode, 567
CanDecode, 567
Clone, 567
Code, 568
Decode, 568
DecodeExtent, 568
GetBufferLength, 568
GetHeaderInfo, 569
GetLossless, 569
ImageRegionReader, 570
IsFrameEncoder, 569
IsRowEncoder, 569
JPEGLSCodec, 566
SetBufferLength, 569
SetLossless, 569
SetLossyError, 569
StartEncode, 569
StopEncode, 570
gdcm::JSON, 570
~JSON, 571
Code, 571
Decode, 571
GetPrettyPrint, 571
JSON, 571
PrettyPrintOff, 571
PrettyPrintOn, 571
SetPrettyPrint, 572
gdcm::KAKADUCodec, 572
~KAKADUCodec, 573
CanCode, 574
CanDecode, 574
Clone, 574
Code, 574
Decode, 574
KAKADUCodec, 573
gdcm::LO, 575
const_iterator, 576
const_reference, 576
const_reverse_iterator, 576
difference_type, 576
IsValid, 578
iterator, 576

LO, 577
pointer, 576
reference, 576
reverse_iterator, 577
size_type, 577
Superclass, 577
value_type, 577
gdcm::LookupTable, 578
~LookupTable, 581
Allocate, 581
BitSample, 584
Clear, 581
Decode, 581
GetBitSample, 581
GetBufferAsRGBA, 582
GetLUTDescriptor, 582
GetLUTLength, 582
GetLUT, 582
GetPointer, 582
IncompleteLUT, 584
InitializeBlueLUT, 582
InitializeGreenLUT, 583
InitializeLUT, 583
InitializeRedLUT, 583
Initialized, 583
Internal, 584
LookupTable, 580, 581
LookupTableType, 580
Print, 583
SetBlueLUT, 583
SetGreenLUT, 584
SetLUT, 584
SetRedLUT, 584
WriteBufferAsRGBA, 584
gdcm::MD5, 592
~MD5, 593
Compute, 593
ComputeFile, 593
MD5, 593
gdcm::Macro, 586
AddMacroEntry, 587
ArrayIncludeMacrosType, 587
Clear, 587
FindMacroEntry, 588
GetMacroEntry, 588
GetName, 588
Macro, 587
MapModuleEntry, 587
operator<<, 588
SetName, 588
Verify, 588
gdcm::Macros, 589
AddMacro, 590
Clear, 590
GetMacro, 590
IsEmpty, 590
Macros, 590
ModuleMapType, 589
operator<<, 590
gdcm::MediaStorage, 594
GetMSString, 600
GetMSType, 600
GetModality, 600
GetModalityDimension, 600
GetNumberOfMSString, 601
GetNumberOfMSType, 601
GetNumberOfModality, 600
GetString, 601
GuessFromModality, 601
IsImage, 601
IsUndefined, 601
MSType, 597
MediaStorage, 600
ObjectType, 599
operator MSType, 602
operator<<, 603
SetFromDataSet, 602
SetFromFile, 602
SetFromHeader, 602
SetFromModality, 602
SetFromSourceImageSequence, 602
gdcm::MemberCommand
~MemberCommand, 605
Execute, 606
m_ConstMemberFunction, 607
m_MemberFunction, 607
m_This, 607
MemberCommand, 605
New, 606
Self, 605
SetCallbackFunction, 606
TConstMemberFunctionPointer, 605
TMemberFunctionPointer, 605
gdcm::MemberCommand< T >, 603
gdcm::MeshPrimitive, 607
~MeshPrimitive, 610
AddPrimitiveData, 610
GetMPType, 610
GetMPTypeString, 610
GetNumberOfPrimitivesData, 610
GetPrimitiveData, 610, 611
GetPrimitiveType, 611
GetPrimitivesData, 611
MPType, 609
MeshPrimitive, 610
PrimitiveData, 612
PrimitiveType, 612
PrimitivesData, 609

SetPrimitiveData, 611
SetPrimitiveType, 612
SetPrimitivesData, 611
gdcm::ModalityPerformedProcedureStepCreateQuery, 612
GetAbstractSyntaxUID, 614
GetRequiredDataSet, 614
ModalityPerformedProcedureStepCreateQuery, 614
QueryFactory, 614
ValidateQuery, 614
gdcm::ModalityPerformedProcedureStepSetQuery, 615
GetAbstractSyntaxUID, 616
GetRequiredDataSet, 616
ModalityPerformedProcedureStepSetQuery, 616
QueryFactory, 617
ValidateQuery, 616
gdcm::ModifiedEvent, 617
gdcm::Module, 618
AddMacro, 619
AddModuleEntry, 619
ArrayIncludeMacrosType, 619
Clear, 620
FindModuleEntryInMacros, 620
GetModuleEntryInMacros, 620
GetName, 620
MapModuleEntry, 619
Module, 619
operator<<, 621
SetName, 620
Verify, 620
gdcm::ModuleEntry, 621
~ModuleEntry, 623
DataElementType, 624
Description, 623
DescriptionField, 624
GetDescription, 623
GetName, 623
GetType, 624
ModuleEntry, 623
Name, 625
operator<<, 624
SetDescription, 624
SetName, 624
SetType, 624
gdcm::Modules, 625
AddModule, 626
Clear, 626
GetModule, 626
IsEmpty, 627
ModuleMapType, 626
Modules, 626
operator<<, 627
gdcm::MovePatientRootQuery, 627
GetAbstractSyntaxUID, 629
GetTagListByLevel, 629
InitializeDataSet, 629
MovePatientRootQuery, 629
QueryFactory, 630
ValidateQuery, 629
gdcm::MoveStudyRootQuery, 630
GetAbstractSyntaxUID, 632
GetTagListByLevel, 632
InitializeDataSet, 632
MoveStudyRootQuery, 632
QueryFactory, 633
ValidateQuery, 632
gdcm::NestedModuleEntries, 641
AddModuleEntry, 643
GetModuleEntry, 643
GetNumberOfModuleEntries, 644
NestedModuleEntries, 643
operator<<, 644
SizeType, 643
gdcm::NoEvent, 650
gdcm::NormalizedNetworkFunctions, 652
ConstructQuery, 653
NAction, 653
NCreate, 653
NDelete, 653
NEventReport, 654
NGet, 654
NSet, 654
gdcm::Object, 657
~Object, 659
Object, 659
operator<<, 660
operator=, 659
Print, 659
Register, 660
SmartPointer, 660
UnRegister, 660
gdcm::OpenSSLCryptoFactory, 661
CreateCMSProvider, 662
InitOpenSSL, 662
OpenSSLCryptoFactory, 662
gdcm::OpenSSLCryptographicMessageSyntax, 662
~OpenSSLCryptographicMessageSyntax, 663
Decrypt, 664
Encrypt, 664
GetCipherType, 664
OpenSSLCryptographicMessageSyntax, 663
ParseCertificateFile, 664
ParseKeyFile, 664
SetCipherType, 664
SetPassword, 665
gdcm::OpenSSLP7CryptoFactory, 665
CreateCMSProvider, 666
OpenSSLP7CryptoFactory, 666

gdcm::OpenSSLP7CryptographicMessageSyntax, 667
~OpenSSLP7CryptographicMessageSyntax, 668
Decrypt, 668
Encrypt, 668
GetCipherType, 669
OpenSSLP7CryptographicMessageSyntax, 668
ParseCertificateFile, 669
ParseKeyFile, 669
SetCipherType, 669
SetPassword, 669
gdcm::Orientation, 670
~Orientation, 671
GetLabel, 672
GetMajorAxisFromPatientRelativeDirectionCosine, 672
GetObliquityThresholdCosineValue, 672
GetType, 672
operator<<, 673
Orientation, 671
OrientationType, 671
Print, 672
SetObliquityThresholdCosineValue, 672
gdcm::Overlay, 673
~Overlay, 676
Decompress, 677
GetBitPosition, 677
GetBitsAllocated, 677
GetColumns, 677
GetDescription, 677
GetGroup, 677
GetOrigin, 677
GetOverlayData, 677
GetOverlayTypeAsString, 678
GetOverlayTypeFromString, 678
GetRows, 678
GetType, 678
GetTypeAsEnum, 678
GetUnpackBuffer, 678
GetUnpackBufferLength, 678
GrabOverlayFromPixelData, 679
IsEmpty, 679
IsInPixelData, 679
IsZero, 679
operator=, 679
Overlay, 676
OverlayType, 676
Print, 679
SetBitPosition, 680
SetBitsAllocated, 680
SetColumns, 680
SetDescription, 680
SetFrameOrigin, 680
SetGroup, 680
SetNumberOfFrames, 681
SetOrigin, 681
SetOverlay, 681
SetRows, 681
SetType, 681
Update, 681
gdcm::PDBElement, 691
GetName, 692
GetValue, 692
NameField, 693
operator<<, 693
operator==, 693
PDBElement, 692
SetName, 693
SetValue, 693
ValueField, 693
gdcm::PDBHeader, 694
~PDBHeader, 695
FindPDBElementByName, 695
GetPDBEEnd, 695
GetPDBElementByName, 695
GetPDBInfoTag, 696
LoadFromDataElement, 696
operator<<, 696
PDBHeader, 695
Print, 696
gdcm::PDFCodec, 697
~PDFCodec, 698
CanCode, 698
CanDecode, 698
Decode, 698
PDFCodec, 698
gdcm::PGXCodec, 705
~PGXCodec, 706
CanCode, 706
CanDecode, 706
Clone, 706
GetHeaderInfo, 707
PGXCodec, 706
Read, 707
Write, 707
gdcm::PNMCodec, 734
~PNMCodec, 736
CanCode, 736
CanDecode, 736
Clone, 736
GetBufferLength, 736
GetHeaderInfo, 736
PNMCodec, 736
Read, 737
SetBufferLength, 737
Write, 737
gdcm::PVRGCodec, 770
~PVRGCodec, 771
CanCode, 771

CanDecode, 771
Clone, 772
Code, 772
Decode, 772
PVRGCodec, 771
SetLossyFlag, 772
gdcm::ParseException, 682
~ParseException, 683
GetLastElement, 684
operator=, 684
ParseException, 683
SetLastElement, 684
gdcm::Parser, 684
~Parser, 686
EndElementHandler, 685
ErrorType, 685
GetBuffer, 686
GetCurrentByteIndex, 686
GetErrorCode, 686
GetErrorString, 686
GetUserData, 686
Parse, 687
ParseBuffer, 687
Parser, 686
Process, 687
SetElementHandler, 687
SetUserData, 687
StartElementHandler, 685
gdcm::Patient, 687
Patient, 688
gdcm::PersonName, 702
Component, 704
GetMaxLength, 703
GetNumberOfComponents, 703
MaxLength, 704
MaxNumberOfComponents, 704
Padding, 704
Print, 703
Separator, 704
SetBlob, 703
SetComponents, 703
gdcm::PhotometricInterpretation, 707
GetPIString, 710
GetPIType, 710
GetSamplesPerPixel, 710
GetString, 711
GetType, 711
IsLossless, 711
IsLossy, 711
IsRetired, 711
IsSameColorSpace, 711
operator PIType, 711
operator<<, 712
PIType, 709
PhotometricInterpretation, 710
gdcm::PixelFormat, 712
Bitmap, 719
GetBitsAllocated, 715
GetBitsStored, 715
GetHighBit, 715
GetMax, 715
GetMin, 716
GetPixelRepresentation, 716
GetPixelSize, 716
GetSamplesPerPixel, 716
GetScalarType, 716
GetScalarTypeAsString, 717
IsCompatible, 717
IsValid, 717
operator ScalarType, 717
operator!=, 717
operator<<, 719
operator==, 717
PixelFormat, 715
Print, 718
ScalarType, 714
SetBitsAllocated, 718
SetBitsStored, 718
SetHighBit, 718
SetPixelRepresentation, 718
SetSamplesPerPixel, 718
SetScalarType, 718
Validate, 719
gdcm::Pixmap, 720
~Pixmap, 721
AreOverlaysInPixelData, 722
Curves, 724
GetCurve, 722
GetIconImage, 722
GetNumberOfCurves, 722
GetNumberOfOverlays, 722
GetOverlay, 723
Icon, 724
Overlays, 724
Pixmap, 721
Print, 723
RemoveOverlay, 723
SetIconImage, 723
SetNumberOfCurves, 723
SetNumberOfOverlays, 723
gdcm::PixmapReader, 724
~PixmapReader, 726
GetPixmap, 726
PixelData, 727
PixmapReader, 726
Read, 726
ReadACRNEMAImage, 727
ReadImage, 727

ReadImageInternal, 727
gdcm::PixmapToPixmapFilter, 728
~PixmapToPixmapFilter, 729
GetInput, 729
GetOutput, 729
GetOutputAsPixmap, 729
PixmapToPixmapFilter, 729
gdcm::PixmapWriter, 730
~PixmapWriter, 732
DoIconImage, 732
GetImage, 732, 733
GetPixmap, 733
PixelData, 734
PixmapWriter, 732
PrepareWrite, 733
SetImage, 733
SetPixmap, 733
Write, 734
gdcm::Preamble, 737
~Preamble, 738
Clear, 739
Create, 739
GetInternal, 739
GetLength, 739
IsEmpty, 739
IsValid, 739
operator<<, 741
operator=, 740
Preamble, 738, 739
Print, 740
Read, 740
Remove, 740
Valid, 740
Write, 740
gdcm::PresentationContext, 741
AbstractSyntax, 744
AddTransferSyntax, 743
GetAbstractSyntax, 743
GetNumberOfTransferSyntaxes, 743
GetPresentationContextID, 743
GetTransferSyntax, 743
ID, 744
operator==, 743
PresentationContext, 743
Print, 744
SetAbstractSyntax, 744
SetPresentationContextID, 744
SizeType, 742
TransferSyntaxArrayType, 742
TransferSyntaxes, 744
gdcm::PresentationContextGenerator, 747
AddFromFile, 749
AddPresentationContext, 749
GenerateFromFilenames, 749
GenerateFromUID, 749
GetDefaultTransferSyntax, 749
GetPresentationContexts, 749
PresentationContextArrayType, 748
PresentationContextGenerator, 748
SetDefaultTransferSyntax, 749
SetMergeModeToAbstractSyntax, 750
SetMergeModeToTransferSyntax, 750
SizeType, 748
gdcm::Printer, 757
~Printer, 759
F, 761
GetPrintStyle, 759
MaxPrintLength, 761
Print, 759
PrintDataElement, 759
PrintDataSet, 759
PrintSQ, 760
PrintStyle, 761
PrintStyles, 758
Printer, 759
SetColor, 760
SetFile, 760
SetStyle, 760
gdcm::PrivateDict, 761
~PrivateDict, 762
AddDictEntry, 762
Dicts, 763
FindDictEntry, 762
GetDictEntry, 762
IsEmpty, 762
LoadDefault, 762
operator<<, 763
PrintXML, 762
PrivateDict, 762
RemoveDictEntry, 763
gdcm::PrivateTag, 763
GetAsDataElement, 765
GetOwner, 765
operator<, 765
operator<<, 766
PrivateTag, 765
ReadFromCommaSeparatedString, 766
SetOwner, 766
gdcm::ProgressEvent, 766
~ProgressEvent, 768
CheckEvent, 769
GetEventName, 769
GetProgress, 769
MakeObject, 769
ProgressEvent, 768
Self, 768
SetProgress, 769
Superclass, 768

gdcm::PythonFilter, 773
~PythonFilter, 773
GetFile, 773
PythonFilter, 773
SetDicts, 773
SetFile, 774
ToPyObject, 774
UseDictAlways, 774
gdcm::QueryBase, 774
~QueryBase, 775
GetAllRequiredTags, 776
GetAllTags, 776
GetHierachicalSearchTags, 776
GetName, 776
GetOptionalTags, 776
GetQueryLevel, 776
GetRequiredTags, 776
GetUniqueTags, 777
gdcm::QueryFactory, 777
GetCharacterFromCurrentLocale, 778
ListCharSets, 778
ProduceCharacterSetDataElement, 778
ProduceQuery, 778
gdcm::QueryImage, 779
GetHierachicalSearchTags, 780
GetName, 780
GetOptionalTags, 780
GetQueryLevel, 780
GetRequiredTags, 780
GetUniqueTags, 780
gdcm::QueryPatient, 781
GetHierachicalSearchTags, 782
GetName, 782
GetOptionalTags, 782
GetQueryLevel, 782
GetRequiredTags, 782
GetUniqueTags, 783
gdcm::QuerySeries, 783
GetHierachicalSearchTags, 784
GetName, 784
GetOptionalTags, 784
GetQueryLevel, 785
GetRequiredTags, 785
GetUniqueTags, 785
gdcm::QueryStudy, 785
GetHierachicalSearchTags, 787
GetName, 787
GetOptionalTags, 787
GetQueryLevel, 787
GetRequiredTags, 787
GetUniqueTags, 787
gdcm::RAWCodec, 788
~RAWCodec, 789
CanCode, 789
CanDecode, 789
Clone, 790
Code, 790
Decode, 790
DecodeByStreams, 790
DecodeBytes, 790
GetHeaderInfo, 791
RAWCodec, 789
gdcm::RLECodec, 806
~RLECodec, 807
AppendFrameEncode, 808
AppendRowEncode, 808
CanCode, 808
CanDecode, 808
Clone, 808
Code, 809
Decode, 809
DecodeByStreams, 809
DecodeExtent, 809
GetBufferLength, 809
GetHeaderInfo, 810
ImageRegionReader, 811
IsFrameEncoder, 810
IsRowEncoder, 810
RLECodec, 807
SetBufferLength, 810
SetLength, 810
StartEncode, 810
StopEncode, 810
gdcm::Reader, 791
~Reader, 794
CanRead, 794
F, 797
GetFile, 794
GetStreamCurrentPosition, 795
GetStreamPtr, 795
Read, 795
ReadDataSet, 795
ReadMetaInformation, 795
ReadPreamble, 795
ReadSelectedPrivateTags, 796
ReadSelectedTags, 796
ReadUpToTag, 796
Reader, 794
SetFile, 796
SetFileName, 796
SetStream, 797
StreamImageReader, 797
gdcm::RealWorldValueMappingContent, 798
CodeMeaning, 798
CodeValue, 798
RealWorldValueIntercept, 798
RealWorldValueSlope, 799
gdcm::Region, 799

~Region, 800
Area, 800
Clone, 800
ComputeBoundingBox, 800
Empty, 800
IsValid, 801
Print, 801
Region, 800
gdcm::Rescaler, 801
~Rescaler, 803
ComputeInterceptSlopePixelType, 803
ComputePixelTypeFromMinMax, 803
GetIntercept, 803
GetSlope, 804
InverseRescale, 804
InverseRescaleFunctionIntoBestFit, 804
Rescale, 804
RescaleFunctionIntoBestFit, 804
Rescaler, 803
SetIntercept, 804
SetMinMaxForPixelType, 805
SetPixelFormat, 805
SetSlope, 805
SetTargetPixelType, 805
SetUseTargetPixelType, 805
gdcm::SHA1, 867
~SHA1, 868
Compute, 868
ComputeFile, 868
SHA1, 868
gdcm::SOPClassUIDToIOD, 881
const, 882
GetIODFromSOPClassUID, 882
GetIOD, 882
GetNumberOfSOPClassToIOD, 882
GetSOPClassUIDFromIOD, 882
GetSOPClassUIDToIODs, 882
GetSOPClassUIDToIOD, 882
gdcm::STATIC_ASSERTION_FAILURE< true >, 895
gdcm::STATIC_ASSERTION_FAILURE< x >, 894
gdcm::Scanner, 814
~Scanner, 818
AddPrivateTag, 818
AddSkipTag, 818
AddTag, 818
Begin, 818
ClearSkipTags, 818
ClearTags, 819
ConstIterator, 817
End, 819
GetAllFilenamesFromTagToValue, 819
GetFilenameFromTagToValue, 819
GetFilenames, 819
GetKeys, 819
GetMapping, 819
GetMappingFromTagToValue, 820
GetMappings, 820
GetOrderedValues, 820
GetValue, 820
GetValues, 820, 821
IsKey, 821
MappingType, 817
New, 821
operator<<, 822
Print, 821
ProcessPublicTag, 821
Scan, 822
Scanner, 818
TagToValue, 817
TagToValueValueType, 817
ValuesType, 817
gdcm::Scanner::ltstr, 585
operator(), 585
gdcm::Segment, 822
~Segment, 825
ALGOType, 824
AddSurface, 825
AnatomicRegion, 828
GetALGOType, 825
GetALGOTypeString, 825
GetAnatomicRegion, 825
GetPropertyCategory, 825, 826
GetPropertyType, 826
GetSegmentAlgorithmName, 826
GetSegmentAlgorithmType, 826
GetSegmentDescription, 826
GetSegmentLabel, 826
GetSegmentNumber, 826
GetSurface, 826
GetSurfaceCount, 827
GetSurfaces, 827
PropertyCategory, 828
PropertyType, 828
Segment, 825
SegmentAlgorithmName, 828
SegmentAlgorithmType, 829
SegmentDescription, 829
SegmentLabel, 829
SegmentNumber, 829
SetAnatomicRegion, 827
SetPropertyCategory, 827
SetPropertyType, 827
SetSegmentAlgorithmName, 827
SetSegmentAlgorithmType, 827
SetSegmentDescription, 828
SetSegmentLabel, 828
SetSegmentNumber, 828
SetSurfaceCount, 828

SurfaceCount, 829
SurfaceVector, 824
Surfaces, 829
gdcm::SegmentHelper, 79
gdcm::SegmentHelper::BasicCodedEntry, 173
BasicCodedEntry, 174
CSD, 175
CSV, 175
CM, 175
CV, 175
IsEmpty, 174
gdcm::SegmentReader, 832
~SegmentReader, 834
GetSegments, 834
Read, 834
ReadSegment, 834
ReadSegments, 834
SegmentMap, 833
SegmentReader, 834
SegmentVector, 833
Segments, 835
gdcm::SegmentWriter, 835
~SegmentWriter, 837
AddSegment, 837
GetNumberOfSegments, 837
GetSegment, 837
GetSegments, 837
PrepareWrite, 837
SegmentVector, 836
SegmentWriter, 837
Segments, 838
SetNumberOfSegments, 837
SetSegments, 838
Write, 838
gdcm::SegmentedPaletteColorLookupTable, 830
~SegmentedPaletteColorLookupTable, 831
Print, 831
SegmentedPaletteColorLookupTable, 831
SetLUT, 831
gdcm::SequenceOfFragments, 838
AddFragment, 841
Begin, 841, 842
Clear, 842
ComputeByteLength, 842
ComputeLength, 842
ConstIterator, 841
End, 842
FragmentVector, 841
GetBuffer, 842
GetFragBuffer, 842
GetFragment, 843
GetLength, 843
GetNumberOfFragments, 843
GetTable, 843
Iterator, 841
New, 843
operator==, 844
Print, 844
Read, 844
ReadPreValue, 844
ReadValue, 844
SequenceOfFragments, 841
SetLength, 844
SizeType, 841
Write, 845
WriteBuffer, 845
gdcm::SequenceOfItems, 845
AddItem, 849
AddNewUndefinedLengthItem, 849
Begin, 849
Clear, 849
ComputeLength, 850
ConstIterator, 848
End, 850
FindDataElement, 850
GetItem, 850
GetLength, 850
GetNumberOfItems, 851
IsUndefinedLength, 851
ItemVector, 848
Items, 853
Iterator, 848
New, 851
operator=, 851
operator==, 851
Print, 851
Read, 852
RemoveItemByIndex, 852
SequenceLengthField, 853
SequenceOfItems, 849
SetLength, 852
SetLengthToUndefined, 852
SetNumberOfItems, 853
SizeType, 848
Write, 853
gdcm::SerieHelper, 854
~SerieHelper, 855
AddFile, 856
AddFileName, 856
AddRestriction, 856
Clear, 856
CreateDefaultUniqueSeriesIdentifier, 856
CreateUniqueSeriesIdentifier, 856
FileNameOrdering, 857
GetFirstSingleSerieUIDFileSet, 857
GetNextSingleSerieUIDFileSet, 857
ImagePositionPatientOrdering, 857
ItFileSetHt, 858

OrderFileList, 857
SerieHelper, 855
SerieRestrictions, 855
SetDirectory, 857
SetLoadMode, 857
SetUseSeriesDetails, 857
SingleSerieUIDFileSetHT, 858
SingleSerieUIDFileSetmap, 855
UserOrdering, 858
gdcm::SerieHelper::Rule, 813
elem, 813
group, 813
op, 813
value, 813
gdcm::Series, 858
Series, 859
gdcm::ServiceClassUser, 860
~ServiceClassUser, 862
GetAETitle, 863
GetCalledAETitle, 863
GetTimeout, 863
InitializeConnection, 863
IsPresentationContextAccepted, 863
New, 863
SendEcho, 863
SendFind, 864
SendMove, 864
SendStore, 864, 865
ServiceClassUser, 862
SetAETitle, 865
SetCalledAETitle, 865
SetHostname, 865
SetPort, 865
SetPortSCP, 866
SetPresentationContexts, 866
SetTimeout, 866
StartAssociation, 866
StopAssociation, 867
gdcm::SimpleMemberCommand
~SimpleMemberCommand, 871
Execute, 872
m_MemberFunction, 872
m_This, 872
New, 872
Self, 871
SetCallbackFunction, 872
SimpleMemberCommand, 871
TMemberFunctionPointer, 871
gdcm::SimpleMemberCommand< T >, 869
gdcm::SimpleSubjectWatcher, 873
~SimpleSubjectWatcher, 874
EndFilter, 874
ShowAbort, 874
ShowAnonymization, 874
ShowData, 874
ShowDataSet, 874
ShowFileName, 874
ShowIteration, 875
ShowProgress, 875
SimpleSubjectWatcher, 874
StartFilter, 875
TestAbortOff, 875
TestAbortOn, 875
gdcm::SmartPointer
~SmartPointer, 878
GetPointer, 878
operator ObjectType *, 878
operator*, 878
operator->, 878
operator=, 879
SmartPointer, 877, 878
gdcm::SmartPointer< ObjectType >, 876
gdcm::Sorter, 883
~Sorter, 885
AddSelect, 885
Filenames, 887
GetFilenames, 885
operator<<, 887
Print, 885
Selection, 887
SelectionMap, 885
SetSortFunction, 886
Sort, 886
SortFunc, 887
SortFunction, 885
Sorter, 885
StableSort, 886
gdcm::Spacing, 887
~Spacing, 890
ComputePixelAspectRatioFromPixelSpacing, 890
Spacing, 890
SpacingType, 889
gdcm::Spectroscopy, 890
Spectroscopy, 891
gdcm::SplitMosaicFilter, 891
~SplitMosaicFilter, 892
ComputeMOSAICDimensions, 892
GetFile, 892
GetImage, 892
SetFile, 892
SetImage, 892
Split, 893
SplitMosaicFilter, 892
gdcm::StartEvent, 893
gdcm::StreamImageReader, 895
~StreamImageReader, 896
CanReadImage, 896
DefinePixelExtent, 896

DefineProperBufferLength, 897
GetDimensionsValueForResolution, 897
GetFile, 897
Read, 897
ReadImageInformation, 898
SetFileName, 898
SetStream, 898
StreamImageReader, 896
gdcm::StreamImageWriter, 899
~StreamImageWriter, 901
CanWriteFile, 902
DefinePixelExtent, 902
DefineProperBufferLength, 902
mElementOffsets, 904
mElementOffsets1, 904
mWriter, 905
mXMax, 905
mXMin, 905
mYMax, 905
mYMin, 905
mZMax, 905
mZMin, 905
mspFile, 904
SetFile, 902
SetFileName, 903
SetStream, 903
StreamImageWriter, 901
Write, 903
WriteImageInformation, 903
WriteImageSubregionRAW, 904
WriteRawHeader, 904
gdcm::StrictScanner, 906
~StrictScanner, 909
AddPrivateTag, 909
AddSkipTag, 909
AddTag, 910
Begin, 910
ClearSkipTags, 910
ClearTags, 910
ConstIterator, 908
End, 910
GetAllFilenamesFromTagToValue, 910
GetFilenameFromTagToValue, 910
GetFilenames, 911
GetKeys, 911
GetMapping, 911
GetMappingFromTagToValue, 911
GetMappings, 911
GetOrderedValues, 911
GetValue, 912
GetValues, 912
IsKey, 912
MappingType, 908
New, 912
operator<<, 913
Print, 913
ProcessPublicTag, 913
Scan, 913
StrictScanner, 909
TagToValue, 909
TagToValueValueType, 909
ValuesType, 909
gdcm::StrictScanner::ltstr, 585
operator(), 586
gdcm::String
const_iterator, 916
const_reference, 916
const_reverse_iterator, 916
difference_type, 916
IsValid, 918
iterator, 916
operator const char *, 918
pointer, 916
reference, 916
reverse_iterator, 916
size_type, 917
String, 917
Trim, 918
Truncate, 918
value_type, 917
gdcm::String< TDelimiter, TMaxLength, TPadChar >, 914
gdcm::StringFilter, 919
~StringFilter, 920
ExecuteQuery, 920
FromString, 920
GetFile, 920, 921
SetDicts, 921
SetFile, 921
StringFilter, 920
ToString, 921
ToStringPair, 921, 922
UseDictAlways, 922
gdcm::Study, 922
Study, 923
gdcm::Subject, 923
~Subject, 924
AddObserver, 925
GetCommand, 925
HasObserver, 925
InvokeEvent, 925
RemoveAllObservers, 926
RemoveObserver, 926
Subject, 924
gdcm::Surface, 926
~Surface, 930
GetAlgorithmFamily, 930
GetAlgorithmName, 930
GetAlgorithmVersion, 930

