GDCM

2.6.3

Generated by Doxygen 1.8.11

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 **37**

8.1 File List 37

9 Namespace Documentation **45**

9.1 gdcm Namespace Reference . . . 45

9.1.1 Detailed Description . . . 60

9.1.2 Typedef Documentation . . . 61

9.1.2.1 AEComp . . . 61

9.1.2.2 ASComp . . . 61

9.1.2.3 BOOL_FUNCTION_PFILE_PFILE_POINTER . . . 61

9.1.2.4 CSComp . . . 61

9.1.2.5 DAComp . . . 61

9.1.2.6 DTComp . . . 61

9.1.2.7 FileList . . . 61

9.1.2.8 IconImage . . . 61

9.1.2.9 LOComp . . . 61

9.1.2.10 LTComp . . . 61

9.1.2.11 MacroEntry . . . 61

9.1.2.12 NestedMacroEntries . . . 61

9.1.2.13 PNComp . . . 61

9.1.2.14 SHComp . . . 61

9.1.2.15 STComp . . . 61

9.1.2.16 TMComp . . . 61

9.1.2.17 UIComp . . . 61

9.1.2.18 UTComp . . . 61

9.1.3 Enumeration Type Documentation . . . 61

9.1.3.1 CompOperators . . . 61

9.1.3.2 ECharSet . . . 62

9.1.3.3 ENQueryType . . . 62

9.1.3.4 EQueryLevel . . . 62

9.1.3.5 EQueryType . . . 63

9.1.3.6 ERootType . . . 63
9.1.3.7 LodModeType . . . 63
9.1.4 Function Documentation . . . 63
9.1.4.1 backslash('\\') . . . 63
9.1.4.2 GetVRFromTag(Tag const &tag) . . . 63
9.1.4.3 operator"!=(const CodeString &ref, const CodeString &cs) . . . 63
9.1.4.4 operator"!=(const DataElement &lhs, const DataElement &rhs) . . . 63
9.1.4.5 operator<<(std::ostream &os, const Version &v) . . . 63
9.1.4.6 operator<<(std::ostream &_os, const NestedModuleEntries &_val) . . . 64
9.1.4.7 operator<<(std::ostream &os, const SwapCode &sc) . . . 64
9.1.4.8 operator<<(std::ostream &os, const FileSet &f) . . . 64
9.1.4.9 operator<<(std::ostream &os, const Region &r) . . . 64
9.1.4.10 operator<<(std::ostream &os, Event &e) . . . 64
9.1.4.11 operator<<(std::ostream &os, const PDBElement &val) . . . 64
9.1.4.12 operator<<(std::ostream &os, const CommandDataSet &val) . . . 64
9.1.4.13 operator<<(std::ostream &os, const Orientation &o) . . . 64
9.1.4.14 operator<<(std::ostream &_os, const IODs &_val) . . . 64
9.1.4.15 operator<<(std::ostream &_os, const Macros &_val) . . . 64
9.1.4.16 operator<<(std::ostream &_os, const Modules &_val) . . . 64
9.1.4.17 operator<<(std::ostream &_os, const Type &val) . . . 64
9.1.4.18 operator<<(std::ostream &_os, const ModuleEntry &_val) . . . 65
9.1.4.19 operator<<(std::ostream &_os, const GroupDict &_val) . . . 65
9.1.4.20 operator<<(std::ostream &os, const PrivateTag &val) . . . 65
9.1.4.21 operator<<(std::ostream &_os, const IOD &_val) . . . 65
9.1.4.22 operator<<(std::ostream &os, const File &val) . . . 65
9.1.4.23 operator<<(std::ostream &_os, const Usage &val) . . . 65
9.1.4.24 operator<<(std::ostream &os, const Sorter &s) . . . 65
9.1.4.25 operator<<(std::ostream &os, const CSAHeaderDictEntry &val) . . . 65

9.1.4.26 operator<<(std::ostream &os, const Preamble &val) . . . 65
9.1.4.27 operator<<(std::ostream &_os, const IODEntry &_val) . . . 65
9.1.4.28 operator<<(std::ostream &_os, const Macro &_val) . . . 65
9.1.4.29 operator<<(std::ostream &os, const CSAHeaderDict &val) . . . 65
9.1.4.30 operator<<(std::ostream &os, const Dicts &d) . . . 65
9.1.4.31 operator<<(std::ostream &os, const PDBHeader &d) . . . 65
9.1.4.32 operator<<(std::ostream &os, const CodeString &str) . . . 66
9.1.4.33 operator<<(std::ostream &os, const Directory &d) . . . 66
9.1.4.34 operator<<(std::ostream &_os, const Module &_val) . . . 66
9.1.4.35 operator<<(std::ostream &os, const PhotometricInterpretation &val) . . . 66
9.1.4.36 operator<<(std::ostream &os, const Global &g) . . . 66
9.1.4.37 operator<<(std::ostream &os, const Object &obj) . . . 66
9.1.4.38 operator<<(std::ostream &os, const BasicOffsetTable &val) . . . 66
9.1.4.39 operator<<(std::ostream &os, const DictEntry &val) . . . 66
9.1.4.40 operator<<(std::ostream &os, const VL &val) . . . 66
9.1.4.41 operator<<(std::ostream &os, const CSAElement &val) . . . 66
9.1.4.42 operator<<(std::ostream &os, const CSAHeader &d) . . . 67
9.1.4.43 operator<<(std::ostream &os, const FileMetaInformation &val) . . . 67
9.1.4.44 operator<<(std::ostream &_os, const TransferSyntax &ts) . . . 67
9.1.4.45 operator<<(std::ostream &_os, const VM &_val) . . . 67
9.1.4.46 operator<<(std::ostream &os, const StrictScanner &s) . . . 67
9.1.4.47 operator<<(std::ostream &os, const Scanner &s) . . . 67
9.1.4.48 operator<<(std::ostream &os, const Dict &val) . . . 67
9.1.4.49 operator<<(std::ostream &_os, const MediaStorage &ms) . . . 67
9.1.4.50 operator<<(std::ostream &_os, const VR &val) . . . 67
9.1.4.51 operator<<(std::ostream &os, const Fragment &val) . . . 67
9.1.4.52 operator<<(std::ostream &os, const PixelFormat &pf) . . . 68
9.1.4.53 operator<<(std::ostream &_os, const UI &_val) . . . 68

9.1.4.54 operator<<(std::ostream &os, const DataElement &val) . . . 68
9.1.4.55 operator<<(std::ostream &_os, const Tag &_val) . . . 68
9.1.4.56 operator<<(std::ostream &os, const Item &val) . . . 68
9.1.4.57 operator<<(std::ostream &os, const DataSet &val) . . . 68
9.1.4.58 operator<<(std::ostream &os, const PrivateDict &val) . . . 68
9.1.4.59 operator<<(std::ostream &_os, const UIDs &uid) . . . 68
9.1.4.60 operator==(const CodeString &ref, const CodeString &cs) . . . 68
9.1.4.61 operator>>(std::istream &is, String< TDelimiter, TMaxLength, TPadChar > &ms) . . . 69
9.1.4.62 operator>>(std::istream &in, ignore_char const &ic) . . . 69
9.1.4.63 operator>>(std::istream &_is, Tag &_val) . . . 69
9.1.4.64 to_string(Float data) . . . 69
9.1.4.65 TYPETOENCODING(SQ, VRBINARY, unsigned char) TYPETOENCODING(UN . . . 69
9.1.5 Variable Documentation . . . 69
9.1.5.1 GlobalInstance . . . 69
9.1.5.2 VRBINARY . . . 69
9.2 gdcm::network Namespace Reference . . . 69
9.2.1 Enumeration Type Documentation . . . 74
9.2.1.1 EEventID . . . 74
9.2.1.2 EStateID . . . 75
9.2.2 Function Documentation . . . 75
9.2.2.1 GetStateIndex(EStateID inState) . . . 75
9.2.3 Variable Documentation . . . 75
9.2.3.1 cMaxEventID . . . 75
9.2.3.2 cMaxStateID . . . 75
9.3 gdcm::SegmentHelper Namespace Reference . . . 76
9.4 gdcm::terminal Namespace Reference . . . 76
9.4.1 Detailed Description . . . 76
9.4.2 Enumeration Type Documentation . . . 77
9.4.2.1 Attribute . . . 77
9.4.2.2 Color . . . 77
9.4.2.3 Mode . . . 77
9.4.3 Function Documentation . . . 77
9.4.3.1 setattribute(Attribute att) . . . 77
9.4.3.2 setbgcolor(Color c) . . . 77
9.4.3.3 setfgcolor(Color c) . . . 77
9.4.3.4 setmode(Mode m) . . . 77

10 Class Documentation **79**

10.1 gdcm::network::AAbortPDU Class Reference . . . 79

10.1.1 Detailed Description . . . 80

10.1.2 Constructor & Destructor Documentation . . . 80

10.1.2.1 AAbortPDU() . . . 80

10.1.3 Member Function Documentation . . . 80

10.1.3.1 IsLastFragment() const . . . 80

10.1.3.2 Print(std::ostream &os) const . . . 80

10.1.3.3 Read(std::istream &is) . . . 80

10.1.3.4 SetReason(const uint8_t r) . . . 80

10.1.3.5 SetSource(const uint8_t s) . . . 80

10.1.3.6 Size() const . . . 80

10.1.3.7 Write(std::ostream &os) const . . . 81

10.2 gdcm::network::AAssociateACPDU Class Reference . . . 81

10.2.1 Detailed Description . . . 82

10.2.2 Member Typedef Documentation . . . 82

10.2.2.1 SizeType . . . 82

10.2.3 Constructor & Destructor Documentation . . . 82

10.2.3.1 AAssociateACPDU() . . . 82

10.2.4 Member Function Documentation . . . 82

10.2.4.1 AddPresentationContextAC(PresentationContextAC const &pcac) . . . 82

10.2.4.2 GetNumberOfPresentationContextAC() const . . . 82

10.2.4.3 GetPresentationContextAC(SizeType i) . . . 82

10.2.4.4 GetUserInformation() const . . . 82

10.2.4.5 InitFromRQ(AAssociateRQPDU const &rqpdu) . . . 83

10.2.4.6 IsLastFragment() const . . . 83

10.2.4.7 Print(std::ostream &os) const . . . 83

10.2.4.8 Read(std::istream &is) . . . 83

10.2.4.9 SetCalledAETitle(const char calledaetitle[16]) . . . 83
10.2.4.10 SetCallingAETitle(const char callingaetitle[16]) . . . 83
10.2.4.11 Size() const . . . 83
10.2.4.12 Write(std::ostream &os) const . . . 83
10.2.5 Friends And Related Function Documentation . . . 83
10.2.5.1 AAssociateRQPDU . . . 83
10.3 gdcm::network::AAssociateRJPDU Class Reference . . . 84
10.3.1 Detailed Description . . . 85
10.3.2 Constructor & Destructor Documentation . . . 85
10.3.2.1 AAssociateRJPDU() . . . 85
10.3.3 Member Function Documentation . . . 85
10.3.3.1 IsLastFragment() const . . . 85
10.3.3.2 Print(std::ostream &os) const . . . 85
10.3.3.3 Read(std::istream &is) . . . 85
10.3.3.4 Size() const . . . 85
10.3.3.5 Write(std::ostream &os) const . . . 85
10.4 gdcm::network::AAssociateRQPDU Class Reference . . . 86
10.4.1 Detailed Description . . . 87
10.4.2 Member Typedef Documentation . . . 88
10.4.2.1 PresentationContextArrayType . . . 88
10.4.2.2 SizeType . . . 88
10.4.3 Constructor & Destructor Documentation . . . 88
10.4.3.1 AAssociateRQPDU() . . . 88
10.4.3.2 AAssociateRQPDU(const AAssociateRQPDU &pdu) . . . 88
10.4.4 Member Function Documentation . . . 88
10.4.4.1 AddPresentationContext(PresentationContextRQ const &pc) . . . 88
10.4.4.2 GetCalledAETitle() const . . . 88
10.4.4.3 GetCallingAETitle() const . . . 88

10.4.4.4 GetNumberOfPresentationContext() const . . . 88
10.4.4.5 GetPresentationContext(SizeType i) const . . . 88
10.4.4.6 GetPresentationContextByAbstractSyntax(AbstractSyntax const &absyn) const . . . 88
10.4.4.7 GetPresentationContextByID(uint8_t i) const . . . 88
10.4.4.8 GetPresentationContexts() . . . 89
10.4.4.9 GetReserved43_74() const . . . 89
10.4.4.10 GetUserInformation() const . . . 89
10.4.4.11 IsAETitleValid(const char title[16]) . . . 89
10.4.4.12 IsLastFragment() const . . . 89
10.4.4.13 Print(std::ostream &os) const . . . 89
10.4.4.14 Read(std::istream &is) . . . 89
10.4.4.15 SetCalledAETitle(const char calledaetitle[16]) . . . 89
10.4.4.16 SetCallingAETitle(const char callingaetitle[16]) . . . 90
10.4.4.17 SetUserInformation(UserInformation const &ui) . . . 90
10.4.4.18 Size() const . . . 90
10.4.4.19 Write(std::ostream &os) const . . . 90
10.4.5 Friends And Related Function Documentation . . . 90
10.4.5.1 AAssociateACPDU . . . 90
10.5 gdcm::AbortEvent Class Reference . . . 90
10.6 gdcm::network::AbstractSyntax Class Reference . . . 91
10.6.1 Detailed Description . . . 92
10.6.2 Constructor & Destructor Documentation . . . 92
10.6.2.1 AbstractSyntax() . . . 92
10.6.3 Member Function Documentation . . . 92
10.6.3.1 GetAsDataElement() const . . . 92
10.6.3.2 GetName() const . . . 92
10.6.3.3 operator==(const AbstractSyntax &as) const . . . 92
10.6.3.4 Print(std::ostream &os) const . . . 92

10.6.3.5 Read(std::istream &is) . . . 92
10.6.3.6 SetName(const char *name) . . . 92
10.6.3.7 SetNameFromUID(UIDs::TSName tsname) . . . 92
10.6.3.8 Size() const . . . 92
10.6.3.9 Write(std::ostream &os) const . . . 93
10.7 gdcm::AnonymizeEvent Class Reference . . . 93
10.7.1 Detailed Description . . . 94
10.7.2 Member Typedef Documentation . . . 94
10.7.2.1 Self . . . 94
10.7.2.2 Superclass . . . 94
10.7.3 Constructor & Destructor Documentation . . . 94
10.7.3.1 AnonymizeEvent(Tag const &tag=0) . . . 94
10.7.3.2 ~AnonymizeEvent() . . . 94
10.7.3.3 AnonymizeEvent(const Self &s) . . . 94
10.7.4 Member Function Documentation . . . 94
10.7.4.1 CheckEvent(const ::gdcm::Event *e) const . . . 94
10.7.4.2 GetEventName() const . . . 94
10.7.4.3 GetTag() const . . . 95
10.7.4.4 MakeObject() const . . . 95
10.7.4.5 SetTag(const Tag &t) . . . 95
10.8 gdcm::Anonymizer Class Reference . . . 95
10.8.1 Detailed Description . . . 97
10.8.2 Constructor & Destructor Documentation . . . 98
10.8.2.1 Anonymizer() . . . 98
10.8.2.2 ~Anonymizer() . . . 98
10.8.3 Member Function Documentation . . . 98
10.8.3.1 BALCPProtect(DataSet &ds, Tag const &tag, const IOD &iod) . . . 98
10.8.3.2 BasicApplicationLevelConfidentialityProfile(bool deidentify=true) . . . 98

10.8.3.3 CanEmptyTag(Tag const &tag, const IOD &iod) const . . . 98
10.8.3.4 ClearInternalUIDs() . . . 98
10.8.3.5 Empty(Tag const &t) . . . 98
10.8.3.6 GetBasicApplicationLevelConfidentialityProfileAttributes() . . . 99
10.8.3.7 GetCryptographicMessageSyntax() const . . . 99
10.8.3.8 GetFile() . . . 99
10.8.3.9 New() . . . 99
10.8.3.10 RecurseDataSet(DataSet &ds) . . . 99
10.8.3.11 Remove(Tag const &t) . . . 99
10.8.3.12 RemoveGroupLength() . . . 99
10.8.3.13 RemovePrivateTags() . . . 99
10.8.3.14 RemoveRetired() . . . 99
10.8.3.15 Replace(Tag const &t, const char *value) . . . 100
10.8.3.16 Replace(Tag const &t, const char *value, VL const &vl) . . . 100
10.8.3.17 SetCryptographicMessageSyntax(CryptographicMessageSyntax *cms) . . . 100
10.8.3.18 SetFile(const File &f) . . . 100
10.9 gdcm::AnyEvent Class Reference . . . 100
10.10 gdcm::network::ApplicationContext Class Reference . . . 102
10.10.1 Detailed Description . . . 102
10.10.2 Constructor & Destructor Documentation . . . 103
10.10.2.1 ApplicationContext() . . . 103
10.10.3 Member Function Documentation . . . 103
10.10.3.1 GetName() const . . . 103
10.10.3.2 Print(std::ostream &os) const . . . 103
10.10.3.3 Read(std::istream &is) . . . 103
10.10.3.4 SetName(const char *name) . . . 103
10.10.3.5 Size() const . . . 103
10.10.3.6 Write(std::ostream &os) const . . . 103

10.11gdcm::ApplicationEntity Class Reference . . . 104
10.11.1 Detailed Description . . . 105
10.11.2 Member Function Documentation . . . 105
10.11.2.1 IsValid() const . . . 105
10.11.2.2 Print(std::ostream &os) const . . . 105
10.11.2.3 SetBlob(const std::vector< char > &v) . . . 105
10.11.2.4 Squeeze() . . . 105
10.11.3 Member Data Documentation . . . 105
10.11.3.1 Internal . . . 105
10.11.3.2 MaxLength . . . 105
10.11.3.3 MaxNumberOfComponents . . . 105
10.11.3.4 Padding . . . 105
10.11.3.5 Separator . . . 105
10.12gdcm::network::AReleaseRPPDU Class Reference . . . 106
10.12.1 Detailed Description . . . 107
10.12.2 Constructor & Destructor Documentation . . . 107
10.12.2.1 AReleaseRPPDU() . . . 107
10.12.3 Member Function Documentation . . . 107
10.12.3.1 IsLastFragment() const . . . 107
10.12.3.2 Print(std::ostream &os) const . . . 107
10.12.3.3 Read(std::istream &is) . . . 107
10.12.3.4 Size() const . . . 107
10.12.3.5 Write(std::ostream &os) const . . . 107
10.13gdcm::network::AReleaseRQPDU Class Reference . . . 108
10.13.1 Detailed Description . . . 109
10.13.2 Constructor & Destructor Documentation . . . 109
10.13.2.1 AReleaseRQPDU() . . . 109
10.13.3 Member Function Documentation . . . 109

10.13.3.1 IsLastFragment() const . . . 109
10.13.3.2 Print(std::ostream &os) const . . . 109
10.13.3.3 Read(std::istream &is) . . . 109
10.13.3.4 Size() const . . . 109
10.13.3.5 Write(std::ostream &os) const . . . 109
10.14gdcm::network::ARTIMTimer Class Reference . . . 109
10.14.1 Detailed Description . . . 110
10.14.2 Constructor & Destructor Documentation . . . 110
10.14.2.1 ARTIMTimer() . . . 110
10.14.3 Member Function Documentation . . . 110
10.14.3.1 GetElapsedTime() const . . . 110
10.14.3.2 GetHasExpired() const . . . 110
10.14.3.3 GetTimeout() const . . . 110
10.14.3.4 SetTimeout(double inTimeout) . . . 110
10.14.3.5 Start() . . . 110
10.14.3.6 Stop() . . . 110
10.15gdcm::ASN1 Class Reference . . . 111
10.15.1 Detailed Description . . . 111
10.15.2 Constructor & Destructor Documentation . . . 111
10.15.2.1 ASN1() . . . 111
10.15.2.2 ∼ASN1() . . . 111
10.15.3 Member Function Documentation . . . 111
10.15.3.1 ParseDump(const char ∗array, size_t length) . . . 111
10.15.3.2 ParseDumpFile(const char ∗filename) . . . 111
10.15.3.3 TestPBKDF2() . . . 111
10.16gdcm::network::AsynchronousOperationsWindowSub Class Reference . . . 112
10.16.1 Detailed Description . . . 112
10.16.2 Constructor & Destructor Documentation . . . 112

10.16.2.1 AsynchronousOperationsWindowSub() . . . 112
10.16.3 Member Function Documentation . . . 112
10.16.3.1 Print(std::ostream &os) const . . . 112
10.16.3.2 Read(std::istream &is) . . . 112
10.16.3.3 Size() const . . . 112
10.16.3.4 Write(std::ostream &os) const . . . 112
10.17 gdcm::Attribute< Group, Element, TVR, TVM > Class Template Reference . . . 113
10.17.1 Detailed Description . . . 114
10.17.2 Member Typedef Documentation . . . 115
10.17.2.1 ArrayType . . . 115
10.17.3 Member Enumeration Documentation . . . 115
10.17.3.1 anonymous enum . . . 115
10.17.4 Member Function Documentation . . . 115
10.17.4.1 GDCM_STATIC_ASSERT(((VR::VRType) TVR &(VR::VRType)(TagToType< Group, Element >::VRType))) . . . 115
10.17.4.2 GDCM_STATIC_ASSERT(((VM::VMType) TVM &(VM::VMType)(TagToType< Group, Element >::VMType))) . . . 115
10.17.4.3 GDCM_STATIC_ASSERT(((((VR::VRType) TVR &VR::VR_VM1)&&((VM::VMType) TVM==VM::VM1))"|"|"!((VR::VRType) TVR &VR::VR_VM1))) . . . 115
10.17.4.4 GetAsDataElement() const . . . 115
10.17.4.5 GetDictVM() . . . 115
10.17.4.6 GetDictVR() . . . 115
10.17.4.7 GetNumberOfValues() const . . . 115
10.17.4.8 GetTag() . . . 116
10.17.4.9 GetValue(unsigned int idx=0) . . . 116
10.17.4.10 GetValue(unsigned int idx=0) const . . . 116
10.17.4.11 GetValues() const . . . 116
10.17.4.12 GetVM() . . . 116
10.17.4.13 GetVR() . . . 116

10.17.4.14operator"!=(const Attribute &att) const 116
10.17.4.15operator<(const Attribute &att) const 116
10.17.4.16operator==(const Attribute &att) const 116
10.17.4.17operator[](unsigned int idx) 117
10.17.4.18operator[](unsigned int idx) const 117
10.17.4.19Print(std::ostream &os) const 117
10.17.4.20Set(DataSet const &ds) 117
10.17.4.21SetByteValue(const ByteValue *bv) 117
10.17.4.22SetByteValueNoSwap(const ByteValue *bv) 117
10.17.4.23SetFromDataElement(DataElement const &de) 117
10.17.4.24SetFromDataSet(DataSet const &ds) 117
10.17.4.25SetValue(ArrayType v, unsigned int idx=0) 118
10.17.4.26SetValues(const ArrayType *array, unsigned int numel=VMType) 118
10.17.5 Member Data Documentation 118
10.17.5.1 Internal 118
10.18gdcm::Attribute< Group, Element, TVR, VM::VM1 > Class Template Reference 118
10.18.1 Member Typedef Documentation 120
10.18.1.1 ArrayType 120
10.18.2 Member Enumeration Documentation 120
10.18.2.1 anonymous enum 120
10.18.3 Member Function Documentation 120
10.18.3.1 GDCM_STATIC_ASSERT(VMToLength< VM::VM1 >::Length==1) 120
10.18.3.2 GDCM_STATIC_ASSERT(((VR::VRType) TVR &(VR::VRType)(TagToType< Group, Element >::VRType))) 120
10.18.3.3 GDCM_STATIC_ASSERT(((VM::VMType) VM::VM1 &(VM::VMType)(TagToType< Group, Element >::VMType))) 120
10.18.3.4 GDCM_STATIC_ASSERT(((((VR::VRType) TVR &VR::VR_VM1)&&((VM::VMType) VM::VM1==VM::VM1))"|"|"!((VR::VRType) TVR &VR::VR_VM1))) 120
10.18.3.5 GetAsDataElement() const 120

10.18.3.6 GetDictVM() . . . 120
10.18.3.7 GetDictVR() . . . 120
10.18.3.8 GetNumberOfValues() const . . . 120
10.18.3.9 GetTag() . . . 121
10.18.3.10 GetValue() . . . 121
10.18.3.11 GetValue() const . . . 121
10.18.3.12 GetValues() const . . . 121
10.18.3.13 GetVM() . . . 121
10.18.3.14 GetVR() . . . 121
10.18.3.15 operator"!=(const Attribute &att) const . . . 121
10.18.3.16 operator<(const Attribute &att) const . . . 121
10.18.3.17 operator==(const Attribute &att) const . . . 121
10.18.3.18 Print(std::ostream &os) const . . . 121
10.18.3.19 Set(DataSet const &ds) . . . 121
10.18.3.20 SetByteValue(const ByteValue *bv) . . . 122
10.18.3.21 SetByteValueNoSwap(const ByteValue *bv) . . . 122
10.18.3.22 SetFromDataElement(DataElement const &de) . . . 122
10.18.3.23 SetFromDataSet(DataSet const &ds) . . . 122
10.18.3.24 SetValue(ArrayType v) . . . 122
10.18.4 Member Data Documentation . . . 122
10.18.4.1 Internal . . . 122
10.19 gdcm::Attribute< Group, Element, TVR, VM::VM1_3 > Class Template Reference . . . 123
10.19.1 Member Function Documentation . . . 124
10.19.1.1 GetVM() const . . . 124
10.20 gdcm::Attribute< Group, Element, TVR, VM::VM1_8 > Class Template Reference . . . 124
10.20.1 Member Function Documentation . . . 125
10.20.1.1 GetVM() const . . . 125
10.21 gdcm::Attribute< Group, Element, TVR, VM::VM1_n > Class Template Reference . . . 125