GetAxisOfRotation, 930
GetCenterOfRotation, 931
GetFiniteVolume, 931
GetManifold, 931
GetMaximumPointDistance, 931
GetMeanPointDistance, 931
GetMeshPrimitive, 931
GetNumberOfSurfacePoints, 931
GetNumberOfVectors, 932
GetPointCoordinatesData, 932
GetPointPositionAccuracy, 932
GetPointsBoundingBoxCoordinates, 932
GetProcessingAlgorithm, 932
GetRecommendedDisplayCIELabValue, 932, 933
GetRecommendedDisplayGrayscaleValue, 933
GetRecommendedPresentationOpacity, 933
GetRecommendedPresentationType, 933
GetSTATESString, 933
GetSTATES, 933
GetSurfaceComments, 933
GetSurfaceNumber, 933
GetSurfaceProcessing, 933
GetSurfaceProcessingDescription, 934
GetSurfaceProcessingRatio, 934
GetVIEWType, 934
GetVIEWTypeString, 934
GetVectorAccuracy, 934
GetVectorCoordinateData, 934
GetVectorDimensionality, 934
STATES, 929
SetAlgorithmFamily, 934
SetAlgorithmName, 935
SetAlgorithmVersion, 935
SetAxisOfRotation, 935
SetCenterOfRotation, 935
SetFiniteVolume, 935
SetManifold, 935
SetMaximumPointDistance, 935
SetMeanPointDistance, 935
SetMeshPrimitive, 936
SetNumberOfSurfacePoints, 936
SetNumberOfVectors, 936
SetPointCoordinatesData, 936
SetPointPositionAccuracy, 936
SetPointsBoundingBoxCoordinates, 936
SetProcessingAlgorithm, 936
SetRecommendedDisplayCIELabValue, 936, 937
SetRecommendedDisplayGrayscaleValue, 937
SetRecommendedPresentationOpacity, 937
SetRecommendedPresentationType, 937
SetSurfaceComments, 937
SetSurfaceNumber, 937
SetSurfaceProcessing, 937
SetSurfaceProcessingDescription, 938
SetSurfaceProcessingRatio, 938
SetVectorAccuracy, 938
SetVectorCoordinateData, 938
SetVectorDimensionality, 938
Surface, 930
VIEWType, 929
gdcm::SurfaceHelper, 938
ColorArray, 939
RGBToRecommendedDisplayCIELab, 940
RGBToRecommendedDisplayGrayscale, 941
RecommendedDisplayCIELabToRGB, 939, 940
gdcm::SurfaceReader, 941
~SurfaceReader, 943
GetNumberOfSurfaces, 943
Read, 943
ReadPointMacro, 944
ReadSurface, 944
ReadSurfaces, 944
SurfaceReader, 943
gdcm::SurfaceWriter, 944
~SurfaceWriter, 946
ComputeNumberOfSurfaces, 946
GetNumberOfSurfaces, 946
NumberOfSurfaces, 947
PrepareWrite, 946
PrepareWritePointMacro, 946
SetNumberOfSurfaces, 946
SurfaceWriter, 946
Write, 947
gdcm::SwapCode, 947
GetIndex, 949
GetSwapCodeString, 949
operator SwapCode::SwapCodeType, 949
operator<<, 949
SwapCode, 948
SwapCodeType, 948
gdcm::SwapperDoOp, 949
Swap, 950
SwapArray, 950
gdcm::SwapperNoOp, 950
Swap, 950
SwapArray, 950
gdcm::System, 951
DeleteDirectory, 952
EncodeBytes, 952
FileExists, 952
FileIsDirectory, 953
FileIsSymlink, 953
FileSize, 953
FileTime, 953
FormatDateTime, 954
GetCWD, 954
GetCurrentDateTime, 954
GetCurrentModuleFileName, 954

GetCurrentProcessFileName, 954
GetCurrentResourcesDirectory, 954
GetHostName, 955
GetLastSystemError, 955
GetLocaleCharset, 955
GetPermissions, 955
GetTimezoneOffsetFromUTC, 955
MakeDirectory, 955
ParseDateTime, 956
RemoveFile, 956
SetPermissions, 956
StrCaseCmp, 956
StrNCaseCmp, 957
StrSep, 957
StrTokR, 957
gdcm::Table, 957
~Table, 958
GetTableEntry, 959
InsertEntry, 959
MapTableEntry, 958
operator<<, 959
Table, 958
gdcm::TableEntry, 959
~TableEntry, 960
TableEntry, 960
gdcm::TableReader, 960
~TableReader, 961
CharacterDataHandler, 961
EndElement, 961
GetDefs, 962
GetFilename, 962
HandleIODEntry, 962
HandleIOD, 962
HandleMacro, 962
HandleMacroEntry, 962
HandleMacroEntryDescription, 962
HandleModule, 962
HandleModuleEntry, 963
HandleModuleEntryDescription, 963
HandleModuleInclude, 963
Read, 963
SetFilename, 963
StartElement, 963
TableReader, 961
gdcm::Tag, 965
bytes, 974
GetElement, 968
GetElementTag, 968
GetGroup, 968
GetLength, 968
GetPrivateCreator, 969
IsGroupLength, 969
IsGroupXX, 969
IsIllegal, 969
IsPrivate, 969
IsPrivateCreator, 969
IsPublic, 970
operator!=, 970
operator<, 970
operator<<, 974
operator<=, 970
operator>>, 974
operator=, 970
operator==, 970
operator[], 971
PrintAsContinuousString, 971
PrintAsContinuousUpperCaseString, 971
PrintAsPipeSeparatedString, 971
Read, 971
ReadFromCommaSeparatedString, 972
ReadFromContinuousString, 972
ReadFromPipeSeparatedString, 972
SetElement, 972
SetElementTag, 972, 973
SetGroup, 973
SetPrivateCreator, 973
Tag, 967, 968
tag, 974
tags, 974
Write, 973
gdcm::TagPath, 974
~TagPath, 975
ConstructFromString, 975
ConstructFromTagList, 975
IsValid, 976
Print, 976
Push, 976
TagPath, 975
gdcm::Testing, 976
~Testing, 978
ComputeFileMD5, 978
ComputeMD5, 978
GetDataExtraRoot, 979
GetDataRoot, 979
GetFileName, 979
GetFileNames, 979
GetLossyFlagFromFile, 980
GetMD5DataImage, 980
GetMD5DataImages, 980
GetMD5FromBrokenFile, 980
GetMD5FromFile, 980
GetMediaStorageDataFile, 980
GetMediaStorageDataFiles, 980
GetMediaStorageFromFile, 981
GetNumberOfFileNames, 981
GetNumberOfMD5DataImages, 981
GetNumberOfMediaStorageDataFiles, 981
GetPixelSpacingDataRoot, 981

GetSelectedPrivateGroupOffsetFromFile, 981
GetSelectedTagsOffsetFromFile, 981
GetSourceDirectory, 982
GetStreamOffsetFromFile, 982
GetTempDirectory, 982
GetTempDirectoryW, 982
GetTempFilename, 982
GetTempFilenameW, 983
MD5DataImagesType, 978
MediaStorageDataFilesType, 978
Print, 983
Testing, 978
gdcm::Trace, 983
~Trace, 985
DebugOff, 985
DebugOn, 985
ErrorOff, 985
ErrorOn, 985
GetDebugFlag, 985
GetDebugStream, 986
GetErrorFlag, 986
GetErrorStream, 986
GetStream, 986
GetWarningFlag, 986
GetWarningStream, 986
SetDebug, 986
SetDebugStream, 986
SetError, 987
SetErrorStream, 987
SetStream, 987
SetStreamToFile, 987
SetWarning, 987
SetWarningStream, 988
Trace, 985
WarningOff, 988
WarningOn, 988
gdcm::TransferSyntax, 988
CanStoreLossy, 991
GetNegociatedType, 992
GetString, 992
GetSwapCode, 992
GetTSString, 992
GetTSType, 992
IsEncapsulated, 992
IsEncoded, 993
IsExplicit, 993
IsImplicit, 993
IsLossless, 993
IsLossy, 993
IsValid, 993
NegociatedType, 990
operator TSType, 993
operator<<, 994
TSType, 990
TransferSyntax, 991
gdcm::Type, 998
GetTypeString, 999
GetTypeType, 1000
operator TypeType, 1000
operator<<, 1000
Type, 999
TypeType, 999
gdcm::UIDGenerator, 1001
Generate, 1002
GenerateUUID, 1002
GetGDCMUID, 1002
GetRoot, 1003
IsValid, 1003
SetRoot, 1003
UIDGenerator, 1002
gdcm::UIDs, 1004
GetName, 1027
GetNumberOfTransferSyntaxStrings, 1027
GetString, 1028
GetTransferSyntaxString, 1028
GetTransferSyntaxStrings, 1028
GetUIDName, 1028
GetUIDString, 1028
operator TSType, 1028
SetFromUID, 1028
TSName, 1015
TSType, 1021
TransferSyntaxStringsType, 1014
gdcm::UNExplicitDataElement, 1090
GetLength, 1091
Read, 1091
ReadPreValue, 1091
ReadValue, 1091
ReadWithLength, 1091
gdcm::UNExplicitImplicitDataElement, 1092
GetLength, 1094
Read, 1094
ReadPreValue, 1094
ReadValue, 1094
gdcm::UUIDGenerator, 1101
Generate, 1101
IsValid, 1101
gdcm::UI, 1000
Internal, 1001
operator<<, 1001
gdcm::Unpacker12Bits, 1094
Pack, 1095
Unpack, 1095
gdcm::Usage, 1095
GetUsageString, 1097
GetUsageType, 1097
operator UsageType, 1097
operator<<, 1097

Usage, 1097
UsageType, 1096
gdcm::UserEvent, 1098
gdcm::VMToLength< T >, 1118
gdcm::VR16ExplicitDataElement, 1124
GetLength, 1126
Read, 1126
ReadPreValue, 1126
ReadValue, 1126
ReadWithLength, 1126
gdcm::VRToEncoding< T >, 1127
gdcm::VRToType< T >, 1127
gdcm::VRVLSize< 0 >, 1128
Read, 1128
Write, 1128
gdcm::VRVLSize< 1 >, 1128
Read, 1129
Write, 1129
gdcm::VRVLSize< T >, 1128
gdcm::Validate, 1102
∼Validate, 1103
F, 1103
GetValidatedFile, 1103
SetFile, 1103
V, 1103
Validate, 1103
Validation, 1103
gdcm::Value, 1104
∼Value, 1105
Clear, 1105
DataElement, 1106
GetLength, 1105
operator==, 1106
SetLength, 1106
SetLengthOnly, 1106
Value, 1105
gdcm::ValueIO< TDE, TSwap, TType >, 1106
gdcm::ValueIO
Read, 1107
Write, 1107
gdcm::Version, 1107
∼Version, 1108
GetBuildVersion, 1108
GetMajorVersion, 1108
GetMinorVersion, 1108
GetVersion, 1108
operator<<, 1109
Print, 1109
Version, 1108
gdcm::VL, 1109
GetLength, 1111
GetVL16Max, 1111
GetVL32Max, 1111
IsOdd, 1111
IsUndefined, 1111
operator uint32_t, 1111
operator<<, 1113
operator++, 1111, 1112
operator+=, 1112
Read, 1112
Read16, 1112
SetToUndefined, 1112
Type, 1110
VL, 1111
Write, 1112
Write16, 1112
gdcm::VM, 1113
Compatible, 1116
GetIndex, 1116
GetLength, 1116
GetNumberOfElementsFromArray, 1116
GetVMString, 1116
GetVMType, 1117
GetVMTypeFromLength, 1117
IsValid, 1117
operator VMType, 1117
operator<<, 1117
VMType, 1115
VM, 1116
gdcm::VR, 1118
CanDisplay, 1121
Compatible, 1121
GetLength, 1121
GetSize, 1122
GetSizeof, 1122
GetVRString, 1122
GetVRStringFromFile, 1122
GetVRType, 1122
GetVRTypeFromFile, 1122
IsASCII2, 1122
IsASCII, 1122
IsBinary, 1123
IsBinary2, 1123
IsDual, 1123
IsSwap, 1123
IsVRFile, 1123
IsValid, 1123
operator VRType, 1123
operator<<, 1124
Read, 1124
VRType, 1120
VR, 1121
Write, 1124
gdcm::WLMFindQuery, 1228
GetAbstractSyntaxUID, 1230
GetTagListByLevel, 1230
GetValidDataSet, 1230
InitializeDataSet, 1230

QueryFactory, 1231
ValidateQuery, 1230
WLMFindQuery, 1229
gdcm::Waveform, 1227
Waveform, 1228
gdcm::Writer, 1231
∼Writer, 1234
CheckFileMetaInformationOff, 1234
CheckFileMetaInformationOn, 1234
GetCheckFileMetaInformation, 1234
GetFile, 1234
GetStreamPtr, 1235
Ofstream, 1237
SetCheckFileMetaInformation, 1235
SetFile, 1235
SetFileName, 1235
SetStream, 1236
SetWriteDataSetOnly, 1236
Stream, 1237
StreamImageWriter, 1237
Write, 1236
Writer, 1234
gdcm::XMLDictReader, 1237
∼XMLDictReader, 1238
CharacterDataHandler, 1239
EndElement, 1239
GetDict, 1239
HandleDescription, 1239
HandleEntry, 1239
StartElement, 1239
XMLDictReader, 1238
gdcm::XMLPrinter, 1240
∼XMLPrinter, 1241
F, 1242
GetPrintStyle, 1241
HandleBulkData, 1241
Print, 1241
PrintDataElement, 1242
PrintDataSet, 1242
PrintSQ, 1242
PrintStyle, 1242
PrintStyles, 1241
SetFile, 1242
SetStyle, 1242
XMLPrinter, 1241
gdcm::XMLPrivateDictReader, 1243
∼XMLPrivateDictReader, 1244
CharacterDataHandler, 1244
EndElement, 1244
GetPrivateDict, 1245
HandleDescription, 1245
HandleEntry, 1245
StartElement, 1245
XMLPrivateDictReader, 1244
gdcm::ignore_char, 459
ignore_char, 459
m_char, 459
gdcm::network, 73
cMaxEventID, 79
cMaxStateID, 79
EEventID, 77
EStateID, 78
GetStateIndex, 78
gdcm::network::AAbortPDU, 83
AAbortPDU, 84
IsLastFragment, 84
Print, 84
Read, 84
SetReason, 84
SetSource, 85
Size, 85
Write, 85
gdcm::network::AAssociateACPDU, 85
AAssociateACPDU, 87
AAssociateRQPDU, 89
AddPresentationContextAC, 87
GetNumberOfPresentationContextAC, 87
GetPresentationContextAC, 87
GetUserInformation, 87
InitFromRQ, 87
IsLastFragment, 88
Print, 88
Read, 88
SetCalledAETitle, 88
SetCallingAETitle, 88
Size, 88
SizeType, 87
Write, 89
gdcm::network::AAssociateRJPDU, 89
AAssociateRJPDU, 90
IsLastFragment, 90
Print, 90
Read, 91
Size, 91
Write, 91
gdcm::network::AAssociateRQPDU, 91
AAssociateACPDU, 97
AAssociateRQPDU, 94
AddPresentationContext, 94
GetCalledAETitle, 94
GetCallingAETitle, 94
GetNumberOfPresentationContext, 94
GetPresentationContext, 94
GetPresentationContextByAbstractSyntax, 95
GetPresentationContextByID, 95
GetPresentationContexts, 95
GetReserved43_74, 95
GetUserInformation, 95

IsAETitleValid, 95
IsLastFragment, 96
PresentationContextArrayType, 93
Print, 96
Read, 96
SetCalledAETitle, 96
SetCallingAETitle, 96
SetUserInformation, 96
Size, 97
SizeType, 93
Write, 97
gdcm::network::ARTIMTimer, 121
ARTIMTimer, 121
GetElapsedTime, 121
GetHasExpired, 121
GetTimeout, 122
SetTimeout, 122
Start, 122
Stop, 122
gdcm::network::AReleaseRPPDU, 116
AReleaseRPPDU, 117
IsLastFragment, 117
Print, 117
Read, 118
Size, 118
Write, 118
gdcm::network::AReleaseRQPDU, 118
AReleaseRQPDU, 120
IsLastFragment, 120
Print, 120
Read, 120
Size, 120
Write, 120
gdcm::network::AbstractSyntax, 99
AbstractSyntax, 99
GetAsDataElement, 99
GetName, 99
operator==, 99
Print, 100
Read, 100
SetName, 100
SetNameFromUID, 100
Size, 100
Write, 100
gdcm::network::ApplicationContext, 112
ApplicationContext, 113
GetName, 113
Print, 113
Read, 113
SetName, 113
Size, 113
Write, 113
gdcm::network::AsynchronousOperationsWindowSub, 124
AsynchronousOperationsWindowSub, 124
Print, 124
Read, 124
Size, 124
Write, 124
gdcm::network::BaseCompositeMessage, 157
~BaseCompositeMessage, 158
ConstructPDV, 158
gdcm::network::BaseNormalizedMessage, 159
~BaseNormalizedMessage, 160
ConstructPDV, 160
gdcm::network::BasePDU, 161
~BasePDU, 162
IsLastFragment, 162
Print, 162
Read, 162
Size, 163
Write, 163
gdcm::network::CEchoRSP, 215
ConstructPDVByDataSet, 216
gdcm::network::CEchoRQ, 213
AffectedSOPClassUID, 215
ConstructPDV, 214
MessageID, 215
gdcm::network::CFind, 216
gdcm::network::CFindCancelRQ, 217
ConstructPDVByDataSet, 218
gdcm::network::CFindRSP, 220
ConstructPDVByDataSet, 221
gdcm::network::CFindRQ, 218
ConstructPDV, 219
gdcm::network::CMoveCancelRq, 221
ConstructPDVByDataSet, 222
gdcm::network::CMoveRSP, 224
ConstructPDVByDataSet, 225
gdcm::network::CMoveRQ, 222
ConstructPDV, 223
gdcm::network::CStoreRSP, 271
ConstructPDV, 272
gdcm::network::CStoreRQ, 269
ConstructPDV, 271
gdcm::network::CompositeMessageFactory, 237
ConstructCEchoRQ, 237
ConstructCFindRQ, 237
ConstructCMoveRQ, 237
ConstructCStoreRSP, 238
ConstructCStoreRQ, 238
gdcm::network::DIMSE, 338
CommandTypes, 339
gdcm::network::ImplementationClassUIDSub, 516
ImplementationClassUIDSub, 517
Print, 517
Read, 517
Size, 517

Write, 517
gdcm::network::ImplementationUIDSub, 517
ImplementationUIDSub, 518
Write, 518
gdcm::network::ImplementationVersionNameSub, 518
ImplementationVersionNameSub, 519
Print, 519
Read, 519
Size, 519
Write, 519
gdcm::network::MaximumLengthSub, 591
GetMaximumLength, 591
MaximumLengthSub, 591
Print, 591
Read, 592
SetMaximumLength, 592
Size, 592
Write, 592
gdcm::network::NActionRSP, 635
ConstructPDVByDataSet, 636
gdcm::network::NActionRQ, 633
ConstructPDV, 634
gdcm::network::NCreateRSP, 637
ConstructPDVByDataSet, 638
gdcm::network::NCreateRQ, 636
ConstructPDV, 637
gdcm::network::NDeleteRSP, 640
ConstructPDVByDataSet, 641
gdcm::network::NDeleteRQ, 639
ConstructPDV, 640
gdcm::network::NEventReportRSP, 646
ConstructPDVByDataSet, 647
gdcm::network::NEventReportRQ, 644
ConstructPDV, 645
gdcm::network::NGetRSP, 648
ConstructPDVByDataSet, 649
gdcm::network::NGetRQ, 647
ConstructPDV, 648
gdcm::network::NSetRSP, 656
ConstructPDVByDataSet, 657
gdcm::network::NSetRQ, 655
ConstructPDV, 656
gdcm::network::NormalizedMessageFactory, 651
ConstructNAction, 651
ConstructNCreate, 651
ConstructNDelete, 651
ConstructNEventReport, 651
ConstructNGet, 652
ConstructNSet, 652
gdcm::network::PDUFactory, 699
ConstructAbortPDU, 700
ConstructPDU, 700
ConstructReleasePDU, 700
CreateCEchoPDU, 700
CreateCFindPDU, 700
CreateCMovePDU, 700
CreateCStoreRQPDU, 700
CreateCStoreRSPPDU, 701
CreateNActionPDU, 701
CreateNCreatePDU, 701
CreateNDeletePDU, 701
CreateNEventReportPDU, 701
CreateNGetPDU, 701
CreateNSetPDU, 701
DetermineEventByPDU, 702
GetPDVs, 702
gdcm::network::PDataTFPDU, 688
AddPresentationDataValue, 690
GetNumberOfPresentationDataValues, 690
GetPresentationDataValue, 690
IsLastFragment, 690
PDataTFPDU, 690
Print, 690
Read, 690
ReadInto, 690
Size, 691
SizeType, 689
Write, 691
gdcm::network::PresentationContextAC, 745
GetPresentationContextID, 745
GetReason, 745
GetTransferSyntax, 746
PresentationContextAC, 745
Print, 746
Read, 746
SetPresentationContextID, 746
SetReason, 746
SetTransferSyntax, 746
Size, 746
Write, 746
gdcm::network::PresentationContextRQ, 750
AddTransferSyntax, 751
GetAbstractSyntax, 751, 752
GetNumberOfTransferSyntaxes, 752
GetPresentationContextID, 752
GetTransferSyntax, 752
GetTransferSyntaxes, 752
operator==, 752
PresentationContextRQ, 751
Print, 752
Read, 752
SetAbstractSyntax, 753
SetPresentationContextID, 753
Size, 753
SizeType, 751
Write, 753
gdcm::network::PresentationDataValue, 753
ConcatenatePDVBlobs, 754

ConcatenatePDVBlobsAsExplicit, 754
GetBlob, 755
GetIsCommand, 755
GetIsLastFragment, 755
GetMessageHeader, 755
GetPresentationContextID, 755
PresentationDataValue, 754
Print, 755
Read, 755
ReadInto, 755
SetBlob, 755
SetCommand, 756
SetDataSet, 756
SetLastFragment, 756
SetMessageHeader, 756
SetPresentationContextID, 756
Size, 756
Write, 756
gdcm::network::RoleSelectionSub, 811
Print, 812
Read, 812
RoleSelectionSub, 811
SetTuple, 812
Size, 812
Write, 812
gdcm::network::SOPClassExtendedNegociationSub, 879
Print, 880
Read, 880
SOPClassExtendedNegociationSub, 880
SetTuple, 880
Size, 880
Write, 881
gdcm::network::ServiceClassApplicationInformation, 859
Print, 859
Read, 859
ServiceClassApplicationInformation, 859
SetTuple, 860
Size, 860
Write, 860
gdcm::network::TableRow, 964
~TableRow, 964
TableRow, 964
transitions, 965
gdcm::network::TransferSyntaxSub, 994
GetName, 995
operator==, 995
Print, 995
Read, 995
SetName, 995
SetNameFromUID, 995
Size, 995
TransferSyntaxSub, 994
Write, 995
gdcm::network::Transition, 996
~Transition, 997
mAction, 997
mEnd, 997
MakeNew, 997
Transition, 997
gdcm::network::ULAction, 1029
~ULAction, 1031
PerformAction, 1032
ULAction, 1031
gdcm::network::ULActionAA1, 1032
PerformAction, 1033
gdcm::network::ULActionAA2, 1033
PerformAction, 1034
gdcm::network::ULActionAA3, 1035
PerformAction, 1035
gdcm::network::ULActionAA4, 1036
PerformAction, 1037
gdcm::network::ULActionAA5, 1037
PerformAction, 1038
gdcm::network::ULActionAA6, 1038
PerformAction, 1039
gdcm::network::ULActionAA7, 1040
PerformAction, 1040
gdcm::network::ULActionAA8, 1041
PerformAction, 1042
gdcm::network::ULActionAE1, 1042
PerformAction, 1043
gdcm::network::ULActionAE2, 1043
PerformAction, 1044
gdcm::network::ULActionAE3, 1045
PerformAction, 1045
gdcm::network::ULActionAE4, 1046
PerformAction, 1047
gdcm::network::ULActionAE5, 1047
PerformAction, 1048
gdcm::network::ULActionAE6, 1048
PerformAction, 1049
gdcm::network::ULActionAE7, 1050
PerformAction, 1050
gdcm::network::ULActionAE8, 1051
PerformAction, 1052
gdcm::network::ULActionAR1, 1052
PerformAction, 1053
gdcm::network::ULActionAR10, 1053
PerformAction, 1054
gdcm::network::ULActionAR2, 1055
PerformAction, 1055
gdcm::network::ULActionAR3, 1056
PerformAction, 1057
gdcm::network::ULActionAR4, 1057
PerformAction, 1058
gdcm::network::ULActionAR5, 1058
PerformAction, 1059
gdcm::network::ULActionAR6, 1060

PerformAction, 1060
gdcm::network::ULActionAR7, 1061
PerformAction, 1062
gdcm::network::ULActionAR8, 1062
PerformAction, 1063
gdcm::network::ULActionAR9, 1063
PerformAction, 1064
gdcm::network::ULActionDT1, 1065
PerformAction, 1065
gdcm::network::ULActionDT2, 1066
PerformAction, 1067
gdcm::network::ULBasicCallback, 1067
~ULBasicCallback, 1068
GetDataSets, 1069
GetResponses, 1069
HandleDataSet, 1069
HandleResponse, 1069
ULBasicCallback, 1068
gdcm::network::ULConnection, 1069
~ULConnection, 1071
AddAcceptedPresentationContext, 1071
FindContext, 1071
GetAcceptedPresentationContexts, 1071
GetConnectionInfo, 1071
GetMaxPDUSize, 1071
GetPresentationContextACByID, 1071
GetPresentationContextIDFromPresentationContext, 1072
GetPresentationContextRQByID, 1072
GetPresentationContexts, 1072
GetProtocol, 1072
GetState, 1072
GetTimer, 1072
InitializeConnection, 1072
InitializeIncomingConnection, 1072
SetMaxPDUSize, 1073
SetPresentationContexts, 1073
SetState, 1073
StopProtocol, 1073
ULActionAE6, 1073
ULConnection, 1071
ULConnectionManager, 1073
gdcm::network::ULConnectionCallback, 1074
~ULConnectionCallback, 1075
DataSetHandled, 1075
DataSetHandles, 1075
HandleDataSet, 1075
HandleResponse, 1075
mImplicit, 1076
ResetHandledDataSet, 1075
SetImplicitFlag, 1075
ULConnectionCallback, 1075
gdcm::network::ULConnectionInfo, 1076
GetCalledAETitle, 1077
GetCalledComputerName, 1077
GetCalledIPAddress, 1077
GetCalledIPPort, 1077
GetCallingAETitle, 1077
GetMaxPDULength, 1077
Initialize, 1077
SetMaxPDULength, 1077
ULConnectionInfo, 1077
gdcm::network::ULConnectionManager, 1078
~ULConnectionManager, 1080
BreakConnection, 1080
BreakConnectionNow, 1080
EstablishConnection, 1080
EstablishConnectionMove, 1081
mConnection, 1084
mSecondaryConnection, 1084
mTransitions, 1084
RunEventLoop, 1081
RunMoveEventLoop, 1081
SendEcho, 1081
SendFind, 1081, 1082
SendMove, 1082
SendNAction, 1082
SendNCreate, 1082
SendNDelete, 1083
SendNEventReport, 1083
SendNGet, 1083
SendNSet, 1083
SendStore, 1084
ULConnectionManager, 1080
gdcm::network::ULEvent, 1085
~ULEvent, 1085
GetDataSetPos, 1086
GetEvent, 1086
GetIStream, 1086
GetPDUs, 1086
SetEvent, 1086
SetPDU, 1086
ULEvent, 1085
gdcm::network::ULTransitionTable, 1086
HandleEvent, 1087
PrintTable, 1087
ULTransitionTable, 1087
gdcm::network::ULWritingCallback, 1088
~ULWritingCallback, 1089
HandleDataSet, 1089
HandleResponse, 1089
SetDirectory, 1089
ULWritingCallback, 1089
gdcm::network::UserInformation, 1099
~UserInformation, 1099
AddRoleSelectionSub, 1100
AddSOPClassExtendedNegociationSub, 1100
GetMaximumLengthSub, 1100

operator=, 1100
Print, 1100
Read, 1100
Size, 1100
UserInformation, 1099
Write, 1101
gdcm::static_assert_test< x >, 894
gdcm::terminal, 79
Attribute, 80
Color, 81
Mode, 81
setattribute, 81
setbgcolor, 81
setfgcolor, 82
setmode, 82
gdcmAAbortPDU.h, 1247
gdcmAAssociateACPDU.h, 1248
gdcmAAssociateRJPDU.h, 1248
gdcmAAssociateRQPDU.h, 1249
gdcmARTIMTimer.h, 1256
gdcmAReleaseRPPDU.h, 1254
gdcmAReleaseRQPDU.h, 1255
gdcmASN1.h, 1257
gdcmAbstractSyntax.h, 1250
gdcmAnonymizeEvent.h, 1251
gdcmAnonymizer.h, 1252
gdcmApplicationContext.h, 1253
gdcmApplicationEntity.h, 1254
gdcmAssertAlwaysMacro
gdcmTrace.h, 1480
gdcmAssertMacro
gdcmTrace.h, 1480
gdcmAsynchronousOperationsWindowSub.h, 1258
gdcmAttribute.h, 1258
gdcmAudioCodec.h, 1260
gdcmBase64.h, 1260
gdcmBaseCompositeMessage.h, 1261
gdcmBaseNormalizedMessage.h, 1262
gdcmBasePDU.h, 1263
gdcmBaseQuery.h, 1264
gdcmBaseRootQuery.h, 1265
gdcmBasicOffsetTable.h, 1266
gdcmBitmap.h, 1267
gdcmBitmapToBitmapFilter.h, 1268
gdcmBoxRegion.h, 1269
gdcmByteBuffer.h, 1269
gdcmByteSwap.h, 1271
gdcmByteSwapFilter.h, 1271
gdcmByteValue.h, 1272
gdcmCAPICryptoFactory.h, 1273
gdcmCAPICryptographicMessageSyntax.h, 1274
gdcmCEchoMessages.h, 1274
gdcmCFindMessages.h, 1275
gdcmCMoveMessages.h, 1276
gdcmCP246ExplicitDataElement.h, 1284
gdcmCSAElement.h, 1287
gdcmCSAHeader.h, 1288
gdcmCSAHeaderDict.h, 1289
gdcmCSAHeaderDictEntry.h, 1290
gdcmCStoreMessages.h, 1291
gdcmCodeString.h, 1280
gdcmCodec.h, 1277
gdcmCoder.h, 1278
gdcmCommand.h, 1281
gdcmCommandDataSet.h, 1282
gdcmCompositeMessageFactory.h, 1283
gdcmCompositeNetworkFunctions.h, 1283
gdcmConstCharWrapper.h, 1284
gdcmCryptoFactory.h, 1285
gdcmCryptographicMessageSyntax.h, 1286
gdcmCurve.h, 1292
gdcmDICOMDIR.h, 1302
gdcmDICOMDIRGenerator.h, 1303
gdcmDIMSE.h, 1309
gdcmDataElement.h, 1293
gdcmDataEvent.h, 1295
gdcmDataSet.h, 1296
gdcmDataSetEvent.h, 1297
gdcmDataSetHelper.h, 1297
gdcmDebugMacro
gdcmTrace.h, 1481
gdcmDecoder.h, 1298
gdcmDefinedTerms.h, 1300
gdcmDeflateStream.h, 1300
gdcmDefs.h, 1301
gdcmDeltaEncodingCodec.h, 1302
gdcmDict.h, 1304
gdcmDictConverter.h, 1305
gdcmDictEntry.h, 1306
gdcmDictPrinter.h, 1307
gdcmDicts.h, 1308
gdcmDirectionCosines.h, 1309
gdcmDirectory.h, 1310
gdcmDirectoryHelper.h, 1311
gdcmDummyValueGenerator.h, 1312
gdcmDumper.h, 1312
gdcmElement.h, 1313
VRDS16ILLEGAL, 1315
gdcmEncapsulatedDocument.h, 1315
gdcmEnumeratedValues.h, 1316
gdcmErrorMacro
gdcmTrace.h, 1481
gdcmEvent.h, 1316
gdcmEventMacro, 1318
gdcmEventMacro
gdcmEvent.h, 1318
gdcmException.h, 1318
gdcmExplicitDataElement.h, 1319

gdcmExplicitImplicitDataElement.h, 1320
gdcmFiducials.h, 1320
gdcmFile.h, 1321
gdcmFileAnonymizer.h, 1322
gdcmFileChangeTransferSyntax.h, 1323
gdcmFileDecompressLookupTable.h, 1323
gdcmFileDerivation.h, 1324
gdcmFileExplicitFilter.h, 1325
gdcmFileMetaInformation.h, 1326
gdcmFileNameEvent.h, 1327
gdcmFileSet.h, 1329
gdcmFileStreamer.h, 1330
gdcmFilename.h, 1327
gdcmFilenameGenerator.h, 1328
gdcmFindPatientRootQuery.h, 1331
gdcmFindStudyRootQuery.h, 1332
gdcmFragment.h, 1332
gdcmGlobal.h, 1334
gdcmGroupDict.h, 1335
gdcmIOD.h, 1352
gdcmIODEntry.h, 1353
gdcmIODs.h, 1355
gdcmIPPSorter.h, 1356
gdcmIconImage.h, 1335
gdcmIconImageFilter.h, 1336
gdcmIconImageGenerator.h, 1337
gdcmImage.h, 1338
gdcmImageApplyLookupTable.h, 1339
gdcmImageChangePhotometricInterpretation.h, 1340
gdcmImageChangePlanarConfiguration.h, 1340
gdcmImageChangeTransferSyntax.h, 1341
gdcmImageCodec.h, 1342
gdcmImageConverter.h, 1343
gdcmImageFragmentSplitter.h, 1343
gdcmImageHelper.h, 1344
gdcmImageReader.h, 1345
gdcmImageRegionReader.h, 1346
gdcmImageToImageFilter.h, 1347
gdcmImageWriter.h, 1348
gdcmImplementationClassUIDSub.h, 1348
gdcmImplementationUIDSub.h, 1349
gdcmImplementationVersionNameSub.h, 1350
gdcmImplicitDataElement.h, 1351
gdcmItem.h, 1357
gdcmJPEG12Codec.h, 1359
gdcmJPEG16Codec.h, 1359
gdcmJPEG2000Codec.h, 1360
gdcmJPEG8Codec.h, 1361
gdcmJPEGCodec.h, 1362
gdcmJPEGLSCodec.h, 1364
gdcmJSON.h, 1364
gdcmKAKADUCodec.h, 1365
gdcmLO.h, 1367
gdcmLegacyMacro.h, 1366
GDCM_LEGACY_BODY, 1367
GDCM_LEGACY_REPLACED_BODY, 1367
GDCM_LEGACY, 1367
gdcmLookupTable.h, 1368
gdcmMD5.h, 1375
gdcmMacro.h, 1369
gdcmMacroEntry.h, 1371
GDCMMACROENTRY_H, 1372
gdcmMacros.h, 1372
gdcmMaximumLengthSub.h, 1374
gdcmMediaStorage.h, 1376
gdcmMeshPrimitive.h, 1377
gdcmModalityPerformedProcedureStepCreateQuery.h, 1378
gdcmModalityPerformedProcedureStepSetQuery.h, 1379
gdcmModule.h, 1379
gdcmModuleEntry.h, 1381
gdcmModules.h, 1383
gdcmMovePatientRootQuery.h, 1384
gdcmMoveStudyRootQuery.h, 1385
gdcmNActionMessages.h, 1385
gdcmNCreateMessages.h, 1386
gdcmNDeleteMessages.h, 1387
gdcmNEventReportMessages.h, 1391
gdcmNGetMessages.h, 1391
gdcmNSetMessages.h, 1393
gdcmNestedModuleEntries.h, 1387
gdcmNetworkEvents.h, 1389
gdcmNetworkStateID.h, 1390
gdcmNormalizedMessageFactory.h, 1392
gdcmNormalizedNetworkFunctions.h, 1393
gdcmObject.h, 1394
gdcmOpenSSLCryptoFactory.h, 1395
gdcmOpenSSLCryptographicMessageSyntax.h, 1396
gdcmOpenSSLP7CryptoFactory.h, 1397
gdcmOpenSSLP7CryptographicMessageSyntax.h, 1397
gdcmOrientation.h, 1399
gdcmOverlay.h, 1399
gdcmPDBElement.h, 1404
gdcmPDBHeader.h, 1405
gdcmPDFCodec.h, 1406
gdcmPDUFactory.h, 1407
gdcmPDataTFPDU.h, 1403
gdcmPGXCodec.h, 1408
gdcmPNMCodec.h, 1415
gdcmPVRGCodec.h, 1425
gdcmParseException.h, 1400
gdcmParser.h, 1402
gdcmPatient.h, 1402
gdcmPersonName.h, 1407
gdcmPhotometricInterpretation.h, 1409
gdcmPixelFormat.h, 1410
gdcmPixmap.h, 1411
gdcmPixmapReader.h, 1412

gdcmPixmapToPixmapFilter.h, 1413
gdcmPixmapWriter.h, 1413
gdcmPreamble.h, 1415
gdcmPresentationContext.h, 1417
gdcmPresentationContextAC.h, 1418
gdcmPresentationContextGenerator.h, 1419
gdcmPresentationContextRQ.h, 1419
gdcmPresentationDataValue.h, 1420
gdcmPrinter.h, 1421
gdcmPrivateTag.h, 1423
gdcmProgressEvent.h, 1424
gdcmPythonFilter.h, 1425
gdcmQueryBase.h, 1426
gdcmQueryFactory.h, 1427
gdcmQueryImage.h, 1428
gdcmQueryPatient.h, 1429
gdcmQuerySeries.h, 1430
gdcmQueryStudy.h, 1431
gdcmRAWCodec.h, 1432
gdcmRLECodec.h, 1436
gdcmReader.h, 1432
gdcmRegion.h, 1434
gdcmRescaler.h, 1435
gdcmRoleSelectionSub.h, 1436
gdcmSHA1.h, 1448
gdcmSOPClassExtendedNegociationSub.h, 1451
gdcmSOPClassUIDToIOD.h, 1452
gdcmScanner.h, 1437
gdcmSegment.h, 1438
gdcmSegmentHelper.h, 1440
gdcmSegmentReader.h, 1441
gdcmSegmentWriter.h, 1442
gdcmSegmentedPaletteColorLookupTable.h, 1439
gdcmSequenceOfFragments.h, 1443
gdcmSequenceOfItems.h, 1444
gdcmSerieHelper.h, 1444
gdcmSeries.h, 1446
gdcmServiceClassApplicationInformation.h, 1447
gdcmServiceClassUser.h, 1448
gdcmSimpleSubjectWatcher.h, 1449
gdcmSmartPointer.h, 1450
gdcmSorter.h, 1453
gdcmSpacing.h, 1455
gdcmSpectroscopy.h, 1455
gdcmSplitMosaicFilter.h, 1456
gdcmStaticAssert.h, 1456
  GDCM_DO_JOIN2, 1457
  GDCM_DO_JOIN, 1457
  GDCM_JOIN, 1457
  GDCM_STATIC_ASSERT, 1457
gdcmStreamImageReader.h, 1458
gdcmStreamImageWriter.h, 1459
gdcmStrictScanner.h, 1459
gdcmString.h, 1460
gdcmStringFilter.h, 1462
gdcmStudy.h, 1462
gdcmSubject.h, 1463
gdcmSurface.h, 1464
gdcmSurfaceHelper.h, 1465
gdcmSurfaceReader.h, 1466
gdcmSurfaceWriter.h, 1467
gdcmSwapCode.h, 1468
gdcmSwapper.h, 1469
gdcmSystem.h, 1469
gdcmTable.h, 1470
gdcmTableEntry.h, 1471
gdcmTableReader.h, 1473
gdcmTag.h, 1474
gdcmTagPath.h, 1475
gdcmTagToVR.h, 1475
gdcmTerminal.h, 1476
gdcmTestDriver.h, 1477
gdcmTesting.h, 1478
gdcmTrace.h, 1479
  GDCM_FUNCTION, 1480
  gdcmAssertAlwaysMacro, 1480
  gdcmAssertMacro, 1480
  gdcmDebugMacro, 1481
  gdcmErrorMacro, 1481
  gdcmWarningMacro, 1483
gdcmTransferSyntax.h, 1483
gdcmTransferSyntaxSub.h, 1485
gdcmType.h, 1486
gdcmTypes.h, 1487
gdcmUIDGenerator.h, 1487
gdcmUIDs.h, 1488
gdcmULAction.h, 1489
gdcmULActionAA.h, 1490
gdcmULActionAE.h, 1491
gdcmULActionAR.h, 1491
gdcmULActionDT.h, 1492
gdcmULBasicCallback.h, 1493
gdcmULConnection.h, 1493
gdcmULConnectionCallback.h, 1494
gdcmULConnectionInfo.h, 1495
gdcmULConnectionManager.h, 1497
gdcmULEvent.h, 1497
gdcmULTransitionTable.h, 1499
gdcmULWritingCallback.h, 1500
gdcmUNExplicitDataElement.h, 1500
gdcmUNExplicitImplicitDataElement.h, 1501
gdcmUUIDGenerator.h, 1505
gdcmUnpacker12Bits.h, 1502
gdcmUsage.h, 1502
gdcmUserInformation.h, 1504
gdcmVL.h, 1508
gdcmVM.h, 1509
  TYPETOLENGTH, 1510

gdcmVR.h, 1511
TYPETOENCODING, 1512
VRTypeTemplateCase, 1512
gdcmVR16ExplicitDataElement.h, 1513
gdcmValidate.h, 1505
gdcmValue.h, 1506
gdcmValueIO.h, 1507
gdcmVersion.h, 1507
gdcmWLMFindQuery.h, 1515
gdcmWarningMacro
gdcmTrace.h, 1483
gdcmWaveform.h, 1514
gdcmWin32.h, 1514
GDCM_EXPORT, 1515
gdcmWriter.h, 1516
gdcmXMLDictReader.h, 1517
gdcmXMLPrinter.h, 1517
gdcmXMLPrivateDictReader.h, 1518
Generate
gdcm::DICOMDIRGenerator, 319
gdcm::DummyValueGenerator, 349
gdcm::FilenameGenerator, 427
gdcm::IconImageGenerator, 457
gdcm::UIDGenerator, 1002
gdcm::UUIDGenerator, 1101
GenerateFromFilenames
gdcm::PresentationContextGenerator, 749
GenerateFromUID
gdcm::PresentationContextGenerator, 749
GenerateUUID
gdcm::UIDGenerator, 1002
Get
gdcm::ByteBuffer, 198
GetAETitle
gdcm::ServiceClassUser, 863
GetALGOType
gdcm::Segment, 825
GetALGOTypeString
gdcm::Segment, 825
GetAbbreviation
gdcm::GroupDict, 451
GetAbstractSyntax
gdcm::PresentationContext, 743
gdcm::network::PresentationContextRQ, 751, 752
GetAbstractSyntaxUID
gdcm::BaseQuery, 166
gdcm::FindPatientRootQuery, 440
gdcm::FindStudyRootQuery, 443
gdcm::ModalityPerformedProcedureStepCreate↩
Query, 614
gdcm::ModalityPerformedProcedureStepSetQuery, 616
gdcm::MovePatientRootQuery, 629
gdcm::MoveStudyRootQuery, 632
gdcm::WLMFindQuery, 1230
GetAcceptedPresentationContexts
gdcm::network::ULConnection, 1071
GetAlgorithmFamily
gdcm::Surface, 930
GetAlgorithmName
gdcm::Surface, 930
GetAlgorithmVersion
gdcm::Surface, 930
GetAllFilenamesFromTagToValue
gdcm::Scanner, 819
gdcm::StrictScanner, 910
GetAllRequiredTags
gdcm::QueryBase, 776
GetAllTags
gdcm::QueryBase, 776
GetAnatomicRegion
gdcm::Segment, 825
GetAsDataElement
gdcm::Attribute, 128
gdcm::Attribute< Group, Element, TVR, VM::VM1 >, 135
gdcm::Attribute< Group, Element, TVR, VM::VM1_n >, 143
gdcm::Element, 353
gdcm::Element< TVR, VM::VM1_n >, 359
gdcm::PrivateTag, 765
gdcm::network::AbstractSyntax, 99
GetAsPoints
gdcm::Curve, 274
GetAsString
gdcm::CodeString, 231
GetAxisOfRotation
gdcm::Surface, 930
GetBasicApplicationLevelConfidentialityProfileAttributes
gdcm::Anonymizer, 108
GetBitPosition
gdcm::Overlay, 677
GetBitSample
gdcm::LookupTable, 581
GetBitsAllocated
gdcm::Overlay, 677
gdcm::PixelFormat, 715
GetBitsStored
gdcm::PixelFormat, 715
GetBlob
gdcm::network::PresentationDataValue, 755
GetBuffer
gdcm::Bitmap, 181
gdcm::ByteValue, 205
gdcm::Parser, 686
gdcm::SequenceOfFragments, 842
GetBuffer2
gdcm::Bitmap, 181

GetBufferAsRGBA
gdcm::LookupTable, 582
GetBufferLength
gdcm::Bitmap, 182
gdcm::JPEGLSCodec, 568
gdcm::PNMCodec, 736
gdcm::RLECodec, 809
GetBuildVersion
gdcm::Version, 1108
GetByteValue
gdcm::CSAElement, 253
gdcm::DataElement, 281
GetCSADataInfo
gdcm::CSAHeader, 260
GetCSAEEnd
gdcm::CSAHeader, 260
GetCSAElementByName
gdcm::CSAHeader, 261
GetCSAHeaderDict
gdcm::Dicts, 336
GetCSAHeaderDictEntry
gdcm::CSAHeaderDict, 265
GetCSAImageHeaderInfoTag
gdcm::CSAHeader, 261
GetCSASeriesHeaderInfoTag
gdcm::CSAHeader, 261
GetCTImageSeriesUIDs
gdcm::DirectoryHelper, 347
GetCWD
gdcm::System, 954
GetCalledAETitle
gdcm::ServiceClassUser, 863
gdcm::network::AAssociateRQPDU, 94
gdcm::network::ULConnectionInfo, 1077
GetCalledComputerName
gdcm::network::ULConnectionInfo, 1077
GetCalledIPAddress
gdcm::network::ULConnectionInfo, 1077
GetCalledIPPort
gdcm::network::ULConnectionInfo, 1077
GetCallingAETitle
gdcm::network::AAssociateRQPDU, 94
gdcm::network::ULConnectionInfo, 1077
GetCenterOfRotation
gdcm::Surface, 931
GetCharacterFromCurrentLocale
gdcm::QueryFactory, 778
GetCheckFileMetaInformation
gdcm::Writer, 1234
GetCipherType
gdcm::CAPICryptographicMessageSyntax, 212
gdcm::CryptographicMessageSyntax, 249
gdcm::OpenSSLCryptographicMessageSyntax, 664
gdcm::OpenSSLP7CryptographicMessageSyntax, 669
GetCodec
gdcm::FileChangeTransferSyntax, 401
GetColorLevel
vtkImageColorViewer, 1191
GetColorWindow
vtkImageColorViewer, 1191
GetColumns
gdcm::Bitmap, 182
gdcm::Overlay, 677
GetCommand
gdcm::Subject, 925
GetConnectionInfo
gdcm::network::ULConnection, 1071
GetConstructorString
gdcm::Dicts, 336
GetContourReferencedFrameOfReferenceClassUID
vtkRTStructSetProperties, 1221
GetContourReferencedFrameOfReferenceInstanceUID
vtkRTStructSetProperties, 1221
GetCryptographicMessageSyntax
gdcm::Anonymizer, 108
GetCurrentByteIndex
gdcm::Parser, 686
GetCurrentDateTime
gdcm::System, 954
GetCurrentModuleFileName
gdcm::System, 954
GetCurrentProcessFileName
gdcm::System, 954
GetCurrentResourcesDirectory
gdcm::System, 954
GetCurve
gdcm::Pixmap, 722
GetCurveDataDescriptor
gdcm::Curve, 275
GetDEEnd
gdcm::DataSet, 299
GetDES
gdcm::DataSet, 299
GetData
gdcm::DataEvent, 293
GetDataElement
gdcm::Bitmap, 182
gdcm::DataSet, 298, 299
gdcm::Item, 539
GetDataExtraRoot
gdcm::Testing, 979
GetDataLength
gdcm::DataEvent, 293
GetDataRoot
gdcm::Testing, 979
GetDataSet

gdcm::CSAHeader, 261
gdcm::DataSetEvent, 307
gdcm::File, 393
GetDataSetPos
gdcm::network::ULEvent, 1086
GetDataSetTransferSyntax
gdcm::FileMetaInformation, 414
GetDataSets
gdcm::network::ULBasicCallback, 1069
GetDataValueRepresentation
gdcm::Curve, 275
GetDebugFlag
gdcm::Trace, 985
GetDebugStream
gdcm::Trace, 986
GetDecodeLength
gdcm::Base64, 156
GetDefaultTransferSyntax
gdcm::PresentationContextGenerator, 749
GetDefs
gdcm::Global, 448
gdcm::TableReader, 962
GetDescription
gdcm::CSAHeaderDictEntry, 267
gdcm::Exception, 384
gdcm::ModuleEntry, 623
gdcm::Overlay, 677
GetDescriptiveName
vtkGDCMImageReader, 1133
vtkGDCMImageReader2, 1145
vtkGDCMImageWriter, 1156
GetDict
gdcm::XMLDictReader, 1239
GetDictEntry
gdcm::Dict, 323
gdcm::Dicts, 337
gdcm::PrivateDict, 762
GetDictEntryByKeyword
gdcm::Dict, 323
GetDictEntryByName
gdcm::Dict, 323
GetDictName
gdcm::DictConverter, 326
GetDictVM
gdcm::Attribute, 128
gdcm::Attribute< Group, Element, TVR, VM::VM1 >, 135
gdcm::Attribute< Group, Element, TVR, VM::VM1_n >, 143
GetDictVR
gdcm::Attribute, 128
gdcm::Attribute< Group, Element, TVR, VM::VM1 >, 135
gdcm::Attribute< Group, Element, TVR, VM::VM1_n >, 144
GetDicts
gdcm::Global, 448, 449
GetDimension
gdcm::Bitmap, 182
GetDimensions
gdcm::Bitmap, 182
gdcm::Curve, 275
gdcm::ImageCodec, 487
GetDimensionsValue
gdcm::ImageHelper, 498
GetDimensionsValueForResolution
gdcm::StreamImageReader, 897
GetDirectionCosines
gdcm::Image, 462
GetDirectionCosinesFromDataSet
gdcm::ImageHelper, 499
GetDirectionCosinesTolerance
gdcm::IPPSorter, 534
GetDirectionCosinesValue
gdcm::ImageHelper, 499
GetDirectories
gdcm::Directory, 345
GetElapsedTime
gdcm::network::ARTIMTimer, 121
GetElement
gdcm::Tag, 968
GetElementTag
gdcm::Tag, 968
GetEncodeLength
gdcm::Base64, 156
GetErrorCode
gdcm::Parser, 686
GetErrorFlag
gdcm::Trace, 986
GetErrorStream
gdcm::Trace, 986
GetErrorString
gdcm::Parser, 686
GetEvent
gdcm::network::ULEvent, 1086
GetEventName
gdcm::AnonymizeEvent, 103
gdcm::DataEvent, 293
gdcm::DataSetEvent, 307
gdcm::Event, 381
gdcm::FileNameEvent, 424
gdcm::ProgressEvent, 769
GetExtension
gdcm::Filename, 420
GetFactoryInstance
gdcm::CryptoFactory, 247
GetFile

gdcm::Anonymizer, 108
gdcm::DICOMDIRGenerator, 320
gdcm::FileDecompressLookupTable, 403
gdcm::FileDerivation, 406
gdcm::FileExplicitFilter, 409
gdcm::IconImageFilter, 454
gdcm::PythonFilter, 773
gdcm::Reader, 794
gdcm::SplitMosaicFilter, 892
gdcm::StreamImageReader, 897
gdcm::StringFilter, 920, 921
gdcm::Writer, 1234
vtkGDCMMedicalImageProperties, 1163
GetFileExtensions
vtkGDCMImageReader, 1133
vtkGDCMImageReader2, 1145
vtkGDCMImageWriter, 1156
GetFileMetaInformationVersion
gdcm::FileMetaInformation, 414
GetFileName
gdcm::FileNameEvent, 424
gdcm::Filename, 420
gdcm::Testing, 979
vtkGDCMImageWriter, 1156
vtkGDCMThreadedImageReader2, 1182
GetFileNames
gdcm::Testing, 979
GetFilename
gdcm::FilenameGenerator, 427
gdcm::TableReader, 962
GetFilenameFromTagToValue
gdcm::Scanner, 819
gdcm::StrictScanner, 910
GetFilenames
gdcm::Directory, 345
gdcm::FilenameGenerator, 427
gdcm::Scanner, 819
gdcm::Sorter, 885
gdcm::StrictScanner, 911
GetFilenamesFromSeriesUIDs
gdcm::DirectoryHelper, 347
GetFiles
gdcm::FileSet, 430
GetFiniteVolume
gdcm::Surface, 931
GetFirstSingleSerieUIDFileSet
gdcm::SerieHelper, 857
GetForcePixelSpacing
gdcm::ImageHelper, 499
GetForceRescaleInterceptSlope
gdcm::ImageHelper, 499
GetFormat
gdcm::CSAHeader, 261
GetFragBuffer
gdcm::SequenceOfFragments, 842
GetFragment
gdcm::SequenceOfFragments, 843
GetFragmentSizeMax
gdcm::ImageFragmentSplitter, 496
GetFrameOfReference
gdcm::DirectoryHelper, 347
GetFullLength
gdcm::FileMetaInformation, 414
GetGDCMDataRoot
vtkGDCMTesting, 1175
GetGDCMImplementationClassUID
gdcm::FileMetaInformation, 415
GetGDCMImplementationVersionName
gdcm::FileMetaInformation, 415
GetGDCMSourceApplicationEntityTitle
gdcm::FileMetaInformation, 415
GetGDCMUID
gdcm::UIDGenerator, 1002
GetGroup
gdcm::Curve, 275
gdcm::Overlay, 677
gdcm::Tag, 968
GetHasExpired
gdcm::network::ARTIMTimer, 121
GetHeader
gdcm::File, 393
GetHeaderInfo
gdcm::ImageCodec, 487
gdcm::JPEG12Codec, 544
gdcm::JPEG16Codec, 547
gdcm::JPEG2000Codec, 551
gdcm::JPEG8Codec, 556
gdcm::JPEGCodec, 561
gdcm::JPEGLSCodec, 569
gdcm::PGXCodec, 707
gdcm::PNMCodec, 736
gdcm::RAWCodec, 791
gdcm::RLECodec, 810
GetHierachicalSearchTags
gdcm::QueryBase, 776
gdcm::QueryImage, 780
gdcm::QueryPatient, 782
gdcm::QuerySeries, 784
gdcm::QueryStudy, 787
GetHighBit
gdcm::PixelFormat, 715
GetHostName
gdcm::System, 955
GetIODEntry
gdcm::IOD, 525
GetIODFromFile
gdcm::Defs, 312
GetIODFromSOPClassUID

gdcm::SOPClassUIDToIOD, 882
GetIODNameFromMediaStorage
gdcm::Defs, 312
GetIODs
gdcm::Defs, 312
GetIOD
gdcm::IODs, 531
gdcm::SOPClassUIDToIOD, 882
GetIStream
gdcm::network::ULEvent, 1086
GetIconImage
gdcm::IconImageFilter, 454
gdcm::IconImageGenerator, 457
gdcm::Pixmap, 722
vtkGDCMImageReader, 1133
vtkGDCMImageReader2, 1145
GetIconImagePort
vtkGDCMImageReader2, 1145
GetIE
gdcm::IODEntry, 527
GetImage
gdcm::ImageReader, 505
gdcm::ImageWriter, 515
gdcm::PixmapWriter, 732, 733
gdcm::SplitMosaicFilter, 892
GetImplementationClassUID
gdcm::FileMetaInformation, 415
GetImplementationVersionName
gdcm::FileMetaInformation, 415
GetIndex
gdcm::SwapCode, 949
gdcm::VM, 1116
GetInitialized
gdcm::CAPICryptographicMessageSyntax, 212
GetInput
gdcm::ImageToImageFilter, 512
gdcm::PixmapToPixmapFilter, 729
vtkImageColorViewer, 1191
GetInputFilename
gdcm::DictConverter, 327
GetInstance
gdcm::Global, 449
GetIntercept
gdcm::Image, 462
gdcm::Rescaler, 803
GetInterfile
gdcm::CSAHeader, 262
GetInternal
gdcm::Preamble, 739
GetIsCommand
gdcm::network::PresentationDataValue, 755
GetIsLastFragment
gdcm::network::PresentationDataValue, 755
GetItem
gdcm::SequenceOfItems, 850
GetKey
gdcm::CSAElement, 253
GetKeys
gdcm::Scanner, 819
gdcm::StrictScanner, 911
GetKeyword
gdcm::DictEntry, 330
GetKeywordFromTag
gdcm::Dict, 323
GetLUTDescriptor
gdcm::LookupTable, 582
GetLUTLength
gdcm::LookupTable, 582
GetLUT
gdcm::Bitmap, 183
gdcm::ImageCodec, 488
gdcm::ImageHelper, 499
gdcm::LookupTable, 582
GetLabel
gdcm::Orientation, 672
GetLastElement
gdcm::ParseException, 684
GetLastSystemError
gdcm::System, 955
GetLength
gdcm::ByteValue, 205
gdcm::CP246ExplicitDataElement, 244
gdcm::DataElement, 282
gdcm::DataSet, 300
gdcm::Element, 353
gdcm::Element< TVR, VM::VM1_n >, 359
gdcm::Element< VR::AS, VM::VM5 >, 369
gdcm::ExplicitDataElement, 387
gdcm::ExplicitImplicitDataElement, 389
gdcm::Fragment, 445
gdcm::ImplicitDataElement, 521
gdcm::Item, 539
gdcm::Preamble, 739
gdcm::SequenceOfFragments, 843
gdcm::SequenceOfItems, 850
gdcm::Tag, 968
gdcm::UNExplicitDataElement, 1091
gdcm::UNExplicitImplicitDataElement, 1094
gdcm::VR16ExplicitDataElement, 1126
gdcm::Value, 1105
gdcm::VL, 1111
gdcm::VM, 1116
gdcm::VR, 1121
GetLocaleCharset
gdcm::System, 955
GetLossless
gdcm::JPEGCodec, 562
gdcm::JPEGLSCodec, 569