10.21.1 Member Typedef Documentation . . . 126
10.21.1.1 ArrayType . . . 126
10.21.2 Constructor & Destructor Documentation . . . 126
10.21.2.1 Attribute() . . . 126
10.21.2.2 ∼Attribute() . . . 127
10.21.3 Member Function Documentation . . . 127
10.21.3.1 GDCM_STATIC_ASSERT(((VR::VRType) TVR &(VR::VRType)(TagToType< Group, Element >::VRType))) . . . 127
10.21.3.2 GDCM_STATIC_ASSERT((VM::VM1_n &(VM::VMType)(TagToType< Group, Element >::VMType))) . . . 127
10.21.3.3 GDCM_STATIC_ASSERT(((((VR::VRType) TVR &VR::VR_VM1)&&((VM::VMType) TagToType< Group, Element >::VMType==VM::VM1))"|"|"!((VR::VRType) TVR &VR::VR_VM1))) . . . 127
10.21.3.4 GetAsDataElement() const . . . 127
10.21.3.5 GetDictVM() . . . 127
10.21.3.6 GetDictVR() . . . 127
10.21.3.7 GetNumberOfValues() const . . . 127
10.21.3.8 GetTag() . . . 127
10.21.3.9 GetValue(unsigned int idx=0) . . . 127
10.21.3.10 GetValue(unsigned int idx=0) const . . . 127
10.21.3.11 GetValues() const . . . 127
10.21.3.12 GetVM() . . . 127
10.21.3.13 GetVR() . . . 128
10.21.3.14 operator[](unsigned int idx) . . . 128
10.21.3.15 operator[](unsigned int idx) const . . . 128
10.21.3.16 Print(std::ostream &os) const . . . 128
10.21.3.17 Set(DataSet const &ds) . . . 128
10.21.3.18 SetByteValue(const ByteValue ∗bv) . . . 128
10.21.3.19 SetFromDataElement(DataElement const &de) . . . 128
10.21.3.20 SetFromDataSet(DataSet const &ds) . . . 128

10.21.3.21SetNumberOfValues(unsigned int numel) . . . 128
10.21.3.22SetValue(unsigned int idx, ArrayType v) . . . 128
10.21.3.23SetValue(ArrayType v) . . . 128
10.21.3.24SetValues(const ArrayType *array, unsigned int numel, bool own=false) . . . 129
10.22gdcm::Attribute< Group, Element, TVR, VM::VM2_2n > Class Template Reference . . . 129
10.22.1 Member Function Documentation . . . 130
10.22.1.1 GetVM() . . . 130
10.23gdcm::Attribute< Group, Element, TVR, VM::VM2_n > Class Template Reference . . . 130
10.23.1 Member Function Documentation . . . 132
10.23.1.1 GetVM() const . . . 132
10.24gdcm::Attribute< Group, Element, TVR, VM::VM3_3n > Class Template Reference . . . 132
10.24.1 Member Function Documentation . . . 133
10.24.1.1 GetVM() . . . 133
10.25gdcm::Attribute< Group, Element, TVR, VM::VM3_n > Class Template Reference . . . 134
10.25.1 Member Function Documentation . . . 135
10.25.1.1 GetVM() . . . 135
10.26gdcm::AudioCodec Class Reference . . . 135
10.26.1 Detailed Description . . . 136
10.26.2 Constructor & Destructor Documentation . . . 136
10.26.2.1 AudioCodec() . . . 136
10.26.2.2 ~AudioCodec() . . . 136
10.26.3 Member Function Documentation . . . 136
10.26.3.1 CanCode(TransferSyntax const &) const . . . 136
10.26.3.2 CanDecode(TransferSyntax const &) const . . . 137
10.26.3.3 Decode(DataElement const &is, DataElement &os) . . . 137
10.27gdcm::Base64 Class Reference . . . 137
10.27.1 Detailed Description . . . 137
10.27.2 Member Function Documentation . . . 137

10.27.2.1 Decode(char *dst, size_t dlen, const char *src, size_t slen) . . . 137
10.27.2.2 Encode(char *dst, size_t dlen, const char *src, size_t slen) . . . 138
10.27.2.3 GetDecodeLength(const char *src, size_t len) . . . 138
10.27.2.4 GetEncodeLength(const char *src, size_t srclen) . . . 138
10.28 gdcm::network::BaseCompositeMessage Class Reference . . . 139
10.28.1 Detailed Description . . . 139
10.28.2 Constructor & Destructor Documentation . . . 140
10.28.2.1 ~BaseCompositeMessage() . . . 140
10.28.3 Member Function Documentation . . . 140
10.28.3.1 ConstructPDV(const ULConnection &inConnection, const BaseRootQuery *inRoot↩ Query)=0 . . . 140
10.29 gdcm::network::BaseNormalizedMessage Class Reference . . . 140
10.29.1 Detailed Description . . . 141
10.29.2 Constructor & Destructor Documentation . . . 142
10.29.2.1 ~BaseNormalizedMessage() . . . 142
10.29.3 Member Function Documentation . . . 142
10.29.3.1 ConstructPDV(const ULConnection &inConnection, const BaseQuery *inQuery)=0 . . . 142
10.30 gdcm::network::BasePDU Class Reference . . . 143
10.30.1 Detailed Description . . . 143
10.30.2 Constructor & Destructor Documentation . . . 144
10.30.2.1 ~BasePDU() . . . 144
10.30.3 Member Function Documentation . . . 144
10.30.3.1 IsLastFragment() const =0 . . . 144
10.30.3.2 Print(std::ostream &os) const =0 . . . 144
10.30.3.3 Read(std::istream &is)=0 . . . 144
10.30.3.4 Size() const =0 . . . 145
10.30.3.5 Write(std::ostream &os) const =0 . . . 145
10.31 gdcm::BaseQuery Class Reference . . . 145

10.31.1 Detailed Description . . . 147
10.31.2 Constructor & Destructor Documentation . . . 147
10.31.2.1 BaseQuery() . . . 147
10.31.2.2 ∼BaseQuery() . . . 147
10.31.3 Member Function Documentation . . . 147
10.31.3.1 AddQueryDataSet(const DataSet &ds) . . . 147
10.31.3.2 GetAbstractSyntaxUID() const =0 . . . 147
10.31.3.3 GetQueryDataSet() const . . . 147
10.31.3.4 GetQueryDataSet() . . . 147
10.31.3.5 GetSOPInstanceUID() const . . . 147
10.31.3.6 Print(std::ostream &os) const . . . 147
10.31.3.7 SetSearchParameter(const Tag &inTag, const DictEntry &inDictEntry, const std::string &inValue) . . . 148
10.31.3.8 SetSearchParameter(const Tag &inTag, const std::string &inValue) . . . 148
10.31.3.9 SetSearchParameter(const std::string &inKeyword, const std::string &inValue) . . . 148
10.31.3.10 SetSOPInstanceUID(const std::string &iSopInstanceUID) . . . 148
10.31.3.11 ValidateQuery(bool inStrict=true) const =0 . . . 148
10.31.3.12 ValidDataSet(const DataSet &dataSetToValid, const DataSet &dataSetReference) const . . . 148
10.31.3.13 WriteHelpFile(std::ostream &os) . . . 148
10.31.3.14 WriteQuery(const std::string &inFileName) . . . 148
10.31.4 Friends And Related Function Documentation . . . 148
10.31.4.1 QueryFactory . . . 148
10.31.5 Member Data Documentation . . . 148
10.31.5.1 mDataSet . . . 148
10.31.5.2 mHelpDescription . . . 148
10.31.5.3 mSopInstanceUID . . . 148
10.32 gdcm::BaseRootQuery Class Reference . . . 149
10.32.1 Detailed Description . . . 150

10.32.2 Constructor & Destructor Documentation . . . 150
10.32.2.1 BaseRootQuery() . . . 150
10.32.2.2 ∼BaseRootQuery() . . . 150
10.32.3 Member Function Documentation . . . 150
10.32.3.1 Construct(ERootType inRootType, EQueryLevel qlevel) . . . 150
10.32.3.2 GetQueryLevelFromQueryRoot(ERootType roottype) . . . 150
10.32.3.3 GetQueryLevelFromString(const char ∗str) . . . 150
10.32.3.4 GetQueryLevelString(EQueryLevel ql) . . . 150
10.32.3.5 GetTagListByLevel(const EQueryLevel &inQueryLevel)=0 . . . 150
10.32.3.6 InitializeDataSet(const EQueryLevel &inQueryLevel)=0 . . . 151
10.32.3.7 ValidateQuery(bool inStrict=true) const =0 . . . 151
10.32.4 Friends And Related Function Documentation . . . 151
10.32.4.1 QueryFactory . . . 151
10.32.5 Member Data Documentation . . . 151
10.32.5.1 mHelpDescription . . . 151
10.32.5.2 mImage . . . 151
10.32.5.3 mPatient . . . 151
10.32.5.4 mRootType . . . 151
10.32.5.5 mSeries . . . 151
10.32.5.6 mStudy . . . 151
10.33gdcm::SegmentHelper::BasicCodedEntry Struct Reference . . . 152
10.33.1 Detailed Description . . . 153
10.33.2 Constructor & Destructor Documentation . . . 153
10.33.2.1 BasicCodedEntry() . . . 153
10.33.2.2 BasicCodedEntry(const char ∗a_CV, const char ∗a_CSD, const char ∗a_CM) . . . 153
10.33.2.3 BasicCodedEntry(const char ∗a_CV, const char ∗a_CSD, const char ∗a_CSV, const char ∗a_CM) . . . 153
10.33.3 Member Function Documentation . . . 153

10.33.3.1 IsEmpty(const bool checkOptionalAttributes=false) const . . . 153
10.33.4 Member Data Documentation . . . 153
10.33.4.1 CM . . . 154
10.33.4.2 CSD . . . 154
10.33.4.3 CSV . . . 154
10.33.4.4 CV . . . 154
10.34gdcm::BasicOffsetTable Class Reference . . . 154
10.34.1 Detailed Description . . . 155
10.34.2 Constructor & Destructor Documentation . . . 155
10.34.2.1 BasicOffsetTable() . . . 155
10.34.3 Member Function Documentation . . . 156
10.34.3.1 Read(std::istream &is) . . . 156
10.34.4 Friends And Related Function Documentation . . . 156
10.34.4.1 operator<< . . . 156
10.35gdcm::Bitmap Class Reference . . . 156
10.35.1 Detailed Description . . . 159
10.35.2 Member Typedef Documentation . . . 159
10.35.2.1 LUTPtr . . . 159
10.35.3 Constructor & Destructor Documentation . . . 159
10.35.3.1 Bitmap() . . . 159
10.35.3.2 ∼Bitmap() . . . 159
10.35.4 Member Function Documentation . . . 159
10.35.4.1 AreOverlaysInPixelData() const . . . 159
10.35.4.2 Clear() . . . 159
10.35.4.3 ComputeLossyFlag() . . . 159
10.35.4.4 GetBuffer(char *buffer) const . . . 159
10.35.4.5 GetBuffer2(std::ostream &os) const . . . 160
10.35.4.6 GetBufferLength() const . . . 160

10.35.4.7 GetColumns() const . . . 160
10.35.4.8 GetDataElement() const . . . 160
10.35.4.9 GetDataElement() . . . 160
10.35.4.10 GetDimension(unsigned int idx) const . . . 160
10.35.4.11 GetDimensions() const . . . 160
10.35.4.12 GetLUT() const . . . 160
10.35.4.13 GetLUT() . . . 161
10.35.4.14 GetNeedByteSwap() const . . . 161
10.35.4.15 GetNumberOfDimensions() const . . . 161
10.35.4.16 GetPhotometricInterpretation() const . . . 161
10.35.4.17 GetPixelFormat() const . . . 161
10.35.4.18 GetPixelFormat() . . . 161
10.35.4.19 GetPlanarConfiguration() const . . . 161
10.35.4.20 GetRows() const . . . 161
10.35.4.21 GetTransferSyntax() const . . . 161
10.35.4.22 IsEmpty() const . . . 162
10.35.4.23 IsLossy() const . . . 162
10.35.4.24 IsTransferSyntaxCompatible(TransferSyntax const &ts) const . . . 162
10.35.4.25 Print(std::ostream &) const . . . 162
10.35.4.26 SetColumns(unsigned int col) . . . 162
10.35.4.27 SetDataElement(DataElement const &de) . . . 162
10.35.4.28 SetDimension(unsigned int idx, unsigned int dim) . . . 162
10.35.4.29 SetDimensions(const unsigned int dims[3]) . . . 162
10.35.4.30 SetLossyFlag(bool f) . . . 162
10.35.4.31 SetLUT(LookupTable const &lut) . . . 163
10.35.4.32 SetNeedByteSwap(bool b) . . . 163
10.35.4.33 SetNumberOfDimensions(unsigned int dim) . . . 163
10.35.4.34 SetPhotometricInterpretation(PhotometricInterpretation const &pi) . . . 163

10.35.4.35 SetPixelFormat(PixelFormat const &pf) . . . 163
10.35.4.36 SetPlanarConfiguration(unsigned int pc) . . . 163
10.35.4.37 SetRows(unsigned int rows) . . . 163
10.35.4.38 SetTransferSyntax(TransferSyntax const &ts) . . . 163
10.35.4.39 TryJPEG2000Codec(char *buffer, bool &lossyflag) const . . . 164
10.35.4.40 TryJPEG2000Codec2(std::ostream &os) const . . . 164
10.35.4.41 TryJPEGCodec(char *buffer, bool &lossyflag) const . . . 164
10.35.4.42 TryJPEGCodec2(std::ostream &os) const . . . 164
10.35.4.43 TryJPEGLSCodec(char *buffer, bool &lossyflag) const . . . 164
10.35.4.44 TryKAKADUCodec(char *buffer, bool &lossyflag) const . . . 164
10.35.4.45 TryPVRGCodec(char *buffer, bool &lossyflag) const . . . 164
10.35.4.46 TryRAWCodec(char *buffer, bool &lossyflag) const . . . 164
10.35.4.47 TryRLECodec(char *buffer, bool &lossyflag) const . . . 164
10.35.5 Friends And Related Function Documentation . . . 164
10.35.5.1 ImageChangeTransferSyntax . . . 164
10.35.5.2 PixmapReader . . . 164
10.35.6 Member Data Documentation . . . 164
10.35.6.1 Dimensions . . . 164
10.35.6.2 LossyFlag . . . 164
10.35.6.3 LUT . . . 164
10.35.6.4 NeedByteSwap . . . 164
10.35.6.5 NumberOfDimensions . . . 164
10.35.6.6 PF . . . 164
10.35.6.7 PI . . . 164
10.35.6.8 PixelData . . . 164
10.35.6.9 PlanarConfiguration . . . 164
10.35.6.10 TS . . . 164
10.36 gdcm::BitmapToBitmapFilter Class Reference . . . 165

10.36.1 Detailed Description . . . 166
10.36.2 Constructor & Destructor Documentation . . . 166
10.36.2.1 BitmapToBitmapFilter() . . . 166
10.36.2.2 ∼BitmapToBitmapFilter() . . . 166
10.36.3 Member Function Documentation . . . 166
10.36.3.1 GetOutput() const . . . 166
10.36.3.2 GetOutputAsBitmap() const . . . 166
10.36.3.3 SetInput(const Bitmap &image) . . . 166
10.36.4 Member Data Documentation . . . 166
10.36.4.1 Input . . . 166
10.36.4.2 Output . . . 166
10.37 gdcm::BoxRegion Class Reference . . . 167
10.37.1 Detailed Description . . . 168
10.37.2 Constructor & Destructor Documentation . . . 168
10.37.2.1 BoxRegion() . . . 168
10.37.2.2 ∼BoxRegion() . . . 168
10.37.2.3 BoxRegion(const BoxRegion &) . . . 168
10.37.3 Member Function Documentation . . . 169
10.37.3.1 Area() const . . . 169
10.37.3.2 BoundingBox(BoxRegion const &b1, BoxRegion const &b2) . . . 169
10.37.3.3 Clone() const . . . 169
10.37.3.4 ComputeBoundingBox() . . . 169
10.37.3.5 Empty() const . . . 169
10.37.3.6 GetXMax() const . . . 169
10.37.3.7 GetXMin() const . . . 169
10.37.3.8 GetYMax() const . . . 170
10.37.3.9 GetYMin() const . . . 170
10.37.3.10 GetZMax() const . . . 170

10.37.3.11 GetZMin() const . . . 170
10.37.3.12 IsValid() const . . . 170
10.37.3.13 operator=(const BoxRegion &) . . . 170
10.37.3.14 Print(std::ostream &os=std::cout) const . . . 170
10.37.3.15 SetDomain(unsigned int xmin, unsigned int xmax, unsigned int ymin, unsigned int ymax, unsigned int zmin, unsigned int zmax) . . . 170
10.38 gdcm::ByteBuffer Class Reference . . . 170
10.38.1 Detailed Description . . . 171
10.38.2 Constructor & Destructor Documentation . . . 171
10.38.2.1 ByteBuffer() . . . 171
10.38.3 Member Function Documentation . . . 171
10.38.3.1 Get(int len) . . . 171
10.38.3.2 GetStart() const . . . 171
10.38.3.3 ShiftEnd(int len) . . . 171
10.38.3.4 UpdatePosition() . . . 171
10.39 gdcm::ByteSwap< T > Class Template Reference . . . 171
10.39.1 Detailed Description . . . 172
10.39.2 Member Function Documentation . . . 172
10.39.2.1 Swap(T &p) . . . 172
10.39.2.2 SwapFromSwapCodeIntoSystem(T &p, SwapCode const &sc) . . . 172
10.39.2.3 SwapRange(T *p, unsigned int num) . . . 172
10.39.2.4 SwapRangeFromSwapCodeIntoSystem(T *p, SwapCode const &sc, std::streamoff num) . . . 172
10.39.2.5 SystemIsBigEndian() . . . 172
10.39.2.6 SystemIsLittleEndian() . . . 172
10.40 gdcm::ByteSwapFilter Class Reference . . . 173
10.40.1 Detailed Description . . . 173
10.40.2 Constructor & Destructor Documentation . . . 173
10.40.2.1 ByteSwapFilter(DataSet &ds) . . . 173

10.40.2.2 ∼ByteSwapFilter() . . . 173
10.40.3 Member Function Documentation . . . 173
10.40.3.1 ByteSwap() . . . 173
10.40.3.2 SetByteSwapTag(bool b) . . . 173
10.41 gdcm::ByteValue Class Reference . . . 174
10.41.1 Detailed Description . . . 175
10.41.2 Constructor & Destructor Documentation . . . 176
10.41.2.1 ByteValue(const char ∗array=0, VL const &vl=0) . . . 176
10.41.2.2 ByteValue(std::vector< char > &v) . . . 176
10.41.2.3 ∼ByteValue() . . . 176
10.41.3 Member Function Documentation . . . 176
10.41.3.1 Append(ByteValue const &bv) . . . 176
10.41.3.2 Clear() . . . 176
10.41.3.3 ComputeLength() const . . . 176
10.41.3.4 Fill(char c) . . . 176
10.41.3.5 GetBuffer(char ∗buffer, unsigned long length) const . . . 176
10.41.3.6 GetLength() const . . . 177
10.41.3.7 GetPointer() const . . . 177
10.41.3.8 IsEmpty() const . . . 177
10.41.3.9 IsPrintable(VL length) const . . . 177
10.41.3.10 operator const std::vector< char > &() const . . . 177
10.41.3.11 operator=(const ByteValue &val) . . . 177
10.41.3.12 operator==(const ByteValue &val) const . . . 177
10.41.3.13 operator==(const Value &val) const . . . 177
10.41.3.14 Print(std::ostream &os) const . . . 178
10.41.3.15 PrintASCII(std::ostream &os, VL maxlength) const . . . 178
10.41.3.16 PrintASCIIXML(std::ostream &os) const . . . 178
10.41.3.17 PrintGroupLength(std::ostream &os) . . . 178

10.41.3.18PrintHex(std::ostream &os, VL maxlength) const . . . 178
10.41.3.19PrintHexXML(std::ostream &os) const . . . 178
10.41.3.20PrintPNXML(std::ostream &os) const . . . 178
10.41.3.21Read(std::istream &is, bool readvalues=true) . . . 178
10.41.3.22Read(std::istream &is) . . . 178
10.41.3.23SetLength(VL vl) . . . 178
10.41.3.24SetLengthOnly(VL vl) . . . 178
10.41.3.25Write(std::ostream &os) const . . . 178
10.41.3.26Write(std::ostream &os) const . . . 178
10.41.3.27WriteBuffer(std::ostream &os) const . . . 178
10.42gdcm::CAPICryptoFactory Class Reference . . . 179
10.42.1 Constructor & Destructor Documentation . . . 179
10.42.1.1 CAPICryptoFactory(CryptoLib id) . . . 179
10.42.2 Member Function Documentation . . . 180
10.42.2.1 CreateCMSProvider() . . . 180
10.43gdcm::CAPICryptographicMessageSyntax Class Reference . . . 180
10.43.1 Constructor & Destructor Documentation . . . 181
10.43.1.1 CAPICryptographicMessageSyntax() . . . 181
10.43.1.2 ∼CAPICryptographicMessageSyntax() . . . 181
10.43.2 Member Function Documentation . . . 181
10.43.2.1 Decrypt(char ∗output, size_t &outlen, const char ∗array, size_t len) const . . . 181
10.43.2.2 Encrypt(char ∗output, size_t &outlen, const char ∗array, size_t len) const . . . 181
10.43.2.3 GetCipherType() const . . . 181
10.43.2.4 GetInitialized() const . . . 182
10.43.2.5 ParseCertificateFile(const char ∗filename) . . . 182
10.43.2.6 ParseKeyFile(const char ∗filename) . . . 182
10.43.2.7 SetCipherType(CipherTypes type) . . . 182
10.43.2.8 SetPassword(const char ∗pass, size_t passLen) . . . 182

10.44gdcm::network::CEchoRQ Class Reference . . . 182
10.44.1 Detailed Description . . . 183
10.44.2 Member Function Documentation . . . 183
10.44.2.1 ConstructPDV(const ULConnection &inConnection, const BaseRootQuery *inRootQuery) . . . 183
10.44.3 Member Data Documentation . . . 183
10.44.3.1 AffectedSOPClassUID . . . 183
10.44.3.2 MessageID . . . 183
10.45gdcm::network::CEchoRSP Class Reference . . . 184
10.45.1 Detailed Description . . . 184
10.45.2 Member Function Documentation . . . 185
10.45.2.1 ConstructPDVByDataSet(const DataSet *inDataSet) . . . 185
10.46gdcm::network::CFind Class Reference . . . 185
10.46.1 Detailed Description . . . 185
10.47gdcm::network::CFindCancelRQ Class Reference . . . 185
10.47.1 Detailed Description . . . 186
10.47.2 Member Function Documentation . . . 186
10.47.2.1 ConstructPDVByDataSet(const DataSet *inDataSet) . . . 186
10.48gdcm::network::CFindRQ Class Reference . . . 186
10.48.1 Detailed Description . . . 187
10.48.2 Member Function Documentation . . . 188
10.48.2.1 ConstructPDV(const ULConnection &inConnection, const BaseRootQuery *inRootQuery) . . . 188
10.49gdcm::network::CFindRSP Class Reference . . . 188
10.49.1 Detailed Description . . . 189
10.49.2 Member Function Documentation . . . 189
10.49.2.1 ConstructPDVByDataSet(const DataSet *inDataSet) . . . 189
10.50gdcm::network::CMoveCancelRq Class Reference . . . 189
10.50.1 Member Function Documentation . . . 190

10.50.1.1 ConstructPDVByDataSet(const DataSet *inDataSet) . . . 190
10.51 gdcm::network::CMoveRQ Class Reference . . . 190
10.51.1 Detailed Description . . . 191
10.51.2 Member Function Documentation . . . 191
10.51.2.1 ConstructPDV(const ULConnection &inConnection, const BaseRootQuery *inRootQuery) . . . 191
10.52 gdcm::network::CMoveRSP Class Reference . . . 192
10.52.1 Detailed Description . . . 192
10.52.2 Member Function Documentation . . . 193
10.52.2.1 ConstructPDVByDataSet(const DataSet *inDataSet) . . . 193
10.53 gdcm::Codec Class Reference . . . 193
10.53.1 Detailed Description . . . 194
10.54 gdcm::Coder Class Reference . . . 194
10.54.1 Detailed Description . . . 195
10.54.2 Constructor & Destructor Documentation . . . 195
10.54.2.1 ~Coder() . . . 195
10.54.3 Member Function Documentation . . . 195
10.54.3.1 CanCode(TransferSyntax const &) const =0 . . . 195
10.54.3.2 Code(DataElement const &in_, DataElement &out_) . . . 195
10.54.3.3 InternalCode(const char *bv, unsigned long len, std::ostream &os) . . . 195
10.55 gdcm::CodeString Class Reference . . . 195
10.55.1 Detailed Description . . . 196
10.55.2 Member Typedef Documentation . . . 197
10.55.2.1 const_iterator . . . 197
10.55.2.2 const_reference . . . 197
10.55.2.3 const_reverse_iterator . . . 197
10.55.2.4 difference_type . . . 197
10.55.2.5 iterator . . . 197

10.55.2.6 pointer . . . 197
10.55.2.7 reference . . . 197
10.55.2.8 reverse_iterator . . . 197
10.55.2.9 size_type . . . 197
10.55.2.10value_type . . . 197
10.55.3 Constructor & Destructor Documentation . . . 197
10.55.3.1 CodeString() . . . 197
10.55.3.2 CodeString(const value_type ∗s) . . . 197
10.55.3.3 CodeString(const value_type ∗s, size_type n) . . . 197
10.55.3.4 CodeString(const InternalClass &s, size_type pos=0, size_type n=InternalClass::npos) 197
10.55.4 Member Function Documentation . . . 197
10.55.4.1 GetAsString() const . . . 197
10.55.4.2 IsValid() const . . . 198
10.55.4.3 Size() const . . . 198
10.55.4.4 TrimInternal() const . . . 198
10.55.5 Friends And Related Function Documentation . . . 198
10.55.5.1 operator"!= . . . 198
10.55.5.2 operator<< . . . 198
10.55.5.3 operator== . . . 198
10.56gdcm::Command Class Reference . . . 198
10.56.1 Detailed Description . . . 199
10.56.2 Constructor & Destructor Documentation . . . 199
10.56.2.1 Command() . . . 199
10.56.2.2 ∼Command() . . . 199
10.56.3 Member Function Documentation . . . 199
10.56.3.1 Execute(Subject ∗caller, const Event &event)=0 . . . 199
10.56.3.2 Execute(const Subject ∗caller, const Event &event)=0 . . . 200
10.57gdcm::CommandDataSet Class Reference . . . 200