GetLossyFlag
gdcm::ImageCodec, 487
GetLossyFlagFromFile
gdcm::Testing, 980
GetMD5DataImage
gdcm::Testing, 980
GetMD5DataImages
gdcm::Testing, 980
GetMD5FromBrokenFile
gdcm::Testing, 980
GetMD5FromFile
gdcm::Testing, 980
GetMD5MetaImage
vtkGDCMTesting, 1175
GetMHDMD5FromFile
vtkGDCMTesting, 1175
GetMPType
gdcm::MeshPrimitive, 610
GetMPTypeString
gdcm::MeshPrimitive, 610
GetMRImageSeriesUIDs
gdcm::DirectoryHelper, 348
GetMSString
gdcm::MediaStorage, 600
GetMSType
gdcm::MediaStorage, 600
GetMTime
vtkImageMapToColors16, 1201
GetMacro
gdcm::Macros, 590
GetMacroEntry
gdcm::Macro, 588
GetMacros
gdcm::Defs, 312, 313
GetMajorAxisFromPatientRelativeDirectionCosine
gdcm::Orientation, 672
GetMajorVersion
gdcm::Version, 1108
GetManifold
gdcm::Surface, 931
GetMapping
gdcm::Scanner, 819
gdcm::StrictScanner, 911
GetMappingFromTagToValue
gdcm::Scanner, 820
gdcm::StrictScanner, 911
GetMappings
gdcm::Scanner, 820
gdcm::StrictScanner, 911
GetMax
gdcm::PixelFormat, 715
GetMaxLength
gdcm::PersonName, 703
GetMaxPDULength
gdcm::network::ULConnectionInfo, 1077
GetMaxPDUSize
gdcm::network::ULConnection, 1071
GetMaximumLength
gdcm::network::MaximumLengthSub, 591
GetMaximumLengthSub
gdcm::network::UserInformation, 1100
GetMaximumPointDistance
gdcm::Surface, 931
GetMeanPointDistance
gdcm::Surface, 931
GetMediaStorage
gdcm::DataSet, 300
gdcm::FileMetaInformation, 415
GetMediaStorageAsString
gdcm::FileMetaInformation, 415
GetMediaStorageDataFile
gdcm::Testing, 980
GetMediaStorageDataFiles
gdcm::Testing, 980
GetMediaStorageFromFile
gdcm::Testing, 981
GetMeshPrimitive
gdcm::Surface, 931
GetMessageHeader
gdcm::network::PresentationDataValue, 755
GetMetaInformationTS
gdcm::FileMetaInformation, 415
GetMin
gdcm::PixelFormat, 716
GetMinorVersion
gdcm::Version, 1108
GetModality
gdcm::MediaStorage, 600
GetModalityDimension
gdcm::MediaStorage, 600
GetModule
gdcm::Modules, 626
GetModuleEntry
gdcm::NestedModuleEntries, 643
GetModuleEntryInMacros
gdcm::Module, 620
GetModules
gdcm::Defs, 313
GetName
gdcm::CSAElement, 253
gdcm::CSAHeaderDictEntry, 267
gdcm::DictEntry, 330
gdcm::Filename, 420
gdcm::GroupDict, 451
gdcm::IODEntry, 527
gdcm::Macro, 588
gdcm::Module, 620
gdcm::ModuleEntry, 623

gdcm::PDBElement, 692
gdcm::QueryBase, 776
gdcm::QueryImage, 780
gdcm::QueryPatient, 782
gdcm::QuerySeries, 784
gdcm::QueryStudy, 787
gdcm::UIDs, 1027
gdcm::network::AbstractSyntax, 99
gdcm::network::ApplicationContext, 113
gdcm::network::TransferSyntaxSub, 995
GetNeedByteSwap
gdcm::Bitmap, 183
gdcm::ImageCodec, 488
GetNegociatedType
gdcm::TransferSyntax, 992
GetNestedDataSet
gdcm::Item, 539
GetNextSingleSerieUIDFileSet
gdcm::SerieHelper, 857
GetNoOfItems
gdcm::CSAElement, 253
GetNumberOfComponents
gdcm::PersonName, 703
GetNumberOfContourReferencedFrameOfReferences
vtkRTStructSetProperties, 1221
GetNumberOfCurves
gdcm::Curve, 275
gdcm::Pixmap, 722
GetNumberOfDimensions
gdcm::Bitmap, 183
gdcm::ImageCodec, 488
GetNumberOfElementsFromArray
gdcm::VM, 1116
GetNumberOfFileNames
gdcm::Testing, 981
GetNumberOfFilenames
gdcm::FilenameGenerator, 427
GetNumberOfFragments
gdcm::SequenceOfFragments, 843
GetNumberOfIODs
gdcm::IOD, 525
GetNumberOfIconImages
gdcm::IconImageFilter, 455
GetNumberOfItems
gdcm::SequenceOfItems, 851
GetNumberOfMD5DataImages
gdcm::Testing, 981
GetNumberOfMD5MetaImages
vtkGDCMTesting, 1175
GetNumberOfMSString
gdcm::MediaStorage, 601
GetNumberOfMSType
gdcm::MediaStorage, 601
GetNumberOfMediaStorageDataFiles
gdcm::Testing, 981
GetNumberOfModality
gdcm::MediaStorage, 600
GetNumberOfModuleEntries
gdcm::NestedModuleEntries, 644
GetNumberOfOverlays
gdcm::Pixmap, 722
GetNumberOfPoints
gdcm::Curve, 275
GetNumberOfPresentationContext
gdcm::network::AAssociateRQPDU, 94
GetNumberOfPresentationContextAC
gdcm::network::AAssociateACPDU, 87
GetNumberOfPresentationDataValues
gdcm::network::PDataTFPDU, 690
GetNumberOfPrimitivesData
gdcm::MeshPrimitive, 610
GetNumberOfReferencedFrameOfReferences
vtkRTStructSetProperties, 1222
GetNumberOfSOPClassToIOD
gdcm::SOPClassUIDToIOD, 882
GetNumberOfSegments
gdcm::SegmentWriter, 837
GetNumberOfStructureSetROIs
vtkRTStructSetProperties, 1222
GetNumberOfSurfacePoints
gdcm::Surface, 931
GetNumberOfSurfaces
gdcm::SurfaceReader, 943
gdcm::SurfaceWriter, 946
GetNumberOfTransferSyntaxStrings
gdcm::UIDs, 1027
GetNumberOfTransferSyntaxes
gdcm::PresentationContext, 743
gdcm::network::PresentationContextRQ, 752
GetNumberOfValues
gdcm::Attribute, 128
gdcm::Attribute< Group, Element, TVR, VM::VM1 >, 135
gdcm::Attribute< Group, Element, TVR, VM::VM1_n >, 144
GetNumberOfVectors
gdcm::Surface, 932
GetObliquityThresholdCosineValue
gdcm::Orientation, 672
GetOffScreenRendering
vtkImageColorViewer, 1191
GetOptionalTags
gdcm::QueryBase, 776
gdcm::QueryImage, 780
gdcm::QueryPatient, 782
gdcm::QuerySeries, 784
gdcm::QueryStudy, 787
GetOrderedValues

gdcm::Scanner, 820
gdcm::StrictScanner, 911
GetOrigin
gdcm::Image, 462, 463
gdcm::Overlay, 677
GetOriginValue
gdcm::ImageHelper, 499
GetOuput
gdcm::ImageConverter, 494
GetOutput
gdcm::BitmapToBitmapFilter, 192
gdcm::ImageToImageFilter, 512
gdcm::PixmapToPixmapFilter, 729
GetOutputAsBitmap
gdcm::BitmapToBitmapFilter, 192
GetOutputAsPixmap
gdcm::PixmapToPixmapFilter, 729
GetOutputFilename
gdcm::DictConverter, 327
GetOutputType
gdcm::DictConverter, 327
GetOverlay
gdcm::Pixmap, 723
vtkGDCMImageReader, 1133
vtkGDCMImageReader2, 1145
GetOverlayData
gdcm::Overlay, 677
GetOverlayPort
vtkGDCMImageReader2, 1145
GetOverlayTypeAsString
gdcm::Overlay, 678
GetOverlayTypeFromString
gdcm::Overlay, 678
GetOverlayVisibility
vtkImageColorViewer, 1191
GetOwner
gdcm::PrivateTag, 765
GetPDBEEnd
gdcm::PDBHeader, 695
GetPDBElementByName
gdcm::PDBHeader, 695
GetPDBInfoTag
gdcm::PDBHeader, 696
GetPDUs
gdcm::network::ULEvent, 1086
GetPDVs
gdcm::network::PDUFactory, 702
GetPIString
gdcm::PhotometricInterpretation, 710
GetPIType
gdcm::PhotometricInterpretation, 710
GetPMSRescaleInterceptSlope
gdcm::ImageHelper, 500
GetPath
gdcm::Filename, 420
GetPattern
gdcm::FilenameGenerator, 428
GetPermissions
gdcm::System, 955
GetPhotometricInterpretation
gdcm::Bitmap, 183
gdcm::ImageChangePhotometricInterpretation, 471
gdcm::ImageCodec, 488
GetPhotometricInterpretationValue
gdcm::ImageHelper, 500
GetPixelFormat
gdcm::Bitmap, 183, 184
gdcm::ImageCodec, 488
GetPixelFormatValue
gdcm::ImageHelper, 500
GetPixelRepresentation
gdcm::PixelFormat, 716
GetPixelSize
gdcm::PixelFormat, 716
GetPixelSpacingDataRoot
gdcm::Testing, 981
GetPixmap
gdcm::FileDecompressLookupTable, 403, 404
gdcm::IconImageGenerator, 457, 458
gdcm::PixmapReader, 726
gdcm::PixmapWriter, 733
GetPlanarConfiguration
gdcm::Bitmap, 184
gdcm::ImageChangePlanarConfiguration, 475
gdcm::ImageCodec, 488
GetPlanarConfigurationValue
gdcm::ImageHelper, 500
GetPointCoordinatesData
gdcm::Surface, 932
GetPointPositionAccuracy
gdcm::Surface, 932
GetPointer
gdcm::ByteValue, 205
gdcm::LookupTable, 582
gdcm::SmartPointer, 878
vtkLookupTable16, 1216
GetPointerFromElement
gdcm::ImageHelper, 500
GetPointsBoundingBoxCoordinates
gdcm::Surface, 932
GetPosition
vtkImageColorViewer, 1191
GetPreamble
gdcm::FileMetaInformation, 415, 416
GetPrefix
gdcm::FilenameGenerator, 428
GetPresentationContext
gdcm::network::AAssociateRQPDU, 94

GetPresentationContextACByID
gdcm::network::ULConnection, 1071
GetPresentationContextAC
gdcm::network::AAssociateACPDU, 87
GetPresentationContextByAbstractSyntax
gdcm::network::AAssociateRQPDU, 95
GetPresentationContextByID
gdcm::network::AAssociateRQPDU, 95
GetPresentationContextIDFromPresentationContext
gdcm::network::ULConnection, 1072
GetPresentationContextID
gdcm::PresentationContext, 743
gdcm::network::PresentationContextAC, 745
gdcm::network::PresentationContextRQ, 752
gdcm::network::PresentationDataValue, 755
GetPresentationContextRQByID
gdcm::network::ULConnection, 1072
GetPresentationContexts
gdcm::PresentationContextGenerator, 749
gdcm::network::AAssociateRQPDU, 95
gdcm::network::ULConnection, 1072
GetPresentationDataValue
gdcm::network::PDataTFPDU, 690
GetPrettyPrint
gdcm::JSON, 571
GetPrimitiveData
gdcm::MeshPrimitive, 610, 611
GetPrimitiveType
gdcm::MeshPrimitive, 611
GetPrimitivesData
gdcm::MeshPrimitive, 611
GetPrintStyle
gdcm::Printer, 759
gdcm::XMLPrinter, 1241
GetPrivateCreator
gdcm::DataSet, 300
gdcm::Tag, 969
GetPrivateDict
gdcm::Dicts, 337
gdcm::XMLPrivateDictReader, 1245
GetProcessingAlgorithm
gdcm::Surface, 932
GetProgress
gdcm::ProgressEvent, 769
GetPropertyCategory
gdcm::Segment, 825, 826
GetPropertyType
gdcm::Segment, 826
GetProtocol
gdcm::network::ULConnection, 1072
GetPublicDict
gdcm::Dicts, 337
GetQuality
gdcm::JPEG2000Codec, 552
gdcm::JPEGCodec, 562
GetQueryDataSet
gdcm::BaseQuery, 166
GetQueryLevel
gdcm::QueryBase, 776
gdcm::QueryImage, 780
gdcm::QueryPatient, 782
gdcm::QuerySeries, 785
gdcm::QueryStudy, 787
GetQueryLevelFromQueryRoot
gdcm::BaseRootQuery, 170
GetQueryLevelFromString
gdcm::BaseRootQuery, 170
GetQueryLevelString
gdcm::BaseRootQuery, 170
GetRAWMD5FromFile
vtkGDCMTesting, 1176
GetRTStructSeriesUIDs
gdcm::DirectoryHelper, 348
GetRate
gdcm::JPEG2000Codec, 552
GetRealWorldValueMappingContent
gdcm::ImageHelper, 500
GetReason
gdcm::network::PresentationContextAC, 745
GetRecommendedDisplayCIELabValue
gdcm::Surface, 932, 933
GetRecommendedDisplayGrayscaleValue
gdcm::Surface, 933
GetRecommendedPresentationOpacity
gdcm::Surface, 933
GetRecommendedPresentationType
gdcm::Surface, 933
GetRef
gdcm::IODEntry, 528
GetReferencedFrameOfReferenceClassUID
vtkRTStructSetProperties, 1222
GetReferencedFrameOfReferenceInstanceUID
vtkRTStructSetProperties, 1222
GetRegion
gdcm::ImageRegionReader, 509
GetRequiredDataSet
gdcm::ModalityPerformedProcedureStepCreate↩
Query, 614
gdcm::ModalityPerformedProcedureStepSetQuery,
616
GetRequiredTags
gdcm::QueryBase, 776
gdcm::QueryImage, 780
gdcm::QueryPatient, 782
gdcm::QuerySeries, 785
gdcm::QueryStudy, 787
GetRescaleInterceptSlopeValue
gdcm::ImageHelper, 500

GetReserved43_74
gdcm::network::AAssociateRQPDU, 95
GetResponses
gdcm::network::ULBasicCallback, 1069
GetRetired
gdcm::DictEntry, 330
GetRoot
gdcm::UIDGenerator, 1003
GetRows
gdcm::Bitmap, 184
gdcm::Overlay, 678
GetSOPClassUIDFromIOD
gdcm::SOPClassUIDToIOD, 882
GetSOPClassUIDToIODs
gdcm::SOPClassUIDToIOD, 882
GetSOPClassUIDToIOD
gdcm::SOPClassUIDToIOD, 882
GetSOPClassUID
gdcm::DirectoryHelper, 348
GetSOPInstanceUID
gdcm::BaseQuery, 166
GetSTATESString
gdcm::Surface, 933
GetSTATES
gdcm::Surface, 933
GetSamplesPerPixel
gdcm::PhotometricInterpretation, 710
gdcm::PixelFormat, 716
GetScalarType
gdcm::PixelFormat, 716
GetScalarTypeAsString
gdcm::PixelFormat, 717
GetScanner
gdcm::DICOMDIRGenerator, 320
GetSegment
gdcm::SegmentWriter, 837
GetSegmentAlgorithmName
gdcm::Segment, 826
GetSegmentAlgorithmType
gdcm::Segment, 826
GetSegmentDescription
gdcm::Segment, 826
GetSegmentLabel
gdcm::Segment, 826
GetSegmentNumber
gdcm::Segment, 826
GetSegments
gdcm::SegmentReader, 834
gdcm::SegmentWriter, 837
GetSelectedPrivateGroupOffsetFromFile
gdcm::Testing, 981
GetSelectedTagsOffsetFromFile
gdcm::Testing, 981
GetSequenceOfFragments
gdcm::DataElement, 282
GetSeriesUIDsBySOPClassUID
gdcm::DirectoryHelper, 348
GetSize
gdcm::VR, 1122
vtkImageColorViewer, 1191
GetSizeof
gdcm::VR, 1122
GetSliceMax
vtkImageColorViewer, 1191
GetSliceMin
vtkImageColorViewer, 1192
GetSliceRange
vtkImageColorViewer, 1192
GetSlope
gdcm::Image, 463
gdcm::Rescaler, 804
GetSourceApplicationEntityTitle
gdcm::FileMetaInformation, 416
GetSourceDirectory
gdcm::Testing, 982
GetSpacing
gdcm::Image, 463
GetSpacingTagFromMediaStorage
gdcm::ImageHelper, 501
GetSpacingValue
gdcm::ImageHelper, 501
GetStart
gdcm::ByteBuffer, 198
GetState
gdcm::network::ULConnection, 1072
GetStateIndex
gdcm::network, 78
GetStream
gdcm::Trace, 986
GetStreamCurrentPosition
gdcm::Reader, 795
GetStreamOffsetFromFile
gdcm::Testing, 982
GetStreamPtr
gdcm::Reader, 795
gdcm::Writer, 1235
GetString
gdcm::MediaStorage, 601
gdcm::PhotometricInterpretation, 711
gdcm::TransferSyntax, 992
gdcm::UIDs, 1028
GetStringValueFromTag
gdcm::DirectoryHelper, 348
GetStructureSetObservationNumber
vtkRTStructSetProperties, 1222
GetStructureSetROIDescription
vtkRTStructSetProperties, 1222
GetStructureSetROIGenerationAlgorithm

vtkRTStructSetProperties, 1222
GetStructureSetROIName
vtkRTStructSetProperties, 1222
GetStructureSetROINumber
vtkRTStructSetProperties, 1223
GetStructureSetROIObservationLabel
vtkRTStructSetProperties, 1223
GetStructureSetROIRefFrameRefUID
vtkRTStructSetProperties, 1223
GetStructureSetRTROIInterpretedType
vtkRTStructSetProperties, 1223
GetSurface
gdcm::Segment, 826
GetSurfaceComments
gdcm::Surface, 933
GetSurfaceCount
gdcm::Segment, 827
GetSurfaceNumber
gdcm::Surface, 933
GetSurfaceProcessing
gdcm::Surface, 933
GetSurfaceProcessingDescription
gdcm::Surface, 934
GetSurfaceProcessingRatio
gdcm::Surface, 934
GetSurfaces
gdcm::Segment, 827
GetSwapCode
gdcm::TransferSyntax, 992
GetSwapCodeString
gdcm::SwapCode, 949
GetSyngoDT
gdcm::CSAElement, 254
GetTSString
gdcm::TransferSyntax, 992
GetTSType
gdcm::TransferSyntax, 992
GetTable
gdcm::SequenceOfFragments, 843
GetTableEntry
gdcm::Table, 959
GetTag
gdcm::AnonymizeEvent, 103
gdcm::Attribute, 129
gdcm::Attribute< Group, Element, TVR, VM::VM1 >, 135
gdcm::Attribute< Group, Element, TVR, VM::VM1_n >, 144
gdcm::DataElement, 282, 283
GetTagListByLevel
gdcm::BaseRootQuery, 171
gdcm::FindPatientRootQuery, 440
gdcm::FindStudyRootQuery, 443
gdcm::MovePatientRootQuery, 629
gdcm::MoveStudyRootQuery, 632
gdcm::WLMFindQuery, 1230
GetTempDirectory
gdcm::Testing, 982
GetTempDirectoryW
gdcm::Testing, 982
GetTempFilename
gdcm::Testing, 982
GetTempFilenameW
gdcm::Testing, 983
GetTimeout
gdcm::ServiceClassUser, 863
gdcm::network::ARTIMTimer, 122
GetTimer
gdcm::network::ULConnection, 1072
GetTimezoneOffsetFromUTC
gdcm::System, 955
GetToplevel
gdcm::Directory, 345
GetTransferSyntax
gdcm::Bitmap, 184
gdcm::ImageChangeTransferSyntax, 480
gdcm::PresentationContext, 743
gdcm::network::PresentationContextAC, 746
gdcm::network::PresentationContextRQ, 752
GetTransferSyntaxString
gdcm::UIDs, 1028
GetTransferSyntaxStrings
gdcm::UIDs, 1028
GetTransferSyntaxes
gdcm::network::PresentationContextRQ, 752
GetType
gdcm::ModuleEntry, 624
gdcm::Orientation, 672
gdcm::Overlay, 678
gdcm::PhotometricInterpretation, 711
GetTypeAsEnum
gdcm::Overlay, 678
GetTypeFromTag
gdcm::Defs, 313
gdcm::IOD, 525
GetTypeOfData
gdcm::Curve, 275
GetTypeOfDataDescription
gdcm::Curve, 275
GetTypeString
gdcm::Type, 999
GetTypeType
gdcm::Type, 1000
GetUIDName
gdcm::UIDs, 1028
GetUIDString
gdcm::UIDs, 1028
GetUniqueTags

gdcm::QueryBase, 777
gdcm::QueryImage, 780
gdcm::QueryPatient, 783
gdcm::QuerySeries, 785
gdcm::QueryStudy, 787
GetUnpackBuffer
gdcm::Overlay, 678
GetUnpackBufferLength
gdcm::Overlay, 678
GetUsage
gdcm::IODEntry, 528
GetUsageString
gdcm::Usage, 1097
GetUsageType
gdcm::IODEntry, 528
gdcm::Usage, 1097
GetUserData
gdcm::Parser, 686
GetUserInformation
gdcm::network::AAssociateACPDU, 87
gdcm::network::AAssociateRQPDU, 95
GetVIEWType
gdcm::Surface, 934
GetVIEWTypeString
gdcm::Surface, 934
GetVL16Max
gdcm::VL, 1111
GetVL32Max
gdcm::VL, 1111
GetVMString
gdcm::VM, 1116
GetVMType
gdcm::VM, 1117
GetVMTypeFromLength
gdcm::VM, 1117
GetVRFromTag
gdcm, 61
GetVRString
gdcm::VR, 1122
GetVRStringFromFile
gdcm::VR, 1122
GetVRType
gdcm::VR, 1122
GetVRTypeFromFile
gdcm::VR, 1122
GetVTKDataRoot
vtkGDCMTesting, 1176
GetValidDataSet
gdcm::WLMFindQuery, 1230
GetValidatedFile
gdcm::Validate, 1103
GetValue
gdcm::Attribute, 129
gdcm::Attribute< Group, Element, TVR, VM::VM1 >, 136
gdcm::Attribute< Group, Element, TVR, VM::VM1_n >, 144
gdcm::CSAElement, 254
gdcm::DataElement, 283
gdcm::Element, 353, 354
gdcm::Element< TVR, VM::VM1_n >, 359
gdcm::PDBElement, 692
gdcm::Scanner, 820
gdcm::StrictScanner, 912
GetValueAsSQ
gdcm::DataElement, 283
GetValues
gdcm::Attribute, 129
gdcm::Attribute< Group, Element, TVR, VM::VM1 >, 136
gdcm::Attribute< Group, Element, TVR, VM::VM1_n >, 144
gdcm::Element, 354
gdcm::Scanner, 820, 821
gdcm::StrictScanner, 912
GetVectorAccuracy
gdcm::Surface, 934
GetVectorCoordinateData
gdcm::Surface, 934
GetVectorDimensionality
gdcm::Surface, 934
GetVersion
gdcm::Version, 1108
GetVL
gdcm::DataElement, 284
GetVM
gdcm::Attribute, 129
gdcm::Attribute< Group, Element, TVR, VM::VM1 >, 136
gdcm::Attribute< Group, Element, TVR, VM::VM1_3 >, 140
gdcm::Attribute< Group, Element, TVR, VM::VM1_8 >, 141
gdcm::Attribute< Group, Element, TVR, VM::VM1_n >, 144
gdcm::Attribute< Group, Element, TVR, VM::VM2_2n >, 148
gdcm::Attribute< Group, Element, TVR, VM::VM2_n >, 149
gdcm::Attribute< Group, Element, TVR, VM::VM3_3n >, 151
gdcm::Attribute< Group, Element, TVR, VM::VM3_n >, 152
gdcm::CSAElement, 254
gdcm::CSAHeaderDictEntry, 267
gdcm::DictEntry, 330
gdcm::Element, 354

gdcm::Element< TVR, VM::VM1_n >, 359
GetVR
gdcm::Attribute, 129
gdcm::Attribute< Group, Element, TVR, VM::VM1 >, 136
gdcm::Attribute< Group, Element, TVR, VM::VM1_n >, 144
gdcm::CSAElement, 254
gdcm::CSAHeaderDictEntry, 267
gdcm::DataElement, 284
gdcm::DictEntry, 330
gdcm::Element, 354
gdcm::Element< TVR, VM::VM1_n >, 359
GetWarningFlag
gdcm::Trace, 986
GetWarningStream
gdcm::Trace, 986
GetWindowName
vtkImageColorViewer, 1192
GetXMax
gdcm::BoxRegion, 196
GetXMin
gdcm::BoxRegion, 196
GetYMax
gdcm::BoxRegion, 196
GetYMin
gdcm::BoxRegion, 196
GetZMax
gdcm::BoxRegion, 196
GetZMin
gdcm::BoxRegion, 196
GetZSpacing
gdcm::IPPSorter, 534
GetZSpacingTagFromMediaStorage
gdcm::ImageHelper, 501
GetZSpacingTolerance
gdcm::IPPSorter, 534
Global
gdcm::Defs, 314
gdcm::Dicts, 338
gdcm::Global, 448
GlobalInstance
gdcm, 73
GrabOverlayFromPixelData
gdcm::Overlay, 679
group
gdcm::SerieHelper::Rule, 813
GroupDict
gdcm::GroupDict, 451
GroupStringVector
gdcm::GroupDict, 451
GuessFromModality
gdcm::MediaStorage, 601
HandleBulkData
gdcm::XMLPrinter, 1241
HandleDataSet
gdcm::network::ULBasicCallback, 1069
gdcm::network::ULConnectionCallback, 1075
gdcm::network::ULWritingCallback, 1089
HandleDescription
gdcm::XMLDictReader, 1239
gdcm::XMLPrivateDictReader, 1245
HandleEntry
gdcm::XMLDictReader, 1239
gdcm::XMLPrivateDictReader, 1245
HandleEvent
gdcm::network::ULTransitionTable, 1087
HandleIODEntry
gdcm::TableReader, 962
HandleIOD
gdcm::TableReader, 962
HandleMacro
gdcm::TableReader, 962
HandleMacroEntry
gdcm::TableReader, 962
HandleMacroEntryDescription
gdcm::TableReader, 962
HandleModule
gdcm::TableReader, 962
HandleModuleEntry
gdcm::TableReader, 963
HandleModuleEntryDescription
gdcm::TableReader, 963
HandleModuleInclude
gdcm::TableReader, 963
HandleResponse
gdcm::network::ULBasicCallback, 1069
gdcm::network::ULConnectionCallback, 1075
gdcm::network::ULWritingCallback, 1089
HasObserver
gdcm::Subject, 925

IODEntry
gdcm::IODEntry, 527
IODMapType
gdcm::IODs, 530
IODMapTypeConstIterator
gdcm::IODs, 530
IODName
gdcm::IODs, 530
IODs
gdcm::IODs, 530
IOD
gdcm::IOD, 525
IPPSorter
gdcm::IPPSorter, 534
Icon

gdcm::Pixmap, 724
IconDataScalarType
vtkGDCMImageReader, 1140
vtkGDCMImageReader2, 1152
IconImage
gdcm, 58
IconImageDataExtent
vtkGDCMImageReader, 1140
vtkGDCMImageReader2, 1152
IconImageFilter
gdcm::IconImageFilter, 454
IconImageGenerator
gdcm::IconImageGenerator, 456
IconNumberOfScalarComponents
vtkGDCMImageReader, 1140
vtkGDCMImageReader2, 1152
ID
gdcm::PresentationContext, 744
ignore_char
gdcm::ignore_char, 459
Image
gdcm::Image, 462
ImageActor
vtkImageColorViewer, 1198
ImageApplyLookupTable
gdcm::ImageApplyLookupTable, 468
ImageChangePhotometricInterpretation
gdcm::ImageChangePhotometricInterpretation, 471
gdcm::ImageCodec, 491
ImageChangePlanarConfiguration
gdcm::ImageChangePlanarConfiguration, 475
ImageChangeTransferSyntax
gdcm::Bitmap, 189
gdcm::ImageChangeTransferSyntax, 479
ImageCodec
gdcm::ImageCodec, 484
ImageConverter
gdcm::ImageConverter, 493
ImageFormat
vtkGDCMImageReader, 1140
vtkGDCMImageReader2, 1152
ImageFragmentSplitter
gdcm::ImageFragmentSplitter, 496
ImageOrientationPatient
vtkGDCMImageReader, 1140
vtkGDCMImageReader2, 1152
ImagePositionPatient
vtkGDCMImageReader, 1140
vtkGDCMImageReader2, 1152
ImagePositionPatientOrdering
gdcm::SerieHelper, 857
ImageReader
gdcm::ImageReader, 505
ImageRegionReader
gdcm::ImageRegionReader, 509
gdcm::JPEG2000Codec, 553
gdcm::JPEGCodec, 564
gdcm::JPEGLSCodec, 570
gdcm::RLECodec, 811
ImageToImageFilter
gdcm::ImageToImageFilter, 512
ImageWriter
gdcm::ImageWriter, 515
ImplementationClassUIDSub
gdcm::network::ImplementationClassUIDSub, 517
ImplementationUIDSub
gdcm::network::ImplementationUIDSub, 518
ImplementationVersionNameSub
gdcm::network::ImplementationVersionNameSub, 519
IncompleteLUT
gdcm::LookupTable, 584
InitFromRQ
gdcm::network::AAssociateACPDU, 87
InitOpenSSL
gdcm::OpenSSLCryptoFactory, 662
Initialize
gdcm::network::ULConnectionInfo, 1077
InitializeBlueLUT
gdcm::LookupTable, 582
InitializeConnection
gdcm::ServiceClassUser, 863
gdcm::network::ULConnection, 1072
InitializeDataSet
gdcm::BaseRootQuery, 171
gdcm::FindPatientRootQuery, 440
gdcm::FindStudyRootQuery, 443
gdcm::MovePatientRootQuery, 629
gdcm::MoveStudyRootQuery, 632
gdcm::WLMFindQuery, 1230
InitializeGreenLUT
gdcm::LookupTable, 583
InitializeIncomingConnection
gdcm::network::ULConnection, 1072
InitializeLUT
gdcm::LookupTable, 583
InitializeRTStructSet
vtkGDCMPolyDataWriter, 1171
InitializeRedLUT
gdcm::LookupTable, 583
Initialized
gdcm::LookupTable, 583
Input
gdcm::BitmapToBitmapFilter, 192
Insert
gdcm::CommandDataSet, 236
gdcm::DataSet, 300
gdcm::FileMetaInformation, 416

gdcm::GroupDict, 452
InsertDataElement
gdcm::DataSet, 300
gdcm::Item, 539
InsertEntry
gdcm::Table, 959
InstallPipeline
vtkImageColorViewer, 1192
Interactor
vtkImageColorViewer, 1198
InteractorStyle
vtkImageColorViewer, 1199
Internal
gdcm::ApplicationEntity, 115
gdcm::Attribute, 132
gdcm::Attribute< Group, Element, TVR, VM::VM1 >, 138
gdcm::Element, 355
gdcm::Element< VR::AS, VM::VM5 >, 369
gdcm::LookupTable, 584
gdcm::UI, 1001
InternalCode
gdcm::Coder, 227
gdcm::JPEG12Codec, 544
gdcm::JPEG16Codec, 547
gdcm::JPEG8Codec, 556
Internals
vtkRTStructSetProperties, 1226
InverseRescale
gdcm::Rescaler, 804
InverseRescaleFunctionIntoBestFit
gdcm::Rescaler, 804
InvokeEvent
gdcm::Subject, 925
IsAETitleValid
gdcm::network::AAssociateRQPDU, 95
IsASCII2
gdcm::VR, 1122
IsASCII
gdcm::VR, 1122
IsBinary
gdcm::VR, 1123
IsBinary2
gdcm::VR, 1123
IsCompatible
gdcm::PixelFormat, 717
IsDual
gdcm::VR, 1123
IsEmpty
gdcm::Bitmap, 184
gdcm::ByteValue, 206
gdcm::CSAElement, 254
gdcm::CSAHeaderDict, 265
gdcm::Curve, 275
gdcm::DataElement, 284
gdcm::DataSet, 301
gdcm::Defs, 313
gdcm::Dict, 324
gdcm::Dicts, 338
gdcm::Filename, 421
gdcm::Macros, 590
gdcm::Modules, 627
gdcm::Overlay, 679
gdcm::Preamble, 739
gdcm::PrivateDict, 762
gdcm::SegmentHelper::BasicCodedEntry, 174
IsEncapsulated
gdcm::TransferSyntax, 992
IsEncoded
gdcm::TransferSyntax, 993
IsExplicit
gdcm::TransferSyntax, 993
IsFrameEncoder
gdcm::ImageCodec, 488
gdcm::JPEG2000Codec, 552
gdcm::JPEGCodec, 562
gdcm::JPEGLSCodec, 569
gdcm::RLECodec, 810
IsGroupLength
gdcm::Tag, 969
IsGroupXX
gdcm::Tag, 969
IsIdentical
gdcm::Filename, 421
IsIllegal
gdcm::Tag, 969
IsImage
gdcm::MediaStorage, 601
IsImplicit
gdcm::TransferSyntax, 993
IsInPixelData
gdcm::Overlay, 679
IsKey
gdcm::Scanner, 821
gdcm::StrictScanner, 912
IsLastFragment
gdcm::network::AAbortPDU, 84
gdcm::network::AAssociateACPDU, 88
gdcm::network::AAssociateRJPDU, 90
gdcm::network::AAssociateRQPDU, 96
gdcm::network::AReleaseRPPDU, 117
gdcm::network::AReleaseRQPDU, 120
gdcm::network::BasePDU, 162
gdcm::network::PDataTFPDU, 690
IsLossless
gdcm::PhotometricInterpretation, 711
gdcm::TransferSyntax, 993
IsLossy

gdcm::Bitmap, 184
gdcm::ImageCodec, 489
gdcm::PhotometricInterpretation, 711
gdcm::TransferSyntax, 993
IsOdd
gdcm::VL, 1111
IsPresentationContextAccepted
gdcm::ServiceClassUser, 863
IsPrintable
gdcm::ByteValue, 206
IsPrivate
gdcm::Tag, 969
IsPrivateCreator
gdcm::Tag, 969
IsPublic
gdcm::Tag, 970
IsRetired
gdcm::PhotometricInterpretation, 711
IsRowEncoder
gdcm::ImageCodec, 489
gdcm::JPEG2000Codec, 552
gdcm::JPEGCodec, 562
gdcm::JPEGLSCodec, 569
gdcm::RLECodec, 810
IsSameColorSpace
gdcm::PhotometricInterpretation, 711
IsStateSuspension
gdcm::JPEG12Codec, 544
gdcm::JPEG16Codec, 547
gdcm::JPEG8Codec, 556
gdcm::JPEGCodec, 562
IsSwap
gdcm::VR, 1123
IsTransferSyntaxCompatible
gdcm::Bitmap, 184
IsUndefined
gdcm::MediaStorage, 601
gdcm::VL, 1111
IsUndefinedLength
gdcm::DataElement, 285
gdcm::SequenceOfItems, 851
IsUnique
gdcm::DictEntry, 331
IsVRFile
gdcm::VR, 1123
IsValid
gdcm::ApplicationEntity, 115
gdcm::BoxRegion, 196
gdcm::CodeString, 231
gdcm::DirectionCosines, 342
gdcm::FileMetaInformation, 416
gdcm::ImageCodec, 489
gdcm::JPEGCodec, 562
gdcm::LO, 578
gdcm::PixelFormat, 717
gdcm::Preamble, 739
gdcm::Region, 801
gdcm::String, 918
gdcm::TagPath, 976
gdcm::TransferSyntax, 993
gdcm::UIDGenerator, 1003
gdcm::UUIDGenerator, 1101
gdcm::VM, 1117
gdcm::VR, 1123
IsZero
gdcm::Overlay, 679
ItFileSetHt
gdcm::SerieHelper, 858
Item
gdcm::Item, 538
ItemVector
gdcm::SequenceOfItems, 848
Items
gdcm::SequenceOfItems, 853
Iterator
gdcm::CSAHeaderDict, 264
gdcm::DataSet, 296
gdcm::Dict, 322
gdcm::SequenceOfFragments, 841
gdcm::SequenceOfItems, 848
iterator
gdcm::CodeString, 229
gdcm::LO, 576
gdcm::String, 916

JPEG12Codec
gdcm::JPEG12Codec, 543
JPEG16Codec
gdcm::JPEG16Codec, 546
JPEG2000Codec
gdcm::JPEG2000Codec, 550
JPEG8Codec
gdcm::JPEG8Codec, 555
JPEGCodec
gdcm::JPEGCodec, 559
JPEGLSCodec
gdcm::JPEGLSCodec, 566
JSON
gdcm::JSON, 571
Join
gdcm::Filename, 421

KAKADUCodec
gdcm::KAKADUCodec, 573
KeyField
gdcm::CSAElement, 257
KeyValuePairArrayType
gdcm::CompositeNetworkFunctions, 239
KeyValuePairType

gdcm::CompositeNetworkFunctions, 239

LOComp
gdcm, 58
LTComp
gdcm, 58
LUTPtr
gdcm::Bitmap, 181
gdcm::ImageCodec, 484
LUT
gdcm::Bitmap, 189
gdcm::ImageCodec, 492
Level
vtkImageMapToWindowLevelColors2, 1208
ListCharSets
gdcm::QueryFactory, 778
LO
gdcm::LO, 577
Load
gdcm::Directory, 345
LoadDefault
gdcm::CSAHeaderDict, 265
gdcm::Dict, 324
gdcm::PrivateDict, 762
LoadDefaults
gdcm::Defs, 313
gdcm::Dicts, 338
LoadFromDataElement
gdcm::CSAHeader, 262
gdcm::PDBHeader, 696
LoadFromFile
gdcm::Defs, 313
LoadIconImage
vtkGDCMImageReader, 1140
vtkGDCMImageReader2, 1152
LoadImageFromFiles
gdcm::DirectoryHelper, 348
LoadOverlays
vtkGDCMImageReader, 1141
vtkGDCMImageReader2, 1152
LoadResourcesFiles
gdcm::Global, 449
LoadSingleFile
vtkGDCMImageReader, 1133
vtkGDCMImageReader2, 1145
Locate
gdcm::Global, 449
LodModeType
gdcm, 61
LookupTable
gdcm::LookupTable, 580, 581
vtkImageMapToColors16, 1204
LookupTableType
gdcm::LookupTable, 580
LossyFlag
gdcm::Bitmap, 189
gdcm::ImageCodec, 491
vtkGDCMImageReader, 1141
vtkGDCMImageReader2, 1153

m_ConstMemberFunction
gdcm::MemberCommand, 607
m_MemberFunction
gdcm::MemberCommand, 607
gdcm::SimpleMemberCommand, 872
m_This
gdcm::MemberCommand, 607
gdcm::SimpleMemberCommand, 872
m_char
gdcm::ignore_char, 459
mAction
gdcm::network::Transition, 997
mConnection
gdcm::network::ULConnectionManager, 1084
MD5
gdcm::MD5, 593
MD5DataImagesType
gdcm::Testing, 978
MD5MetaImagesType
vtkGDCMTesting, 1175
mDataSet
gdcm::BaseQuery, 168
mElementOffsets
gdcm::StreamImageWriter, 904
mElementOffsets1
gdcm::StreamImageWriter, 904
mEnd
gdcm::network::Transition, 997
mHelpDescription
gdcm::BaseQuery, 168
gdcm::BaseRootQuery, 172
mImage
gdcm::BaseRootQuery, 172
mImplicit
gdcm::network::ULConnectionCallback, 1076
MPType
gdcm::MeshPrimitive, 609
mPatient
gdcm::BaseRootQuery, 172
mRootType
gdcm::BaseRootQuery, 172
MSType
gdcm::MediaStorage, 597
mSecondaryConnection
gdcm::network::ULConnectionManager, 1084
mSeries
gdcm::BaseRootQuery, 172
mSopInstanceUID

gdcm::BaseQuery, 168
mStudy
gdcm::BaseRootQuery, 172
mTransitions
gdcm::network::ULConnectionManager, 1084
mWriter
gdcm::StreamImageWriter, 905
mXMax
gdcm::StreamImageWriter, 905
mXMin
gdcm::StreamImageWriter, 905
mYMax
gdcm::StreamImageWriter, 905
mYMin
gdcm::StreamImageWriter, 905
mZMax
gdcm::StreamImageWriter, 905
mZMin
gdcm::StreamImageWriter, 905
Macro
gdcm::Macro, 587
MacroEntry
gdcm, 58
Macros
gdcm::Macros, 590
MakeDirectory
gdcm::System, 955
MakeNew
gdcm::network::Transition, 997
MakeObject
gdcm::AnonymizeEvent, 103
gdcm::DataEvent, 293
gdcm::DataSetEvent, 307
gdcm::Event, 381
gdcm::FileNameEvent, 425
gdcm::ProgressEvent, 769
MapCSAHeaderDictEntry
gdcm::CSAHeaderDict, 264
MapDictEntry
gdcm::Dict, 322
MapIODEntry
gdcm::IOD, 524
MapModuleEntry
gdcm::Macro, 587
gdcm::Module, 619
MapScalarsThroughTable2
vtkLookupTable16, 1216
MapTableEntry
gdcm::Table, 958
MappingType
gdcm::Scanner, 817
gdcm::StrictScanner, 908
MaxLength
gdcm::ApplicationEntity, 115
gdcm::PersonName, 704
MaxNumberOfComponents
gdcm::ApplicationEntity, 115
gdcm::PersonName, 704
MaxPrintLength
gdcm::Printer, 761
MaximumLengthSub
gdcm::network::MaximumLengthSub, 591
MediaStorage
gdcm::MediaStorage, 600
MediaStorageDataFilesType
gdcm::Testing, 978
MedicalImageProperties
vtkGDCMImageReader, 1141
vtkGDCMPolyDataReader, 1168
vtkGDCMPolyDataWriter, 1173
MemberCommand
gdcm::MemberCommand, 605
MeshPrimitive
gdcm::MeshPrimitive, 610
MessageID
gdcm::network::CEchoRQ, 215
MetaInformationTS
gdcm::FileMetaInformation, 418
ModalityPerformedProcedureStepCreateQuery
gdcm::ModalityPerformedProcedureStepCreate↩
Query, 614
ModalityPerformedProcedureStepSetQuery
gdcm::ModalityPerformedProcedureStepSetQuery,
616
Mode
gdcm::terminal, 81
Module
gdcm::Module, 619
ModuleEntry
gdcm::ModuleEntry, 623
ModuleMapType
gdcm::Macros, 589
gdcm::Modules, 626
Modules
gdcm::Modules, 626
MovePatientRootQuery
gdcm::MovePatientRootQuery, 629
MoveStudyRootQuery
gdcm::MoveStudyRootQuery, 632
mspFile
gdcm::StreamImageWriter, 904

NAction
gdcm::NormalizedNetworkFunctions, 653
NCreate
gdcm::NormalizedNetworkFunctions, 653
NDelete
gdcm::NormalizedNetworkFunctions, 653

NEventReport
gdcm::NormalizedNetworkFunctions, 654
NGet
gdcm::NormalizedNetworkFunctions, 654
NSet
gdcm::NormalizedNetworkFunctions, 654
Name
gdcm::ModuleEntry, 625
NameField
gdcm::CSAElement, 257
gdcm::PDBElement, 693
NeedByteSwap
gdcm::Bitmap, 189
gdcm::ImageCodec, 492
NeedOverlayCleanup
gdcm::ImageCodec, 492
NegociatedType
gdcm::TransferSyntax, 990
NestedMacroEntries
gdcm, 58
NestedModuleEntries
gdcm::NestedModuleEntries, 643
New
gdcm::Anonymizer, 108
gdcm::FileChangeTransferSyntax, 401
gdcm::FileStreamer, 434
gdcm::MemberCommand, 606
gdcm::Scanner, 821
gdcm::SequenceOfFragments, 843
gdcm::SequenceOfItems, 851
gdcm::ServiceClassUser, 863
gdcm::SimpleMemberCommand, 872
gdcm::StrictScanner, 912
vtkGDCMImageReader, 1133
vtkGDCMImageReader2, 1145
vtkGDCMImageWriter, 1157
vtkGDCMMedicalImageProperties, 1164
vtkGDCMPolyDataReader, 1166
vtkGDCMPolyDataWriter, 1171
vtkGDCMTesting, 1176
vtkGDCMThreadedImageReader, 1178
vtkGDCMThreadedImageReader2, 1182
vtkImageColorViewer, 1192
vtkImageMapToColors16, 1201
vtkImageMapToWindowLevelColors2, 1206
vtkImagePlanarComponentsToComponents, 1210
vtkImageRGBToYBR, 1212
vtkImageYBRToRGB, 1214
vtkLookupTable16, 1216
vtkRTStructSetProperties, 1223
NoOfItemsField
gdcm::CSAElement, 257
Normalize
gdcm::DirectionCosines, 342
NumberOfDimensions
gdcm::Bitmap, 189
gdcm::ImageCodec, 492
NumberOfIconImages
vtkGDCMImageReader, 1141
vtkGDCMImageReader2, 1153
NumberOfOverlays
vtkGDCMImageReader, 1141
vtkGDCMImageReader2, 1153
NumberOfSurfaces
gdcm::SurfaceWriter, 947

Object
gdcm::Object, 659
ObjectType
gdcm::MediaStorage, 599
Ofstream
gdcm::Writer, 1237
op
gdcm::SerieHelper::Rule, 813
OpenSSLCryptoFactory
gdcm::OpenSSLCryptoFactory, 662
OpenSSLCryptographicMessageSyntax
gdcm::OpenSSLCryptographicMessageSyntax, 663
OpenSSLP7CryptoFactory
gdcm::OpenSSLP7CryptoFactory, 666
OpenSSLP7CryptographicMessageSyntax
gdcm::OpenSSLP7CryptographicMessageSyntax, 668
operator const char *
gdcm::ConstCharWrapper, 243
gdcm::Filename, 421
gdcm::String, 918
operator const double *
gdcm::DirectionCosines, 342
operator const std::vector< char > &
gdcm::ByteValue, 206
operator MSType
gdcm::MediaStorage, 602
operator ObjectType *
gdcm::SmartPointer, 878
operator PIType
gdcm::PhotometricInterpretation, 711
operator ScalarType
gdcm::PixelFormat, 717
operator SwapCode::SwapCodeType
gdcm::SwapCode, 949
operator TSType
gdcm::TransferSyntax, 993
gdcm::UIDs, 1028
operator TypeType
gdcm::Type, 1000
operator uint32_t
gdcm::VL, 1111

operator UsageType
gdcm::Usage, 1097
operator VMType
gdcm::VM, 1117
operator VRType
gdcm::VR, 1123
operator!=
gdcm, 61, 62
gdcm::Attribute, 130
gdcm::Attribute< Group, Element, TVR, VM::VM1 >, 136
gdcm::CodeString, 231
gdcm::PixelFormat, 717
gdcm::Tag, 970
operator<
gdcm::Attribute, 130
gdcm::Attribute< Group, Element, TVR, VM::VM1 >, 136
gdcm::CSAElement, 255
gdcm::CSAHeaderDictEntry, 267
gdcm::DataElement, 285
gdcm::PrivateTag, 765
gdcm::Tag, 970
operator<<
gdcm, 62–71
gdcm::BasicOffsetTable, 177
gdcm::CSAElement, 256
gdcm::CSAHeader, 263
gdcm::CSAHeaderDict, 265
gdcm::CSAHeaderDictEntry, 268
gdcm::CodeString, 231
gdcm::CommandDataSet, 236
gdcm::DataElement, 289
gdcm::DataSet, 304
gdcm::Dict, 324
gdcm::DictEntry, 332
gdcm::Dicts, 338
gdcm::Directory, 346
gdcm::File, 394
gdcm::FileMetaInformation, 418
gdcm::FileSet, 431
gdcm::Fragment, 447
gdcm::Global, 450
gdcm::GroupDict, 452
gdcm::IODEntry, 529
gdcm::IODs, 531
gdcm::IOD, 526
gdcm::Item, 540
gdcm::Macro, 588
gdcm::Macros, 590
gdcm::MediaStorage, 603
gdcm::Module, 621
gdcm::ModuleEntry, 624
gdcm::Modules, 627
gdcm::NestedModuleEntries, 644
gdcm::Object, 660
gdcm::Orientation, 673
gdcm::PDBElement, 693
gdcm::PDBHeader, 696
gdcm::PhotometricInterpretation, 712
gdcm::PixelFormat, 719
gdcm::Preamble, 741
gdcm::PrivateDict, 763
gdcm::PrivateTag, 766
gdcm::Scanner, 822
gdcm::Sorter, 887
gdcm::StrictScanner, 913
gdcm::SwapCode, 949
gdcm::Table, 959
gdcm::Tag, 974
gdcm::TransferSyntax, 994
gdcm::Type, 1000
gdcm::UI, 1001
gdcm::Usage, 1097
gdcm::Version, 1109
gdcm::VL, 1113
gdcm::VM, 1117
gdcm::VR, 1124
operator<=
gdcm::Tag, 970
operator>>
gdcm, 72
gdcm::Tag, 974
operator*
gdcm::SmartPointer, 878
operator()
gdcm::DataSet, 301
gdcm::Scanner::ltstr, 585
gdcm::StrictScanner::ltstr, 586
operator++
gdcm::VL, 1111, 1112
operator+=
gdcm::VL, 1112
operator->
gdcm::SmartPointer, 878
operator=
gdcm::BoxRegion, 196
gdcm::ByteValue, 206
gdcm::CSAElement, 255
gdcm::DataElement, 285
gdcm::DataSet, 301
gdcm::Element< TVR, VM::VM1_n >, 360
gdcm::Object, 659
gdcm::Overlay, 679
gdcm::ParseException, 684
gdcm::Preamble, 740
gdcm::SequenceOfItems, 851
gdcm::SmartPointer, 879

gdcm::Tag, 970
gdcm::network::UserInformation, 1100
operator==
gdcm, 71
gdcm::Attribute, 130
gdcm::Attribute< Group, Element, TVR, VM::VM1 >, 137
gdcm::ByteValue, 206
gdcm::CSAElement, 255
gdcm::CodeString, 232
gdcm::DataElement, 285
gdcm::PDBElement, 693
gdcm::PixelFormat, 717
gdcm::PresentationContext, 743
gdcm::SequenceOfFragments, 844
gdcm::SequenceOfItems, 851
gdcm::Tag, 970
gdcm::Value, 1106
gdcm::network::AbstractSyntax, 99
gdcm::network::PresentationContextRQ, 752
gdcm::network::TransferSyntaxSub, 995
operator[]
gdcm::Attribute, 130
gdcm::Attribute< Group, Element, TVR, VM::VM1_n >, 145
gdcm::DataSet, 301
gdcm::Element, 354
gdcm::Element< TVR, VM::VM1_n >, 360
gdcm::Tag, 971
OrderFileList
gdcm::SerieHelper, 857
Orientation
gdcm::Orientation, 671
OrientationType
gdcm::Orientation, 671
Output
gdcm::BitmapToBitmapFilter, 192
OutputFormat
vtkImageMapToColors16, 1204
OutputTypes
gdcm::DictConverter, 325
Overlay
gdcm::Overlay, 676
OverlayImageActor
vtkImageColorViewer, 1199
OverlayType
gdcm::Overlay, 676
Overlays
gdcm::Pixmap, 724