10.57.1 Detailed Description . . . 201
10.57.2 Constructor & Destructor Documentation . . . 201
10.57.2.1 CommandDataSet() . . . 201
10.57.2.2 ∼CommandDataSet() . . . 201
10.57.3 Member Function Documentation . . . 201
10.57.3.1 Insert(const DataElement &de) . . . 201
10.57.3.2 Read(std::istream &is) . . . 201
10.57.3.3 Replace(const DataElement &de) . . . 202
10.57.3.4 Write(std::ostream &os) const . . . 202
10.57.4 Friends And Related Function Documentation . . . 202
10.57.4.1 operator<< . . . 202
10.58gdcm::network::CompositeMessageFactory Class Reference . . . 202
10.58.1 Detailed Description . . . 202
10.58.2 Member Function Documentation . . . 203
10.58.2.1 ConstructCEchoRQ(const ULConnection &inConnection) . . . 203
10.58.2.2 ConstructCFindRQ(const ULConnection &inConnection, const BaseRootQuery ∗in↩ RootQuery) . . . 203
10.58.2.3 ConstructCMoveRQ(const ULConnection &inConnection, const BaseRootQuery ∗inRootQuery) . . . 203
10.58.2.4 ConstructCStoreRQ(const ULConnection &inConnection, const File &file, bool write↩ DataSet=true) . . . 203
10.58.2.5 ConstructCStoreRSP(const DataSet ∗inDataSet, const BasePDU ∗inPC) . . . 203
10.59gdcm::CompositeNetworkFunctions Class Reference . . . 203
10.59.1 Detailed Description . . . 204
10.59.2 Member Typedef Documentation . . . 204
10.59.2.1 KeyValuePairArrayType . . . 204
10.59.2.2 KeyValuePairType . . . 204
10.59.3 Member Function Documentation . . . 204
10.59.3.1 CEcho(const char ∗remote, uint16_t portno, const char ∗aetitle=NULL, const char ∗call=NULL) . . . 204

10.59.3.2 CFind(const char *remote, uint16_t portno, const BaseRootQuery *query, std::vector< DataSet > &retDataSets, const char *aetitle=NULL, const char *call=NULL) . . . 205

10.59.3.3 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) . . . 205

10.59.3.4 ConstructQuery(ERootType inRootType, EQueryLevel inQueryLevel, const DataSet &queryds, EQueryType queryType=eFind) . . . 205

10.59.3.5 ConstructQuery(ERootType inRootType, EQueryLevel inQueryLevel, const KeyValuePairArrayType &keys, EQueryType queryType=eFind) . . . 206

10.59.3.6 CStore(const char *remote, uint16_t portno, const Directory::FilenamesType &filenames, const char *aetitle=NULL, const char *call=NULL) . . . 206

10.60 gdcm::ConstCharWrapper Class Reference . . . 206

10.60.1 Detailed Description . . . 207

10.60.2 Constructor & Destructor Documentation . . . 207

10.60.2.1 ConstCharWrapper(const char *i=0) . . . 207

10.60.3 Member Function Documentation . . . 207

10.60.3.1 operator const char *() const . . . 207

10.61 gdcm::CP246ExplicitDataElement Class Reference . . . 207

10.61.1 Detailed Description . . . 208

10.61.2 Member Function Documentation . . . 209

10.61.2.1 GetLength() const . . . 209

10.61.2.2 Read(std::istream &is) . . . 209

10.61.2.3 ReadPreValue(std::istream &is) . . . 209

10.61.2.4 ReadValue(std::istream &is, bool readvalues=true) . . . 209

10.61.2.5 ReadWithLength(std::istream &is, VL &length) . . . 209

10.62 gdcm::CryptoFactory Class Reference . . . 209

10.62.1 Detailed Description . . . 210

10.62.2 Member Enumeration Documentation . . . 210

10.62.2.1 CryptoLib . . . 210

10.62.3 Constructor & Destructor Documentation . . . 210

10.62.3.1 CryptoFactory(CryptoLib id) . . . 210
10.62.3.2 CryptoFactory() . . . 211
10.62.3.3 ∼CryptoFactory() . . . 211
10.62.4 Member Function Documentation . . . 211
10.62.4.1 CreateCMSProvider()=0 . . . 211
10.62.4.2 GetFactoryInstance(CryptoLib id=DEFAULT) . . . 211
10.63gdcm::CryptographicMessageSyntax Class Reference . . . 211
10.63.1 Member Enumeration Documentation . . . 212
10.63.1.1 CipherTypes . . . 212
10.63.2 Constructor & Destructor Documentation . . . 212
10.63.2.1 CryptographicMessageSyntax() . . . 212
10.63.2.2 ∼CryptographicMessageSyntax() . . . 212
10.63.3 Member Function Documentation . . . 212
10.63.3.1 Decrypt(char ∗output, size_t &outlen, const char ∗array, size_t len) const =0 . . . 212
10.63.3.2 Encrypt(char ∗output, size_t &outlen, const char ∗array, size_t len) const =0 . . . 212
10.63.3.3 GetCipherType() const =0 . . . 213
10.63.3.4 ParseCertificateFile(const char ∗filename)=0 . . . 213
10.63.3.5 ParseKeyFile(const char ∗filename)=0 . . . 213
10.63.3.6 SetCipherType(CipherTypes type)=0 . . . 213
10.63.3.7 SetPassword(const char ∗pass, size_t passLen)=0 . . . 213
10.64gdcm::CSAElement Class Reference . . . 213
10.64.1 Detailed Description . . . 215
10.64.2 Member Typedef Documentation . . . 215
10.64.2.1 DataPtr . . . 215
10.64.3 Constructor & Destructor Documentation . . . 215
10.64.3.1 CSAElement(unsigned int kf=0) . . . 215
10.64.3.2 CSAElement(const CSAElement &_val) . . . 215
10.64.4 Member Function Documentation . . . 215

10.64.4.1 GetByteValue() const . . . 215
10.64.4.2 GetKey() const . . . 216
10.64.4.3 GetName() const . . . 216
10.64.4.4 GetNoOfItems() const . . . 216
10.64.4.5 GetSyngoDT() const . . . 216
10.64.4.6 GetValue() const . . . 216
10.64.4.7 GetValue() . . . 216
10.64.4.8 GetVM() const . . . 216
10.64.4.9 GetVR() const . . . 216
10.64.4.10 IsEmpty() const . . . 217
10.64.4.11 operator<(const CSAElement &de) const . . . 217
10.64.4.12 operator=(const CSAElement &de) . . . 217
10.64.4.13 operator==(const CSAElement &de) const . . . 217
10.64.4.14 SetByteValue(const char *array, VL length) . . . 217
10.64.4.15 SetKey(unsigned int key) . . . 217
10.64.4.16 SetName(const char *name) . . . 217
10.64.4.17 SetNoOfItems(unsigned int items) . . . 217
10.64.4.18 SetSyngoDT(unsigned int syngodt) . . . 217
10.64.4.19 SetValue(Value const &vl) . . . 217
10.64.4.20 SetVM(const VM &vm) . . . 217
10.64.4.21 SetVR(VR const &vr) . . . 217
10.64.5 Friends And Related Function Documentation . . . 217
10.64.5.1 operator<< . . . 217
10.64.6 Member Data Documentation . . . 217
10.64.6.1 DataField . . . 217
10.64.6.2 KeyField . . . 218
10.64.6.3 NameField . . . 218
10.64.6.4 NoOfItemsField . . . 218

10.64.6.5 SyngoDTField . . . 218
10.64.6.6 ValueMultiplicityField . . . 218
10.64.6.7 VRField . . . 218
10.65gdcm::CSAHeader Class Reference . . . 218
10.65.1 Detailed Description . . . 219
10.65.2 Member Enumeration Documentation . . . 220
10.65.2.1 CSAHeaderType . . . 220
10.65.3 Constructor & Destructor Documentation . . . 220
10.65.3.1 CSAHeader() . . . 220
10.65.3.2 ∼CSAHeader() . . . 220
10.65.4 Member Function Documentation . . . 220
10.65.4.1 FindCSAElementByName(const char ∗name) . . . 220
10.65.4.2 GetCSADataInfo() . . . 221
10.65.4.3 GetCSAEEnd() const . . . 221
10.65.4.4 GetCSAElementByName(const char ∗name) . . . 221
10.65.4.5 GetCSAImageHeaderInfoTag() . . . 221
10.65.4.6 GetCSASeriesHeaderInfoTag() . . . 221
10.65.4.7 GetDataSet() const . . . 221
10.65.4.8 GetFormat() const . . . 222
10.65.4.9 GetInterfile() const . . . 222
10.65.4.10LoadFromDataElement(DataElement const &de) . . . 222
10.65.4.11Print(std::ostream &os) const . . . 222
10.65.4.12Read(std::istream &is) . . . 222
10.65.4.13Write(std::ostream &os) const . . . 222
10.65.5 Friends And Related Function Documentation . . . 222
10.65.5.1 operator<< . . . 222
10.66gdcm::CSAHeaderDict Class Reference . . . 223
10.66.1 Detailed Description . . . 223

10.66.2 Member Typedef Documentation . . . 224
10.66.2.1 ConstIterator . . . 224
10.66.2.2 Iterator . . . 224
10.66.2.3 MapCSAHeaderDictEntry . . . 224
10.66.3 Constructor & Destructor Documentation . . . 224
10.66.3.1 CSAHeaderDict() . . . 224
10.66.4 Member Function Documentation . . . 224
10.66.4.1 AddCSAHeaderDictEntry(const CSAHeaderDictEntry &de) . . . 224
10.66.4.2 Begin() const . . . 224
10.66.4.3 End() const . . . 224
10.66.4.4 GetCSAHeaderDictEntry(const char ∗name) const . . . 224
10.66.4.5 IsEmpty() const . . . 224
10.66.4.6 LoadDefault() . . . 224
10.66.5 Friends And Related Function Documentation . . . 224
10.66.5.1 Dicts . . . 224
10.66.5.2 operator<< . . . 224
10.67 gdcm::CSAHeaderDictEntry Class Reference . . . 225
10.67.1 Detailed Description . . . 225
10.67.2 Constructor & Destructor Documentation . . . 226
10.67.2.1 CSAHeaderDictEntry(const char ∗name="", VR const &vr=VR::INVALID, VM const &vm=VM::VM0, const char ∗desc="") . . . 226
10.67.3 Member Function Documentation . . . 226
10.67.3.1 GetDescription() const . . . 226
10.67.3.2 GetName() const . . . 226
10.67.3.3 GetVM() const . . . 226
10.67.3.4 GetVR() const . . . 226
10.67.3.5 operator<(const CSAHeaderDictEntry &entry) const . . . 226
10.67.3.6 SetDescription(const char ∗desc) . . . 226

10.67.3.7 SetName(const char *name) . . . 226
10.67.3.8 SetVM(VM const &vm) . . . 226
10.67.3.9 SetVR(const VR &vr) . . . 226
10.67.4 Friends And Related Function Documentation . . . 226
10.67.4.1 operator<< . . . 226
10.68 gdcm::CSAHeaderDictException Class Reference . . . 227
10.69 gdcm::network::CStoreRQ Class Reference . . . 227
10.69.1 Detailed Description . . . 228
10.69.2 Member Function Documentation . . . 228
10.69.2.1 ConstructPDV(const ULConnection &inConnection, const File &file, bool writeDataSet=true) . . . 228
10.70 gdcm::network::CStoreRSP Class Reference . . . 229
10.70.1 Detailed Description . . . 229
10.70.2 Member Function Documentation . . . 230
10.70.2.1 ConstructPDV(const DataSet *inDataSet, const BasePDU *inPC) . . . 230
10.71 gdcm::Curve Class Reference . . . 230
10.71.1 Detailed Description . . . 231
10.71.2 Constructor & Destructor Documentation . . . 232
10.71.2.1 Curve() . . . 232
10.71.2.2 ~Curve() . . . 232
10.71.2.3 Curve(Curve const &ov) . . . 232
10.71.3 Member Function Documentation . . . 232
10.71.3.1 Decode(std::istream &is, std::ostream &os) . . . 232
10.71.3.2 GetAsPoints(float *array) const . . . 232
10.71.3.3 GetCurveDataDescriptor() const . . . 232
10.71.3.4 GetDataValueRepresentation() const . . . 232
10.71.3.5 GetDimensions() const . . . 232
10.71.3.6 GetGroup() const . . . 232

10.71.3.7 GetNumberOfCurves(DataSet const &ds) . . . 232
10.71.3.8 GetNumberOfPoints() const . . . 232
10.71.3.9 GetTypeOfData() const . . . 232
10.71.3.10 GetTypeOfDataDescription() const . . . 232
10.71.3.11 IsEmpty() const . . . 232
10.71.3.12 Print(std::ostream &) const . . . 232
10.71.3.13 SetCoordinateStartValue(unsigned short v) . . . 233
10.71.3.14 SetCoordinateStepValue(unsigned short v) . . . 233
10.71.3.15 SetCurve(const char *array, unsigned int length) . . . 233
10.71.3.16 SetCurveDataDescriptor(const uint16_t *values, size_t num) . . . 233
10.71.3.17 SetCurveDescription(const char *curvedescription) . . . 233
10.71.3.18 SetDataValueRepresentation(unsigned short datavaluerepresentation) . . . 233
10.71.3.19 SetDimensions(unsigned short dimensions) . . . 233
10.71.3.20 SetGroup(unsigned short group) . . . 233
10.71.3.21 SetNumberOfPoints(unsigned short numberofpoints) . . . 233
10.71.3.22 SetTypeOfData(const char *typeofdata) . . . 233
10.71.3.23 Update(const DataElement &de) . . . 233
10.72 gdcm::DataElement Class Reference . . . 233
10.72.1 Detailed Description . . . 236
10.72.2 Member Typedef Documentation . . . 237
10.72.2.1 ValuePtr . . . 237
10.72.3 Constructor & Destructor Documentation . . . 237
10.72.3.1 DataElement(const Tag &t=Tag(0), const VL &vl=0, const VR &vr=VR::INVALID) . . . 237
10.72.3.2 DataElement(const DataElement &_val) . . . 237
10.72.4 Member Function Documentation . . . 237
10.72.4.1 Clear() . . . 237
10.72.4.2 Empty() . . . 237
10.72.4.3 GetByteValue() const . . . 237

10.72.4.4 GetLength() const . . . 238
10.72.4.5 GetSequenceOfFragments() const . . . 238
10.72.4.6 GetSequenceOfFragments() . . . 238
10.72.4.7 GetTag() const . . . 238
10.72.4.8 GetTag() . . . 238
10.72.4.9 GetValue() const . . . 238
10.72.4.10 GetValue() . . . 239
10.72.4.11 GetValueAsSQ() const . . . 239
10.72.4.12 GetVL() const . . . 239
10.72.4.13 GetVL() . . . 239
10.72.4.14 GetVR() const . . . 239
10.72.4.15 IsEmpty() const . . . 240
10.72.4.16 IsUndefinedLength() const . . . 240
10.72.4.17 operator<(const DataElement &de) const . . . 240
10.72.4.18 operator=(const DataElement &de) . . . 240
10.72.4.19 operator==(const DataElement &de) const . . . 240
10.72.4.20 Read(std::istream &is) . . . 240
10.72.4.21 ReadOrSkip(std::istream &is, std::set< Tag > const &skiptags) . . . 240
10.72.4.22 ReadPreValue(std::istream &is, std::set< Tag > const &skiptags) . . . 240
10.72.4.23 ReadValue(std::istream &is, std::set< Tag > const &skiptags) . . . 240
10.72.4.24 ReadValueWithLength(std::istream &is, VL &length, std::set< Tag > const &skiptags) 240
10.72.4.25 ReadWithLength(std::istream &is, VL &length) . . . 240
10.72.4.26 SetByteValue(const char *array, VL length) . . . 240
10.72.4.27 SetTag(const Tag &t) . . . 241
10.72.4.28 SetValue(Value const &vl) . . . 241
10.72.4.29 SetValueFieldLength(VL vl, bool readvalues) . . . 241
10.72.4.30 SetVL(const VL &vl) . . . 241
10.72.4.31 SetVLToUndefined() . . . 242

10.72.4.32SetVR(VR const &vr) . . . 242
10.72.4.33Write(std::ostream &os) const . . . 242
10.72.5 Friends And Related Function Documentation . . . 242
10.72.5.1 operator<< . . . 242
10.72.6 Member Data Documentation . . . 242
10.72.6.1 TagField . . . 242
10.72.6.2 ValueField . . . 242
10.72.6.3 ValueLengthField . . . 243
10.72.6.4 VRField . . . 243
10.73gdcm::DataElementException Class Reference . . . 243
10.74gdcm::DataEvent Class Reference . . . 244
10.74.1 Detailed Description . . . 245
10.74.2 Member Typedef Documentation . . . 245
10.74.2.1 Self . . . 245
10.74.2.2 Superclass . . . 245
10.74.3 Constructor & Destructor Documentation . . . 245
10.74.3.1 DataEvent(const char *bytes=0, size_t len=0) . . . 245
10.74.3.2 ~DataEvent() . . . 245
10.74.3.3 DataEvent(const Self &s) . . . 245
10.74.4 Member Function Documentation . . . 245
10.74.4.1 CheckEvent(const ::gdcm::Event *e) const . . . 245
10.74.4.2 GetData() const . . . 245
10.74.4.3 GetDataLength() const . . . 245
10.74.4.4 GetEventName() const . . . 245
10.74.4.5 MakeObject() const . . . 246
10.74.4.6 SetData(const char *bytes, size_t len) . . . 246
10.75gdcm::DataSet Class Reference . . . 246
10.75.1 Detailed Description . . . 248

10.75.2 Member Typedef Documentation . . . 249
10.75.2.1 ConstIterator . . . 249
10.75.2.2 DataElementSet . . . 249
10.75.2.3 Iterator . . . 249
10.75.2.4 SizeType . . . 249
10.75.3 Member Function Documentation . . . 249
10.75.3.1 Begin() const . . . 249
10.75.3.2 Begin() . . . 249
10.75.3.3 Clear() . . . 249
10.75.3.4 ComputeDataElement(const PrivateTag &t) const . . . 249
10.75.3.5 ComputeGroupLength(Tag const &tag) const . . . 249
10.75.3.6 End() const . . . 249
10.75.3.7 End() . . . 250
10.75.3.8 FindDataElement(const PrivateTag &t) const . . . 250
10.75.3.9 FindDataElement(const Tag &t) const . . . 250
10.75.3.10 FindNextDataElement(const Tag &t) const . . . 250
10.75.3.11 GetDataElement(const Tag &t) const . . . 250
10.75.3.12 GetDataElement(const PrivateTag &t) const . . . 251
10.75.3.13 GetDEEnd() const . . . 251
10.75.3.14 GetDES() const . . . 251
10.75.3.15 GetDES() . . . 251
10.75.3.16 GetLength() const . . . 251
10.75.3.17 GetMediaStorage() const . . . 251
10.75.3.18 GetPrivateCreator(const Tag &t) const . . . 251
10.75.3.19 Insert(const DataElement &de) . . . 251
10.75.3.20 InsertDataElement(const DataElement &de) . . . 252
10.75.3.21 IsEmpty() const . . . 252
10.75.3.22 operator()(uint16_t group, uint16_t element) const . . . 252

10.75.3.23operator=(DataSet const &val) . . . 252
10.75.3.24operator[](const Tag &t) const . . . 252
10.75.3.25Print(std::ostream &os, std::string const &indent="""") const . . . 252
10.75.3.26Read(std::istream &is) . . . 252
10.75.3.27ReadNested(std::istream &is) . . . 252
10.75.3.28ReadSelectedPrivateTags(std::istream &is, const std::set< PrivateTag > &tags, bool readvalues=true) . . . 252
10.75.3.29ReadSelectedPrivateTagsWithLength(std::istream &is, const std::set< PrivateTag > &tags, VL &length, bool readvalues=true) . . . 252
10.75.3.30ReadSelectedTags(std::istream &is, const std::set< Tag > &tags, bool readvalues=true) 252
10.75.3.31ReadSelectedTagsWithLength(std::istream &is, const std::set< Tag > &tags, VL &length, bool readvalues=true) . . . 252
10.75.3.32ReadUpToTag(std::istream &is, const Tag &t, std::set< Tag > const &skiptags) . . . 252
10.75.3.33ReadUpToTagWithLength(std::istream &is, const Tag &t, std::set< Tag > const &skiptags, VL &length) . . . 252
10.75.3.34ReadWithLength(std::istream &is, VL &length) . . . 252
10.75.3.35Remove(const Tag &tag) . . . 252
10.75.3.36Replace(const DataElement &de) . . . 253
10.75.3.37ReplaceEmpty(const DataElement &de) . . . 253
10.75.3.38Size() const . . . 253
10.75.3.39Write(std::ostream &os) const . . . 253
10.75.4 Friends And Related Function Documentation . . . 253
10.75.4.1 CSAHeader . . . 253
10.75.4.2 operator<< . . . 253
10.76gdcm::DataSetEvent Class Reference . . . 254
10.76.1 Detailed Description . . . 255
10.76.2 Member Typedef Documentation . . . 255
10.76.2.1 Self . . . 255
10.76.2.2 Superclass . . . 255
10.76.3 Constructor & Destructor Documentation . . . 255

10.76.3.1 DataSetEvent(DataSet const *ds=NULL) . . . 255
10.76.3.2 ~DataSetEvent() . . . 255
10.76.3.3 DataSetEvent(const Self &s) . . . 255
10.76.4 Member Function Documentation . . . 255
10.76.4.1 CheckEvent(const ::gdcm::Event *e) const . . . 255
10.76.4.2 GetDataSet() const . . . 255
10.76.4.3 GetEventName() const . . . 255
10.76.4.4 MakeObject() const . . . 256
10.77gdcm::DataSetHelper Class Reference . . . 256
10.77.1 Detailed Description . . . 256
10.77.2 Member Function Documentation . . . 256
10.77.2.1 ComputeVR(File const &file, DataSet const &ds, const Tag &tag) . . . 256
10.78gdcm::Decoder Class Reference . . . 257
10.78.1 Detailed Description . . . 257
10.78.2 Constructor & Destructor Documentation . . . 258
10.78.2.1 ~Decoder() . . . 258
10.78.3 Member Function Documentation . . . 258
10.78.3.1 CanDecode(TransferSyntax const &) const =0 . . . 258
10.78.3.2 Decode(DataElement const &, DataElement &) . . . 258
10.78.3.3 DecodeByStreams(std::istream &, std::ostream &) . . . 258
10.79gdcm::DefinedTerms Class Reference . . . 258
10.79.1 Detailed Description . . . 259
10.79.2 Constructor & Destructor Documentation . . . 259
10.79.2.1 DefinedTerms() . . . 259
10.80gdcm::Defs Class Reference . . . 259
10.80.1 Detailed Description . . . 260
10.80.2 Constructor & Destructor Documentation . . . 260
10.80.2.1 Defs() . . . 260

10.80.2.2 ~Defs() . . . 260
10.80.3 Member Function Documentation . . . 260
10.80.3.1 GetIODFromFile(const File &file) const . . . 260
10.80.3.2 GetIODNameFromMediaStorage(MediaStorage const &ms) . . . 260
10.80.3.3 GetIODs() const . . . 261
10.80.3.4 GetIODs() . . . 261
10.80.3.5 GetMacros() const . . . 261
10.80.3.6 GetMacros() . . . 261
10.80.3.7 GetModules() const . . . 261
10.80.3.8 GetModules() . . . 261
10.80.3.9 GetTypeFromTag(const File &file, const Tag &tag) const . . . 261
10.80.3.10 IsEmpty() const . . . 261
10.80.3.11 LoadDefaults() . . . 261
10.80.3.12 LoadFromFile(const char *filename) . . . 261
10.80.3.13 Verify(const File &file) const . . . 261
10.80.3.14 Verify(const DataSet &ds) const . . . 261
10.80.4 Friends And Related Function Documentation . . . 261
10.80.4.1 Global . . . 261
10.81 gdcm::DeltaEncodingCodec Class Reference . . . 262
10.81.1 Detailed Description . . . 263
10.81.2 Constructor & Destructor Documentation . . . 263
10.81.2.1 DeltaEncodingCodec() . . . 263
10.81.2.2 ~DeltaEncodingCodec() . . . 263
10.81.3 Member Function Documentation . . . 263
10.81.3.1 CanDecode(TransferSyntax const &ts) . . . 263
10.81.3.2 Decode(DataElement const &is, DataElement &os) . . . 263
10.81.3.3 Decode(std::istream &is, std::ostream &os) . . . 263
10.82 gdcm::DICOMDIR Class Reference . . . 263

10.82.1 Detailed Description . . . 264
10.82.2 Constructor & Destructor Documentation . . . 264
10.82.2.1 DICOMDIR() . . . 264
10.82.2.2 DICOMDIR(const FileSet &fs) . . . 264
10.83gdcm::DICOMDIRGenerator Class Reference . . . 264
10.83.1 Detailed Description . . . 265
10.83.2 Member Typedef Documentation . . . 265
10.83.2.1 FilenamesType . . . 265
10.83.2.2 FilenameType . . . 265
10.83.3 Constructor & Destructor Documentation . . . 265
10.83.3.1 DICOMDIRGenerator() . . . 265
10.83.3.2 ∼DICOMDIRGenerator() . . . 265
10.83.4 Member Function Documentation . . . 265
10.83.4.1 AddImageDirectoryRecord() . . . 266
10.83.4.2 AddPatientDirectoryRecord() . . . 266
10.83.4.3 AddSeriesDirectoryRecord() . . . 266
10.83.4.4 AddStudyDirectoryRecord() . . . 266
10.83.4.5 Generate() . . . 266
10.83.4.6 GetFile() . . . 266
10.83.4.7 GetScanner() . . . 266
10.83.4.8 SetDescriptor(const char ∗d) . . . 266
10.83.4.9 SetFile(const File &f) . . . 266
10.83.4.10SetFilenames(FilenamesType const &fns) . . . 266
10.83.4.11SetRootDirectory(FilenameType const &root) . . . 266
10.84gdcm::Dict Class Reference . . . 267
10.84.1 Detailed Description . . . 267
10.84.2 Member Typedef Documentation . . . 268
10.84.2.1 ConstIterator . . . 268

10.84.2.2 Iterator . . . 268
10.84.2.3 MapDictEntry . . . 268
10.84.3 Constructor & Destructor Documentation . . . 268
10.84.3.1 Dict() . . . 268
10.84.4 Member Function Documentation . . . 268
10.84.4.1 AddDictEntry(const Tag &tag, const DictEntry &de) . . . 268
10.84.4.2 Begin() const . . . 268
10.84.4.3 End() const . . . 268
10.84.4.4 GetDictEntry(const Tag &tag) const . . . 268
10.84.4.5 GetDictEntryByKeyword(const char *keyword, Tag &tag) const . . . 268
10.84.4.6 GetDictEntryByName(const char *name, Tag &tag) const . . . 269
10.84.4.7 GetKeywordFromTag(Tag const &tag) const . . . 269
10.84.4.8 IsEmpty() const . . . 269
10.84.4.9 LoadDefault() . . . 269
10.84.5 Friends And Related Function Documentation . . . 269
10.84.5.1 Dicts . . . 269
10.84.5.2 operator<< . . . 269
10.85gdcm::DictConverter Class Reference . . . 269
10.85.1 Detailed Description . . . 270
10.85.2 Member Enumeration Documentation . . . 270
10.85.2.1 OutputTypes . . . 270
10.85.3 Constructor & Destructor Documentation . . . 271
10.85.3.1 DictConverter() . . . 271
10.85.3.2 ~DictConverter() . . . 271
10.85.4 Member Function Documentation . . . 271
10.85.4.1 AddGroupLength() . . . 271
10.85.4.2 Convert() . . . 271
10.85.4.3 ConvertToCXX(const char *raw, std::string &cxx) . . . 271

10.85.4.4 ConvertToXML(const char *raw, std::string &cxx) . . . 271
10.85.4.5 GetDictName() const . . . 271
10.85.4.6 GetInputFilename() const . . . 271
10.85.4.7 GetOutputFilename() const . . . 271
10.85.4.8 GetOutputType() const . . . 271
10.85.4.9 Readuint16(const char *raw, uint16_t &ov) . . . 271
10.85.4.10 ReadVM(const char *raw, VM::VMType &type) . . . 271
10.85.4.11 ReadVR(const char *raw, VR::VRType &type) . . . 271
10.85.4.12 SetDictName(const char *name) . . . 271
10.85.4.13 SetInputFileName(const char *filename) . . . 271
10.85.4.14 SetOutputFileName(const char *filename) . . . 271
10.85.4.15 SetOutputType(int type) . . . 271
10.85.4.16 WriteFooter() . . . 271
10.85.4.17 WriteHeader() . . . 271
10.86 gdcm::DictEntry Class Reference . . . 272
10.86.1 Detailed Description . . . 272
10.86.2 Constructor & Destructor Documentation . . . 273
10.86.2.1 DictEntry(const char *name="", const char *keyword="", VR const &vr=VR::INV↩ ALID, VM const &vm=VM::VM0, bool ret=false) . . . 273
10.86.3 Member Function Documentation . . . 273
10.86.3.1 GetKeyword() const . . . 273
10.86.3.2 GetName() const . . . 273
10.86.3.3 GetRetired() const . . . 273
10.86.3.4 GetVM() const . . . 273
10.86.3.5 GetVR() const . . . 273
10.86.3.6 IsUnique() const . . . 274
10.86.3.7 SetElementXX(bool v) . . . 274
10.86.3.8 SetGroupXX(bool v) . . . 274

10.86.3.9 SetKeyword(const char *keyword) . . . 274
10.86.3.10 SetName(const char *name) . . . 274
10.86.3.11 SetRetired(bool retired) . . . 274
10.86.3.12 SetVM(VM const &vm) . . . 274
10.86.3.13 SetVR(const VR &vr) . . . 274
10.86.4 Friends And Related Function Documentation . . . 274
10.86.4.1 Dict . . . 274
10.86.4.2 operator<< . . . 274
10.87 gdcm::DictPrinter Class Reference . . . 275
10.87.1 Detailed Description . . . 276
10.87.2 Constructor & Destructor Documentation . . . 276
10.87.2.1 DictPrinter() . . . 276
10.87.2.2 ~DictPrinter() . . . 276
10.87.3 Member Function Documentation . . . 276
10.87.3.1 Print(std::ostream &os) . . . 276
10.87.3.2 PrintDataElement2(std::ostream &os, const DataSet &ds, const DataElement &ide) . . . 276
10.87.3.3 PrintDataSet2(std::ostream &os, const DataSet &ds) . . . 276
10.88 gdcm::Dicts Class Reference . . . 276
10.88.1 Detailed Description . . . 277
10.88.2 Member Enumeration Documentation . . . 278
10.88.2.1 ConstructorType . . . 278
10.88.3 Constructor & Destructor Documentation . . . 278
10.88.3.1 Dicts() . . . 278
10.88.3.2 ~Dicts() . . . 278
10.88.4 Member Function Documentation . . . 278
10.88.4.1 GetConstructorString(ConstructorType type) . . . 278
10.88.4.2 GetCSAHeaderDict() const . . . 278
10.88.4.3 GetDictEntry(const Tag &tag, const char *owner=NULL) const . . . 278

10.88.4.4 GetDictEntry(const PrivateTag &tag) const . . . 279
10.88.4.5 GetPrivateDict() const . . . 279
10.88.4.6 GetPrivateDict() . . . 279
10.88.4.7 GetPublicDict() const . . . 279
10.88.4.8 IsEmpty() const . . . 279
10.88.4.9 LoadDefaults() . . . 279
10.88.5 Friends And Related Function Documentation . . . 279
10.88.5.1 Global . . . 279
10.88.5.2 operator<< . . . 279
10.89gdcm::network::DIMSE Class Reference . . . 279
10.89.1 Detailed Description . . . 280
10.89.2 Member Enumeration Documentation . . . 280
10.89.2.1 CommandTypes . . . 280
10.90gdcm::DirectionCosines Class Reference . . . 281
10.90.1 Detailed Description . . . 282
10.90.2 Constructor & Destructor Documentation . . . 282
10.90.2.1 DirectionCosines() . . . 282
10.90.2.2 DirectionCosines(const double dircos[6]) . . . 282
10.90.2.3 ∼DirectionCosines() . . . 282
10.90.3 Member Function Documentation . . . 282
10.90.3.1 ComputeDistAlongNormal(const double ipp[3]) const . . . 282
10.90.3.2 Cross(double z[3]) const . . . 282
10.90.3.3 CrossDot(DirectionCosines const &dc) const . . . 282
10.90.3.4 Dot() const . . . 282
10.90.3.5 IsValid() const . . . 282
10.90.3.6 Normalize() . . . 283
10.90.3.7 operator const double ∗() const . . . 283
10.90.3.8 Print(std::ostream &) const . . . 283

10.90.3.9 SetFromString(const char *str) . . . 283
10.91 gdcm::Directory Class Reference . . . 283
10.91.1 Detailed Description . . . 284
10.91.2 Member Typedef Documentation . . . 284
10.91.2.1 FilenamesType . . . 284
10.91.2.2 FilenameType . . . 285
10.91.3 Constructor & Destructor Documentation . . . 285
10.91.3.1 Directory() . . . 285
10.91.3.2 ~Directory() . . . 285
10.91.4 Member Function Documentation . . . 285
10.91.4.1 Explore(FilenameType const &name, bool recursive) . . . 285
10.91.4.2 GetDirectories() const . . . 285
10.91.4.3 GetFilenames() const . . . 285
10.91.4.4 GetToplevel() const . . . 285
10.91.4.5 Load(FilenameType const &name, bool recursive=false) . . . 285
10.91.4.6 Print(std::ostream &os=std::cout) const . . . 286
10.91.5 Friends And Related Function Documentation . . . 286
10.91.5.1 operator<< . . . 286
10.92 gdcm::DirectoryHelper Class Reference . . . 286
10.92.1 Detailed Description . . . 287
10.92.2 Member Function Documentation . . . 287
10.92.2.1 GetCTImageSeriesUIDs(const std::string &inDirectory) . . . 287
10.92.2.2 GetFilenamesFromSeriesUIDs(const std::string &inDirectory, const std::string &inSeriesUID) . . . 287
10.92.2.3 GetFrameOfReference(const std::vector< DataSet > &inDS) . . . 287
10.92.2.4 GetMRImageSeriesUIDs(const std::string &inDirectory) . . . 287
10.92.2.5 GetRTStructSeriesUIDs(const std::string &inDirectory) . . . 287
10.92.2.6 GetSeriesUIDsBySOPClassUID(const std::string &inDirectory, const std::string &inSOPClassUID) . . . 287

10.92.2.7 GetSOPClassUID(const std::vector< DataSet > &inDS) . . . 287
10.92.2.8 GetStringValueFromTag(const Tag &t, const DataSet &ds) . . . 287
10.92.2.9 LoadImageFromFiles(const std::string &inDirectory, const std::string &inSeriesUID) . . . 287
10.92.2.10 RetrieveSOPInstanceUIDFromIndex(int inIndex, const std::vector< DataSet > &in↩ DS) . . . 287
10.92.2.11 RetrieveSOPInstanceUIDFromZPosition(double inZPos, const std::vector< DataSet > &inDS) . . . 287
10.93 gdcm::DummyValueGenerator Class Reference . . . 288
10.93.1 Detailed Description . . . 288
10.93.2 Member Function Documentation . . . 288
10.93.2.1 Generate(const char *input) . . . 288
10.94 gdcm::Dumper Class Reference . . . 288
10.94.1 Detailed Description . . . 290
10.94.2 Constructor & Destructor Documentation . . . 290
10.94.2.1 Dumper() . . . 290
10.94.2.2 ~Dumper() . . . 290
10.95 gdcm::Element< TVR, TVM > Class Template Reference . . . 290
10.95.1 Detailed Description . . . 292
10.95.2 Member Typedef Documentation . . . 292
10.95.2.1 Type . . . 292
10.95.3 Member Function Documentation . . . 292
10.95.3.1 GetAsDataElement() const . . . 292
10.95.3.2 GetLength() const . . . 292
10.95.3.3 GetValue(unsigned int idx=0) const . . . 292
10.95.3.4 GetValue(unsigned int idx=0) . . . 292
10.95.3.5 GetValues() const . . . 292
10.95.3.6 GetVM() . . . 293
10.95.3.7 GetVR() . . . 293
10.95.3.8 operator[](unsigned int idx) const . . . 293

10.95.3.9 Print(std::ostream &_os) const . . . 293
10.95.3.10 Read(std::istream &_is) . . . 293
10.95.3.11 Set(Value const &v) . . . 293
10.95.3.12 SetFromDataElement(DataElement const &de) . . . 293
10.95.3.13 SetNoSwap(Value const &v) . . . 293
10.95.3.14 SetValue(typename VRToType< TVR >::Type v, unsigned int idx=0) . . . 293
10.95.3.15 Write(std::ostream &_os) const . . . 293
10.95.4 Member Data Documentation . . . 293
10.95.4.1 Internal . . . 293
10.96 gdcm::Element< TVR, VM::VM1_2 > Class Template Reference . . . 294
10.96.1 Member Typedef Documentation . . . 295
10.96.1.1 Parent . . . 295
10.96.2 Member Function Documentation . . . 295
10.96.2.1 SetLength(int len) . . . 295
10.97 gdcm::Element< TVR, VM::VM1_n > Class Template Reference . . . 295
10.97.1 Member Typedef Documentation . . . 296
10.97.1.1 Type . . . 296
10.97.2 Constructor & Destructor Documentation . . . 296
10.97.2.1 Element() . . . 296
10.97.2.2 ~Element() . . . 296
10.97.2.3 Element(const Element &_val) . . . 296
10.97.3 Member Function Documentation . . . 296
10.97.3.1 GetAsDataElement() const . . . 296
10.97.3.2 GetLength() const . . . 297
10.97.3.3 GetValue(unsigned int idx=0) const . . . 297
10.97.3.4 GetValue(unsigned int idx=0) . . . 297
10.97.3.5 GetVM() . . . 297
10.97.3.6 GetVR() . . . 297

10.97.3.7 operator=(const Element &_val) . . . 297
10.97.3.8 operator[](unsigned int idx) const . . . 297
10.97.3.9 Print(std::ostream &_os) const . . . 297
10.97.3.10 Read(std::istream &_is) . . . 297
10.97.3.11 Set(Value const &v) . . . 297
10.97.3.12 SetArray(const Type *array, unsigned long len, bool save=false) . . . 297
10.97.3.13 SetFromDataElement(DataElement const &de) . . . 297
10.97.3.14 SetLength(unsigned long len) . . . 297
10.97.3.15 SetNoSwap(Value const &v) . . . 297
10.97.3.16 SetValue(typename VRToType< TVR >::Type v, unsigned int idx=0) . . . 298
10.97.3.17 Write(std::ostream &_os) const . . . 298
10.97.3.18 WriteASCII(std::ostream &os) const . . . 298
10.98 gdcm::Element< TVR, VM::VM2_2n > Class Template Reference . . . 298
10.98.1 Member Typedef Documentation . . . 299
10.98.1.1 Parent . . . 299
10.98.2 Member Function Documentation . . . 299
10.98.2.1 SetLength(int len) . . . 299
10.99 gdcm::Element< TVR, VM::VM2_n > Class Template Reference . . . 300
10.99.1 Member Typedef Documentation . . . 301
10.99.1.1 Parent . . . 301
10.99.2 Member Function Documentation . . . 301
10.99.2.1 SetLength(int len) . . . 301
10.100 gdcm::Element< TVR, VM::VM3_3n > Class Template Reference . . . 301
10.100.1 Member Typedef Documentation . . . 302
10.100.1.1 Parent . . . 302
10.100.2 Member Function Documentation . . . 302
10.100.2.1 SetLength(int len) . . . 302
10.101 gdcm::Element< TVR, VM::VM3_n > Class Template Reference . . . 303

10.101.1 Member Typedef Documentation . . . 304
10.101.1.1 Parent . . . 304
10.101.2 Member Function Documentation . . . 304
10.101.2.1 SetLength(int len) . . . 304
10.102 gdcm::Element< VR::AS, VM::VM5 > Class Template Reference . . . 304
10.102.1 Member Function Documentation . . . 304
10.102.1.1 GetLength() const . . . 304
10.102.1.2 Print(std::ostream &_os) const . . . 304
10.102.2 Member Data Documentation . . . 304
10.102.2.1 Internal . . . 304
10.103 gdcm::Element< VR::OB, VM::VM1 > Class Template Reference . . . 305
10.104 gdcm::Element< VR::OW, VM::VM1 > Class Template Reference . . . 306
10.105 gdcm::ElementDisableCombinations< TVR, TVM > Class Template Reference . . . 308
10.105.1 Detailed Description . . . 309
10.106 gdcm::ElementDisableCombinations< VR::OB, VM::VM1_n > Class Template Reference . . . 309
10.107 gdcm::ElementDisableCombinations< VR::OW, VM::VM1_n > Class Template Reference . . . 309
10.108 gdcm::EncapsulatedDocument Class Reference . . . 309
10.108.1 Detailed Description . . . 310
10.108.2 Constructor & Destructor Documentation . . . 310
10.108.2.1 EncapsulatedDocument() . . . 310
10.109 gdcm::EncodingImplementation< T > Class Template Reference . . . 310
10.109.1 Detailed Description . . . 310
10.110 gdcm::EncodingImplementation< VR::VRASCII > Class Template Reference . . . 310
10.110.1 Member Function Documentation . . . 311
10.110.1.1 Read(T *data, unsigned long length, std::istream &_is) . . . 311
10.110.1.2 ReadComputeLength(T *data, unsigned int &length, std::istream &_is) . . . 311
10.110.1.3 ReadNoSwap(T *data, unsigned long length, std::istream &_is) . . . 311
10.110.1.4 Write(const T *data, unsigned long length, std::ostream &_os) . . . 311

10.110.1.5 Write(const float *data, unsigned long length, std::ostream &_os) . . . 311
10.110.1.6 Write(const double *data, unsigned long length, std::ostream &_os) . . . 311
10.111 gdcm::EncodingImplementation< VR::VRBINARY > Class Template Reference . . . 312
10.111.1 Member Function Documentation . . . 312
10.111.1.1 Read(T *data, unsigned long length, std::istream &_is) . . . 312
10.111.1.2 ReadComputeLength(T *data, unsigned int &length, std::istream &_is) . . . 312
10.111.1.3 ReadNoSwap(T *data, unsigned long length, std::istream &_is) . . . 312
10.111.1.4 Write(const T *data, unsigned long length, std::ostream &_os) . . . 312
10.112 gdcm::EndEvent Class Reference . . . 312
10.113 gdcm::EnumeratedValues Class Reference . . . 314
10.113.1 Detailed Description . . . 314
10.113.2 Constructor & Destructor Documentation . . . 314
10.113.2.1 EnumeratedValues() . . . 314
10.114 gdcm::Event Class Reference . . . 314
10.114.1 Detailed Description . . . 316
10.114.2 Constructor & Destructor Documentation . . . 316
10.114.2.1 Event() . . . 316
10.114.2.2 Event(const Event &) . . . 316
10.114.2.3 ~Event() . . . 316
10.114.3 Member Function Documentation . . . 316
10.114.3.1 CheckEvent(const Event *) const =0 . . . 316
10.114.3.2 GetEventName(void) const =0 . . . 316
10.114.3.3 MakeObject() const =0 . . . 316
10.114.3.4 Print(std::ostream &os) const . . . 317
10.115 gdcm::Exception Class Reference . . . 317
10.115.1 Detailed Description . . . 318
10.115.2 Constructor & Destructor Documentation . . . 318

10.115.2.1 Exception(const char *desc=""None"", const char *file=__FILE__, unsigned int lineNumber=__LINE__, const char *func="""") . . . 318
10.115.2.2 ~Exception() . . . 318
10.115.3 Member Function Documentation . . . 318
10.115.3.1 GetDescription() const . . . 318
10.115.3.2 what() const . . . 319
10.116 gdcm::ExitEvent Class Reference . . . 319
10.117 gdcm::ExplicitDataElement Class Reference . . . 320
10.117.1 Detailed Description . . . 321
10.117.2 Member Function Documentation . . . 322
10.117.2.1 GetLength() const . . . 322
10.117.2.2 Read(std::istream &is) . . . 322
10.117.2.3 ReadPreValue(std::istream &is) . . . 322
10.117.2.4 ReadValue(std::istream &is, bool readvalues=true) . . . 322
10.117.2.5 ReadWithLength(std::istream &is, VL &length) . . . 322
10.117.2.6 Write(std::ostream &os) const . . . 322
10.118 gdcm::ExplicitImplicitDataElement Class Reference . . . 322
10.118.1 Detailed Description . . . 323
10.118.2 Member Function Documentation . . . 324
10.118.2.1 GetLength() const . . . 324
10.118.2.2 Read(std::istream &is) . . . 324
10.118.2.3 ReadPreValue(std::istream &is) . . . 324
10.118.2.4 ReadValue(std::istream &is, bool readvalues=true) . . . 324
10.118.2.5 ReadWithLength(std::istream &is, VL &length) . . . 324
10.119 gdcm::Fiducials Class Reference . . . 324
10.119.1 Detailed Description . . . 324
10.119.2 Constructor & Destructor Documentation . . . 324
10.119.2.1 Fiducials() . . . 324

10.120 gdcm::File Class Reference . . . 325
10.120.1 Detailed Description . . . 326
10.120.2 Constructor & Destructor Documentation . . . 327
10.120.2.1 File() . . . 327
10.120.2.2 ~File() . . . 327
10.120.3 Member Function Documentation . . . 327
10.120.3.1 GetDataSet() const . . . 327
10.120.3.2 GetDataSet() . . . 327
10.120.3.3 GetHeader() const . . . 327
10.120.3.4 GetHeader() . . . 327
10.120.3.5 Read(std::istream &is) . . . 328
10.120.3.6 SetDataSet(const DataSet &ds) . . . 328
10.120.3.7 SetHeader(const FileMetaInformation &fmi) . . . 328
10.120.3.8 Write(std::ostream &os) const . . . 328
10.120.4 Friends And Related Function Documentation . . . 328
10.120.4.1 operator<< . . . 328
10.121 gdcm::FileAnonymizer Class Reference . . . 328
10.121.1 Detailed Description . . . 329
10.121.2 Constructor & Destructor Documentation . . . 330
10.121.2.1 FileAnonymizer() . . . 330
10.121.2.2 ~FileAnonymizer() . . . 330
10.121.3 Member Function Documentation . . . 330
10.121.3.1 Empty(Tag const &t) . . . 330
10.121.3.2 Remove(Tag const &t) . . . 330
10.121.3.3 Replace(Tag const &t, const char *value_str) . . . 330
10.121.3.4 Replace(Tag const &t, const char *value_data, VL const &vl) . . . 331
10.121.3.5 SetInputFileName(const char *filename_native) . . . 331
10.121.3.6 SetOutputFileName(const char *filename_native) . . . 331

10.121.3.7 Write() . . . 331
10.122 gdcm::FileChangeTransferSyntax Class Reference . . . 332
10.122.1 Detailed Description . . . 333
10.122.2 Constructor & Destructor Documentation . . . 333
10.122.2.1 FileChangeTransferSyntax() . . . 333
10.122.2.2 ~FileChangeTransferSyntax() . . . 333
10.122.3 Member Function Documentation . . . 333
10.122.3.1 Change() . . . 333
10.122.3.2 GetCodec() . . . 334
10.122.3.3 New() . . . 334
10.122.3.4 SetInputFileName(const char *filename_native) . . . 334
10.122.3.5 SetOutputFileName(const char *filename_native) . . . 334
10.122.3.6 SetTransferSyntax(TransferSyntax const &ts) . . . 334
10.123 gdcm::FileDecompressLookupTable Class Reference . . . 334
10.123.1 Detailed Description . . . 336
10.123.2 Constructor & Destructor Documentation . . . 336
10.123.2.1 FileDecompressLookupTable() . . . 336
10.123.2.2 ~FileDecompressLookupTable() . . . 336
10.123.3 Member Function Documentation . . . 336
10.123.3.1 Change() . . . 336
10.123.3.2 GetFile() . . . 336
10.123.3.3 GetPixmap() const . . . 336
10.123.3.4 GetPixmap() . . . 336
10.123.3.5 SetFile(const File &f) . . . 336
10.123.3.6 SetPixmap(Pixmap const &img) . . . 336
10.124 gdcm::FileDerivation Class Reference . . . 337
10.124.1 Detailed Description . . . 337
10.124.2 Constructor & Destructor Documentation . . . 338

10.124.2.1 FileDerivation() . . . 338
10.124.2.2 ~FileDerivation() . . . 338
10.124.3 Member Function Documentation . . . 338
10.124.3.1 AddDerivationDescription() . . . 338
10.124.3.2 AddPurposeOfReferenceCodeSequence(DataSet &ds) . . . 338
10.124.3.3 AddReference(const char *referencedsopclassuid, const char *referencedsopinstanceuid) 338
10.124.3.4 AddSourceImageSequence() . . . 338
10.124.3.5 Derive() . . . 338
10.124.3.6 GetFile() . . . 338
10.124.3.7 GetFile() const . . . 339
10.124.3.8 SetDerivationCodeSequenceCodeValue(unsigned int codevalue) . . . 339
10.124.3.9 SetDerivationDescription(const char *dd) . . . 339
10.124.3.10 SetFile(const File &f) . . . 339
10.124.3.11 SetPurposeOfReferenceCodeSequenceCodeValue(unsigned int codevalue) . . . 339
10.125 gdcm::FileExplicitFilter Class Reference . . . 339
10.125.1 Detailed Description . . . 340
10.125.2 Constructor & Destructor Documentation . . . 341
10.125.2.1 FileExplicitFilter() . . . 341
10.125.2.2 ~FileExplicitFilter() . . . 341
10.125.3 Member Function Documentation . . . 341
10.125.3.1 Change() . . . 341
10.125.3.2 ChangeFMI() . . . 341
10.125.3.3 GetFile() . . . 341
10.125.3.4 ProcessDataSet(DataSet &ds, Dicts const &dicts) . . . 341
10.125.3.5 SetChangePrivateTags(bool b) . . . 341
10.125.3.6 SetFile(const File &f) . . . 341
10.125.3.7 SetRecomputeItemLength(bool b) . . . 341
10.125.3.8 SetRecomputeSequenceLength(bool b) . . . 342

10.125.3.9 SetUseVRUN(bool b) . . . 342
10.126 gdcm::FileMetaInformation Class Reference . . . 342
10.126.1 Detailed Description . . . 344
10.126.2 Constructor & Destructor Documentation . . . 344
10.126.2.1 FileMetaInformation() . . . 344
10.126.2.2 ~FileMetaInformation() . . . 344
10.126.2.3 FileMetaInformation(FileMetaInformation const &fmi) . . . 344
10.126.3 Member Function Documentation . . . 345
10.126.3.1 AppendImplementationClassUID(const char *imp) . . . 345
10.126.3.2 ComputeDataSetMediaStorageSOPClass() . . . 345
10.126.3.3 ComputeDataSetTransferSyntax() . . . 345
10.126.3.4 Default() . . . 345
10.126.3.5 FillFromDataSet(DataSet const &ds) . . . 345
10.126.3.6 GetDataSetTransferSyntax() const . . . 345
10.126.3.7 GetFileMetaInformationVersion() . . . 345
10.126.3.8 GetFullLength() const . . . 345
10.126.3.9 GetGDCMImplementationClassUID() . . . 345
10.126.3.10 GetGDCMImplementationVersionName() . . . 345
10.126.3.11 GetGDCMSourceApplicationEntityTitle() . . . 345
10.126.3.12 GetImplementationClassUID() . . . 345
10.126.3.13 GetImplementationVersionName() . . . 345
10.126.3.14 GetMediaStorage() const . . . 345
10.126.3.15 GetMediaStorageAsString() const . . . 345
10.126.3.16 GetMetaInformationTS() const . . . 345
10.126.3.17 GetPreamble() const . . . 345
10.126.3.18 GetPreamble() . . . 346
10.126.3.19 GetSourceApplicationEntityTitle() . . . 346
10.126.3.20 Insert(const DataElement &de) . . . 346