PDBElement
gdcm::PDBElement, 692
PDBHeader
gdcm::PDBHeader, 695
PDFCodec
gdcm::PDFCodec, 698
PDataTFPDU
gdcm::network::PDataTFPDU, 690
PGXCodec
gdcm::PGXCodec, 706
PIType
gdcm::PhotometricInterpretation, 709
PNComp
gdcm, 59
PNMCodec
gdcm::PNMCodec, 736
PVRGCodec
gdcm::PVRGCodec, 771
Pack
gdcm::Unpacker12Bits, 1095
Padding
gdcm::ApplicationEntity, 116
gdcm::PersonName, 704
Parent
gdcm::Element< TVR, VM::VM1_2 >, 357
gdcm::Element< TVR, VM::VM2_2n >, 363
gdcm::Element< TVR, VM::VM2_n >, 365
gdcm::Element< TVR, VM::VM3_3n >, 366
gdcm::Element< TVR, VM::VM3_n >, 368
Parse
gdcm::Parser, 687
ParseBuffer
gdcm::Parser, 687
ParseCertificateFile
gdcm::CAPICryptographicMessageSyntax, 212
gdcm::CryptographicMessageSyntax, 250
gdcm::OpenSSLCryptographicMessageSyntax, 664
gdcm::OpenSSLP7CryptographicMessageSyntax, 669
ParseDateTime
gdcm::System, 956
ParseDump
gdcm::ASN1, 123
ParseDumpFile
gdcm::ASN1, 123
ParseException
gdcm::ParseException, 683
ParseKeyFile
gdcm::CAPICryptographicMessageSyntax, 212
gdcm::CryptographicMessageSyntax, 250
gdcm::OpenSSLCryptographicMessageSyntax, 664
gdcm::OpenSSLP7CryptographicMessageSyntax, 669
Parser
gdcm::Parser, 686
PassAlphaToOutput
vtkImageMapToColors16, 1204
Patient

gdcm::Patient, 688
PerformAction
gdcm::network::ULAction, 1032
gdcm::network::ULActionAA1, 1033
gdcm::network::ULActionAA2, 1034
gdcm::network::ULActionAA3, 1035
gdcm::network::ULActionAA4, 1037
gdcm::network::ULActionAA5, 1038
gdcm::network::ULActionAA6, 1039
gdcm::network::ULActionAA7, 1040
gdcm::network::ULActionAA8, 1042
gdcm::network::ULActionAE1, 1043
gdcm::network::ULActionAE2, 1044
gdcm::network::ULActionAE3, 1045
gdcm::network::ULActionAE4, 1047
gdcm::network::ULActionAE5, 1048
gdcm::network::ULActionAE6, 1049
gdcm::network::ULActionAE7, 1050
gdcm::network::ULActionAE8, 1052
gdcm::network::ULActionAR1, 1053
gdcm::network::ULActionAR10, 1054
gdcm::network::ULActionAR2, 1055
gdcm::network::ULActionAR3, 1057
gdcm::network::ULActionAR4, 1058
gdcm::network::ULActionAR5, 1059
gdcm::network::ULActionAR6, 1060
gdcm::network::ULActionAR7, 1062
gdcm::network::ULActionAR8, 1063
gdcm::network::ULActionAR9, 1064
gdcm::network::ULActionDT1, 1065
gdcm::network::ULActionDT2, 1067
PF
gdcm::Bitmap, 189
gdcm::ImageCodec, 492
PhotometricInterpretation
gdcm::PhotometricInterpretation, 710
PI
gdcm::Bitmap, 190
gdcm::ImageCodec, 492
PixelData
gdcm::Bitmap, 190
gdcm::PixmapReader, 727
gdcm::PixmapWriter, 734
PixelFormat
gdcm::PixelFormat, 715
Pixmap
gdcm::Pixmap, 721
PixmapReader
gdcm::Bitmap, 189
gdcm::PixmapReader, 726
PixmapToPixmapFilter
gdcm::PixmapToPixmapFilter, 729
PixmapWriter
gdcm::PixmapWriter, 732
PlanarConfiguration
gdcm::Bitmap, 190
gdcm::ImageCodec, 492
vtkGDCMImageReader, 1141
vtkGDCMImageReader2, 1153
pointer
gdcm::CodeString, 229
gdcm::LO, 576
gdcm::String, 916
Preamble
gdcm::Preamble, 738, 739
PrepareWrite
gdcm::PixmapWriter, 733
gdcm::SegmentWriter, 837
gdcm::SurfaceWriter, 946
PrepareWritePointMacro
gdcm::SurfaceWriter, 946
Prepend
gdcm::Global, 449
PresentationContext
gdcm::PresentationContext, 743
PresentationContextAC
gdcm::network::PresentationContextAC, 745
PresentationContextArrayType
gdcm::PresentationContextGenerator, 748
gdcm::network::AAssociateRQPDU, 93
PresentationContextGenerator
gdcm::PresentationContextGenerator, 748
PresentationContextRQ
gdcm::network::PresentationContextRQ, 751
PresentationDataValue
gdcm::network::PresentationDataValue, 754
PrettyPrintOff
gdcm::JSON, 571
PrettyPrintOn
gdcm::JSON, 571
PrimitiveData
gdcm::MeshPrimitive, 612
PrimitiveType
gdcm::MeshPrimitive, 612
PrimitivesData
gdcm::MeshPrimitive, 609
Print
gdcm::ApplicationEntity, 115
gdcm::Attribute, 130
gdcm::Attribute< Group, Element, TVR, VM::VM1 >, 137
gdcm::Attribute< Group, Element, TVR, VM::VM1_n >, 145
gdcm::BaseQuery, 166
gdcm::Bitmap, 185
gdcm::BoxRegion, 197
gdcm::ByteValue, 207
gdcm::CSAHeader, 262

gdcm::Curve, 276
gdcm::DataSet, 301
gdcm::DictPrinter, 334
gdcm::DirectionCosines, 342
gdcm::Directory, 346
gdcm::Element, 354
gdcm::Element< TVR, VM::VM1_n >, 360
gdcm::Element< VR::AS, VM::VM5 >, 369
gdcm::Event, 382
gdcm::Image, 463
gdcm::LookupTable, 583
gdcm::Object, 659
gdcm::Orientation, 672
gdcm::Overlay, 679
gdcm::PDBHeader, 696
gdcm::PersonName, 703
gdcm::PixelFormat, 718
gdcm::Pixmap, 723
gdcm::Preamble, 740
gdcm::PresentationContext, 744
gdcm::Printer, 759
gdcm::Region, 801
gdcm::Scanner, 821
gdcm::SegmentedPaletteColorLookupTable, 831
gdcm::SequenceOfFragments, 844
gdcm::SequenceOfItems, 851
gdcm::Sorter, 885
gdcm::StrictScanner, 913
gdcm::TagPath, 976
gdcm::Testing, 983
gdcm::Version, 1109
gdcm::XMLPrinter, 1241
gdcm::network::AAbortPDU, 84
gdcm::network::AAssociateACPDU, 88
gdcm::network::AAssociateRJPDU, 90
gdcm::network::AAssociateRQPDU, 96
gdcm::network::AReleaseRPPDU, 117
gdcm::network::AReleaseRQPDU, 120
gdcm::network::AbstractSyntax, 100
gdcm::network::ApplicationContext, 113
gdcm::network::AsynchronousOperationsWindow↩
Sub, 124
gdcm::network::BasePDU, 162
gdcm::network::ImplementationClassUIDSub, 517
gdcm::network::ImplementationVersionNameSub, 519
gdcm::network::MaximumLengthSub, 591
gdcm::network::PDataTFPDU, 690
gdcm::network::PresentationContextAC, 746
gdcm::network::PresentationContextRQ, 752
gdcm::network::PresentationDataValue, 755
gdcm::network::RoleSelectionSub, 812
gdcm::network::SOPClassExtendedNegociationSub, 880
gdcm::network::ServiceClassApplicationInformation, 859
gdcm::network::TransferSyntaxSub, 995
gdcm::network::UserInformation, 1100
PrintASCIIXML
gdcm::ByteValue, 207
PrintASCII
gdcm::ByteValue, 207
PrintAsContinuousString
gdcm::Tag, 971
PrintAsContinuousUpperCaseString
gdcm::Tag, 971
PrintAsPipeSeparatedString
gdcm::Tag, 971
PrintDataElement
gdcm::Printer, 759
gdcm::XMLPrinter, 1242
PrintDataElement2
gdcm::DictPrinter, 334
PrintDataSet
gdcm::Printer, 759
gdcm::XMLPrinter, 1242
PrintDataSet2
gdcm::DictPrinter, 334
PrintGroupLength
gdcm::ByteValue, 207
PrintHex
gdcm::ByteValue, 207
PrintHexXML
gdcm::ByteValue, 207
PrintPNXML
gdcm::ByteValue, 207
PrintSelf
vtkGDCMImageReader, 1133
vtkGDCMImageReader2, 1146
vtkGDCMImageWriter, 1157
vtkGDCMMedicalImageProperties, 1164
vtkGDCMPolyDataReader, 1167
vtkGDCMPolyDataWriter, 1171
vtkGDCMTesting, 1176
vtkGDCMThreadedImageReader, 1179
vtkGDCMThreadedImageReader2, 1182
vtkImageColorViewer, 1192
vtkImageMapToColors16, 1202
vtkImageMapToWindowLevelColors2, 1206
vtkImagePlanarComponentsToComponents, 1210
vtkImageRGBToYBR, 1212
vtkImageYBRToRGB, 1214
vtkLookupTable16, 1217
vtkRTStructSetProperties, 1223
PrintSQ
gdcm::Printer, 760
gdcm::XMLPrinter, 1242
PrintStyle

gdcm::Printer, 761
gdcm::XMLPrinter, 1242
PrintStyles
gdcm::Printer, 758
gdcm::XMLPrinter, 1241
PrintTable
gdcm::network::ULTransitionTable, 1087
PrintXML
gdcm::PrivateDict, 762
Printer
gdcm::Printer, 759
PrivateDict
gdcm::PrivateDict, 762
PrivateTag
gdcm::PrivateTag, 765
Process
gdcm::Parser, 687
ProcessDataSet
gdcm::FileExplicitFilter, 409
ProcessPublicTag
gdcm::Scanner, 821
gdcm::StrictScanner, 913
ProcessRequest
vtkGDCMImageReader2, 1146
ProduceCharacterSetDataElement
gdcm::QueryFactory, 778
ProduceQuery
gdcm::QueryFactory, 778
ProgressEvent
gdcm::ProgressEvent, 768
PropertyCategory
gdcm::Segment, 828
PropertyType
gdcm::Segment, 828
Push
gdcm::TagPath, 976
PushBackFile
vtkGDCMMedicalImageProperties, 1164
PythonFilter
gdcm::PythonFilter, 773

Quality
gdcm::JPEGCodec, 564
QueryFactory
gdcm::BaseQuery, 168
gdcm::BaseRootQuery, 172
gdcm::FindPatientRootQuery, 441
gdcm::FindStudyRootQuery, 443
gdcm::ModalityPerformedProcedureStepCreate↩
Query, 614
gdcm::ModalityPerformedProcedureStepSetQuery,
617
gdcm::MovePatientRootQuery, 630
gdcm::MoveStudyRootQuery, 633
gdcm::WLMFindQuery, 1231

RAWCodec
gdcm::RAWCodec, 789
README.txt, 1519
RGB2YBR
gdcm::ImageChangePhotometricInterpretation, 472
RGBPixelsToRGBPlanes
gdcm::ImageChangePlanarConfiguration, 475
RGBPlanesToRGBPixels
gdcm::ImageChangePlanarConfiguration, 476
RGBToRecommendedDisplayCIELab
gdcm::SurfaceHelper, 940
RGBToRecommendedDisplayGrayscale
gdcm::SurfaceHelper, 941
RLECodec
gdcm::RLECodec, 807
RTStructSetProperties
vtkGDCMPolyDataReader, 1169
vtkGDCMPolyDataWriter, 1173
Read
gdcm::BasicOffsetTable, 177
gdcm::ByteValue, 208
gdcm::CP246ExplicitDataElement, 244
gdcm::CSAHeader, 262
gdcm::CommandDataSet, 236
gdcm::DataElement, 285
gdcm::DataSet, 301
gdcm::Element, 354
gdcm::Element< TVR, VM::VM1_n >, 360
gdcm::EncodingImplementation< VR::VRASCII >,
375
gdcm::EncodingImplementation< VR::VRBINARY
>, 377
gdcm::ExplicitDataElement, 387
gdcm::ExplicitImplicitDataElement, 389
gdcm::File, 394
gdcm::FileMetaInformation, 416
gdcm::Fragment, 446
gdcm::ImageReader, 506
gdcm::ImageRegionReader, 510
gdcm::ImplicitDataElement, 521
gdcm::Item, 539
gdcm::PGXCodec, 707
gdcm::PNMCodec, 737
gdcm::PixmapReader, 726
gdcm::Preamble, 740
gdcm::Reader, 795
gdcm::SegmentReader, 834
gdcm::SequenceOfFragments, 844
gdcm::SequenceOfItems, 852
gdcm::StreamImageReader, 897
gdcm::SurfaceReader, 943
gdcm::TableReader, 963

gdcm::Tag, 971
gdcm::UNExplicitDataElement, 1091
gdcm::UNExplicitImplicitDataElement, 1094
gdcm::VR16ExplicitDataElement, 1126
gdcm::VRVLSize< 0 >, 1128
gdcm::VRVLSize< 1 >, 1129
gdcm::ValueIO, 1107
gdcm::VL, 1112
gdcm::VR, 1124
gdcm::network::AAbortPDU, 84
gdcm::network::AAssociateACPDU, 88
gdcm::network::AAssociateRJPDU, 91
gdcm::network::AAssociateRQPDU, 96
gdcm::network::AReleaseRPPDU, 118
gdcm::network::AReleaseRQPDU, 120
gdcm::network::AbstractSyntax, 100
gdcm::network::ApplicationContext, 113
gdcm::network::AsynchronousOperationsWindowSub, 124
gdcm::network::BasePDU, 162
gdcm::network::ImplementationClassUIDSub, 517
gdcm::network::ImplementationVersionNameSub, 519
gdcm::network::MaximumLengthSub, 592
gdcm::network::PDataTFPDU, 690
gdcm::network::PresentationContextAC, 746
gdcm::network::PresentationContextRQ, 752
gdcm::network::PresentationDataValue, 755
gdcm::network::RoleSelectionSub, 812
gdcm::network::SOPClassExtendedNegociationSub, 880
gdcm::network::ServiceClassApplicationInformation, 859
gdcm::network::TransferSyntaxSub, 995
gdcm::network::UserInformation, 1100
Read16
gdcm::VL, 1112
ReadACRNEMAImage
gdcm::ImageReader, 506
gdcm::PixmapReader, 727
ReadBacktrack
gdcm::Fragment, 446
ReadCompat
gdcm::FileMetaInformation, 416
ReadCompatInternal
gdcm::FileMetaInformation, 416
ReadComputeLength
gdcm::EncodingImplementation< VR::VRASCII >, 375
gdcm::EncodingImplementation< VR::VRBINARY >, 377
ReadDataSet
gdcm::Reader, 795
ReadFiles
vtkGDCMThreadedImageReader, 1179
ReadFromCommaSeparatedString
gdcm::PrivateTag, 766
gdcm::Tag, 972
ReadFromContinuousString
gdcm::Tag, 972
ReadFromPipeSeparatedString
gdcm::Tag, 972
ReadImage
gdcm::ImageReader, 506
gdcm::PixmapReader, 727
ReadImageInformation
gdcm::StreamImageReader, 898
ReadImageInternal
gdcm::PixmapReader, 727
ReadInformation
gdcm::ImageRegionReader, 510
ReadInto
gdcm::network::PDataTFPDU, 690
gdcm::network::PresentationDataValue, 755
ReadIntoBuffer
gdcm::ImageRegionReader, 510
ReadMetaInformation
gdcm::Reader, 795
ReadNested
gdcm::DataSet, 301
ReadNoSwap
gdcm::EncodingImplementation< VR::VRASCII >, 375
gdcm::EncodingImplementation< VR::VRBINARY >, 377
ReadOrSkip
gdcm::DataElement, 286
ReadPointMacro
gdcm::SurfaceReader, 944
ReadPreValue
gdcm::CP246ExplicitDataElement, 245
gdcm::DataElement, 286
gdcm::ExplicitDataElement, 387
gdcm::ExplicitImplicitDataElement, 389
gdcm::Fragment, 446
gdcm::ImplicitDataElement, 521
gdcm::SequenceOfFragments, 844
gdcm::UNExplicitDataElement, 1091
gdcm::UNExplicitImplicitDataElement, 1094
gdcm::VR16ExplicitDataElement, 1126
ReadPreamble
gdcm::Reader, 795
ReadSegment
gdcm::SegmentReader, 834
ReadSegments
gdcm::SegmentReader, 834
ReadSelectedPrivateTags
gdcm::DataSet, 302

gdcm::Reader, 796
ReadSelectedPrivateTagsWithLength
gdcm::DataSet, 302
ReadSelectedTags
gdcm::DataSet, 302
gdcm::Reader, 796
ReadSelectedTagsWithLength
gdcm::DataSet, 302
ReadSurface
gdcm::SurfaceReader, 944
ReadSurfaces
gdcm::SurfaceReader, 944
ReadUpToTag
gdcm::DataSet, 302
gdcm::Reader, 796
ReadUpToTagWithLength
gdcm::DataSet, 302
ReadValue
gdcm::CP246ExplicitDataElement, 245
gdcm::DataElement, 286
gdcm::ExplicitDataElement, 387
gdcm::ExplicitImplicitDataElement, 389
gdcm::Fragment, 446
gdcm::ImplicitDataElement, 521
gdcm::SequenceOfFragments, 844
gdcm::UNExplicitDataElement, 1091
gdcm::UNExplicitImplicitDataElement, 1094
gdcm::VR16ExplicitDataElement, 1126
ReadValueWithLength
gdcm::DataElement, 286
gdcm::ImplicitDataElement, 522
ReadVM
gdcm::DictConverter, 327
ReadVR
gdcm::DictConverter, 327
ReadWithLength
gdcm::CP246ExplicitDataElement, 245
gdcm::DataElement, 286
gdcm::DataSet, 303
gdcm::ExplicitDataElement, 387
gdcm::ExplicitImplicitDataElement, 389
gdcm::ImplicitDataElement, 522
gdcm::UNExplicitDataElement, 1091
gdcm::VR16ExplicitDataElement, 1126
Reader
gdcm::Reader, 794
Readuint16
gdcm::DictConverter, 327
RealWorldValueIntercept
gdcm::RealWorldValueMappingContent, 798
RealWorldValueSlope
gdcm::RealWorldValueMappingContent, 799
RecommendedDisplayCIELabToRGB
gdcm::SurfaceHelper, 939, 940
RecurseDataSet
gdcm::Anonymizer, 108
reference
gdcm::CodeString, 230
gdcm::LO, 576
gdcm::String, 916
ReferenceFrameOfReferenceUID
vtkRTStructSetProperties, 1226
ReferenceSeriesInstanceUID
vtkRTStructSetProperties, 1226
Region
gdcm::Region, 800
Register
gdcm::Object, 660
Remove
gdcm::Anonymizer, 108
gdcm::DataSet, 303
gdcm::FileAnonymizer, 397
gdcm::Preamble, 740
RemoveAllObservers
gdcm::Subject, 926
RemoveDictEntry
gdcm::PrivateDict, 763
RemoveFile
gdcm::System, 956
RemoveGroupLength
gdcm::Anonymizer, 109
RemoveItemByIndex
gdcm::SequenceOfItems, 852
RemoveObserver
gdcm::Subject, 926
RemoveOverlay
gdcm::Pixmap, 723
RemovePrivateTags
gdcm::Anonymizer, 109
RemoveRetired
gdcm::Anonymizer, 109
Render
vtkImageColorViewer, 1193
RenderWindow
vtkImageColorViewer, 1199
Renderer
vtkImageColorViewer, 1199
Replace
gdcm::Anonymizer, 109
gdcm::CommandDataSet, 236
gdcm::DataSet, 303
gdcm::FileAnonymizer, 397
gdcm::FileMetaInformation, 417
ReplaceEmpty
gdcm::DataSet, 303
RequestData
vtkGDCMImageReader2, 1146
vtkGDCMPolyDataReader, 1167

vtkImageMapToColors16, 1202
vtkImageMapToWindowLevelColors2, 1207
vtkImagePlanarComponentsToComponents, 1210
RequestData_HemodynamicWaveformStorage
vtkGDCMPolyDataReader, 1167
RequestData_RTStructureSetStorage
vtkGDCMPolyDataReader, 1167
RequestDataCompat
vtkGDCMImageReader, 1134
vtkGDCMImageReader2, 1146
vtkGDCMThreadedImageReader, 1179
RequestInformation
vtkGDCMImageReader2, 1146
vtkGDCMPolyDataReader, 1167
vtkGDCMThreadedImageReader2, 1182
vtkImageMapToColors16, 1202
vtkImageMapToWindowLevelColors2, 1207
RequestInformation_HemodynamicWaveformStorage
vtkGDCMPolyDataReader, 1167
RequestInformation_RTStructureSetStorage
vtkGDCMPolyDataReader, 1167
RequestInformationCompat
vtkGDCMImageReader, 1134
vtkGDCMImageReader2, 1146
RequestPaddedCompositePixelCode
gdcm::ImageCodec, 492
RequestPlanarConfiguration
gdcm::ImageCodec, 492
Rescale
gdcm::Rescaler, 804
RescaleFunctionIntoBestFit
gdcm::Rescaler, 804
Rescaler
gdcm::Rescaler, 803
ReserveDataElement
gdcm::FileStreamer, 434
ReserveGroupDataElement
gdcm::FileStreamer, 434
ResetHandledDataSet
gdcm::network::ULConnectionCallback, 1075
RetrieveSOPInstanceUIDFromIndex
gdcm::DirectoryHelper, 348
RetrieveSOPInstanceUIDFromZPosition
gdcm::DirectoryHelper, 349
reverse_iterator
gdcm::CodeString, 230
gdcm::LO, 577
gdcm::String, 916
RoleSelectionSub
gdcm::network::RoleSelectionSub, 811
RunEventLoop
gdcm::network::ULConnectionManager, 1081
RunMoveEventLoop
gdcm::network::ULConnectionManager, 1081
SHA1
gdcm::SHA1, 868
SHComp
gdcm, 59
SOPClassExtendedNegociationSub
gdcm::network::SOPClassExtendedNegociationSub, 880
SOPInstanceUID
vtkRTStructSetProperties, 1226
STATES
gdcm::Surface, 929
STComp
gdcm, 59
ScalarType
gdcm::PixelFormat, 714
Scale
vtkGDCMImageReader, 1141
vtkGDCMImageReader2, 1153
Scan
gdcm::Scanner, 822
gdcm::StrictScanner, 913
Scanner
gdcm::Scanner, 818
Segment
gdcm::Segment, 825
SegmentAlgorithmName
gdcm::Segment, 828
SegmentAlgorithmType
gdcm::Segment, 829
SegmentDescription
gdcm::Segment, 829
SegmentLabel
gdcm::Segment, 829
SegmentMap
gdcm::SegmentReader, 833
SegmentNumber
gdcm::Segment, 829
SegmentReader
gdcm::SegmentReader, 834
SegmentVector
gdcm::SegmentReader, 833
gdcm::SegmentWriter, 836
SegmentWriter
gdcm::SegmentWriter, 837
SegmentedPaletteColorLookupTable
gdcm::SegmentedPaletteColorLookupTable, 831
Segments
gdcm::SegmentReader, 835
gdcm::SegmentWriter, 838
Selection
gdcm::Sorter, 887
SelectionMap
gdcm::Sorter, 885
Self

gdcm::AnonymizeEvent, 102
gdcm::DataEvent, 292
gdcm::DataSetEvent, 306
gdcm::FileNameEvent, 424
gdcm::MemberCommand, 605
gdcm::ProgressEvent, 768
gdcm::SimpleMemberCommand, 871
SendEcho
gdcm::ServiceClassUser, 863
gdcm::network::ULConnectionManager, 1081
SendFind
gdcm::ServiceClassUser, 864
gdcm::network::ULConnectionManager, 1081, 1082
SendMove
gdcm::ServiceClassUser, 864
gdcm::network::ULConnectionManager, 1082
SendNAction
gdcm::network::ULConnectionManager, 1082
SendNCreate
gdcm::network::ULConnectionManager, 1082
SendNDelete
gdcm::network::ULConnectionManager, 1083
SendNEventReport
gdcm::network::ULConnectionManager, 1083
SendNGet
gdcm::network::ULConnectionManager, 1083
SendNSet
gdcm::network::ULConnectionManager, 1083
SendStore
gdcm::ServiceClassUser, 864, 865
gdcm::network::ULConnectionManager, 1084
Separator
gdcm::ApplicationEntity, 116
gdcm::PersonName, 704
SequenceLengthField
gdcm::SequenceOfItems, 853
SequenceOfFragments
gdcm::SequenceOfFragments, 841
SequenceOfItems
gdcm::SequenceOfItems, 849
SerieHelper
gdcm::SerieHelper, 855
SerieRestrictions
gdcm::SerieHelper, 855
Series
gdcm::Series, 859
SeriesInstanceUID
vtkRTStructSetProperties, 1226
ServiceClassApplicationInformation
gdcm::network::ServiceClassApplicationInformation, 859
ServiceClassUser
gdcm::ServiceClassUser, 862
Set
gdcm::Attribute, 131
gdcm::Attribute< Group, Element, TVR, VM::VM1 >, 137
gdcm::Attribute< Group, Element, TVR, VM::VM1_n >, 145
gdcm::Element, 355
gdcm::Element< TVR, VM::VM1_n >, 360
SetAETitle
gdcm::ServiceClassUser, 865
SetAbstractSyntax
gdcm::PresentationContext, 744
gdcm::network::PresentationContextRQ, 753
SetAlgorithmFamily
gdcm::Surface, 934
SetAlgorithmName
gdcm::Surface, 935
SetAlgorithmVersion
gdcm::Surface, 935
SetAnatomicRegion
gdcm::Segment, 827
SetArray
gdcm::Element< TVR, VM::VM1_n >, 360
SetAxisOfRotation
gdcm::Surface, 935
SetBitPosition
gdcm::Overlay, 680
SetBitSample
gdcm::JPEGCodec, 563
SetBitsAllocated
gdcm::Overlay, 680
gdcm::PixelFormat, 718
SetBitsStored
gdcm::PixelFormat, 718
SetBlob
gdcm::ApplicationEntity, 115
gdcm::PersonName, 703
gdcm::network::PresentationDataValue, 755
SetBlueLUT
gdcm::LookupTable, 583
SetBufferLength
gdcm::JPEGLSCodec, 569
gdcm::PNMCodec, 737
gdcm::RLECodec, 810
SetByteSwapTag
gdcm::ByteSwapFilter, 201
SetByteValue
gdcm::Attribute, 131
gdcm::Attribute< Group, Element, TVR, VM::VM1 >, 137
gdcm::Attribute< Group, Element, TVR, VM::VM1_n >, 145
gdcm::CSAElement, 255
gdcm::DataElement, 286
SetByteValueNoSwap

gdcm::Attribute, 131
gdcm::Attribute< Group, Element, TVR, VM::VM1 >, 137
SetCallbackFunction
gdcm::MemberCommand, 606
gdcm::SimpleMemberCommand, 872
SetCalledAETitle
gdcm::ServiceClassUser, 865
gdcm::network::AAssociateACPDU, 88
gdcm::network::AAssociateRQPDU, 96
SetCallingAETitle
gdcm::network::AAssociateACPDU, 88
gdcm::network::AAssociateRQPDU, 96
SetCenterOfRotation
gdcm::Surface, 935
SetChangePrivateTags
gdcm::FileExplicitFilter, 409
SetCheckFileMetaInformation
gdcm::Writer, 1235
SetCipherType
gdcm::CAPICryptographicMessageSyntax, 213
gdcm::CryptographicMessageSyntax, 250
gdcm::OpenSSLCryptographicMessageSyntax, 664
gdcm::OpenSSLP7CryptographicMessageSyntax, 669
SetColor
gdcm::Printer, 760
SetColorLevel
vtkImageColorViewer, 1193
SetColorWindow
vtkImageColorViewer, 1193
SetColumns
gdcm::Bitmap, 185
gdcm::Overlay, 680
SetCommand
gdcm::network::PresentationDataValue, 756
SetComponents
gdcm::PersonName, 703
SetCompressIconImage
gdcm::ImageChangeTransferSyntax, 480
SetComputeZSpacing
gdcm::IPPSorter, 534
SetCoordinateStartValue
gdcm::Curve, 276
SetCoordinateStepValue
gdcm::Curve, 276
SetCryptographicMessageSyntax
gdcm::Anonymizer, 110
SetCurve
gdcm::Curve, 276
vtkGDCMImageReader, 1134
vtkGDCMImageReader2, 1146
SetCurveDataDescriptor
gdcm::Curve, 276
SetCurveDescription
gdcm::Curve, 276
SetData
gdcm::DataEvent, 293
SetDataElement
gdcm::Bitmap, 185
SetDataSet
gdcm::File, 394
gdcm::network::PresentationDataValue, 756
SetDataSetTransferSyntax
gdcm::FileMetaInformation, 417
SetDataValueRepresentation
gdcm::Curve, 276
SetDebug
gdcm::Trace, 986
SetDebugStream
gdcm::Trace, 986
SetDefaultTransferSyntax
gdcm::PresentationContextGenerator, 749
SetDerivationCodeSequenceCodeValue
gdcm::FileDerivation, 406
SetDerivationDescription
gdcm::FileDerivation, 407
SetDescription
gdcm::CSAHeaderDictEntry, 267
gdcm::ModuleEntry, 624
gdcm::Overlay, 680
SetDescriptor
gdcm::DICOMDIRGenerator, 320
SetDictName
gdcm::DictConverter, 327
SetDicts
gdcm::PythonFilter, 773
gdcm::StringFilter, 921
SetDimension
gdcm::Bitmap, 185
SetDimensions
gdcm::Bitmap, 185
gdcm::Curve, 276
gdcm::ImageCodec, 489
SetDimensionsValue
gdcm::ImageHelper, 501
SetDirectionCosines
gdcm::Image, 463, 464
vtkGDCMImageWriter, 1157
SetDirectionCosinesFromImageOrientationPatient
vtkGDCMImageWriter, 1157
SetDirectionCosinesTolerance
gdcm::IPPSorter, 534
SetDirectionCosinesValue
gdcm::ImageHelper, 501
SetDirectory
gdcm::SerieHelper, 857
gdcm::network::ULWritingCallback, 1089

SetDisplayId
vtkImageColorViewer, 1193
SetDomain
gdcm::BoxRegion, 197
SetDropDuplicatePositions
gdcm::IPPSorter, 535
SetElement
gdcm::Tag, 972
SetElementHandler
gdcm::Parser, 687
SetElementTag
gdcm::Tag, 972, 973
SetElementXX
gdcm::DictEntry, 331
SetError
gdcm::Trace, 987
SetErrorStream
gdcm::Trace, 987
SetEvent
gdcm::network::ULEvent, 1086
SetFile
gdcm::Anonymizer, 110
gdcm::DICOMDIRGenerator, 320
gdcm::FileDecompressLookupTable, 404
gdcm::FileDerivation, 407
gdcm::FileExplicitFilter, 409
gdcm::IconImageFilter, 455
gdcm::Printer, 760
gdcm::PythonFilter, 774
gdcm::Reader, 796
gdcm::SplitMosaicFilter, 892
gdcm::StreamImageWriter, 902
gdcm::StringFilter, 921
gdcm::Validate, 1103
gdcm::Writer, 1235
gdcm::XMLPrinter, 1242
SetFileName
gdcm::FileNameEvent, 425
gdcm::Reader, 796
gdcm::StreamImageReader, 898
gdcm::StreamImageWriter, 903
gdcm::Writer, 1235
vtkGDCMThreadedImageReader2, 1183
SetFileNames
vtkGDCMImageReader, 1134
vtkGDCMImageWriter, 1157
vtkGDCMThreadedImageReader2, 1183
SetFilePattern
vtkGDCMImageReader, 1134
vtkGDCMImageReader2, 1147
SetFilePrefix
vtkGDCMImageReader, 1134
vtkGDCMImageReader2, 1147
SetFilename
gdcm::TableReader, 963
SetFilenames
gdcm::DICOMDIRGenerator, 320
SetFiles
gdcm::FileSet, 430
SetFiniteVolume
gdcm::Surface, 935
SetForce
gdcm::ImageChangeTransferSyntax, 480
gdcm::ImageFragmentSplitter, 496
SetForcePixelSpacing
gdcm::ImageHelper, 501
SetForceRescaleInterceptSlope
gdcm::ImageHelper, 502
SetFragmentSizeMax
gdcm::ImageFragmentSplitter, 496
SetFrameOrigin
gdcm::Overlay, 680
SetFromDataElement
gdcm::Attribute, 131
gdcm::Attribute< Group, Element, TVR, VM::VM1 >, 137
gdcm::Attribute< Group, Element, TVR, VM::VM1_n >, 145
gdcm::Element, 355
gdcm::Element< TVR, VM::VM1_n >, 360
SetFromDataSet
gdcm::Attribute, 131
gdcm::Attribute< Group, Element, TVR, VM::VM1 >, 138
gdcm::Attribute< Group, Element, TVR, VM::VM1_n >, 145
gdcm::MediaStorage, 602
SetFromFile
gdcm::MediaStorage, 602
SetFromHeader
gdcm::MediaStorage, 602
SetFromModality
gdcm::MediaStorage, 602
SetFromSourceImageSequence
gdcm::MediaStorage, 602
SetFromString
gdcm::DirectionCosines, 343
SetFromUID
gdcm::UIDs, 1028
SetGreenLUT
gdcm::LookupTable, 584
SetGroup
gdcm::Curve, 277
gdcm::Overlay, 680
gdcm::Tag, 973
SetGroupXX
gdcm::DictEntry, 331
SetHeader

gdcm::File, 394
SetHighBit
gdcm::PixelFormat, 718
SetHostname
gdcm::ServiceClassUser, 865
SetIconImage
gdcm::Pixmap, 723
SetIE
gdcm::IODEntry, 528
SetImage
gdcm::PixmapWriter, 733
gdcm::SplitMosaicFilter, 892
SetImplementationClassUID
gdcm::FileMetaInformation, 417
SetImplementationVersionName
gdcm::FileMetaInformation, 417
SetImplicitFlag
gdcm::network::ULConnectionCallback, 1075
SetInput
gdcm::BitmapToBitmapFilter, 192
gdcm::ImageConverter, 494
vtkImageColorViewer, 1193
SetInputConnection
vtkImageColorViewer, 1193
SetInputFileName
gdcm::DictConverter, 327
gdcm::FileAnonymizer, 398
gdcm::FileChangeTransferSyntax, 401
SetIntercept
gdcm::Image, 464
gdcm::Rescaler, 804
SetKey
gdcm::CSAElement, 255
SetKeyword
gdcm::DictEntry, 331
SetLUT
gdcm::Bitmap, 186
gdcm::ImageCodec, 489
gdcm::LookupTable, 584
gdcm::SegmentedPaletteColorLookupTable, 831
SetLastElement
gdcm::ParseException, 684
SetLastFragment
gdcm::network::PresentationDataValue, 756
SetLength
gdcm::ByteValue, 208
gdcm::Element< TVR, VM::VM1_2 >, 357
gdcm::Element< TVR, VM::VM1_n >, 361
gdcm::Element< TVR, VM::VM2_2n >, 363
gdcm::Element< TVR, VM::VM2_n >, 365
gdcm::Element< TVR, VM::VM3_3n >, 366
gdcm::Element< TVR, VM::VM3_n >, 368
gdcm::RLECodec, 810
gdcm::SequenceOfFragments, 844
gdcm::SequenceOfItems, 852
gdcm::Value, 1106
SetLengthOnly
gdcm::ByteValue, 208
gdcm::Value, 1106
SetLengthToUndefined
gdcm::SequenceOfItems, 852
SetLoadMode
gdcm::SerieHelper, 857
SetLookupTable
vtkImageMapToColors16, 1202
SetLossless
gdcm::JPEGCodec, 563
gdcm::JPEGLSCodec, 569
SetLossyError
gdcm::JPEGLSCodec, 569
SetLossyFlag
gdcm::Bitmap, 186
gdcm::ImageCodec, 489
gdcm::PVRGCodec, 772
SetManifold
gdcm::Surface, 935
SetMaxPDULength
gdcm::network::ULConnectionInfo, 1077
SetMaxPDUSize
gdcm::network::ULConnection, 1073
SetMaximumLength
gdcm::network::MaximumLengthSub, 592
SetMaximumPointDistance
gdcm::Surface, 935
SetMeanPointDistance
gdcm::Surface, 935
SetMedicalImageProperties
vtkGDCMImageReader, 1134
vtkGDCMImageReader2, 1147
vtkGDCMImageWriter, 1157
vtkGDCMPolyDataWriter, 1171
SetMergeModeToAbstractSyntax
gdcm::PresentationContextGenerator, 750
SetMergeModeToTransferSyntax
gdcm::PresentationContextGenerator, 750
SetMeshPrimitive
gdcm::Surface, 936
SetMessageHeader
gdcm::network::PresentationDataValue, 756
SetMinMaxForPixelType
gdcm::Rescaler, 805
SetName
gdcm::CSAElement, 255
gdcm::CSAHeaderDictEntry, 268
gdcm::DictEntry, 331
gdcm::IODEntry, 528
gdcm::Macro, 588
gdcm::Module, 620

gdcm::ModuleEntry, 624
gdcm::PDBElement, 693
gdcm::network::AbstractSyntax, 100
gdcm::network::ApplicationContext, 113
gdcm::network::TransferSyntaxSub, 995
SetNameFromUID
gdcm::network::AbstractSyntax, 100
gdcm::network::TransferSyntaxSub, 995
SetNeedByteSwap
gdcm::Bitmap, 186
gdcm::ImageCodec, 490
SetNeedOverlayCleanup
gdcm::ImageCodec, 490
SetNestedDataSet
gdcm::Item, 540
SetNoOfItems
gdcm::CSAElement, 256
SetNoSwap
gdcm::Element, 355
gdcm::Element< TVR, VM::VM1_n >, 361
SetNumberOfCurves
gdcm::Pixmap, 723
SetNumberOfDimensions
gdcm::Bitmap, 186
gdcm::ImageCodec, 490
SetNumberOfFilenames
gdcm::FilenameGenerator, 428
SetNumberOfFrames
gdcm::Overlay, 681
SetNumberOfInputPorts
vtkGDCMPolyDataWriter, 1172
SetNumberOfItems
gdcm::SequenceOfItems, 853
SetNumberOfOverlays
gdcm::Pixmap, 723
SetNumberOfPoints
gdcm::Curve, 277
SetNumberOfResolutions
gdcm::JPEG2000Codec, 552
SetNumberOfSegments
gdcm::SegmentWriter, 837
SetNumberOfSurfacePoints
gdcm::Surface, 936
SetNumberOfSurfaces
gdcm::SurfaceWriter, 946
SetNumberOfTableValues
vtkLookupTable16, 1217
SetNumberOfValues
gdcm::Attribute< Group, Element, TVR, VM::VM1_n >, 146
SetNumberOfVectors
gdcm::Surface, 936
SetObliquityThresholdCosineValue
gdcm::Orientation, 672
SetOffScreenRendering
vtkImageColorViewer, 1193
SetOrigin
gdcm::Image, 464
gdcm::Overlay, 681
SetOriginValue
gdcm::ImageHelper, 502
SetOutputDimensions
gdcm::IconImageGenerator, 458
SetOutputFileName
gdcm::DictConverter, 328
gdcm::FileAnonymizer, 398
gdcm::FileChangeTransferSyntax, 401
gdcm::FileStreamer, 435
SetOutputFormatToLuminance
vtkImageMapToColors16, 1202
SetOutputFormatToLuminanceAlpha
vtkImageMapToColors16, 1202
SetOutputFormatToRGBA
vtkImageMapToColors16, 1202
SetOutputFormatToRGB
vtkImageMapToColors16, 1202
SetOutputType
gdcm::DictConverter, 328
SetOutsideValuePixel
gdcm::IconImageGenerator, 458
SetOverlay
gdcm::Overlay, 681
SetOverlayVisibility
vtkImageColorViewer, 1194
SetOwner
gdcm::PrivateTag, 766
SetPDU
gdcm::network::ULEvent, 1086
SetPMSRescaleInterceptSlope
gdcm::ImageHelper, 502
SetParentId
vtkImageColorViewer, 1194
SetPassword
gdcm::CAPICryptographicMessageSyntax, 213
gdcm::CryptographicMessageSyntax, 250
gdcm::OpenSSLCryptographicMessageSyntax, 665
gdcm::OpenSSLP7CryptographicMessageSyntax, 669
SetPattern
gdcm::FilenameGenerator, 428
SetPermissions
gdcm::System, 956
SetPhotometricInterpretation
gdcm::Bitmap, 186
gdcm::ImageChangePhotometricInterpretation, 472
gdcm::ImageCodec, 490
SetPixelFormat
gdcm::Bitmap, 186

gdcm::ImageCodec, 490
gdcm::JPEGCodec, 563
gdcm::Rescaler, 805
SetPixelMinMax
gdcm::IconImageGenerator, 458
SetPixelRepresentation
gdcm::PixelFormat, 718
SetPixmap
gdcm::FileDecompressLookupTable, 404
gdcm::IconImageGenerator, 458
gdcm::PixmapWriter, 733
SetPlanarConfiguration
gdcm::Bitmap, 187
gdcm::ImageChangePlanarConfiguration, 476
gdcm::ImageCodec, 490
SetPointCoordinatesData
gdcm::Surface, 936
SetPointPositionAccuracy
gdcm::Surface, 936
SetPointsBoundingBoxCoordinates
gdcm::Surface, 936
SetPort
gdcm::ServiceClassUser, 865
SetPortSCP
gdcm::ServiceClassUser, 866
SetPosition
vtkImageColorViewer, 1194
SetPreamble
gdcm::FileMetaInformation, 417
SetPrefix
gdcm::FilenameGenerator, 428
SetPresentationContextID
gdcm::PresentationContext, 744
gdcm::network::PresentationContextAC, 746
gdcm::network::PresentationContextRQ, 753
gdcm::network::PresentationDataValue, 756
SetPresentationContexts
gdcm::ServiceClassUser, 866
gdcm::network::ULConnection, 1073
SetPrettyPrint
gdcm::JSON, 572
SetPrimitiveData
gdcm::MeshPrimitive, 611
SetPrimitiveType
gdcm::MeshPrimitive, 612
SetPrimitivesData
gdcm::MeshPrimitive, 611
SetPrivateCreator
gdcm::Tag, 973
SetProcessingAlgorithm
gdcm::Surface, 936
SetProgress
gdcm::ProgressEvent, 769
SetPropertyCategory
gdcm::Segment, 827
SetPropertyType
gdcm::Segment, 827
SetPurposeOfReferenceCodeSequenceCodeValue
gdcm::FileDerivation, 407
SetQuality
gdcm::JPEG2000Codec, 552
gdcm::JPEGCodec, 563
SetRTStructSetProperties
vtkGDCMPolyDataWriter, 1172
SetRate
gdcm::JPEG2000Codec, 552
SetReason
gdcm::network::AAbortPDU, 84
gdcm::network::PresentationContextAC, 746
SetRecommendedDisplayCIELabValue
gdcm::Surface, 936, 937
SetRecommendedDisplayGrayscaleValue
gdcm::Surface, 937
SetRecommendedPresentationOpacity
gdcm::Surface, 937
SetRecommendedPresentationType
gdcm::Surface, 937
SetRecomputeItemLength
gdcm::FileExplicitFilter, 410
SetRecomputeSequenceLength
gdcm::FileExplicitFilter, 410
SetRedLUT
gdcm::LookupTable, 584
SetRef
gdcm::IODEntry, 528
SetRegion
gdcm::ImageRegionReader, 510
SetRenderWindow
vtkImageColorViewer, 1194
SetRenderer
vtkImageColorViewer, 1194
SetRescaleInterceptSlopeValue
gdcm::ImageHelper, 502
SetRetired
gdcm::DictEntry, 331
SetReversible
gdcm::JPEG2000Codec, 553
SetRoot
gdcm::UIDGenerator, 1003
SetRootDirectory
gdcm::DICOMDIRGenerator, 320
SetRows
gdcm::Bitmap, 187
gdcm::Overlay, 681
SetSOPInstanceUID
gdcm::BaseQuery, 167
SetSamplesPerPixel
gdcm::PixelFormat, 718

SetScalarType
gdcm::PixelFormat, 718
SetSearchParameter
gdcm::BaseQuery, 166, 167
SetSegmentAlgorithmName
gdcm::Segment, 827
SetSegmentAlgorithmType
gdcm::Segment, 827
SetSegmentDescription
gdcm::Segment, 828
SetSegmentLabel
gdcm::Segment, 828
SetSegmentNumber
gdcm::Segment, 828
SetSegments
gdcm::SegmentWriter, 838
SetSize
vtkImageColorViewer, 1194, 1195
SetSlice
vtkImageColorViewer, 1195
SetSliceOrientation
vtkImageColorViewer, 1195
SetSliceOrientationToXY
vtkImageColorViewer, 1195
SetSliceOrientationToXZ
vtkImageColorViewer, 1195
SetSliceOrientationToYZ
vtkImageColorViewer, 1195
SetSlope
gdcm::Image, 464
gdcm::Rescaler, 805
SetSortFunction
gdcm::Sorter, 886
SetSource
gdcm::network::AAbortPDU, 85
SetSourceApplicationEntityTitle
gdcm::FileMetaInformation, 417
SetSpacing
gdcm::Image, 465
SetSpacingValue
gdcm::ImageHelper, 502
SetState
gdcm::network::ULConnection, 1073
SetStream
gdcm::Reader, 797
gdcm::StreamImageReader, 898
gdcm::StreamImageWriter, 903
gdcm::Trace, 987
gdcm::Writer, 1236
SetStreamToFile
gdcm::Trace, 987
SetStyle
gdcm::Printer, 760
gdcm::XMLPrinter, 1242
SetSurfaceComments
gdcm::Surface, 937
SetSurfaceCount
gdcm::Segment, 828
SetSurfaceNumber
gdcm::Surface, 937
SetSurfaceProcessing
gdcm::Surface, 937
SetSurfaceProcessingDescription
gdcm::Surface, 938
SetSurfaceProcessingRatio
gdcm::Surface, 938
SetSyngoDT
gdcm::CSAElement, 256
SetTag
gdcm::AnonymizeEvent, 103
gdcm::DataElement, 287
SetTargetPixelType
gdcm::Rescaler, 805
SetTemplateFileName
gdcm::FileStreamer, 435
SetTileSize
gdcm::JPEG2000Codec, 553
SetTimeout
gdcm::ServiceClassUser, 866
gdcm::network::ARTIMTimer, 122
SetToUndefined
gdcm::VL, 1112
SetTransferSyntax
gdcm::Bitmap, 187
gdcm::FileChangeTransferSyntax, 401
gdcm::ImageChangeTransferSyntax, 480
gdcm::network::PresentationContextAC, 746
SetTuple
gdcm::network::RoleSelectionSub, 812
gdcm::network::SOPClassExtendedNegociationSub, 880
gdcm::network::ServiceClassApplicationInformation, 860
SetType
gdcm::ModuleEntry, 624
gdcm::Overlay, 681
SetTypeOfData
gdcm::Curve, 277
SetUsage
gdcm::IODEntry, 528
SetUseSeriesDetails
gdcm::SerieHelper, 857
SetUseTargetPixelType
gdcm::Rescaler, 805
SetUseVRUN
gdcm::FileExplicitFilter, 410
SetUserCodec
gdcm::ImageChangeTransferSyntax, 480

SetUserData
gdcm::Parser, 687
SetUserInformation
gdcm::network::AAssociateRQPDU, 96
SetVLToUndefined
gdcm::DataElement, 288
SetValue
gdcm::Attribute, 132
gdcm::Attribute< Group, Element, TVR, VM::VM1 >, 138
gdcm::Attribute< Group, Element, TVR, VM::VM1_n >, 146
gdcm::CSAElement, 256
gdcm::DataElement, 287
gdcm::Element, 355
gdcm::Element< TVR, VM::VM1_n >, 361
gdcm::PDBElement, 693
SetValueFieldLength
gdcm::DataElement, 287
SetValues
gdcm::Attribute, 132
gdcm::Attribute< Group, Element, TVR, VM::VM1_n >, 146
SetVectorAccuracy
gdcm::Surface, 938
SetVectorCoordinateData
gdcm::Surface, 938
SetVectorDimensionality
gdcm::Surface, 938
SetVL
gdcm::DataElement, 288
SetVM
gdcm::CSAElement, 256
gdcm::CSAHeaderDictEntry, 268
gdcm::DictEntry, 332
SetVR
gdcm::CSAElement, 256
gdcm::CSAHeaderDictEntry, 268
gdcm::DataElement, 288
gdcm::DictEntry, 332
SetWarning
gdcm::Trace, 987
SetWarningStream
gdcm::Trace, 988
SetWindowId
vtkImageColorViewer, 1196
SetWriteDataSetOnly
gdcm::Writer, 1236
SetZSpacingTolerance
gdcm::IPPSorter, 535
setattribute
gdcm::terminal, 81
setbgcolor
gdcm::terminal, 81
setfgcolor
gdcm::terminal, 82
setmode
gdcm::terminal, 82
SetupInteractor
vtkImageColorViewer, 1196
Shift
vtkGDCMImageReader, 1141
vtkGDCMImageReader2, 1153
ShiftEnd
gdcm::ByteBuffer, 198
ShowAbort
gdcm::SimpleSubjectWatcher, 874
ShowAnonymization
gdcm::SimpleSubjectWatcher, 874
ShowData
gdcm::SimpleSubjectWatcher, 874
ShowDataSet
gdcm::SimpleSubjectWatcher, 874
ShowFileName
gdcm::SimpleSubjectWatcher, 874
ShowIteration
gdcm::SimpleSubjectWatcher, 875
ShowProgress
gdcm::SimpleSubjectWatcher, 875
SimpleMemberCommand
gdcm::SimpleMemberCommand, 871
SimpleSubjectWatcher
gdcm::SimpleSubjectWatcher, 874
SingleSerieUIDFileSetHT
gdcm::SerieHelper, 858
SingleSerieUIDFileSetmap
gdcm::SerieHelper, 855
Size
gdcm::CodeString, 231
gdcm::DataSet, 304
gdcm::GroupDict, 452
gdcm::network::AAbortPDU, 85
gdcm::network::AAssociateACPDU, 88
gdcm::network::AAssociateRJPDU, 91
gdcm::network::AAssociateRQPDU, 97
gdcm::network::AReleaseRPPDU, 118
gdcm::network::AReleaseRQPDU, 120
gdcm::network::AbstractSyntax, 100
gdcm::network::ApplicationContext, 113
gdcm::network::AsynchronousOperationsWindowSub, 124
gdcm::network::BasePDU, 163
gdcm::network::ImplementationClassUIDSub, 517
gdcm::network::ImplementationVersionNameSub, 519
gdcm::network::MaximumLengthSub, 592
gdcm::network::PDataTFPDU, 691
gdcm::network::PresentationContextAC, 746

gdcm::network::PresentationContextRQ, 753
gdcm::network::PresentationDataValue, 756
gdcm::network::RoleSelectionSub, 812
gdcm::network::SOPClassExtendedNegociationSub, 880
gdcm::network::ServiceClassApplicationInformation, 860
gdcm::network::TransferSyntaxSub, 995
gdcm::network::UserInformation, 1100
size_type
gdcm::CodeString, 230
gdcm::LO, 577
gdcm::String, 917
SizeType
gdcm::DataSet, 297
gdcm::FilenameGenerator, 426
gdcm::IOD, 524
gdcm::NestedModuleEntries, 643
gdcm::PresentationContext, 742
gdcm::PresentationContextGenerator, 748
gdcm::SequenceOfFragments, 841
gdcm::SequenceOfItems, 848
gdcm::network::AAssociateACPDU, 87
gdcm::network::AAssociateRQPDU, 93
gdcm::network::PDataTFPDU, 689
gdcm::network::PresentationContextRQ, 751
Slice
vtkImageColorViewer, 1199
SliceOrientation
vtkImageColorViewer, 1199
SmartPointer
gdcm::Object, 660
gdcm::SmartPointer, 877, 878
Sort
gdcm::IPPSorter, 535
gdcm::Sorter, 886
SortFunc
gdcm::Sorter, 887
SortFunction
gdcm::Sorter, 885
Sorter
gdcm::Sorter, 885
Spacing
gdcm::Spacing, 890
SpacingType
gdcm::Spacing, 889
Spectroscopy
gdcm::Spectroscopy, 891
Split
gdcm::ImageFragmentSplitter, 496
gdcm::SplitMosaicFilter, 893
SplitExtent
vtkGDCMThreadedImageReader2, 1183
SplitMosaicFilter
gdcm::SplitMosaicFilter, 892
Squeeze
gdcm::ApplicationEntity, 115
StableSort
gdcm::Sorter, 886
Start
gdcm::network::ARTIMTimer, 122
StartAssociation
gdcm::ServiceClassUser, 866
StartDataElement
gdcm::FileStreamer, 435
StartElement
gdcm::TableReader, 963
gdcm::XMLDictReader, 1239
gdcm::XMLPrivateDictReader, 1245
StartElementHandler
gdcm::Parser, 685
StartEncode
gdcm::ImageCodec, 491
gdcm::JPEG2000Codec, 553
gdcm::JPEGCodec, 563
gdcm::JPEGLSCodec, 569
gdcm::RLECodec, 810
StartFilter
gdcm::SimpleSubjectWatcher, 875
StartGroupDataElement
gdcm::FileStreamer, 435
Stop
gdcm::network::ARTIMTimer, 122
StopAssociation
gdcm::ServiceClassUser, 867
StopDataElement
gdcm::FileStreamer, 436
StopEncode
gdcm::ImageCodec, 491
gdcm::JPEG2000Codec, 553
gdcm::JPEGCodec, 563
gdcm::JPEGLSCodec, 570
gdcm::RLECodec, 810
StopGroupDataElement
gdcm::FileStreamer, 436
StopProtocol
gdcm::network::ULConnection, 1073
StrCaseCmp
gdcm::System, 956
StrNCaseCmp
gdcm::System, 957
StrSep
gdcm::System, 957
StrTokR
gdcm::System, 957
Stream
gdcm::Writer, 1237
StreamImageReader

gdcm::Reader, 797
gdcm::StreamImageReader, 896
StreamImageWriter
gdcm::StreamImageWriter, 901
gdcm::Writer, 1237
StrictScanner
gdcm::StrictScanner, 909
String
gdcm::String, 917
StringFilter
gdcm::StringFilter, 920
StructureSetDate
vtkRTStructSetProperties, 1226
StructureSetLabel
vtkRTStructSetProperties, 1227
StructureSetName
vtkRTStructSetProperties, 1227
StructureSetTime
vtkRTStructSetProperties, 1227
Study
gdcm::Study, 923
StudyInstanceUID
vtkRTStructSetProperties, 1227
Subject
gdcm::Subject, 924
Superclass
gdcm::AnonymizeEvent, 102
gdcm::DataEvent, 292
gdcm::DataSetEvent, 306
gdcm::FileNameEvent, 424
gdcm::LO, 577
gdcm::ProgressEvent, 768
Surface
gdcm::Surface, 930
SurfaceCount
gdcm::Segment, 829
SurfaceReader
gdcm::SurfaceReader, 943
SurfaceVector
gdcm::Segment, 824
SurfaceWriter
gdcm::SurfaceWriter, 946
Surfaces
gdcm::Segment, 829
Swap
gdcm::ByteSwap, 199
gdcm::SwapperDoOp, 950
gdcm::SwapperNoOp, 950
SwapArray
gdcm::SwapperDoOp, 950
gdcm::SwapperNoOp, 950
SwapCode
gdcm::SwapCode, 948
SwapCodeType
gdcm::SwapCode, 948
SwapFromSwapCodeIntoSystem
gdcm::ByteSwap, 199
SwapRange
gdcm::ByteSwap, 199
SwapRangeFromSwapCodeIntoSystem
gdcm::ByteSwap, 200
SyngoDTField
gdcm::CSAElement, 257
SystemIsBigEndian
gdcm::ByteSwap, 200
SystemIsLittleEndian
gdcm::ByteSwap, 200