10.126.3.21 IsValid() const . . . 346
10.126.3.22 Read(std::istream &is) . . . 346
10.126.3.23 ReadCompat(std::istream &is) . . . 346
10.126.3.24 ReadCompatInternal(std::istream &is) . . . 346
10.126.3.25 Replace(const DataElement &de) . . . 346
10.126.3.26 SetDataSetTransferSyntax(const TransferSyntax &ts) . . . 346
10.126.3.27 SetImplementationClassUID(const char *imp) . . . 346
10.126.3.28 SetImplementationVersionName(const char *version) . . . 347
10.126.3.29 SetPreamble(const Preamble &p) . . . 347
10.126.3.30 SetSourceApplicationEntityTitle(const char *title) . . . 347
10.126.3.31 Write(std::ostream &os) const . . . 347
10.126.4 Friends And Related Function Documentation . . . 347
10.126.4.1 operator<< . . . 347
10.126.5 Member Data Documentation . . . 347
10.126.5.1 DataSetMS . . . 347
10.126.5.2 DataSetTS . . . 347
10.126.5.3 MetaInformationTS . . . 347
10.127 gdcm::Filename Class Reference . . . 347
10.127.1 Detailed Description . . . 348
10.127.2 Constructor & Destructor Documentation . . . 348
10.127.2.1 Filename(const char *filename="") . . . 348
10.127.3 Member Function Documentation . . . 348
10.127.3.1 EndWith(const char ending[]) const . . . 348
10.127.3.2 GetExtension() . . . 349
10.127.3.3 GetFileName() const . . . 349
10.127.3.4 GetName() . . . 349
10.127.3.5 GetPath() . . . 349
10.127.3.6 IsEmpty() const . . . 349

10.127.3.7 IsIdentical(Filename const &fn) const . . . 349
10.127.3.8 Join(const char *path, const char *filename) . . . 349
10.127.3.9 operator const char *() const . . . 349
10.127.3.10 ToUnixSlashes() . . . 350
10.127.3.11 ToWindowsSlashes() . . . 350
10.128 gdcm::FileNameEvent Class Reference . . . 350
10.128.1 Detailed Description . . . 351
10.128.2 Member Typedef Documentation . . . 352
10.128.2.1 Self . . . 352
10.128.2.2 Superclass . . . 352
10.128.3 Constructor & Destructor Documentation . . . 352
10.128.3.1 FileNameEvent(const char *s="") . . . 352
10.128.3.2 ~FileNameEvent() . . . 352
10.128.3.3 FileNameEvent(const Self &s) . . . 352
10.128.4 Member Function Documentation . . . 352
10.128.4.1 CheckEvent(const ::gdcm::Event *e) const . . . 352
10.128.4.2 GetEventName() const . . . 352
10.128.4.3 GetFileName() const . . . 352
10.128.4.4 MakeObject() const . . . 352
10.128.4.5 SetFileName(const char *f) . . . 352
10.129 gdcm::FilenameGenerator Class Reference . . . 353
10.129.1 Detailed Description . . . 353
10.129.2 Member Typedef Documentation . . . 354
10.129.2.1 FilenamesType . . . 354
10.129.2.2 FilenameType . . . 354
10.129.2.3 SizeType . . . 354
10.129.3 Constructor & Destructor Documentation . . . 354
10.129.3.1 FilenameGenerator() . . . 354

10.129.3.2 ~FilenameGenerator() . . . 354
10.129.4 Member Function Documentation . . . 354
10.129.4.1 Generate() . . . 354
10.129.4.2 GetFilename(SizeType n) const . . . 354
10.129.4.3 GetFilenames() const . . . 354
10.129.4.4 GetNumberOfFilenames() const . . . 354
10.129.4.5 GetPattern() const . . . 355
10.129.4.6 GetPrefix() const . . . 355
10.129.4.7 SetNumberOfFilenames(SizeType nfiles) . . . 355
10.129.4.8 SetPattern(const char *pattern) . . . 355
10.129.4.9 SetPrefix(const char *prefix) . . . 355
10.130 gdcm::FileSet Class Reference . . . 355
10.130.1 Detailed Description . . . 356
10.130.2 Member Typedef Documentation . . . 356
10.130.2.1 FilesType . . . 356
10.130.2.2 FileType . . . 356
10.130.3 Constructor & Destructor Documentation . . . 356
10.130.3.1 FileSet() . . . 356
10.130.4 Member Function Documentation . . . 356
10.130.4.1 AddFile(File const &) . . . 356
10.130.4.2 AddFile(const char *filename) . . . 356
10.130.4.3 GetFiles() const . . . 357
10.130.4.4 SetFiles(FilesType const &files) . . . 357
10.130.5 Friends And Related Function Documentation . . . 357
10.130.5.1 operator<< . . . 357
10.131 gdcm::FileStreamer Class Reference . . . 357
10.131.1 Detailed Description . . . 359
10.131.2 Constructor & Destructor Documentation . . . 359

10.131.2.1 FileStreamer() . . . 359
10.131.2.2 ~FileStreamer() . . . 359
10.131.3 Member Function Documentation . . . 359
10.131.3.1 AppendToDataElement(const Tag &t, const char *array, size_t len) . . . 359
10.131.3.2 AppendToGroupDataElement(const PrivateTag &pt, const char *array, size_t len) . . . 359
10.131.3.3 CheckDataElement(const Tag &t) . . . 359
10.131.3.4 CheckTemplateFileName(bool check) . . . 360
10.131.3.5 New() . . . 360
10.131.3.6 ReserveDataElement(size_t len) . . . 360
10.131.3.7 ReserveGroupDataElement(unsigned short ndataelement) . . . 360
10.131.3.8 SetOutputFileName(const char *filename_native) . . . 360
10.131.3.9 SetTemplateFileName(const char *filename_native) . . . 360
10.131.3.10 StartDataElement(const Tag &t) . . . 360
10.131.3.11 StartGroupDataElement(const PrivateTag &pt, size_t maxsizede=0, uint8_t startoffset=0) . . . 360
10.131.3.12 StopDataElement(const Tag &t) . . . 361
10.131.3.13 StopGroupDataElement(const PrivateTag &pt) . . . 361
10.132 gdcm::FileWithName Class Reference . . . 361
10.132.1 Detailed Description . . . 362
10.132.2 Constructor & Destructor Documentation . . . 362
10.132.2.1 FileWithName(File &f) . . . 362
10.132.3 Member Data Documentation . . . 362
10.132.3.1 filename . . . 362
10.133 gdcm::FindPatientRootQuery Class Reference . . . 363
10.133.1 Detailed Description . . . 364
10.133.2 Constructor & Destructor Documentation . . . 364
10.133.2.1 FindPatientRootQuery() . . . 364
10.133.3 Member Function Documentation . . . 364

10.133.3.1 GetAbstractSyntaxUID() const . . . 364
10.133.3.2 GetTagListByLevel(const EQueryLevel &inQueryLevel) . . . 364
10.133.3.3 InitializeDataSet(const EQueryLevel &inQueryLevel) . . . 364
10.133.3.4 ValidateQuery(bool inStrict=true) const . . . 364
10.133.4 Friends And Related Function Documentation . . . 365
10.133.4.1 QueryFactory . . . 365
10.134 gdcm::FindStudyRootQuery Class Reference . . . 365
10.134.1 Detailed Description . . . 366
10.134.2 Constructor & Destructor Documentation . . . 366
10.134.2.1 FindStudyRootQuery() . . . 366
10.134.3 Member Function Documentation . . . 366
10.134.3.1 GetAbstractSyntaxUID() const . . . 366
10.134.3.2 GetTagListByLevel(const EQueryLevel &inQueryLevel) . . . 367
10.134.3.3 InitializeDataSet(const EQueryLevel &inQueryLevel) . . . 367
10.134.3.4 ValidateQuery(bool inStrict=true) const . . . 367
10.134.4 Friends And Related Function Documentation . . . 367
10.134.4.1 QueryFactory . . . 367
10.135 gdcm::Fragment Class Reference . . . 367
10.135.1 Detailed Description . . . 369
10.135.2 Constructor & Destructor Documentation . . . 369
10.135.2.1 Fragment() . . . 369
10.135.3 Member Function Documentation . . . 369
10.135.3.1 ComputeLength() const . . . 369
10.135.3.2 GetLength() const . . . 369
10.135.3.3 Read(std::istream &is) . . . 369
10.135.3.4 ReadBacktrack(std::istream &is) . . . 369
10.135.3.5 ReadPreValue(std::istream &is) . . . 370
10.135.3.6 ReadValue(std::istream &is) . . . 370

10.135.3.7 Write(std::ostream &os) const . . . 370
10.135.4 Friends And Related Function Documentation . . . 370
10.135.4.1 operator<< . . . 370
10.136 gdcm::Global Class Reference . . . 370
10.136.1 Detailed Description . . . 371
10.136.2 Constructor & Destructor Documentation . . . 371
10.136.2.1 Global() . . . 371
10.136.2.2 ~Global() . . . 371
10.136.3 Member Function Documentation . . . 371
10.136.3.1 Append(const char *path) . . . 371
10.136.3.2 GetDefs() const . . . 371
10.136.3.3 GetDicts() const . . . 372
10.136.3.4 GetDicts() . . . 372
10.136.3.5 GetInstance() . . . 372
10.136.3.6 LoadResourcesFiles() . . . 372
10.136.3.7 Locate(const char *resfile) const . . . 372
10.136.3.8 Prepend(const char *path) . . . 372
10.136.4 Friends And Related Function Documentation . . . 373
10.136.4.1 operator<< . . . 373
10.137 gdcm::GroupDict Class Reference . . . 373
10.137.1 Detailed Description . . . 373
10.137.2 Member Typedef Documentation . . . 374
10.137.2.1 GroupStringVector . . . 374
10.137.3 Constructor & Destructor Documentation . . . 374
10.137.3.1 GroupDict() . . . 374
10.137.3.2 ~GroupDict() . . . 374
10.137.4 Member Function Documentation . . . 374
10.137.4.1 Add(std::string const &abbreviation, std::string const &name) . . . 374

10.137.4.2 GetAbbreviation(uint16_t num) const . . . 374
10.137.4.3 GetName(uint16_t num) const . . . 374
10.137.4.4 Insert(uint16_t num, std::string const &abbreviation, std::string const &name) . . . 374
10.137.4.5 Size() const . . . 374
10.137.5 Friends And Related Function Documentation . . . 374
10.137.5.1 operator<< . . . 374
10.138 gdcm::IconImageFilter Class Reference . . . 375
10.138.1 Detailed Description . . . 375
10.138.2 Constructor & Destructor Documentation . . . 376
10.138.2.1 IconImageFilter() . . . 376
10.138.2.2 ~IconImageFilter() . . . 376
10.138.3 Member Function Documentation . . . 376
10.138.3.1 Extract() . . . 376
10.138.3.2 ExtractIconImages() . . . 376
10.138.3.3 ExtractVeproIconImages() . . . 376
10.138.3.4 GetFile() . . . 376
10.138.3.5 GetFile() const . . . 376
10.138.3.6 GetIconImage(unsigned int i) const . . . 376
10.138.3.7 GetNumberOfIconImages() const . . . 377
10.138.3.8 SetFile(const File &f) . . . 377
10.139 gdcm::IconImageGenerator Class Reference . . . 377
10.139.1 Detailed Description . . . 378
10.139.2 Constructor & Destructor Documentation . . . 378
10.139.2.1 IconImageGenerator() . . . 378
10.139.2.2 ~IconImageGenerator() . . . 378
10.139.3 Member Function Documentation . . . 378
10.139.3.1 AutoPixelMinMax(bool b) . . . 378
10.139.3.2 ConvertRGBToPaletteColor(bool b) . . . 378

10.139.3.3 Generate() . . . 379
10.139.3.4 GetIconImage() const . . . 379
10.139.3.5 GetPixmap() . . . 379
10.139.3.6 GetPixmap() const . . . 379
10.139.3.7 SetOutputDimensions(const unsigned int dims[2]) . . . 379
10.139.3.8 SetOutsideValuePixel(double v) . . . 379
10.139.3.9 SetPixelMinMax(double min, double max) . . . 379
10.139.3.10 SetPixmap(const Pixmap &p) . . . 380
10.140 gdcm::ignore_char Struct Reference . . . 380
10.140.1 Constructor & Destructor Documentation . . . 380
10.140.1.1 ignore_char(char c) . . . 380
10.140.2 Member Data Documentation . . . 380
10.140.2.1 m_char . . . 380
10.141 gdcm::Image Class Reference . . . 381
10.141.1 Detailed Description . . . 382
10.141.2 Constructor & Destructor Documentation . . . 383
10.141.2.1 Image() . . . 383
10.141.2.2 ~Image() . . . 383
10.141.3 Member Function Documentation . . . 383
10.141.3.1 GetDirectionCosines() const . . . 383
10.141.3.2 GetDirectionCosines(unsigned int idx) const . . . 383
10.141.3.3 GetIntercept() const . . . 383
10.141.3.4 GetOrigin() const . . . 383
10.141.3.5 GetOrigin(unsigned int idx) const . . . 383
10.141.3.6 GetSlope() const . . . 383
10.141.3.7 GetSpacing() const . . . 383
10.141.3.8 GetSpacing(unsigned int idx) const . . . 384
10.141.3.9 Print(std::ostream &os) const . . . 384

10.141.3.11SetDirectionCosines(const float *dircos) . . . 384
10.141.3.12SetDirectionCosines(const double *dircos) . . . 384
10.141.3.13SetDirectionCosines(unsigned int idx, double dircos) . . . 384
10.141.3.14SetIntercept(double intercept) . . . 384
10.141.3.15SetOrigin(const float *ori) . . . 384
10.141.3.16SetOrigin(const double *ori) . . . 384
10.141.3.17SetOrigin(unsigned int idx, double ori) . . . 384
10.141.3.18SetSlope(double slope) . . . 384
10.141.3.19SetSpacing(const double *spacing) . . . 384
10.141.3.20SetSpacing(unsigned int idx, double spacing) . . . 384
10.142gdcm::ImageApplyLookupTable Class Reference . . . 385
10.142.1Detailed Description . . . 386
10.142.2Constructor & Destructor Documentation . . . 387
10.142.2.1ImageApplyLookupTable() . . . 387
10.142.2.2~ImageApplyLookupTable() . . . 387
10.142.3Member Function Documentation . . . 387
10.142.3.1Apply() . . . 387
10.143gdcm::ImageChangePhotometricInterpretation Class Reference . . . 387
10.143.1Detailed Description . . . 389
10.143.2Constructor & Destructor Documentation . . . 389
10.143.2.1ImageChangePhotometricInterpretation() . . . 389
10.143.2.2~ImageChangePhotometricInterpretation() . . . 389
10.143.3Member Function Documentation . . . 389
10.143.3.1Change() . . . 389
10.143.3.2ChangeMonochrome() . . . 389
10.143.3.3GetPhotometricInterpretation() const . . . 389
10.143.3.4RGB2YBR(T ybr[3], const T rgb[3]) . . . 389
10.143.3.5SetPhotometricInterpretation(PhotometricInterpretation const &pi) . . . 390

10.143.3.6 YBR2RGB(T rgb[3], const T ybr[3]) . . . 390
10.144 gdcm::ImageChangePlanarConfiguration Class Reference . . . 390
10.144.1 Detailed Description . . . 392
10.144.2 Constructor & Destructor Documentation . . . 392
10.144.2.1 ImageChangePlanarConfiguration() . . . 392
10.144.2.2 ~ImageChangePlanarConfiguration() . . . 392
10.144.3 Member Function Documentation . . . 392
10.144.3.1 Change() . . . 392
10.144.3.2 GetPlanarConfiguration() const . . . 392
10.144.3.3 RGBPixelsToRGBPlanes(T *r, T *g, T *b, const T *rgb, size_t s) . . . 392
10.144.3.4 RGBPlanesToRGBPixels(T *out, const T *r, const T *g, const T *b, size_t s) . . . 392
10.144.3.5 SetPlanarConfiguration(unsigned int pc) . . . 393
10.145 gdcm::ImageChangeTransferSyntax Class Reference . . . 393
10.145.1 Detailed Description . . . 395
10.145.2 Constructor & Destructor Documentation . . . 395
10.145.2.1 ImageChangeTransferSyntax() . . . 395
10.145.2.2 ~ImageChangeTransferSyntax() . . . 395
10.145.3 Member Function Documentation . . . 395
10.145.3.1 Change() . . . 395
10.145.3.2 GetTransferSyntax() const . . . 396
10.145.3.3 SetCompressIconImage(bool b) . . . 396
10.145.3.4 SetForce(bool f) . . . 396
10.145.3.5 SetTransferSyntax(const TransferSyntax &ts) . . . 396
10.145.3.6 SetUserCodec(ImageCodec *ic) . . . 396
10.145.3.7 TryJPEG2000Codec(const DataElement &pixelde, Bitmap const &input, Bitmap &output) . . . 396
10.145.3.8 TryJPEGCodec(const DataElement &pixelde, Bitmap const &input, Bitmap &output) . . . 396
10.145.3.9 TryJPEGLSCodec(const DataElement &pixelde, Bitmap const &input, Bitmap &output) 396

10.145.3.17TryRAWCodec(const DataElement &pixelde, Bitmap const &input, Bitmap &output) . . 396
10.145.3.18TryRLECodec(const DataElement &pixelde, Bitmap const &input, Bitmap &output) . . 396
10.146gdcm::ImageCodec Class Reference 397
10.146.1Detailed Description 399
10.146.2Member Typedef Documentation 399
10.146.2.1LUTPtr 399
10.146.3Constructor & Destructor Documentation 399
10.146.3.1ImageCodec() 399
10.146.3.2~ImageCodec() 399
10.146.4Member Function Documentation 399
10.146.4.1AppendFrameEncode(std::ostream &out, const char *data, size_t datalen) 399
10.146.4.2AppendRowEncode(std::ostream &out, const char *data, size_t datalen) 400
10.146.4.3CanCode(TransferSyntax const &) const 400
10.146.4.4CanDecode(TransferSyntax const &) const 400
10.146.4.5Clone() const =0 400
10.146.4.6Decode(DataElement const &is_, DataElement &os) 400
10.146.4.7DecodeByStreams(std::istream &is_, std::ostream &os) 400
10.146.4.8DoByteSwap(std::istream &is_, std::ostream &os) 401
10.146.4.9DoInvertMonochrome(std::istream &is_, std::ostream &os) 401
10.146.4.10DoOverlayCleanup(std::istream &is_, std::ostream &os) 401
10.146.4.11DoPaddedCompositePixelCode(std::istream &is_, std::ostream &os) 401
10.146.4.12DoPlanarConfiguration(std::istream &is_, std::ostream &os) 401
10.146.4.13DoSimpleCopy(std::istream &is_, std::ostream &os) 401
10.146.4.14DoYBR(std::istream &is_, std::ostream &os) 401
10.146.4.15GetDimensions() const 401
10.146.4.16GetHeaderInfo(std::istream &is_, TransferSyntax &ts) 401
10.146.4.17GetLossyFlag() const 401
10.146.4.18GetLUT() const 401

10.146.4.19 GetNeedByteSwap() const . . . 401
10.146.4.20 GetNumberOfDimensions() const . . . 401
10.146.4.21 GetPhotometricInterpretation() const . . . 401
10.146.4.22 GetPixelFormat() . . . 401
10.146.4.23 GetPixelFormat() const . . . 402
10.146.4.24 GetPlanarConfiguration() const . . . 402
10.146.4.25 IsFrameEncoder() . . . 402
10.146.4.26 IsLossy() const . . . 402
10.146.4.27 IsRowEncoder() . . . 402
10.146.4.28 IsValid(PhotometricInterpretation const &pi) . . . 402
10.146.4.29 SetDimensions(const unsigned int d[3]) . . . 402
10.146.4.30 SetDimensions(const std::vector< unsigned int > &d) . . . 402
10.146.4.31 SetLossyFlag(bool l) . . . 402
10.146.4.32 SetLUT(LookupTable const &lut) . . . 402
10.146.4.33 SetNeedByteSwap(bool b) . . . 402
10.146.4.34 SetNeedOverlayCleanup(bool b) . . . 402
10.146.4.35 SetNumberOfDimensions(unsigned int dim) . . . 402
10.146.4.36 SetPhotometricInterpretation(PhotometricInterpretation const &pi) . . . 402
10.146.4.37 SetPixelFormat(PixelFormat const &pf) . . . 403
10.146.4.38 SetPlanarConfiguration(unsigned int pc) . . . 403
10.146.4.39 StartEncode(std::ostream &os) . . . 403
10.146.4.40 StopEncode(std::ostream &os) . . . 403
10.146.5 Friends And Related Function Documentation . . . 403
10.146.5.1 FileChangeTransferSyntax . . . 403
10.146.5.2 ImageChangePhotometricInterpretation . . . 403
10.146.6 Member Data Documentation . . . 403
10.146.6.1 Dimensions . . . 403
10.146.6.2 LossyFlag . . . 403

10.146.6.3 LUT . . . 403
10.146.6.4 NeedByteSwap . . . 403
10.146.6.5 NeedOverlayCleanup . . . 404
10.146.6.6 NumberOfDimensions . . . 404
10.146.6.7 PF . . . 404
10.146.6.8 PI . . . 404
10.146.6.9 PlanarConfiguration . . . 404
10.146.6.10 RequestPaddedCompositePixelCode . . . 404
10.146.6.11 RequestPlanarConfiguration . . . 404
10.147 gdcm::ImageConverter Class Reference . . . 404
10.147.1 Detailed Description . . . 404
10.147.2 Constructor & Destructor Documentation . . . 405
10.147.2.1 ImageConverter() . . . 405
10.147.2.2 ~ImageConverter() . . . 405
10.147.3 Member Function Documentation . . . 405
10.147.3.1 Convert() . . . 405
10.147.3.2 GetOuput() const . . . 405
10.147.3.3 SetInput(Image const &input) . . . 405
10.148 gdcm::ImageFragmentSplitter Class Reference . . . 405
10.148.1 Detailed Description . . . 407
10.148.2 Constructor & Destructor Documentation . . . 407
10.148.2.1 ImageFragmentSplitter() . . . 407
10.148.2.2 ~ImageFragmentSplitter() . . . 407
10.148.3 Member Function Documentation . . . 407
10.148.3.1 GetFragmentSizeMax() const . . . 407
10.148.3.2 SetForce(bool f) . . . 407
10.148.3.3 SetFragmentSizeMax(unsigned int fragsize) . . . 407
10.148.3.4 Split() . . . 407

10.149gdcm::ImageHelper Class Reference . . . 407
10.149.1Detailed Description . . . 408
10.149.2Member Function Documentation . . . 409
10.149.2.1ComputeMediaStorageFromModality(const char *modality, unsigned int dimension=2, PixelFormat const &pf=PixelFormat(), PhotometricInterpretation const &pi=PhotometricInterpretation(), double rescaleintercept=0, double rescaleslope=1) . . . 409
10.149.2.2ComputeSpacingFromImagePositionPatient(const std::vector< double > &imageposition, std::vector< double > &spacing) . . . 409
10.149.2.3GetDimensionsValue(const File &f) . . . 409
10.149.2.4GetDirectionCosinesFromDataSet(DataSet const &ds, std::vector< double > &dircos) 409
10.149.2.5GetDirectionCosinesValue(File const &f) . . . 409
10.149.2.6GetForcePixelSpacing() . . . 409
10.149.2.7GetForceRescaleInterceptSlope() . . . 409
10.149.2.8GetLUT(File const &f) . . . 409
10.149.2.9GetOriginValue(File const &f) . . . 409
10.149.2.10GetPhotometricInterpretationValue(File const &f) . . . 410
10.149.2.11GetPixelFormatValue(const File &f) . . . 410
10.149.2.12GetPlanarConfigurationValue(const File &f) . . . 410
10.149.2.13GetPMSRescaleInterceptSlope() . . . 410
10.149.2.14GetPointerFromElement(Tag const &tag, File const &f) . . . 410
10.149.2.15GetRealWorldValueMappingContent(File const &f, RealWorldValueMappingContent &rwvmc) . . . 410
10.149.2.16GetRescaleInterceptSlopeValue(File const &f) . . . 410
10.149.2.17GetSpacingTagFromMediaStorage(MediaStorage const &ms) . . . 410
10.149.2.18GetSpacingValue(File const &f) . . . 410
10.149.2.19GetZSpacingTagFromMediaStorage(MediaStorage const &ms) . . . 410
10.149.2.20SetDimensionsValue(File &f, const Pixmap &img) . . . 410
10.149.2.21SetDirectionCosinesValue(DataSet &ds, const std::vector< double > &dircos) . . . 410
10.149.2.22SetForcePixelSpacing(bool) . . . 411
10.149.2.23SetForceRescaleInterceptSlope(bool) . . . 411

10.149.2.24 SetOriginValue(DataSet &ds, const Image &img) . . . 411
10.149.2.25 SetPMSRescaleInterceptSlope(bool) . . . 411
10.149.2.26 SetRescaleInterceptSlopeValue(File &f, const Image &img) . . . 411
10.149.2.27 SetSpacingValue(DataSet &ds, const std::vector< double > &spacing) . . . 411
10.150 gdcm::ImageReader Class Reference . . . 411
10.150.1 Detailed Description . . . 414
10.150.2 Constructor & Destructor Documentation . . . 414
10.150.2.1 ImageReader() . . . 414
10.150.2.2 ~ImageReader() . . . 414
10.150.3 Member Function Documentation . . . 414
10.150.3.1 GetImage() const . . . 414
10.150.3.2 GetImage() . . . 415
10.150.3.3 Read() . . . 415
10.150.3.4 ReadACRNEMAImage() . . . 415
10.150.3.5 ReadImage(MediaStorage const &ms) . . . 415
10.151 gdcm::ImageRegionReader Class Reference . . . 416
10.151.1 Detailed Description . . . 418
10.151.2 Constructor & Destructor Documentation . . . 418
10.151.2.1 ImageRegionReader() . . . 418
10.151.2.2 ~ImageRegionReader() . . . 418
10.151.3 Member Function Documentation . . . 418
10.151.3.1 ComputeBufferLength() const . . . 418
10.151.3.2 GetRegion() const . . . 418
10.151.3.3 Read() . . . 418
10.151.3.4 ReadInformation() . . . 419
10.151.3.5 ReadIntoBuffer(char *inreadbuffer, size_t buflen) . . . 419
10.151.3.6 SetRegion(Region const &region) . . . 419
10.152 gdcm::ImageToImageFilter Class Reference . . . 419