TConstMemberFunctionPointer
gdcm::MemberCommand, 605
TMComp
gdcm, 59
TMemberFunctionPointer
gdcm::MemberCommand, 605
gdcm::SimpleMemberCommand, 871
TSName
gdcm::UIDs, 1015
TSType
gdcm::TransferSyntax, 990
gdcm::UIDs, 1021
TYPETOENCODING
gdcm, 72
gdcmVR.h, 1512
TYPETOLENGTH
gdcmVM.h, 1510
Table
gdcm::Table, 958
Table16
vtkLookupTable16, 1217
TableEntry
gdcm::TableEntry, 960
TableReader
gdcm::TableReader, 961
TableRow
gdcm::network::TableRow, 964
Tag
gdcm::Tag, 967, 968
tag
gdcm::Tag, 974
TagField
gdcm::DataElement, 289
TagPath
gdcm::TagPath, 975
TagToValue
gdcm::Scanner, 817
gdcm::StrictScanner, 909
TagToValueValueType
gdcm::Scanner, 817

gdcm::StrictScanner, 909
tags
gdcm::Tag, 974
TestAbortOff
gdcm::SimpleSubjectWatcher, 875
TestAbortOn
gdcm::SimpleSubjectWatcher, 875
TestPBKDF2
gdcm::ASN1, 123
Testing
gdcm::Testing, 978
TestsList.txt, 1519
ThreadedExecute
vtkImageRGBToYBR, 1212
vtkImageYBRToRGB, 1214
ThreadedRequestData
vtkGDCMThreadedImageReader2, 1183
vtkImageMapToColors16, 1203
vtkImageMapToWindowLevelColors2, 1207
to_string
gdcm, 72
ToPyObject
gdcm::PythonFilter, 774
ToString
gdcm::StringFilter, 921
ToStringPair
gdcm::StringFilter, 921, 922
ToUnixSlashes
gdcm::Filename, 421
ToWindowsSlashes
gdcm::Filename, 421
Trace
gdcm::Trace, 985
TransferSyntax
gdcm::TransferSyntax, 991
TransferSyntaxArrayType
gdcm::PresentationContext, 742
TransferSyntaxStringsType
gdcm::UIDs, 1014
TransferSyntaxSub
gdcm::network::TransferSyntaxSub, 994
TransferSyntaxes
gdcm::PresentationContext, 744
Transition
gdcm::network::Transition, 997
transitions
gdcm::network::TableRow, 965
Trim
gdcm::String, 918
TrimInternal
gdcm::CodeString, 231
Truncate
gdcm::String, 918
TryJPEG2000Codec
gdcm::Bitmap, 187
gdcm::ImageChangeTransferSyntax, 481
TryJPEG2000Codec2
gdcm::Bitmap, 187
TryJPEGCodec
gdcm::Bitmap, 188
gdcm::ImageChangeTransferSyntax, 481
TryJPEGCodec2
gdcm::Bitmap, 188
TryJPEGLSCodec
gdcm::Bitmap, 188
gdcm::ImageChangeTransferSyntax, 481
TryKAKADUCodec
gdcm::Bitmap, 188
TryPVRGCodec
gdcm::Bitmap, 188
TryRAWCodec
gdcm::Bitmap, 188
gdcm::ImageChangeTransferSyntax, 481
TryRLECodec
gdcm::Bitmap, 188
gdcm::ImageChangeTransferSyntax, 481
TS
gdcm::Bitmap, 190
Type
gdcm::Element, 353
gdcm::Element< TVR, VM::VM1_n >, 358
gdcm::Type, 999
gdcm::VL, 1110
TypeType
gdcm::Type, 999

UIComp
gdcm, 59
UIDGenerator
gdcm::UIDGenerator, 1002
ULAction
gdcm::network::ULAction, 1031
ULActionAE6
gdcm::network::ULConnection, 1073
ULBasicCallback
gdcm::network::ULBasicCallback, 1068
ULConnection
gdcm::network::ULConnection, 1071
ULConnectionCallback
gdcm::network::ULConnectionCallback, 1075
ULConnectionInfo
gdcm::network::ULConnectionInfo, 1077
ULConnectionManager
gdcm::network::ULConnection, 1073
gdcm::network::ULConnectionManager, 1080
ULEvent
gdcm::network::ULEvent, 1085
ULTransitionTable

gdcm::network::ULTransitionTable, 1087
ULWritingCallback
gdcm::network::ULWritingCallback, 1089
UTComp
gdcm, 59
UnInstallPipeline
vtkImageColorViewer, 1196
UnRegister
gdcm::Object, 660
Unpack
gdcm::Unpacker12Bits, 1095
Update
gdcm::Curve, 277
gdcm::Overlay, 681
UpdateDisplayExtent
vtkImageColorViewer, 1196
UpdateOrientation
vtkImageColorViewer, 1196
UpdatePosition
gdcm::ByteBuffer, 198
Usage
gdcm::Usage, 1097
UsageType
gdcm::Usage, 1096
UseDictAlways
gdcm::PythonFilter, 774
gdcm::StringFilter, 922
UserInformation
gdcm::network::UserInformation, 1099
UserOrdering
gdcm::SerieHelper, 858

V
gdcm::Validate, 1103
VIEWType
gdcm::Surface, 929
VMType
gdcm::VM, 1115
VRBINARY
gdcm, 73
VRDS16ILLEGAL
gdcmElement.h, 1315
VRField
gdcm::CSAElement, 257
gdcm::DataElement, 289
VRType
gdcm::VR, 1120
VRTypeTemplateCase
gdcmVR.h, 1512
VTK_CMYK
vtkGDCMImageReader.h, 1520
vtkGDCMImageReader2.h, 1521
VTK_INVERSE_LUMINANCE
vtkGDCMImageReader.h, 1520
vtkGDCMImageReader2.h, 1521
VTK_LEGACY
vtkImageColorViewer, 1196, 1197
VTK_LOOKUP_TABLE
vtkGDCMImageReader.h, 1520
vtkGDCMImageReader2.h, 1521
VTK_YBR
vtkGDCMImageReader.h, 1520
vtkGDCMImageReader2.h, 1521
Valid
gdcm::Preamble, 740
ValidDataSet
gdcm::BaseQuery, 167
Validate
gdcm::PixelFormat, 719
gdcm::Validate, 1103
ValidateQuery
gdcm::BaseQuery, 167
gdcm::BaseRootQuery, 171
gdcm::FindPatientRootQuery, 440
gdcm::FindStudyRootQuery, 443
gdcm::ModalityPerformedProcedureStepCreate↩
Query, 614
gdcm::ModalityPerformedProcedureStepSetQuery,
616
gdcm::MovePatientRootQuery, 629
gdcm::MoveStudyRootQuery, 632
gdcm::WLMFindQuery, 1230
Validation
gdcm::Validate, 1103
Value
gdcm::Value, 1105
value
gdcm::SerieHelper::Rule, 813
value_type
gdcm::CodeString, 230
gdcm::LO, 577
gdcm::String, 917
ValueField
gdcm::DataElement, 289
gdcm::PDBElement, 693
ValueLengthField
gdcm::DataElement, 289
ValueMultiplicityField
gdcm::CSAElement, 257
ValuePtr
gdcm::DataElement, 281
ValuesType
gdcm::Scanner, 817
gdcm::StrictScanner, 909
Verify
gdcm::Defs, 314
gdcm::Macro, 588
gdcm::Module, 620

Version
gdcm::Version, 1108
VL
gdcm::VL, 1111
VM
gdcm::VM, 1116
VR
gdcm::VR, 1121
vtkBooleanMacro
vtkGDCMImageReader, 1135
vtkGDCMImageReader2, 1147
vtkGDCMImageWriter, 1158
vtkGDCMThreadedImageReader, 1179
vtkGDCMThreadedImageReader2, 1183, 1184
vtkImageColorViewer, 1197
vtkImageMapToColors16, 1203
vtkGDCMImageReader, 1129
∼vtkGDCMImageReader, 1132
ApplyInverseVideo, 1139
ApplyLookupTable, 1139
ApplyPlanarConfiguration, 1139
ApplyShiftScale, 1139
ApplyYBRToRGB, 1139
CanReadFile, 1132
Curve, 1139
DirectionCosines, 1140
ExecuteData, 1132
ExecuteInformation, 1132
FileNames, 1140
FillMedicalImageInformation, 1132
ForceRescale, 1140
GetDescriptiveName, 1133
GetFileExtensions, 1133
GetIconImage, 1133
GetOverlay, 1133
IconDataScalarType, 1140
IconImageDataExtent, 1140
IconNumberOfScalarComponents, 1140
ImageFormat, 1140
ImageOrientationPatient, 1140
ImagePositionPatient, 1140
LoadIconImage, 1140
LoadOverlays, 1141
LoadSingleFile, 1133
LossyFlag, 1141
MedicalImageProperties, 1141
New, 1133
NumberOfIconImages, 1141
NumberOfOverlays, 1141
PlanarConfiguration, 1141
PrintSelf, 1133
RequestDataCompat, 1134
RequestInformationCompat, 1134
Scale, 1141
SetCurve, 1134
SetFileNames, 1134
SetFilePattern, 1134
SetFilePrefix, 1134
SetMedicalImageProperties, 1134
Shift, 1141
vtkBooleanMacro, 1135
vtkGDCMImageReader, 1132
vtkGDCMMedicalImageProperties, 1164
vtkGetMacro, 1135–1137
vtkGetObjectMacro, 1137
vtkGetStringMacro, 1137, 1138
vtkGetVector3Macro, 1138
vtkGetVector6Macro, 1138
vtkSetMacro, 1138
vtkSetVector6Macro, 1139
vtkTypeRevisionMacro, 1139
vtkGDCMImageReader.h, 1519
VTK_CMYK, 1520
VTK_INVERSE_LUMINANCE, 1520
VTK_LOOKUP_TABLE, 1520
VTK_YBR, 1520
vtkGDCMImageReader2, 1142
∼vtkGDCMImageReader2, 1144
ApplyInverseVideo, 1151
ApplyLookupTable, 1151
ApplyPlanarConfiguration, 1151
ApplyShiftScale, 1151
ApplyYBRToRGB, 1151
CanReadFile, 1144
Curve, 1151
DirectionCosines, 1152
FillMedicalImageInformation, 1144
ForceRescale, 1152
GetDescriptiveName, 1145
GetFileExtensions, 1145
GetIconImage, 1145
GetIconImagePort, 1145
GetOverlay, 1145
GetOverlayPort, 1145
IconDataScalarType, 1152
IconImageDataExtent, 1152
IconNumberOfScalarComponents, 1152
ImageFormat, 1152
ImageOrientationPatient, 1152
ImagePositionPatient, 1152
LoadIconImage, 1152
LoadOverlays, 1152
LoadSingleFile, 1145
LossyFlag, 1153
New, 1145
NumberOfIconImages, 1153
NumberOfOverlays, 1153
PlanarConfiguration, 1153

PrintSelf, 1146
ProcessRequest, 1146
RequestData, 1146
RequestDataCompat, 1146
RequestInformation, 1146
RequestInformationCompat, 1146
Scale, 1153
SetCurve, 1146
SetFilePattern, 1147
SetFilePrefix, 1147
SetMedicalImageProperties, 1147
Shift, 1153
vtkBooleanMacro, 1147
vtkGDCMImageReader2, 1144
vtkGDCMMedicalImageProperties, 1164
vtkGetMacro, 1148, 1149
vtkGetObjectMacro, 1149
vtkGetStringMacro, 1149, 1150
vtkGetVector3Macro, 1150
vtkGetVector6Macro, 1150
vtkSetMacro, 1150
vtkSetVector6Macro, 1151
vtkTypeRevisionMacro, 1151
vtkGDCMImageReader2.h, 1520
VTK_CMYK, 1521
VTK_INVERSE_LUMINANCE, 1521
VTK_LOOKUP_TABLE, 1521
VTK_YBR, 1521
vtkGDCMImageWriter, 1154
~vtkGDCMImageWriter, 1156
CompressionTypes, 1156
GetDescriptiveName, 1156
GetFileExtensions, 1156
GetFileName, 1156
New, 1157
PrintSelf, 1157
SetDirectionCosines, 1157
SetDirectionCosinesFromImageOrientationPatient, 1157
SetFileNames, 1157
SetMedicalImageProperties, 1157
vtkBooleanMacro, 1158
vtkGDCMImageWriter, 1156
vtkGDCMMedicalImageProperties, 1164
vtkGetMacro, 1158, 1159
vtkGetObjectMacro, 1159
vtkGetStringMacro, 1159, 1160
vtkSetMacro, 1160, 1161
vtkSetStringMacro, 1161
vtkTypeRevisionMacro, 1161
Write, 1161
WriteGDCMData, 1161
WriteSlice, 1161
vtkGDCMImageWriter.h, 1522
vtkGDCMMedicalImageProperties, 1162
~vtkGDCMMedicalImageProperties, 1163
Clear, 1163
GetFile, 1163
New, 1164
PrintSelf, 1164
PushBackFile, 1164
vtkGDCMImageReader, 1164
vtkGDCMImageReader2, 1164
vtkGDCMImageWriter, 1164
vtkGDCMMedicalImageProperties, 1163
vtkTypeRevisionMacro, 1164
vtkGDCMMedicalImageProperties.h, 1522
vtkGDCMPolyDataReader, 1165
~vtkGDCMPolyDataReader, 1166
FileName, 1168
FillMedicalImageInformation, 1166
MedicalImageProperties, 1168
New, 1166
PrintSelf, 1167
RTStructSetProperties, 1169
RequestData, 1167
RequestData_HemodynamicWaveformStorage, 1167
RequestData_RTStructureSetStorage, 1167
RequestInformation, 1167
RequestInformation_HemodynamicWaveform↩ Storage, 1167
RequestInformation_RTStructureSetStorage, 1167
vtkGDCMPolyDataReader, 1166
vtkGetObjectMacro, 1168
vtkGetStringMacro, 1168
vtkSetStringMacro, 1168
vtkTypeRevisionMacro, 1168
vtkGDCMPolyDataReader.h, 1523
vtkGDCMPolyDataWriter, 1169
~vtkGDCMPolyDataWriter, 1171
InitializeRTStructSet, 1171
MedicalImageProperties, 1173
New, 1171
PrintSelf, 1171
RTStructSetProperties, 1173
SetMedicalImageProperties, 1171
SetNumberOfInputPorts, 1172
SetRTStructSetProperties, 1172
vtkGDCMPolyDataWriter, 1171
vtkTypeRevisionMacro, 1172
WriteData, 1172
WriteRTSTRUCTData, 1172
WriteRTSTRUCTInfo, 1172
vtkGDCMPolyDataWriter.h, 1523
vtkGDCMTesting, 1173
~vtkGDCMTesting, 1175
GetGDCMDataRoot, 1175

GetMD5MetaImage, 1175
GetMHDMD5FromFile, 1175
GetNumberOfMD5MetaImages, 1175
GetRAWMD5FromFile, 1176
GetVTKDataRoot, 1176
MD5MetaImagesType, 1175
New, 1176
PrintSelf, 1176
vtkGDCMTesting, 1175
vtkTypeRevisionMacro, 1176
vtkGDCMTesting.h, 1524
vtkGDCMThreadedImageReader, 1177
~vtkGDCMThreadedImageReader, 1178
ExecuteData, 1178
ExecuteInformation, 1178
New, 1178
PrintSelf, 1179
ReadFiles, 1179
RequestDataCompat, 1179
vtkBooleanMacro, 1179
vtkGDCMThreadedImageReader, 1178
vtkGetMacro, 1179
vtkSetMacro, 1179, 1180
vtkTypeRevisionMacro, 1180
vtkGDCMThreadedImageReader.h, 1524
vtkGDCMThreadedImageReader2, 1180
~vtkGDCMThreadedImageReader2, 1182
GetFileName, 1182
New, 1182
PrintSelf, 1182
RequestInformation, 1182
SetFileName, 1183
SetFileNames, 1183
SplitExtent, 1183
ThreadedRequestData, 1183
vtkBooleanMacro, 1183, 1184
vtkGDCMThreadedImageReader2, 1182
vtkGetMacro, 1184, 1185
vtkGetObjectMacro, 1185
vtkGetVector3Macro, 1185
vtkGetVector6Macro, 1185
vtkSetMacro, 1185, 1186
vtkSetVector3Macro, 1186, 1187
vtkSetVector6Macro, 1187
vtkTypeRevisionMacro, 1187
vtkGDCMThreadedImageReader2.h, 1525
vtkGetMacro
vtkGDCMImageReader, 1135–1137
vtkGDCMImageReader2, 1148, 1149
vtkGDCMImageWriter, 1158, 1159
vtkGDCMThreadedImageReader, 1179
vtkGDCMThreadedImageReader2, 1184, 1185
vtkImageColorViewer, 1197
vtkImageMapToColors16, 1203
vtkImageMapToWindowLevelColors2, 1207
vtkGetObjectMacro
vtkGDCMImageReader, 1137
vtkGDCMImageReader2, 1149
vtkGDCMImageWriter, 1159
vtkGDCMPolyDataReader, 1168
vtkGDCMThreadedImageReader2, 1185
vtkImageColorViewer, 1197, 1198
vtkImageMapToColors16, 1203
vtkGetStringMacro
vtkGDCMImageReader, 1137, 1138
vtkGDCMImageReader2, 1149, 1150
vtkGDCMImageWriter, 1159, 1160
vtkGDCMPolyDataReader, 1168
vtkRTStructSetProperties, 1223, 1224
vtkGetVector3Macro
vtkGDCMImageReader, 1138
vtkGDCMImageReader2, 1150
vtkGDCMThreadedImageReader2, 1185
vtkGetVector6Macro
vtkGDCMImageReader, 1138
vtkGDCMImageReader2, 1150
vtkGDCMThreadedImageReader2, 1185
vtkImageColorViewer, 1187
~vtkImageColorViewer, 1190
AddInput, 1191
AddInputConnection, 1191
FirstRender, 1198
GetColorLevel, 1191
GetColorWindow, 1191
GetInput, 1191
GetOffScreenRendering, 1191
GetOverlayVisibility, 1191
GetPosition, 1191
GetSize, 1191
GetSliceMax, 1191
GetSliceMin, 1192
GetSliceRange, 1192
GetWindowName, 1192
ImageActor, 1198
InstallPipeline, 1192
Interactor, 1198
InteractorStyle, 1199
New, 1192
OverlayImageActor, 1199
PrintSelf, 1192
Render, 1193
RenderWindow, 1199
Renderer, 1199
SetColorLevel, 1193
SetColorWindow, 1193
SetDisplayId, 1193
SetInput, 1193
SetInputConnection, 1193

SetOffScreenRendering, 1193
SetOverlayVisibility, 1194
SetParentId, 1194
SetPosition, 1194
SetRenderWindow, 1194
SetRenderer, 1194
SetSize, 1194, 1195
SetSlice, 1195
SetSliceOrientation, 1195
SetSliceOrientationToXY, 1195
SetSliceOrientationToXZ, 1195
SetSliceOrientationToYZ, 1195
SetWindowId, 1196
SetupInteractor, 1196
Slice, 1199
SliceOrientation, 1199
UnInstallPipeline, 1196
UpdateDisplayExtent, 1196
UpdateOrientation, 1196
VTK_LEGACY, 1196, 1197
vtkBooleanMacro, 1197
vtkGetMacro, 1197
vtkGetObjectMacro, 1197, 1198
vtkImageColorViewer, 1190
vtkImageColorViewerCallback, 1198
vtkTypeRevisionMacro, 1198
WindowLevel, 1199
vtkImageColorViewer.h, 1526
vtkImageColorViewerCallback
vtkImageColorViewer, 1198
vtkImageMapToColors16, 1200
~vtkImageMapToColors16, 1201
ActiveComponent, 1204
DataWasPassed, 1204
GetMTime, 1201
LookupTable, 1204
New, 1201
OutputFormat, 1204
PassAlphaToOutput, 1204
PrintSelf, 1202
RequestData, 1202
RequestInformation, 1202
SetLookupTable, 1202
SetOutputFormatToLuminance, 1202
SetOutputFormatToLuminanceAlpha, 1202
SetOutputFormatToRGBA, 1202
SetOutputFormatToRGB, 1202
ThreadedRequestData, 1203
vtkBooleanMacro, 1203
vtkGetMacro, 1203
vtkGetObjectMacro, 1203
vtkImageMapToColors16, 1201
vtkSetMacro, 1203, 1204
vtkTypeRevisionMacro, 1204
vtkImageMapToColors16.h, 1526
vtkImageMapToWindowLevelColors2, 1205
~vtkImageMapToWindowLevelColors2, 1206
Level, 1208
New, 1206
PrintSelf, 1206
RequestData, 1207
RequestInformation, 1207
ThreadedRequestData, 1207
vtkGetMacro, 1207
vtkImageMapToWindowLevelColors2, 1206
vtkSetMacro, 1207, 1208
vtkTypeRevisionMacro, 1208
Window, 1208
vtkImageMapToWindowLevelColors2.h, 1527
vtkImagePlanarComponentsToComponents, 1209
~vtkImagePlanarComponentsToComponents, 1210
New, 1210
PrintSelf, 1210
RequestData, 1210
vtkImagePlanarComponentsToComponents, 1210
vtkTypeRevisionMacro, 1210
vtkImagePlanarComponentsToComponents.h, 1527
vtkImageRGBToYBR.h, 1528
vtkImageRGBToYBR, 1211
~vtkImageRGBToYBR, 1212
New, 1212
PrintSelf, 1212
ThreadedExecute, 1212
vtkImageRGBToYBR, 1212
vtkTypeRevisionMacro, 1212
vtkImageYBRToRGB.h, 1528
vtkImageYBRToRGB, 1213
~vtkImageYBRToRGB, 1214
New, 1214
PrintSelf, 1214
ThreadedExecute, 1214
vtkImageYBRToRGB, 1214
vtkTypeRevisionMacro, 1214
vtkLookupTable16, 1215
~vtkLookupTable16, 1216
Build, 1216
GetPointer, 1216
MapScalarsThroughTable2, 1216
New, 1216
PrintSelf, 1217
SetNumberOfTableValues, 1217
Table16, 1217
vtkLookupTable16, 1216
vtkTypeRevisionMacro, 1217
WritePointer, 1217
vtkLookupTable16.h, 1529
vtkRTStructSetProperties, 1218
~vtkRTStructSetProperties, 1220

AddContourReferencedFrameOfReference, 1220
AddReferencedFrameOfReference, 1220
AddStructureSetROIObservation, 1221
AddStructureSetROI, 1220
Clear, 1221
DeepCopy, 1221
GetContourReferencedFrameOfReferenceClassUID, 1221
GetContourReferencedFrameOfReferenceInstanceUID, 1221
GetNumberOfContourReferencedFrameOfReferences, 1221
GetNumberOfReferencedFrameOfReferences, 1222
GetNumberOfStructureSetROIs, 1222
GetReferencedFrameOfReferenceClassUID, 1222
GetReferencedFrameOfReferenceInstanceUID, 1222
GetStructureSetObservationNumber, 1222
GetStructureSetROIDescription, 1222
GetStructureSetROIGenerationAlgorithm, 1222
GetStructureSetROIName, 1222
GetStructureSetROINumber, 1223
GetStructureSetROIObservationLabel, 1223
GetStructureSetROIRefFrameRefUID, 1223
GetStructureSetRTROIInterpretedType, 1223
Internals, 1226
New, 1223
PrintSelf, 1223
ReferenceFrameOfReferenceUID, 1226
ReferenceSeriesInstanceUID, 1226
SOPInstanceUID, 1226
SeriesInstanceUID, 1226
StructureSetDate, 1226
StructureSetLabel, 1227
StructureSetName, 1227
StructureSetTime, 1227
StudyInstanceUID, 1227
vtkGetStringMacro, 1223, 1224
vtkRTStructSetProperties, 1220
vtkSetStringMacro, 1225, 1226
vtkTypeRevisionMacro, 1226
vtkRTStructSetProperties.h, 1529
vtkSetMacro
vtkGDCMImageReader, 1138
vtkGDCMImageReader2, 1150
vtkGDCMImageWriter, 1160, 1161
vtkGDCMThreadedImageReader, 1179, 1180
vtkGDCMThreadedImageReader2, 1185, 1186
vtkImageMapToColors16, 1203, 1204
vtkImageMapToWindowLevelColors2, 1207, 1208
vtkSetStringMacro
vtkGDCMImageWriter, 1161
vtkGDCMPolyDataReader, 1168
vtkRTStructSetProperties, 1225, 1226
vtkSetVector3Macro
vtkGDCMThreadedImageReader2, 1186, 1187
vtkSetVector6Macro
vtkGDCMImageReader, 1139
vtkGDCMImageReader2, 1151
vtkGDCMThreadedImageReader2, 1187
vtkTypeRevisionMacro
vtkGDCMImageReader, 1139
vtkGDCMImageReader2, 1151
vtkGDCMImageWriter, 1161
vtkGDCMMedicalImageProperties, 1164
vtkGDCMPolyDataReader, 1168
vtkGDCMPolyDataWriter, 1172
vtkGDCMTesting, 1176
vtkGDCMThreadedImageReader, 1180
vtkGDCMThreadedImageReader2, 1187
vtkImageColorViewer, 1198
vtkImageMapToColors16, 1204
vtkImageMapToWindowLevelColors2, 1208
vtkImagePlanarComponentsToComponents, 1210
vtkImageRGBToYBR, 1212
vtkImageYBRToRGB, 1214
vtkLookupTable16, 1217
vtkRTStructSetProperties, 1226

WLMFindQuery
gdcm::WLMFindQuery, 1229
WarningOff
gdcm::Trace, 988
WarningOn
gdcm::Trace, 988
Waveform
gdcm::Waveform, 1228
what
gdcm::Exception, 384
Window
vtkImageMapToWindowLevelColors2, 1208
WindowLevel
vtkImageColorViewer, 1199
Write
gdcm::ByteValue, 208
gdcm::CSAHeader, 262
gdcm::CommandDataSet, 236
gdcm::DataElement, 288
gdcm::DataSet, 304
gdcm::Element, 355
gdcm::Element< TVR, VM::VM1_n >, 361
gdcm::EncodingImplementation< VR::VRASCII >, 375, 376
gdcm::EncodingImplementation< VR::VRBINARY >, 377
gdcm::ExplicitDataElement, 387
gdcm::File, 394
gdcm::FileAnonymizer, 398
gdcm::FileMetaInformation, 418

gdcm::Fragment, 446
gdcm::ImageWriter, 516
gdcm::ImplicitDataElement, 522
gdcm::Item, 540
gdcm::PGXCodec, 707
gdcm::PNMCodec, 737
gdcm::PixmapWriter, 734
gdcm::Preamble, 740
gdcm::SegmentWriter, 838
gdcm::SequenceOfFragments, 845
gdcm::SequenceOfItems, 853
gdcm::StreamImageWriter, 903
gdcm::SurfaceWriter, 947
gdcm::Tag, 973
gdcm::VRVLSize< 0 >, 1128
gdcm::VRVLSize< 1 >, 1129
gdcm::ValueIO, 1107
gdcm::VL, 1112
gdcm::VR, 1124
gdcm::Writer, 1236
gdcm::network::AAbortPDU, 85
gdcm::network::AAssociateACPDU, 89
gdcm::network::AAssociateRJPDU, 91
gdcm::network::AAssociateRQPDU, 97
gdcm::network::AReleaseRPPDU, 118
gdcm::network::AReleaseRQPDU, 120
gdcm::network::AbstractSyntax, 100
gdcm::network::ApplicationContext, 113
gdcm::network::AsynchronousOperationsWindowSub, 124
gdcm::network::BasePDU, 163
gdcm::network::ImplementationClassUIDSub, 517
gdcm::network::ImplementationUIDSub, 518
gdcm::network::ImplementationVersionNameSub, 519
gdcm::network::MaximumLengthSub, 592
gdcm::network::PDataTFPDU, 691
gdcm::network::PresentationContextAC, 746
gdcm::network::PresentationContextRQ, 753
gdcm::network::PresentationDataValue, 756
gdcm::network::RoleSelectionSub, 812
gdcm::network::SOPClassExtendedNegociationSub, 881
gdcm::network::ServiceClassApplicationInformation, 860
gdcm::network::TransferSyntaxSub, 995
gdcm::network::UserInformation, 1101
vtkGDCMImageWriter, 1161
Write16
gdcm::VL, 1112
WriteASCII
gdcm::Element< TVR, VM::VM1_n >, 361
WriteBuffer
gdcm::ByteValue, 209
gdcm::SequenceOfFragments, 845
WriteBufferAsRGBA
gdcm::LookupTable, 584
WriteData
vtkGDCMPolyDataWriter, 1172
WriteFooter
gdcm::DictConverter, 328
WriteGDCMData
vtkGDCMImageWriter, 1161
WriteHeader
gdcm::DictConverter, 328
WriteHelpFile
gdcm::BaseQuery, 167
WriteImageInformation
gdcm::StreamImageWriter, 903
WriteImageSubregionRAW
gdcm::StreamImageWriter, 904
WritePointer
vtkLookupTable16, 1217
WriteQuery
gdcm::BaseQuery, 167
WriteRTSTRUCTData
vtkGDCMPolyDataWriter, 1172
WriteRTSTRUCTInfo
vtkGDCMPolyDataWriter, 1172
WriteRawHeader
gdcm::StreamImageWriter, 904
WriteSlice
vtkGDCMImageWriter, 1161
Writer
gdcm::Writer, 1234

XMLDictReader
gdcm::XMLDictReader, 1238
XMLPrinter
gdcm::XMLPrinter, 1241
XMLPrivateDictReader
gdcm::XMLPrivateDictReader, 1244

YBR2RGB
gdcm::ImageChangePhotometricInterpretation, 472

ZSpacing
gdcm::IPPSorter, 536
ZTolerance
gdcm::IPPSorter, 536