10.152.1 Detailed Description . . . 420
10.152.2 Constructor & Destructor Documentation . . . 421
10.152.2.1 ImageToImageFilter() . . . 421
10.152.2.2 ~ImageToImageFilter() . . . 421
10.152.3 Member Function Documentation . . . 421
10.152.3.1 GetInput() . . . 421
10.152.3.2 GetOutput() const . . . 421
10.153 gdcm::ImageWriter Class Reference . . . 421
10.153.1 Detailed Description . . . 423
10.153.2 Constructor & Destructor Documentation . . . 423
10.153.2.1 ImageWriter() . . . 423
10.153.2.2 ~ImageWriter() . . . 423
10.153.3 Member Function Documentation . . . 423
10.153.3.1 ComputeTargetMediaStorage() . . . 423
10.153.3.2 GetImage() const . . . 423
10.153.3.3 GetImage() . . . 423
10.153.3.4 Write() . . . 424
10.154 gdcm::network::ImplementationClassUIDSub Class Reference . . . 424
10.154.1 Detailed Description . . . 424
10.154.2 Constructor & Destructor Documentation . . . 424
10.154.2.1 ImplementationClassUIDSub() . . . 424
10.154.3 Member Function Documentation . . . 424
10.154.3.1 Print(std::ostream &os) const . . . 424
10.154.3.2 Read(std::istream &is) . . . 424
10.154.3.3 Size() const . . . 424
10.154.3.4 Write(std::ostream &os) const . . . 424
10.155 gdcm::network::ImplementationUIDSub Class Reference . . . 425
10.155.1 Detailed Description . . . 425

10.155.2 Constructor & Destructor Documentation . . . 425
10.155.2.1 ImplementationUIDSub() . . . 425
10.155.3 Member Function Documentation . . . 425
10.155.3.1 Write(std::ostream &os) const . . . 425
10.156 gdcm::network::ImplementationVersionNameSub Class Reference . . . 425
10.156.1 Detailed Description . . . 426
10.156.2 Constructor & Destructor Documentation . . . 426
10.156.2.1 ImplementationVersionNameSub() . . . 426
10.156.3 Member Function Documentation . . . 426
10.156.3.1 Print(std::ostream &os) const . . . 426
10.156.3.2 Read(std::istream &is) . . . 426
10.156.3.3 Size() const . . . 426
10.156.3.4 Write(std::ostream &os) const . . . 426
10.157 gdcm::ImplicitDataElement Class Reference . . . 426
10.157.1 Detailed Description . . . 427
10.157.2 Member Function Documentation . . . 428
10.157.2.1 GetLength() const . . . 428
10.157.2.2 Read(std::istream &is) . . . 428
10.157.2.3 ReadPreValue(std::istream &is) . . . 428
10.157.2.4 ReadValue(std::istream &is, bool readvalues=true) . . . 428
10.157.2.5 ReadValueWithLength(std::istream &is, VL &length, bool readvalues=true) . . . 428
10.157.2.6 ReadWithLength(std::istream &is, VL &length, bool readvalues=true) . . . 428
10.157.2.7 Write(std::ostream &os) const . . . 428
10.158 gdcm::InitializeEvent Class Reference . . . 428
10.159 gdcm::IOD Class Reference . . . 429
10.159.1 Detailed Description . . . 430
10.159.2 Member Typedef Documentation . . . 430
10.159.2.1 MapIODEntry . . . 430

10.159.2.2 SizeType . . . 430
10.159.3 Constructor & Destructor Documentation . . . 430
10.159.3.1 IOD() . . . 430
10.159.4 Member Function Documentation . . . 430
10.159.4.1 AddIODEntry(const IODEntry &iode) . . . 430
10.159.4.2 Clear() . . . 430
10.159.4.3 GetIODEntry(SizeType idx) const . . . 430
10.159.4.4 GetNumberOfIODs() const . . . 431
10.159.4.5 GetTypeFromTag(const Defs &defs, const Tag &tag) const . . . 431
10.159.5 Friends And Related Function Documentation . . . 431
10.159.5.1 operator<< . . . 431
10.160 gdcm::IODEntry Class Reference . . . 431
10.160.1 Detailed Description . . . 432
10.160.2 Constructor & Destructor Documentation . . . 432
10.160.2.1 IODEntry(const char *name="", const char *ref="", const char *usag="") . . . 432
10.160.3 Member Function Documentation . . . 432
10.160.3.1 GetIE() const . . . 432
10.160.3.2 GetName() const . . . 432
10.160.3.3 GetRef() const . . . 432
10.160.3.4 GetUsage() const . . . 433
10.160.3.5 GetUsageType() const . . . 433
10.160.3.6 SetIE(const char *ie) . . . 433
10.160.3.7 SetName(const char *name) . . . 433
10.160.3.8 SetRef(const char *ref) . . . 433
10.160.3.9 SetUsage(const char *usag) . . . 433
10.160.4 Friends And Related Function Documentation . . . 433
10.160.4.1 operator<< . . . 433
10.161 gdcm::IODs Class Reference . . . 433

10.161.1 Detailed Description . . . 434
10.161.2 Member Typedef Documentation . . . 434
10.161.2.1 IODMapType . . . 434
10.161.2.2 IODMapTypeConstIterator . . . 434
10.161.2.3 IODName . . . 434
10.161.3 Constructor & Destructor Documentation . . . 434
10.161.3.1 IODs() . . . 434
10.161.4 Member Function Documentation . . . 434
10.161.4.1 AddIOD(const char *name, const IOD &module) . . . 434
10.161.4.2 Begin() const . . . 434
10.161.4.3 Clear() . . . 435
10.161.4.4 End() const . . . 435
10.161.4.5 GetIOD(const char *name) const . . . 435
10.161.5 Friends And Related Function Documentation . . . 435
10.161.5.1 operator<< . . . 435
10.162 gdcm::IPPSorter Class Reference . . . 435
10.162.1 Detailed Description . . . 436
10.162.2 Constructor & Destructor Documentation . . . 437
10.162.2.1 IPPSorter() . . . 437
10.162.3 Member Function Documentation . . . 437
10.162.3.1 GetDirectionCosinesTolerance() const . . . 437
10.162.3.2 GetZSpacing() const . . . 437
10.162.3.3 GetZSpacingTolerance() const . . . 438
10.162.3.4 SetComputeZSpacing(bool b) . . . 438
10.162.3.5 SetDirectionCosinesTolerance(double tol) . . . 438
10.162.3.6 SetDropDuplicatePositions(bool b) . . . 438
10.162.3.7 SetZSpacingTolerance(double tol) . . . 438
10.162.3.8 Sort(std::vector< std::string > const &filenames) . . . 438

10.162.4 Member Data Documentation . . . 439
10.162.4.1 ComputeZSpacing . . . 439
10.162.4.2 DirCosTolerance . . . 439
10.162.4.3 DropDuplicatePositions . . . 439
10.162.4.4 ZSpacing . . . 439
10.162.4.5 ZTolerance . . . 439
10.163 gdcm::Item Class Reference . . . 439
10.163.1 Detailed Description . . . 441
10.163.2 Constructor & Destructor Documentation . . . 441
10.163.2.1 Item() . . . 441
10.163.2.2 Item(Item const &val) . . . 441
10.163.3 Member Function Documentation . . . 441
10.163.3.1 Clear() . . . 441
10.163.3.2 FindDataElement(const Tag &t) const . . . 441
10.163.3.3 GetDataElement(const Tag &t) const . . . 442
10.163.3.4 GetLength() const . . . 442
10.163.3.5 GetNestedDataSet() const . . . 442
10.163.3.6 GetNestedDataSet() . . . 442
10.163.3.7 InsertDataElement(const DataElement &de) . . . 442
10.163.3.8 Read(std::istream &is) . . . 442
10.163.3.9 SetNestedDataSet(const DataSet &nested) . . . 442
10.163.3.10 Write(std::ostream &os) const . . . 442
10.163.4 Friends And Related Function Documentation . . . 442
10.163.4.1 operator<< . . . 442
10.164 gdcm::IterationEvent Class Reference . . . 443
10.165 gdcm::JPEG12Codec Class Reference . . . 444
10.165.1 Detailed Description . . . 445
10.165.2 Constructor & Destructor Documentation . . . 445

10.165.2.1 JPEG12Codec() . . . 445
10.165.2.2 ~JPEG12Codec() . . . 445
10.165.3 Member Function Documentation . . . 445
10.165.3.1 DecodeByStreams(std::istream &is, std::ostream &os) . . . 445
10.165.3.2 EncodeBuffer(std::ostream &os, const char *data, size_t datalen) . . . 445
10.165.3.3 GetHeaderInfo(std::istream &is, TransferSyntax &ts) . . . 445
10.165.3.4 InternalCode(const char *input, unsigned long len, std::ostream &os) . . . 446
10.165.3.5 IsStateSuspension() const . . . 446
10.166 gdcm::JPEG16Codec Class Reference . . . 446
10.166.1 Detailed Description . . . 447
10.166.2 Constructor & Destructor Documentation . . . 447
10.166.2.1 JPEG16Codec() . . . 447
10.166.2.2 ~JPEG16Codec() . . . 447
10.166.3 Member Function Documentation . . . 447
10.166.3.1 DecodeByStreams(std::istream &is, std::ostream &os) . . . 447
10.166.3.2 EncodeBuffer(std::ostream &os, const char *data, size_t datalen) . . . 448
10.166.3.3 GetHeaderInfo(std::istream &is, TransferSyntax &ts) . . . 448
10.166.3.4 InternalCode(const char *input, unsigned long len, std::ostream &os) . . . 448
10.166.3.5 IsStateSuspension() const . . . 448
10.167 gdcm::JPEG2000Codec Class Reference . . . 448
10.167.1 Detailed Description . . . 450
10.167.2 Constructor & Destructor Documentation . . . 450
10.167.2.1 JPEG2000Codec() . . . 450
10.167.2.2 ~JPEG2000Codec() . . . 450
10.167.3 Member Function Documentation . . . 450
10.167.3.1 AppendFrameEncode(std::ostream &out, const char *data, size_t datalen) . . . 450
10.167.3.2 AppendRowEncode(std::ostream &out, const char *data, size_t datalen) . . . 450
10.167.3.3 CanCode(TransferSyntax const &ts) const . . . 450

10.167.3.4 CanDecode(TransferSyntax const &ts) const . . . 451
10.167.3.5 Clone() const . . . 451
10.167.3.6 Code(DataElement const &in, DataElement &out) . . . 451
10.167.3.7 Decode(DataElement const &is, DataElement &os) . . . 451
10.167.3.8 DecodeByStreams(std::istream &is, std::ostream &os) . . . 451
10.167.3.9 DecodeExtent(char *buffer, unsigned int xmin, unsigned int xmax, unsigned int ymin, unsigned int ymax, unsigned int zmin, unsigned int zmax, std::istream &is) . . . 451
10.167.3.10 GetHeaderInfo(std::istream &is, TransferSyntax &ts) . . . 451
10.167.3.11 GetQuality(unsigned int idx=0) const . . . 451
10.167.3.12 GetRate(unsigned int idx=0) const . . . 451
10.167.3.13 IsFrameEncoder() . . . 451
10.167.3.14 IsRowEncoder() . . . 452
10.167.3.15 SetNumberOfResolutions(unsigned int nres) . . . 452
10.167.3.16 SetQuality(unsigned int idx, double q) . . . 452
10.167.3.17 SetRate(unsigned int idx, double rate) . . . 452
10.167.3.18 SetReversible(bool res) . . . 452
10.167.3.19 SetTileSize(unsigned int tx, unsigned int ty) . . . 452
10.167.3.20 StartEncode(std::ostream &) . . . 452
10.167.3.21 StopEncode(std::ostream &) . . . 452
10.167.4 Friends And Related Function Documentation . . . 452
10.167.4.1 Bitmap . . . 452
10.167.4.2 ImageRegionReader . . . 452
10.168 gdcm::JPEG8Codec Class Reference . . . 453
10.168.1 Detailed Description . . . 454
10.168.2 Constructor & Destructor Documentation . . . 454
10.168.2.1 JPEG8Codec() . . . 454
10.168.2.2 ~JPEG8Codec() . . . 454
10.168.3 Member Function Documentation . . . 454

10.168.3.1 DecodeByStreams(std::istream &is, std::ostream &os) . . . 454
10.168.3.2 EncodeBuffer(std::ostream &os, const char *data, size_t datalen) . . . 454
10.168.3.3 GetHeaderInfo(std::istream &is, TransferSyntax &ts) . . . 454
10.168.3.4 InternalCode(const char *input, unsigned long len, std::ostream &os) . . . 455
10.168.3.5 IsStateSuspension() const . . . 455
10.169 gdcm::JPEGCodec Class Reference . . . 455
10.169.1 Detailed Description . . . 457
10.169.2 Constructor & Destructor Documentation . . . 457
10.169.2.1 JPEGCodec() . . . 457
10.169.2.2 ~JPEGCodec() . . . 457
10.169.3 Member Function Documentation . . . 457
10.169.3.1 AppendFrameEncode(std::ostream &out, const char *data, size_t datalen) . . . 457
10.169.3.2 AppendRowEncode(std::ostream &out, const char *data, size_t datalen) . . . 458
10.169.3.3 CanCode(TransferSyntax const &ts) const . . . 458
10.169.3.4 CanDecode(TransferSyntax const &ts) const . . . 458
10.169.3.5 Clone() const . . . 458
10.169.3.6 Code(DataElement const &in, DataElement &out) . . . 458
10.169.3.7 ComputeOffsetTable(bool b) . . . 458
10.169.3.8 Decode(DataElement const &is, DataElement &os) . . . 458
10.169.3.9 DecodeByStreams(std::istream &is, std::ostream &os) . . . 458
10.169.3.10 DecodeExtent(char *buffer, unsigned int xmin, unsigned int xmax, unsigned int ymin, unsigned int ymax, unsigned int zmin, unsigned int zmax, std::istream &is) . . . 459
10.169.3.11 EncodeBuffer(std::ostream &out, const char *inbuffer, size_t inlen) . . . 459
10.169.3.12 GetHeaderInfo(std::istream &is, TransferSyntax &ts) . . . 459
10.169.3.13 GetLossless() const . . . 459
10.169.3.14 GetQuality() const . . . 459
10.169.3.15 IsFrameEncoder() . . . 459
10.169.3.16 IsRowEncoder() . . . 459

10.169.3.17 IsStateSuspension() const . . . 459
10.169.3.18 IsValid(PhotometricInterpretation const &pi) . . . 459
10.169.3.19 SetBitSample(int bit) . . . 460
10.169.3.20 SetLossless(bool l) . . . 460
10.169.3.21 SetPixelFormat(PixelFormat const &pf) . . . 460
10.169.3.22 SetQuality(double q) . . . 460
10.169.3.23 StartEncode(std::ostream &) . . . 460
10.169.3.24 StopEncode(std::ostream &) . . . 460
10.169.4 Friends And Related Function Documentation . . . 460
10.169.4.1 ImageRegionReader . . . 460
10.169.5 Member Data Documentation . . . 460
10.169.5.1 BitSample . . . 460
10.169.5.2 Quality . . . 460
10.170 gdcm::JPEGLSCodec Class Reference . . . 461
10.170.1 Detailed Description . . . 462
10.170.2 Constructor & Destructor Documentation . . . 463
10.170.2.1 JPEGLSCodec() . . . 463
10.170.2.2 ~JPEGLSCodec() . . . 463
10.170.3 Member Function Documentation . . . 463
10.170.3.1 AppendFrameEncode(std::ostream &out, const char *data, size_t datalen) . . . 463
10.170.3.2 AppendRowEncode(std::ostream &out, const char *data, size_t datalen) . . . 463
10.170.3.3 CanCode(TransferSyntax const &ts) const . . . 463
10.170.3.4 CanDecode(TransferSyntax const &ts) const . . . 463
10.170.3.5 Clone() const . . . 463
10.170.3.6 Code(DataElement const &in, DataElement &out) . . . 463
10.170.3.7 Decode(DataElement const &is, DataElement &os) . . . 464
10.170.3.8 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) . . . 464

10.170.3.9 DecodeExtent(char *buffer, unsigned int xmin, unsigned int xmax, unsigned int ymin, unsigned int ymax, unsigned int zmin, unsigned int zmax, std::istream &is) 464
10.170.3.10 GetBufferLength() const 464
10.170.3.11 GetHeaderInfo(std::istream &is, TransferSyntax &ts) 464
10.170.3.12 GetLossless() const 464
10.170.3.13 JPEGFrameEncoder() 464
10.170.3.14 JPEGRowEncoder() 464
10.170.3.15 SetBufferLength(unsigned long l) 464
10.170.3.16 SetLossless(bool l) 464
10.170.3.17 SetLossyError(int error) 464
10.170.3.18 StartEncode(std::ostream &) 464
10.170.3.19 StopEncode(std::ostream &) 465
10.170.4 Friends And Related Function Documentation 465
10.170.4.1 ImageRegionReader 465
10.171 gdcm::JSON Class Reference 465
10.171.1 Detailed Description 465
10.171.2 Constructor & Destructor Documentation 465
10.171.2.1 JSON() 465
10.171.2.2 ~JSON() 465
10.171.3 Member Function Documentation 465
10.171.3.1 Code(DataSet const &in, std::ostream &os) 465
10.171.3.2 Decode(std::istream &is, DataSet &out) 466
10.171.3.3 GetPrettyPrint() const 466
10.171.3.4 PrettyPrintOff() 466
10.171.3.5 PrettyPrintOn() 466
10.171.3.6 SetPrettyPrint(bool onoff) 466
10.172 gdcm::KAKADUCodec Class Reference 466
10.172.1 Detailed Description 467

10.172.2 Constructor & Destructor Documentation . . . 467
10.172.2.1 KAKADUCodec() . . . 467
10.172.2.2 ~KAKADUCodec() . . . 467
10.172.3 Member Function Documentation . . . 467
10.172.3.1 CanCode(TransferSyntax const &ts) const . . . 467
10.172.3.2 CanDecode(TransferSyntax const &ts) const . . . 468
10.172.3.3 Clone() const . . . 468
10.172.3.4 Code(DataElement const &in, DataElement &out) . . . 468
10.172.3.5 Decode(DataElement const &is, DataElement &os) . . . 468
10.173 gdcm::LO Class Reference . . . 468
10.173.1 Detailed Description . . . 469
10.173.2 Member Typedef Documentation . . . 470
10.173.2.1 const_iterator . . . 470
10.173.2.2 const_reference . . . 470
10.173.2.3 const_reverse_iterator . . . 470
10.173.2.4 difference_type . . . 470
10.173.2.5 iterator . . . 470
10.173.2.6 pointer . . . 470
10.173.2.7 reference . . . 470
10.173.2.8 reverse_iterator . . . 470
10.173.2.9 size_type . . . 470
10.173.2.10 Superclass . . . 470
10.173.2.11 value_type . . . 470
10.173.3 Constructor & Destructor Documentation . . . 470
10.173.3.1 LO() . . . 470
10.173.3.2 LO(const value_type *s) . . . 470
10.173.3.3 LO(const value_type *s, size_type n) . . . 470
10.173.3.4 LO(const Superclass &s, size_type pos=0, size_type n=npos) . . . 470

10.173.4 Member Function Documentation . . . 470
10.173.4.1 IsValid() const . . . 470
10.174 gdcm::LookupTable Class Reference . . . 471
10.174.1 Detailed Description . . . 472
10.174.2 Member Enumeration Documentation . . . 473
10.174.2.1 LookupTableType . . . 473
10.174.3 Constructor & Destructor Documentation . . . 473
10.174.3.1 LookupTable() . . . 473
10.174.3.2 ~LookupTable() . . . 473
10.174.3.3 LookupTable(LookupTable const &lut) . . . 473
10.174.4 Member Function Documentation . . . 473
10.174.4.1 Allocate(unsigned short bitsample=8) . . . 473
10.174.4.2 Clear() . . . 473
10.174.4.3 Decode(std::istream &is, std::ostream &os) const . . . 473
10.174.4.4 Decode(char *outputbuffer, size_t outlen, const char *inputbuffer, size_t inlen) const . . . 473
10.174.4.5 GetBitSample() const . . . 473
10.174.4.6 GetBufferAsRGBA(unsigned char *rgba) const . . . 474
10.174.4.7 GetLUT(LookupTableType type, unsigned char *array, unsigned int &length) const . . . 474
10.174.4.8 GetLUTDescriptor(LookupTableType type, unsigned short &length, unsigned short &subscript, unsigned short &bitsize) const . . . 474
10.174.4.9 GetLUTLength(LookupTableType type) const . . . 474
10.174.4.10 GetPointer() const . . . 474
10.174.4.11 InitializeBlueLUT(unsigned short length, unsigned short subscript, unsigned short bitsize) . . . 474
10.174.4.12 Initialized() const . . . 474
10.174.4.13 InitializeGreenLUT(unsigned short length, unsigned short subscript, unsigned short bitsize) . . . 474
10.174.4.14 InitializeLUT(LookupTableType type, unsigned short length, unsigned short subscript, unsigned short bitsize) . . . 474

10.174.4.15 InitializeRedLUT(unsigned short length, unsigned short subscript, unsigned short bitsize) . . . 474
10.174.4.16 Print(std::ostream &) const . . . 475
10.174.4.17 SetBlueLUT(const unsigned char *blue, unsigned int length) . . . 475
10.174.4.18 SetGreenLUT(const unsigned char *green, unsigned int length) . . . 475
10.174.4.19 SetLUT(LookupTableType type, const unsigned char *array, unsigned int length) . . . 475
10.174.4.20 SetRedLUT(const unsigned char *red, unsigned int length) . . . 475
10.174.4.21 WriteBufferAsRGBA(const unsigned char *rgba) . . . 475
10.174.5 Member Data Documentation . . . 475
10.174.5.1 BitSample . . . 475
10.174.5.2 IncompleteLUT . . . 475
10.174.5.3 Internal . . . 475
10.175 gdcm::Scanner::ltstr Struct Reference . . . 475
10.175.1 Member Function Documentation . . . 476
10.175.1.1 operator()(const char *s1, const char *s2) const . . . 476
10.176 gdcm::StrictScanner::ltstr Struct Reference . . . 476
10.176.1 Member Function Documentation . . . 476
10.176.1.1 operator()(const char *s1, const char *s2) const . . . 476
10.177 gdcm::Macro Class Reference . . . 476
10.177.1 Detailed Description . . . 477
10.177.2 Member Typedef Documentation . . . 477
10.177.2.1 ArrayIncludeMacrosType . . . 477
10.177.2.2 MapModuleEntry . . . 477
10.177.3 Constructor & Destructor Documentation . . . 477
10.177.3.1 Macro() . . . 477
10.177.4 Member Function Documentation . . . 478
10.177.4.1 AddMacroEntry(const Tag &tag, const MacroEntry &module) . . . 478
10.177.4.2 Clear() . . . 478

10.177.4.3 FindMacroEntry(const Tag &tag) const . . . 478
10.177.4.4 GetMacroEntry(const Tag &tag) const . . . 478
10.177.4.5 GetName() const . . . 478
10.177.4.6 SetName(const char *name) . . . 478
10.177.4.7 Verify(const DataSet &ds, Usage const &usage) const . . . 478
10.177.5 Friends And Related Function Documentation . . . 478
10.177.5.1 operator<< . . . 478
10.178 gdcm::Macros Class Reference . . . 478
10.178.1 Detailed Description . . . 479
10.178.2 Member Typedef Documentation . . . 479
10.178.2.1 ModuleMapType . . . 479
10.178.3 Constructor & Destructor Documentation . . . 479
10.178.3.1 Macros() . . . 479
10.178.4 Member Function Documentation . . . 480
10.178.4.1 AddMacro(const char *ref, const Macro &module) . . . 480
10.178.4.2 Clear() . . . 480
10.178.4.3 GetMacro(const char *name) const . . . 480
10.178.4.4 IsEmpty() const . . . 480
10.178.5 Friends And Related Function Documentation . . . 480
10.178.5.1 operator<< . . . 480
10.179 gdcm::network::MaximumLengthSub Class Reference . . . 480
10.179.1 Detailed Description . . . 480
10.179.2 Constructor & Destructor Documentation . . . 481
10.179.2.1 MaximumLengthSub() . . . 481
10.179.3 Member Function Documentation . . . 481
10.179.3.1 GetMaximumLength() const . . . 481
10.179.3.2 Print(std::ostream &os) const . . . 481
10.179.3.3 Read(std::istream &is) . . . 481

10.179.3.4 SetMaximumLength(uint32_t maximumlength) . . . 481
10.179.3.5 Size() const . . . 481
10.179.3.6 Write(std::ostream &os) const . . . 481
10.180 gdcm::MD5 Class Reference . . . 481
10.180.1 Detailed Description . . . 482
10.180.2 Constructor & Destructor Documentation . . . 482
10.180.2.1 MD5() . . . 482
10.180.2.2 ~MD5() . . . 482
10.180.3 Member Function Documentation . . . 482
10.180.3.1 Compute(const char *buffer, unsigned long buf_len, char digest_str[33]) . . . 482
10.180.3.2 ComputeFile(const char *filename, char digest_str[33]) . . . 482
10.181 gdcm::MediaStorage Class Reference . . . 482
10.181.1 Detailed Description . . . 486
10.181.2 Member Enumeration Documentation . . . 486
10.181.2.1 MSType . . . 486
10.181.2.2 ObjectType . . . 488
10.181.3 Constructor & Destructor Documentation . . . 489
10.181.3.1 MediaStorage(MSType type=MS_END) . . . 489
10.181.4 Member Function Documentation . . . 489
10.181.4.1 GetModality() const . . . 489
10.181.4.2 GetModalityDimension() const . . . 489
10.181.4.3 GetMSString(MSType ts) . . . 489
10.181.4.4 GetMSType(const char *str) . . . 489
10.181.4.5 GetNumberOfModality() . . . 489
10.181.4.6 GetNumberOfMSString() . . . 489
10.181.4.7 GetNumberOfMSType() . . . 489
10.181.4.8 GetString() const . . . 489
10.181.4.9 GuessFromModality(const char *modality, unsigned int dimension=2) . . . 490

10.181.4.16 IsImage(MSType ts) . . . 490
10.181.4.17 IsUndefined() const . . . 490
10.181.4.18 operator MSType() const . . . 490
10.181.4.19 SetFromDataSet(DataSet const &ds) . . . 490
10.181.4.20 SetFromFile(File const &file) . . . 490
10.181.4.21 SetFromHeader(FileMetaInformation const &fmi) . . . 491
10.181.4.22 SetFromModality(DataSet const &ds) . . . 491
10.181.4.23 SetFromSourceImageSequence(DataSet const &ds) . . . 491
10.181.5 Friends And Related Function Documentation . . . 491
10.181.5.1 operator<< . . . 491
10.182 gdcm::MemberCommand< T > Class Template Reference . . . 491
10.182.1 Detailed Description . . . 493
10.182.2 Member Typedef Documentation . . . 493
10.182.2.1 Self . . . 493
10.182.2.2 TConstMemberFunctionPointer . . . 493
10.182.2.3 TMemberFunctionPointer . . . 493
10.182.3 Constructor & Destructor Documentation . . . 493
10.182.3.1 MemberCommand() . . . 493
10.182.3.2 ~MemberCommand() . . . 493
10.182.4 Member Function Documentation . . . 493
10.182.4.1 Execute(Subject *caller, const Event &event) . . . 493
10.182.4.2 Execute(const Subject *caller, const Event &event) . . . 494
10.182.4.3 New() . . . 494
10.182.4.4 SetCallbackFunction(T *object, TMemberFunctionPointer memberFunction) . . . 494
10.182.4.5 SetCallbackFunction(T *object, TConstMemberFunctionPointer memberFunction) . . . 494
10.182.5 Member Data Documentation . . . 494
10.182.5.1 m_ConstMemberFunction . . . 494
10.182.5.2 m_MemberFunction . . . 494

10.182.5.3 m_This . . . 494
10.183 gdcm::MeshPrimitive Class Reference . . . 495
10.183.1 Detailed Description . . . 496
10.183.2 Member Typedef Documentation . . . 497
10.183.2.1 PrimitivesData . . . 497
10.183.3 Member Enumeration Documentation . . . 497
10.183.3.1 MPType . . . 497
10.183.4 Constructor & Destructor Documentation . . . 497
10.183.4.1 MeshPrimitive() . . . 497
10.183.4.2 ~MeshPrimitive() . . . 497
10.183.5 Member Function Documentation . . . 497
10.183.5.1 AddPrimitiveData(DataElement const &de) . . . 497
10.183.5.2 GetMPType(const char *type) . . . 497
10.183.5.3 GetMPTypeString(const MPType type) . . . 497
10.183.5.4 GetNumberOfPrimitivesData() const . . . 497
10.183.5.5 GetPrimitiveData() const . . . 497
10.183.5.6 GetPrimitiveData() . . . 498
10.183.5.7 GetPrimitiveData(const unsigned int idx) const . . . 498
10.183.5.8 GetPrimitiveData(const unsigned int idx) . . . 498
10.183.5.9 GetPrimitivesData() const . . . 498
10.183.5.10 GetPrimitivesData() . . . 498
10.183.5.11 GetPrimitiveType() const . . . 498
10.183.5.12 SetPrimitiveData(DataElement const &de) . . . 498
10.183.5.13 SetPrimitiveData(const unsigned int idx, DataElement const &de) . . . 498
10.183.5.14 SetPrimitivesData(PrimitivesData const &DEs) . . . 498
10.183.5.15 SetPrimitiveType(const MPType type) . . . 498
10.183.6 Member Data Documentation . . . 498
10.183.6.1 PrimitiveData . . . 498

10.183.6.2 PrimitiveType . . . 498
10.184 gdcm::ModalityPerformedProcedureStepCreateQuery Class Reference . . . 499
10.184.1 Detailed Description . . . 500
10.184.2 Constructor & Destructor Documentation . . . 501
10.184.2.1 ModalityPerformedProcedureStepCreateQuery(const std::string &iSopInstanceUID) . . . 501
10.184.3 Member Function Documentation . . . 501
10.184.3.1 GetAbstractSyntaxUID() const . . . 501
10.184.3.2 GetRequiredDataSet() const . . . 501
10.184.3.3 ValidateQuery(bool inStrict=true) const . . . 501
10.184.4 Friends And Related Function Documentation . . . 501
10.184.4.1 QueryFactory . . . 501
10.185 gdcm::ModalityPerformedProcedureStepSetQuery Class Reference . . . 501
10.185.1 Detailed Description . . . 503
10.185.2 Constructor & Destructor Documentation . . . 503
10.185.2.1 ModalityPerformedProcedureStepSetQuery(const std::string &iSopInstanceUID) . . . 503
10.185.3 Member Function Documentation . . . 503
10.185.3.1 GetAbstractSyntaxUID() const . . . 503
10.185.3.2 GetRequiredDataSet() const . . . 503
10.185.3.3 ValidateQuery(bool inStrict=true) const . . . 503
10.185.4 Friends And Related Function Documentation . . . 503
10.185.4.1 QueryFactory . . . 503
10.186 gdcm::ModifiedEvent Class Reference . . . 504
10.187 gdcm::Module Class Reference . . . 505
10.187.1 Detailed Description . . . 505
10.187.2 Member Typedef Documentation . . . 506
10.187.2.1 ArrayIncludeMacrosType . . . 506
10.187.2.2 MapModuleEntry . . . 506
10.187.3 Constructor & Destructor Documentation . . . 506

10.187.3.1 Module() . . . 506
10.187.4 Member Function Documentation . . . 506
10.187.4.1 AddMacro(const char *include) . . . 506
10.187.4.2 AddModuleEntry(const Tag &tag, const ModuleEntry &module) . . . 506
10.187.4.3 Clear() . . . 506
10.187.4.4 FindModuleEntryInMacros(Macros const &macros, const Tag &tag) const . . . 506
10.187.4.5 GetModuleEntryInMacros(Macros const &macros, const Tag &tag) const . . . 506
10.187.4.6 GetName() const . . . 507
10.187.4.7 SetName(const char *name) . . . 507
10.187.4.8 Verify(const DataSet &ds, Usage const &usage) const . . . 507
10.187.5 Friends And Related Function Documentation . . . 507
10.187.5.1 operator<< . . . 507
10.188 gdcm::ModuleEntry Class Reference . . . 507
10.188.1 Detailed Description . . . 509
10.188.2 Member Typedef Documentation . . . 509
10.188.2.1 Description . . . 509
10.188.3 Constructor & Destructor Documentation . . . 509
10.188.3.1 ModuleEntry(const char *name="""", const char *type=""3"", const char *description="""") 509
10.188.3.2 ~ModuleEntry() . . . 509
10.188.4 Member Function Documentation . . . 509
10.188.4.1 GetDescription() const . . . 509
10.188.4.2 GetName() const . . . 509
10.188.4.3 GetType() const . . . 509
10.188.4.4 SetDescription(const char *d) . . . 510
10.188.4.5 SetName(const char *name) . . . 510
10.188.4.6 SetType(const Type &type) . . . 510
10.188.5 Friends And Related Function Documentation . . . 510
10.188.5.1 operator<< . . . 510

10.188.6 Member Data Documentation . . . 510
10.188.6.1 DataElementType . . . 510
10.188.6.2 DescriptionField . . . 510
10.188.6.3 Name . . . 510
10.189 gdcm::Modules Class Reference . . . 510
10.189.1 Detailed Description . . . 511
10.189.2 Member Typedef Documentation . . . 511
10.189.2.1 ModuleMapType . . . 511
10.189.3 Constructor & Destructor Documentation . . . 511
10.189.3.1 Modules() . . . 511
10.189.4 Member Function Documentation . . . 511
10.189.4.1 AddModule(const char *ref, const Module &module) . . . 511
10.189.4.2 Clear() . . . 511
10.189.4.3 GetModule(const char *name) const . . . 511
10.189.4.4 IsEmpty() const . . . 512
10.189.5 Friends And Related Function Documentation . . . 512
10.189.5.1 operator<< . . . 512
10.190 gdcm::MovePatientRootQuery Class Reference . . . 512
10.190.1 Detailed Description . . . 513
10.190.2 Constructor & Destructor Documentation . . . 513
10.190.2.1 MovePatientRootQuery() . . . 513
10.190.3 Member Function Documentation . . . 513
10.190.3.1 GetAbstractSyntaxUID() const . . . 513
10.190.3.2 GetTagListByLevel(const EQueryLevel &inQueryLevel) . . . 514
10.190.3.3 InitializeDataSet(const EQueryLevel &inQueryLevel) . . . 514
10.190.3.4 ValidateQuery(bool inStrict=true) const . . . 514
10.190.4 Friends And Related Function Documentation . . . 514
10.190.4.1 QueryFactory . . . 514

10.191 gdcm::MoveStudyRootQuery Class Reference . . . 515
10.191.1 Detailed Description . . . 516
10.191.2 Constructor & Destructor Documentation . . . 516
10.191.2.1 MoveStudyRootQuery() . . . 516
10.191.3 Member Function Documentation . . . 516
10.191.3.1 GetAbstractSyntaxUID() const . . . 516
10.191.3.2 GetTagListByLevel(const EQueryLevel &inQueryLevel) . . . 516
10.191.3.3 InitializeDataSet(const EQueryLevel &inQueryLevel) . . . 516
10.191.3.4 ValidateQuery(bool inStrict=true) const . . . 516
10.191.4 Friends And Related Function Documentation . . . 517
10.191.4.1 QueryFactory . . . 517
10.192 gdcm::network::NActionRQ Class Reference . . . 517
10.192.1 Detailed Description . . . 518
10.192.2 Member Function Documentation . . . 518
10.192.2.1 ConstructPDV(const ULConnection &inConnection, const BaseQuery *inQuery) . . . 518
10.193 gdcm::network::NActionRSP Class Reference . . . 518
10.193.1 Detailed Description . . . 519
10.193.2 Member Function Documentation . . . 519
10.193.2.1 ConstructPDVByDataSet(const DataSet *inDataSet) . . . 519
10.194 gdcm::network::NCreateRQ Class Reference . . . 519
10.194.1 Detailed Description . . . 520
10.194.2 Member Function Documentation . . . 521
10.194.2.1 ConstructPDV(const ULConnection &inConnection, const BaseQuery *inQuery) . . . 521
10.195 gdcm::network::NCreateRSP Class Reference . . . 521
10.195.1 Detailed Description . . . 522
10.195.2 Member Function Documentation . . . 522
10.195.2.1 ConstructPDVByDataSet(const DataSet *inDataSet) . . . 522
10.196 gdcm::network::NDeleteRQ Class Reference . . . 522

10.196.1 Detailed Description . . . 523
10.196.2 Member Function Documentation . . . 523
10.196.2.1 ConstructPDV(const ULConnection &inConnection, const BaseQuery *inQuery) . . . 523
10.197 gdcm::network::NDeleteRSP Class Reference . . . 524
10.197.1 Detailed Description . . . 524
10.197.2 Member Function Documentation . . . 525
10.197.2.1 ConstructPDVByDataSet(const DataSet *inDataSet) . . . 525
10.198 gdcm::NestedModuleEntries Class Reference . . . 525
10.198.1 Detailed Description . . . 527
10.198.2 Member Typedef Documentation . . . 527
10.198.2.1 SizeType . . . 527
10.198.3 Constructor & Destructor Documentation . . . 527
10.198.3.1 NestedModuleEntries(const char *name="""", const char *type=""3"", const char *description="""") . . . 527
10.198.4 Member Function Documentation . . . 527
10.198.4.1 AddModuleEntry(const ModuleEntry &me) . . . 527
10.198.4.2 GetModuleEntry(SizeType idx) const . . . 527
10.198.4.3 GetModuleEntry(SizeType idx) . . . 527
10.198.4.4 GetNumberOfModuleEntries() . . . 527
10.198.5 Friends And Related Function Documentation . . . 527
10.198.5.1 operator<< . . . 527
10.199 gdcm::network::NEventReportRQ Class Reference . . . 528
10.199.1 Detailed Description . . . 528
10.199.2 Member Function Documentation . . . 529
10.199.2.1 ConstructPDV(const ULConnection &inConnection, const BaseQuery *inQuery) . . . 529
10.200 gdcm::network::NEventReportRSP Class Reference . . . 529
10.200.1 Detailed Description . . . 530
10.200.2 Member Function Documentation . . . 530

10.200.2.1 ConstructPDVByDataSet(const DataSet *inDataSet) . . . 530
10.201 gdcm::network::NGetRQ Class Reference . . . 530
10.201.1 Detailed Description . . . 531
10.201.2 Member Function Documentation . . . 531
10.201.2.1 ConstructPDV(const ULConnection &inConnection, const BaseQuery *inQuery) . . . 531
10.202 gdcm::network::NGetRSP Class Reference . . . 532
10.202.1 Detailed Description . . . 532
10.202.2 Member Function Documentation . . . 533
10.202.2.1 ConstructPDVByDataSet(const DataSet *inDataSet) . . . 533
10.203 gdcm::NoEvent Class Reference . . . 533
10.203.1 Detailed Description . . . 534
10.204 gdcm::network::NormalizedMessageFactory Class Reference . . . 534
10.204.1 Member Function Documentation . . . 534
10.204.1.1 ConstructNAction(const ULConnection &inConnection, const BaseQuery *inQuery) . 534
10.204.1.2 ConstructNCreate(const ULConnection &inConnection, const BaseQuery *inQuery) . 534
10.204.1.3 ConstructNDelete(const ULConnection &inConnection, const BaseQuery *inQuery) . 534
10.204.1.4 ConstructNEventReport(const ULConnection &inConnection, const BaseQuery *in↩ Query) . . . 534
10.204.1.5 ConstructNGet(const ULConnection &inConnection, const BaseQuery *inQuery) . . . 534
10.204.1.6 ConstructNSet(const ULConnection &inConnection, const BaseQuery *inQuery) . . . 534
10.205 gdcm::NormalizedNetworkFunctions Class Reference . . . 535
10.205.1 Detailed Description . . . 535
10.205.2 Member Function Documentation . . . 536
10.205.2.1 ConstructQuery(const std::string &sopInstanceUID, const DataSet &queryds, EN↩ QueryType queryType=eCreateMMPS) . . . 536
10.205.2.2 NAction(const char *remote, uint16_t portno, const BaseQuery *query, std::vector< DataSet > &retDataSets, const char *aetitle, const char *call) . . . 536
10.205.2.3 NCreate(const char *remote, uint16_t portno, BaseQuery *query, std::vector< DataSet > &retDataSets, const char *aetitle, const char *call) . . . 536

10.205.2.4 NDelete(const char *remote, uint16_t portno, const BaseQuery *query, std::vector< DataSet > &retDataSets, const char *aetitle, const char *call) . . . 536
10.205.2.5 NEventReport(const char *remote, uint16_t portno, const BaseQuery *query, std::vector< DataSet > &retDataSets, const char *aetitle, const char *call) . . . 536
10.205.2.6 NGet(const char *remote, uint16_t portno, const BaseQuery *query, std::vector< DataSet > &retDataSets, const char *aetitle, const char *call) . . . 536
10.205.2.7 NSet(const char *remote, uint16_t portno, const BaseQuery *query, std::vector< DataSet > &retDataSets, const char *aetitle, const char *call) . . . 536
10.206 gdcm::network::NSetRQ Class Reference . . . 536
10.206.1 Detailed Description . . . 537
10.206.2 Member Function Documentation . . . 537
10.206.2.1 ConstructPDV(const ULConnection &inConnection, const BaseQuery *inQuery) . . . 537
10.207 gdcm::network::NSetRSP Class Reference . . . 538
10.207.1 Detailed Description . . . 538
10.207.2 Member Function Documentation . . . 539
10.207.2.1 ConstructPDVByDataSet(const DataSet *inDataSet) . . . 539
10.208 gdcm::Object Class Reference . . . 539
10.208.1 Detailed Description . . . 540
10.208.2 Constructor & Destructor Documentation . . . 540
10.208.2.1 Object() . . . 540
10.208.2.2 ~Object() . . . 540
10.208.2.3 Object(const Object &) . . . 540
10.208.3 Member Function Documentation . . . 541
10.208.3.1 operator=(const Object &) . . . 541
10.208.3.2 Print(std::ostream &) const . . . 541
10.208.3.3 Register() . . . 541
10.208.3.4 UnRegister() . . . 541
10.208.4 Friends And Related Function Documentation . . . 541
10.208.4.1 operator<< . . . 541
10.208.4.2 SmartPointer . . . 541

10.209gdcm::OpenSSLCryptoFactory Class Reference . . . 541
10.209.1Constructor & Destructor Documentation . . . 542
10.209.1.1OpenSSLCryptoFactory(CryptoLib id) . . . 542
10.209.2Member Function Documentation . . . 542
10.209.2.1CreateCMSProvider() . . . 542
10.209.2.2InitOpenSSL() . . . 542
10.210gdcm::OpenSSLCryptographicMessageSyntax Class Reference . . . 543
10.210.1Constructor & Destructor Documentation . . . 544
10.210.1.1OpenSSLCryptographicMessageSyntax() . . . 544
10.210.1.2~OpenSSLCryptographicMessageSyntax() . . . 544
10.210.2Member Function Documentation . . . 544
10.210.2.1Decrypt(char *output, size_t &outlen, const char *array, size_t len) const . . . 544
10.210.2.2Encrypt(char *output, size_t &outlen, const char *array, size_t len) const . . . 544
10.210.2.3GetCipherType() const . . . 544
10.210.2.4ParseCertificateFile(const char *filename) . . . 544
10.210.2.5ParseKeyFile(const char *filename) . . . 544
10.210.2.6SetCipherType(CipherTypes type) . . . 545
10.210.2.7SetPassword(const char *pass, size_t passLen) . . . 545
10.211gdcm::OpenSSLP7CryptoFactory Class Reference . . . 545
10.211.1Constructor & Destructor Documentation . . . 546
10.211.1.1OpenSSLP7CryptoFactory(CryptoLib id) . . . 546
10.211.2Member Function Documentation . . . 546
10.211.2.1CreateCMSProvider() . . . 546
10.212gdcm::OpenSSLP7CryptographicMessageSyntax Class Reference . . . 546
10.212.1Detailed Description . . . 547
10.212.2Constructor & Destructor Documentation . . . 548
10.212.2.1OpenSSLP7CryptographicMessageSyntax() . . . 548
10.212.2.2~OpenSSLP7CryptographicMessageSyntax() . . . 548

10.212.3 Member Function Documentation . . . 548
10.212.3.1 Decrypt(char *output, size_t &outlen, const char *array, size_t len) const . . . 548
10.212.3.2 Encrypt(char *output, size_t &outlen, const char *array, size_t len) const . . . 548
10.212.3.3 GetCipherType() const . . . 548
10.212.3.4 ParseCertificateFile(const char *filename) . . . 548
10.212.3.5 ParseKeyFile(const char *filename) . . . 548
10.212.3.6 SetCipherType(CipherTypes type) . . . 548
10.212.3.7 SetPassword(const char *, size_t) . . . 549
10.213 gdcm::Orientation Class Reference . . . 549
10.213.1 Detailed Description . . . 550
10.213.2 Member Enumeration Documentation . . . 550
10.213.2.1 OrientationType . . . 550
10.213.3 Constructor & Destructor Documentation . . . 550
10.213.3.1 Orientation() . . . 550
10.213.3.2 ~Orientation() . . . 550
10.213.4 Member Function Documentation . . . 550
10.213.4.1 GetLabel(OrientationType type) . . . 550
10.213.4.2 GetMajorAxisFromPatientRelativeDirectionCosine(double x, double y, double z) . . . 550
10.213.4.3 GetObliquityThresholdCosineValue() . . . 550
10.213.4.4 GetType(const double dircos[6]) . . . 550
10.213.4.5 Print(std::ostream &) const . . . 551
10.213.4.6 SetObliquityThresholdCosineValue(double val) . . . 551
10.213.5 Friends And Related Function Documentation . . . 551
10.213.5.1 operator<< . . . 551
10.214 gdcm::Overlay Class Reference . . . 551
10.214.1 Detailed Description . . . 554
10.214.2 Member Enumeration Documentation . . . 554
10.214.2.1 OverlayType . . . 554

10.214.3 Constructor & Destructor Documentation . . . 554
10.214.3.1 Overlay() . . . 554
10.214.3.2 ~Overlay() . . . 554
10.214.3.3 Overlay(Overlay const &ov) . . . 554
10.214.4 Member Function Documentation . . . 554
10.214.4.1 Decompress(std::ostream &os) const . . . 554
10.214.4.2 GetBitPosition() const . . . 554
10.214.4.3 GetBitsAllocated() const . . . 555
10.214.4.4 GetColumns() const . . . 555
10.214.4.5 GetDescription() const . . . 555
10.214.4.6 GetGroup() const . . . 555
10.214.4.7 GetOrigin() const . . . 555
10.214.4.8 GetOverlayData() const . . . 555
10.214.4.9 GetOverlayTypeAsString(OverlayType ot) . . . 555
10.214.4.10 GetOverlayTypeFromString(const char *) . . . 555
10.214.4.11 GetRows() const . . . 555
10.214.4.12 GetType() const . . . 555
10.214.4.13 GetTypeAsEnum() const . . . 556
10.214.4.14 GetUnpackBuffer(char *buffer, size_t len) const . . . 556
10.214.4.15 GetUnpackBufferLength() const . . . 556
10.214.4.16 GrabOverlayFromPixelData(DataSet const &ds) . . . 556
10.214.4.17 IsEmpty() const . . . 556
10.214.4.18 IsInPixelData() const . . . 556
10.214.4.19 IsInPixelData(bool b) . . . 556
10.214.4.20 IsZero() const . . . 556
10.214.4.21 operator=(Overlay const &ov) . . . 556
10.214.4.22 Print(std::ostream &) const . . . 556
10.214.4.23 SetBitPosition(unsigned short bitposition) . . . 556

10.214.4.24 SetBitsAllocated(unsigned short bitsallocated) . . . 557
10.214.4.25 SetColumns(unsigned short columns) . . . 557
10.214.4.26 SetDescription(const char *description) . . . 557
10.214.4.27 SetFrameOrigin(unsigned short frameorigin) . . . 557
10.214.4.28 SetGroup(unsigned short group) . . . 557
10.214.4.29 SetNumberOfFrames(unsigned int numberofframes) . . . 557
10.214.4.30 SetOrigin(const signed short origin[2]) . . . 557
10.214.4.31 SetOverlay(const char *array, size_t length) . . . 557
10.214.4.32 SetRows(unsigned short rows) . . . 557
10.214.4.33 SetType(const char *type) . . . 557
10.214.4.34 Update(const DataElement &de) . . . 558
10.215 gdcm::ParseException Class Reference . . . 558
10.215.1 Detailed Description . . . 559
10.215.2 Constructor & Destructor Documentation . . . 559
10.215.2.1 ParseException() . . . 559
10.215.2.2 ~ParseException() . . . 559
10.215.3 Member Function Documentation . . . 559
10.215.3.1 GetLastElement() const . . . 559
10.215.3.2 operator=(const ParseException &orig) . . . 559
10.215.3.3 SetLastElement(DataElement &de) . . . 560
10.216 gdcm::Parser Class Reference . . . 560
10.216.1 Detailed Description . . . 561
10.216.2 Member Typedef Documentation . . . 561
10.216.2.1 EndElementHandler . . . 561
10.216.2.2 StartElementHandler . . . 561
10.216.3 Member Enumeration Documentation . . . 561
10.216.3.1 ErrorType . . . 561
10.216.4 Constructor & Destructor Documentation . . . 562

10.216.4.1 Parser() . . . 562
10.216.4.2 ~Parser() . . . 562
10.216.5 Member Function Documentation . . . 562
10.216.5.1 GetBuffer(int len) . . . 562
10.216.5.2 GetCurrentByteIndex() const . . . 562
10.216.5.3 GetErrorCode() const . . . 562
10.216.5.4 GetErrorString(ErrorType const &err) . . . 562
10.216.5.5 GetUserData() const . . . 562
10.216.5.6 Parse(const char *s, int len, bool isFinal) . . . 562
10.216.5.7 ParseBuffer(int len, bool isFinal) . . . 562
10.216.5.8 Process() . . . 562
10.216.5.9 SetElementHandler(StartElementHandler start, EndElementHandler end) . . . 562
10.216.5.10 SetUserData(void *userData) . . . 562
10.217 gdcm::Patient Class Reference . . . 562
10.217.1 Detailed Description . . . 563
10.217.2 Constructor & Destructor Documentation . . . 563
10.217.2.1 Patient() . . . 563
10.218 gdcm::network::PDataTFPDU Class Reference . . . 563
10.218.1 Detailed Description . . . 564
10.218.2 Member Typedef Documentation . . . 564
10.218.2.1 SizeType . . . 564
10.218.3 Constructor & Destructor Documentation . . . 564
10.218.3.1 PDataTFPDU() . . . 564
10.218.4 Member Function Documentation . . . 564
10.218.4.1 AddPresentationDataValue(PresentationDataValue const &pdv) . . . 564
10.218.4.2 GetNumberOfPresentationDataValues() const . . . 564
10.218.4.3 GetPresentationDataValue(SizeType i) const . . . 564
10.218.4.4 IsLastFragment() const . . . 564

10.218.4.5 Print(std::ostream &os) const . . . 565
10.218.4.6 Read(std::istream &is) . . . 565
10.218.4.7 ReadInto(std::istream &is, std::ostream &os) . . . 565
10.218.4.8 Size() const . . . 565
10.218.4.9 Write(std::ostream &os) const . . . 565
10.219 gdcm::PDBElement Class Reference . . . 565
10.219.1 Detailed Description . . . 566
10.219.2 Constructor & Destructor Documentation . . . 566
10.219.2.1 PDBElement() . . . 566
10.219.3 Member Function Documentation . . . 566
10.219.3.1 GetName() const . . . 566
10.219.3.2 GetValue() const . . . 566
10.219.3.3 operator==(const PDBElement &de) const . . . 567
10.219.3.4 SetName(const char *name) . . . 567
10.219.3.5 SetValue(const char *value) . . . 567
10.219.4 Friends And Related Function Documentation . . . 567
10.219.4.1 operator<< . . . 567
10.219.5 Member Data Documentation . . . 567
10.219.5.1 NameField . . . 567
10.219.5.2 ValueField . . . 567
10.220 gdcm::PDBHeader Class Reference . . . 567
10.220.1 Detailed Description . . . 568
10.220.2 Constructor & Destructor Documentation . . . 568
10.220.2.1 PDBHeader() . . . 568
10.220.2.2 ~PDBHeader() . . . 568
10.220.3 Member Function Documentation . . . 568
10.220.3.1 FindPDBElementByName(const char *name) . . . 568
10.220.3.2 GetPDBEEnd() const . . . 569

10.220.3.3 GetPDBElementByName(const char *name) . . . 569
10.220.3.4 GetPDBInfoTag() . . . 569
10.220.3.5 LoadFromDataElement(DataElement const &de) . . . 569
10.220.3.6 Print(std::ostream &os) const . . . 569
10.220.4 Friends And Related Function Documentation . . . 569
10.220.4.1 operator<< . . . 569
10.221 gdcm::PDFCodec Class Reference . . . 570
10.221.1 Detailed Description . . . 571
10.221.2 Constructor & Destructor Documentation . . . 571
10.221.2.1 PDFCodec() . . . 571
10.221.2.2 ~PDFCodec() . . . 571
10.221.3 Member Function Documentation . . . 571
10.221.3.1 CanCode(TransferSyntax const &) const . . . 571
10.221.3.2 CanDecode(TransferSyntax const &) const . . . 571
10.221.3.3 Decode(DataElement const &is, DataElement &os) . . . 571
10.222 gdcm::network::PDUFactory Class Reference . . . 572
10.222.1 Detailed Description . . . 572
10.222.2 Member Function Documentation . . . 573
10.222.2.1 ConstructAbortPDU() . . . 573
10.222.2.2 ConstructPDU(uint8_t itemtype) . . . 573
10.222.2.3 ConstructReleasePDU() . . . 573
10.222.2.4 CreateCEchoPDU(const ULConnection &inConnection) . . . 573
10.222.2.5 CreateCFindPDU(const ULConnection &inConnection, const BaseRootQuery *in↩ RootQuery) . . . 573
10.222.2.6 CreateCMovePDU(const ULConnection &inConnection, const BaseRootQuery *in↩ RootQuery) . . . 573
10.222.2.7 CreateCStoreRQPDU(const ULConnection &inConnection, const File &file, bool writeDataSet=true) . . . 573
10.222.2.8 CreateCStoreRSPPDU(const DataSet *inDataSet, const BasePDU *inPC) . . . 573

10.222.2.9 CreateNActionPDU(const ULConnection &inConnection, const BaseQuery *inQuery) 573
10.222.2.10 CreateNCreatePDU(const ULConnection &inConnection, const BaseQuery *inQuery) 573
10.222.2.11 CreateNDeletePDU(const ULConnection &inConnection, const BaseQuery *inQuery) 573
10.222.2.12 CreateNEventReportPDU(const ULConnection &inConnection, const BaseQuery *inQuery) 573
10.222.2.13 CreateNGetPDU(const ULConnection &inConnection, const BaseQuery *inQuery) 573
10.222.2.14 CreateNSetPDU(const ULConnection &inConnection, const BaseQuery *inQuery) 573
10.222.2.15 DetermineEventByPDU(const BasePDU *inPDU) 573
10.222.2.16 GetPDVs(const std::vector< BasePDU * > &inDataPDUs) 573
10.223 gdcm::PersonName Class Reference 574
10.223.1 Detailed Description 574
10.223.2 Member Function Documentation 574
10.223.2.1 GetMaxLength() const 574
10.223.2.2 GetNumberOfComponents() const 574
10.223.2.3 Print(std::ostream &os) const 574
10.223.2.4 SetBlob(const std::vector< char > &v) 574
10.223.2.5 SetComponents(const char *comp1="", const char *comp2="", const char *comp3="", const char *comp4="", const char *comp5="") 575
10.223.2.6 SetComponents(const char *components[]) 575
10.223.3 Member Data Documentation 575
10.223.3.1 Component 575
10.223.3.2 MaxLength 575
10.223.3.3 MaxNumberOfComponents 575
10.223.3.4 Padding 575
10.223.3.5 Separator 575
10.224 gdcm::PGXCodec Class Reference 575
10.224.1 Detailed Description 576
10.224.2 Constructor & Destructor Documentation 576
10.224.2.1 PGXCodec() 576

10.224.2.2 ~PGXCodec() . . . 576
10.224.3 Member Function Documentation . . . 576
10.224.3.1 CanCode(TransferSyntax const &ts) const . . . 576
10.224.3.2 CanDecode(TransferSyntax const &ts) const . . . 577
10.224.3.3 Clone() const . . . 577
10.224.3.4 GetHeaderInfo(std::istream &is, TransferSyntax &ts) . . . 577
10.224.3.5 Read(const char *filename, DataElement &out) const . . . 577
10.224.3.6 Write(const char *filename, const DataElement &out) const . . . 577
10.225 gdcm::PhotometricInterpretation Class Reference . . . 577
10.225.1 Detailed Description . . . 578
10.225.2 Member Enumeration Documentation . . . 578
10.225.2.1 PIType . . . 578
10.225.3 Constructor & Destructor Documentation . . . 579
10.225.3.1 PhotometricInterpretation(PIType pi=UNKNOWN) . . . 579
10.225.4 Member Function Documentation . . . 579
10.225.4.1 GetPIString(PIType pi) . . . 579
10.225.4.2 GetPIType(const char *pi) . . . 579
10.225.4.3 GetSamplesPerPixel() const . . . 579
10.225.4.4 GetString() const . . . 579
10.225.4.5 GetType() const . . . 579
10.225.4.6 IsLossless() const . . . 579
10.225.4.7 IsLossy() const . . . 579
10.225.4.8 IsRetired(PIType pi) . . . 579
10.225.4.9 IsSameColorSpace(PhotometricInterpretation const &pi) const . . . 579
10.225.4.10 operator PIType() const . . . 579
10.225.5 Friends And Related Function Documentation . . . 579
10.225.5.1 operator<< . . . 579
10.226 gdcm::PixelFormat Class Reference . . . 580

10.226.1 Detailed Description . . . 581
10.226.2 Member Enumeration Documentation . . . 582
10.226.2.1 ScalarType . . . 582
10.226.3 Constructor & Destructor Documentation . . . 582
10.226.3.1 PixelFormat(unsigned short samplesperpixel=1, unsigned short bitsallocated=8, unsigned short bitsstored=8, unsigned short highbit=7, unsigned short pixelrepresentation=0) . . . 582
10.226.3.2 PixelFormat(ScalarType st) . . . 582
10.226.4 Member Function Documentation . . . 582
10.226.4.1 GetBitsAllocated() const . . . 582
10.226.4.2 GetBitsStored() const . . . 583
10.226.4.3 GetHighBit() const . . . 583
10.226.4.4 GetMax() const . . . 583
10.226.4.5 GetMin() const . . . 583
10.226.4.6 GetPixelRepresentation() const . . . 583
10.226.4.7 GetPixelSize() const . . . 583
10.226.4.8 GetSamplesPerPixel() const . . . 584
10.226.4.9 GetScalarType() const . . . 584
10.226.4.10 GetScalarTypeAsString() const . . . 584
10.226.4.11 IsCompatible(const TransferSyntax &ts) const . . . 584
10.226.4.12 IsValid() const . . . 584
10.226.4.13 operator ScalarType() const . . . 584
10.226.4.14 operator"!=(ScalarType st) const . . . 584
10.226.4.15 operator"!=(const PixelFormat &pf) const . . . 584
10.226.4.16 operator==(ScalarType st) const . . . 584
10.226.4.17 operator==(const PixelFormat &pf) const . . . 584
10.226.4.18 Print(std::ostream &os) const . . . 584
10.226.4.19 SetBitsAllocated(unsigned short ba) . . . 585
10.226.4.20 SetBitsStored(unsigned short bs) . . . 585

10.226.4.23 SetHighBit(unsigned short hb) . . . 585
10.226.4.24 SetPixelRepresentation(unsigned short pr) . . . 585
10.226.4.25 SetSamplesPerPixel(unsigned short spp) . . . 585
10.226.4.26 SetScalarType(ScalarType st) . . . 585
10.226.4.27 Validate() . . . 585
10.226.5 Friends And Related Function Documentation . . . 585
10.226.5.1 Bitmap . . . 585
10.226.5.2 operator<< . . . 585
10.227 gdcm::Pixmap Class Reference . . . 586
10.227.1 Detailed Description . . . 587
10.227.2 Constructor & Destructor Documentation . . . 588
10.227.2.1 Pixmap() . . . 588
10.227.2.2 ~Pixmap() . . . 588
10.227.3 Member Function Documentation . . . 588
10.227.3.1 AreOverlaysInPixelData() const . . . 588
10.227.3.2 GetCurve(size_t i=0) . . . 588
10.227.3.3 GetCurve(size_t i=0) const . . . 588
10.227.3.4 GetIconImage() const . . . 588
10.227.3.5 GetIconImage() . . . 588
10.227.3.6 GetNumberOfCurves() const . . . 588
10.227.3.7 GetNumberOfOverlays() const . . . 588
10.227.3.8 GetOverlay(size_t i=0) . . . 588
10.227.3.9 GetOverlay(size_t i=0) const . . . 588
10.227.3.10 Print(std::ostream &) const . . . 588
10.227.3.11 RemoveOverlay(size_t i) . . . 589
10.227.3.12 SetIconImage(IconImage const &ii) . . . 589
10.227.3.13 SetNumberOfCurves(size_t n) . . . 589
10.227.3.14 SetNumberOfOverlays(size_t n) . . . 589

10.227.4 Member Data Documentation . . . 589
10.227.4.1 Curves . . . 589
10.227.4.2 Icon . . . 589
10.227.4.3 Overlays . . . 589
10.228 gdcm::PixmapReader Class Reference . . . 589
10.228.1 Detailed Description . . . 591
10.228.2 Constructor & Destructor Documentation . . . 591
10.228.2.1 PixmapReader() . . . 591
10.228.2.2 ~PixmapReader() . . . 591
10.228.3 Member Function Documentation . . . 591
10.228.3.1 GetPixmap() const . . . 591
10.228.3.2 GetPixmap() . . . 591
10.228.3.3 Read() . . . 591
10.228.3.4 ReadACRNEMAImage() . . . 592
10.228.3.5 ReadImage(MediaStorage const &ms) . . . 592
10.228.3.6 ReadImageInternal(MediaStorage const &ms, bool handlepixeldata=true) . . . 592
10.228.4 Member Data Documentation . . . 592
10.228.4.1 PixelData . . . 592
10.229 gdcm::PixmapToPixmapFilter Class Reference . . . 592
10.229.1 Detailed Description . . . 593
10.229.2 Constructor & Destructor Documentation . . . 594
10.229.2.1 PixmapToPixmapFilter() . . . 594
10.229.2.2 ~PixmapToPixmapFilter() . . . 594
10.229.3 Member Function Documentation . . . 594
10.229.3.1 GetInput() . . . 594
10.229.3.2 GetOutput() const . . . 594
10.229.3.3 GetOutputAsPixmap() const . . . 594
10.230 gdcm::PixmapWriter Class Reference . . . 594

10.230.1 Detailed Description . . . 596
10.230.2 Constructor & Destructor Documentation . . . 596
10.230.2.1 PixmapWriter() . . . 596
10.230.2.2 ~PixmapWriter() . . . 596
10.230.3 Member Function Documentation . . . 596
10.230.3.1 DoIconImage(DataSet &ds, Pixmap const &image) . . . 596
10.230.3.2 GetImage() const . . . 596
10.230.3.3 GetImage() . . . 596
10.230.3.4 GetPixmap() const . . . 597
10.230.3.5 GetPixmap() . . . 597
10.230.3.6 PrepareWrite() . . . 597
10.230.3.7 PrepareWrite(MediaStorage const &refms) . . . 597
10.230.3.8 SetImage(Pixmap const &img) . . . 597
10.230.3.9 SetPixmap(Pixmap const &img) . . . 597
10.230.3.10 Write() . . . 597
10.230.4 Member Data Documentation . . . 597
10.230.4.1 PixelData . . . 597
10.231 gdcm::PNMCodec Class Reference . . . 598
10.231.1 Detailed Description . . . 599
10.231.2 Constructor & Destructor Documentation . . . 599
10.231.2.1 PNMCodec() . . . 599
10.231.2.2 ~PNMCodec() . . . 599
10.231.3 Member Function Documentation . . . 599
10.231.3.1 CanCode(TransferSyntax const &ts) const . . . 599
10.231.3.2 CanDecode(TransferSyntax const &ts) const . . . 599
10.231.3.3 Clone() const . . . 600
10.231.3.4 GetBufferLength() const . . . 600
10.231.3.5 GetHeaderInfo(std::istream &is, TransferSyntax &ts) . . . 600

10.231.3.6 Read(const char *filename, DataElement &out) const . . . 600
10.231.3.7 SetBufferLength(unsigned long l) . . . 600
10.231.3.8 Write(const char *filename, const DataElement &out) const . . . 600
10.232 gdcm::Preamble Class Reference . . . 600
10.232.1 Detailed Description . . . 601
10.232.2 Constructor & Destructor Documentation . . . 601
10.232.2.1 Preamble() . . . 601
10.232.2.2 ~Preamble() . . . 601
10.232.2.3 Preamble(Preamble const &) . . . 601
10.232.3 Member Function Documentation . . . 601
10.232.3.1 Clear() . . . 601
10.232.3.2 Create() . . . 601
10.232.3.3 GetInternal() const . . . 601
10.232.3.4 GetLength() const . . . 601
10.232.3.5 IsEmpty() const . . . 601
10.232.3.6 IsValid() const . . . 601
10.232.3.7 operator=(Preamble const &) . . . 601
10.232.3.8 Print(std::ostream &os) const . . . 601
10.232.3.9 Read(std::istream &is) . . . 601
10.232.3.10 Remove() . . . 601
10.232.3.11 Valid() . . . 601
10.232.3.12 Write(std::ostream &os) const . . . 601
10.232.4 Friends And Related Function Documentation . . . 601
10.232.4.1 operator<< . . . 601
10.233 gdcm::PresentationContext Class Reference . . . 602
10.233.1 Detailed Description . . . 603
10.233.2 Member Typedef Documentation . . . 603
10.233.2.1 SizeType . . . 603

10.233.2.2 TransferSyntaxArrayType . . . 603
10.233.3 Constructor & Destructor Documentation . . . 603
10.233.3.1 PresentationContext() . . . 603
10.233.3.2 PresentationContext(UIDs::TSName asname, UIDs::TSName tsname=UIDs::ImplicitVRLittleEndianDefaultTransferSyntaxforDICOM) . . . 603
10.233.4 Member Function Documentation . . . 603
10.233.4.1 AddTransferSyntax(const char *tsstr) . . . 603
10.233.4.2 GetAbstractSyntax() const . . . 603
10.233.4.3 GetNumberOfTransferSyntaxes() const . . . 603
10.233.4.4 GetPresentationContextID() const . . . 603
10.233.4.5 GetTransferSyntax(SizeType i) const . . . 603
10.233.4.6 operator==(const PresentationContext &pc) const . . . 603
10.233.4.7 Print(std::ostream &os) const . . . 604
10.233.4.8 SetAbstractSyntax(const char *absyn) . . . 604
10.233.4.9 SetPresentationContextID(uint8_t id) . . . 604
10.233.5 Member Data Documentation . . . 604
10.233.5.1 AbstractSyntax . . . 604
10.233.5.2 ID . . . 604
10.233.5.3 TransferSyntaxes . . . 604
10.234 gdcm::network::PresentationContextAC Class Reference . . . 604
10.234.1 Detailed Description . . . 605
10.234.2 Constructor & Destructor Documentation . . . 605
10.234.2.1 PresentationContextAC() . . . 605
10.234.3 Member Function Documentation . . . 605
10.234.3.1 GetPresentationContextID() const . . . 605
10.234.3.2 GetReason() const . . . 605
10.234.3.3 GetTransferSyntax() const . . . 605
10.234.3.4 Print(std::ostream &os) const . . . 605

10.234.3.5 Read(std::istream &is) . . . 605
10.234.3.6 SetPresentationContextID(uint8_t id) . . . 605
10.234.3.7 SetReason(uint8_t r) . . . 605
10.234.3.8 SetTransferSyntax(TransferSyntaxSub const &ts) . . . 605
10.234.3.9 Size() const . . . 605
10.234.3.10 Write(std::ostream &os) const . . . 605
10.235 gdcm::PresentationContextGenerator Class Reference . . . 605
10.235.1 Detailed Description . . . 606
10.235.2 Member Typedef Documentation . . . 607
10.235.2.1 PresentationContextArrayType . . . 607
10.235.2.2 SizeType . . . 607
10.235.3 Constructor & Destructor Documentation . . . 607
10.235.3.1 PresentationContextGenerator() . . . 607
10.235.4 Member Function Documentation . . . 607
10.235.4.1 AddFromFile(const File &file) . . . 607
10.235.4.2 AddPresentationContext(const char *absyn, const char *ts) . . . 607
10.235.4.3 GenerateFromFilenames(const Directory::FilenamesType &files) . . . 607
10.235.4.4 GenerateFromUID(UIDs::TSName asname) . . . 607
10.235.4.5 GetDefaultTransferSyntax() const . . . 607
10.235.4.6 GetPresentationContexts() . . . 607
10.235.4.7 SetDefaultTransferSyntax(const TransferSyntax &ts) . . . 608
10.235.4.8 SetMergeModeToAbstractSyntax() . . . 608
10.235.4.9 SetMergeModeToTransferSyntax() . . . 608
10.236 gdcm::network::PresentationContextRQ Class Reference . . . 608
10.236.1 Detailed Description . . . 609
10.236.2 Member Typedef Documentation . . . 609
10.236.2.1 SizeType . . . 609
10.236.3 Constructor & Destructor Documentation . . . 609

10.236.3.1 PresentationContextRQ() . . . 609
10.236.3.2 PresentationContextRQ(UIDs::TSName asname, UIDs::TSName tsname=UIDs::ImplicitVRLittleEndianDefaultTransferSyntaxforDICOM) . . . 609
10.236.3.3 PresentationContextRQ(const PresentationContext &pc) . . . 609
10.236.4 Member Function Documentation . . . 609
10.236.4.1 AddTransferSyntax(TransferSyntaxSub const &ts) . . . 609
10.236.4.2 GetAbstractSyntax() const . . . 609
10.236.4.3 GetAbstractSyntax() . . . 609
10.236.4.4 GetNumberOfTransferSyntaxes() const . . . 609
10.236.4.5 GetPresentationContextID() const . . . 609
10.236.4.6 GetTransferSyntax(SizeType i) const . . . 609
10.236.4.7 GetTransferSyntax(SizeType i) . . . 609
10.236.4.8 GetTransferSyntaxes() const . . . 610
10.236.4.9 operator==(const PresentationContextRQ &pc) const . . . 610
10.236.4.10 Print(std::ostream &os) const . . . 610
10.236.4.11 Read(std::istream &is) . . . 610
10.236.4.12 SetAbstractSyntax(AbstractSyntax const &absyn) . . . 610
10.236.4.13 SetPresentationContextID(uint8_t id) . . . 610
10.236.4.14 Size() const . . . 610
10.236.4.15 Write(std::ostream &os) const . . . 610
10.237 dcm::network::PresentationDataValue Class Reference . . . 610
10.237.1 Detailed Description . . . 611
10.237.2 Constructor & Destructor Documentation . . . 611
10.237.2.1 PresentationDataValue() . . . 611
10.237.3 Member Function Documentation . . . 611
10.237.3.1 ConcatenatePDVBlobs(const std::vector< PresentationDataValue > &inPDVs) . . . 611
10.237.3.2 ConcatenatePDVBlobsAsExplicit(const std::vector< PresentationDataValue > &inPDVs) . . . 611
10.237.3.3 GetBlob() const . . . 611

10.237.3.4 GetIsCommand() const . . . 611
10.237.3.5 GetIsLastFragment() const . . . 611
10.237.3.6 GetMessageHeader() const . . . 611
10.237.3.7 GetPresentationContextID() const . . . 611
10.237.3.8 Print(std::ostream &os) const . . . 611
10.237.3.9 Read(std::istream &is) . . . 611
10.237.3.10 ReadInto(std::istream &is, std::ostream &os) . . . 611
10.237.3.11 SetBlob(const std::string &partialblob) . . . 611
10.237.3.12 SetCommand(bool inCommand) . . . 611
10.237.3.13 SetDataSet(const DataSet &ds) . . . 611
10.237.3.14 SetLastFragment(bool inLast) . . . 612
10.237.3.15 SetMessageHeader(uint8_t messageheader) . . . 612
10.237.3.16 SetPresentationContextID(uint8_t id) . . . 612
10.237.3.17 Size() const . . . 612
10.237.3.18 Write(std::ostream &os) const . . . 612
10.238 gdcm::Printer Class Reference . . . 612
10.238.1 Detailed Description . . . 614
10.238.2 Member Enumeration Documentation . . . 614
10.238.2.1 PrintStyles . . . 614
10.238.3 Constructor & Destructor Documentation . . . 614
10.238.3.1 Printer() . . . 614
10.238.3.2 ~Printer() . . . 614
10.238.4 Member Function Documentation . . . 614
10.238.4.1 GetPrintStyle() const . . . 614
10.238.4.2 Print(std::ostream &os) . . . 615
10.238.4.3 PrintDataElement(std::ostringstream &os, const Dicts &dicts, const DataSet &ds, const DataElement &de, std::ostream &out, std::string const &indent) . . . 615
10.238.4.4 PrintDataSet(const DataSet &ds, std::ostream &os, const std::string &s="") . . . 615

10.238.4.5 PrintSQ(const SequenceOfItems *sqi, std::ostream &os, std::string const &indent) . . 615
10.238.4.6 SetColor(bool c) . . . 615
10.238.4.7 SetFile(File const &f) . . . 615
10.238.4.8 SetStyle(PrintStyles ps) . . . 615
10.238.5 Member Data Documentation . . . 616
10.238.5.1 F . . . 616
10.238.5.2 MaxPrintLength . . . 616
10.238.5.3 PrintStyle . . . 616
10.239 gdcm::PrivateDict Class Reference . . . 616
10.239.1 Detailed Description . . . 616
10.239.2 Constructor & Destructor Documentation . . . 617
10.239.2.1 PrivateDict() . . . 617
10.239.2.2 ~PrivateDict() . . . 617
10.239.3 Member Function Documentation . . . 617
10.239.3.1 AddDictEntry(const PrivateTag &tag, const DictEntry &de) . . . 617
10.239.3.2 FindDictEntry(const PrivateTag &tag) const . . . 617
10.239.3.3 GetDictEntry(const PrivateTag &tag) const . . . 617
10.239.3.4 IsEmpty() const . . . 617
10.239.3.5 LoadDefault() . . . 617
10.239.3.6 PrintXML() const . . . 617
10.239.3.7 RemoveDictEntry(const PrivateTag &tag) . . . 617
10.239.4 Friends And Related Function Documentation . . . 617
10.239.4.1 Dicts . . . 617
10.239.4.2 operator<< . . . 617
10.240 gdcm::PrivateTag Class Reference . . . 618
10.240.1 Detailed Description . . . 619
10.240.2 Constructor & Destructor Documentation . . . 619
10.240.2.1 PrivateTag(uint16_t group=0, uint16_t element=0, const char *owner="") . . . 619

10.240.2.2 PrivateTag(Tag const &t, const char *owner="") . . . 619
10.240.3 Member Function Documentation . . . 619
10.240.3.1 GetAsDataElement() const . . . 619
10.240.3.2 GetOwner() const . . . 619
10.240.3.3 operator<(const PrivateTag &_val) const . . . 619
10.240.3.4 ReadFromCommaSeparatedString(const char *str) . . . 619
10.240.3.5 SetOwner(const char *owner) . . . 620
10.240.4 Friends And Related Function Documentation . . . 620
10.240.4.1 operator<< . . . 620
10.241 gdcm::ProgressEvent Class Reference . . . 620
10.241.1 Detailed Description . . . 621
10.241.2 Member Typedef Documentation . . . 622
10.241.2.1 Self . . . 622
10.241.2.2 Superclass . . . 622
10.241.3 Constructor & Destructor Documentation . . . 622
10.241.3.1 ProgressEvent(double p=0) . . . 622
10.241.3.2 ~ProgressEvent() . . . 622
10.241.3.3 ProgressEvent(const Self &s) . . . 622
10.241.4 Member Function Documentation . . . 622
10.241.4.1 CheckEvent(const ::gdcm::Event *e) const . . . 622
10.241.4.2 GetEventName() const . . . 622
10.241.4.3 GetProgress() const . . . 622
10.241.4.4 MakeObject() const . . . 622
10.241.4.5 SetProgress(double p) . . . 622
10.242 gdcm::PVRGCodec Class Reference . . . 623
10.242.1 Detailed Description . . . 624
10.242.2 Constructor & Destructor Documentation . . . 624
10.242.2.1 PVRGCodec() . . . 624

10.242.2.2 ~PVRGCodec() . . . 624
10.242.3 Member Function Documentation . . . 624
10.242.3.1 CanCode(TransferSyntax const &ts) const . . . 624
10.242.3.2 CanDecode(TransferSyntax const &ts) const . . . 624
10.242.3.3 Clone() const . . . 625
10.242.3.4 Code(DataElement const &in, DataElement &out) . . . 625
10.242.3.5 Decode(DataElement const &is, DataElement &os) . . . 625
10.242.3.6 SetLossyFlag(bool l) . . . 625
10.243 gdcm::PythonFilter Class Reference . . . 625
10.243.1 Detailed Description . . . 626
10.243.2 Constructor & Destructor Documentation . . . 626
10.243.2.1 PythonFilter() . . . 626
10.243.2.2 ~PythonFilter() . . . 626
10.243.3 Member Function Documentation . . . 626
10.243.3.1 GetFile() . . . 626
10.243.3.2 GetFile() const . . . 626
10.243.3.3 SetDicts(const Dicts &dicts) . . . 626
10.243.3.4 SetFile(const File &f) . . . 626
10.243.3.5 ToPyObject(const Tag &t) const . . . 626
10.243.3.6 UseDictAlways(bool) . . . 626
10.244 gdcm::QueryBase Class Reference . . . 626
10.244.1 Detailed Description . . . 627
10.244.2 Constructor & Destructor Documentation . . . 627
10.244.2.1 ~QueryBase() . . . 627
10.244.3 Member Function Documentation . . . 627
10.244.3.1 GetAllRequiredTags(const ERootType &inRootType) const . . . 627
10.244.3.2 GetAllTags(const ERootType &inRootType) const . . . 628
10.244.3.3 GetHierachicalSearchTags(const ERootType &inRootType) const =0 . . . 628

10.244.3.4 GetName() const =0 . . . 628
10.244.3.5 GetOptionalTags(const ERootType &inRootType) const =0 . . . 628
10.244.3.6 GetQueryLevel() const =0 . . . 628
10.244.3.7 GetRequiredTags(const ERootType &inRootType) const =0 . . . 628
10.244.3.8 GetUniqueTags(const ERootType &inRootType) const =0 . . . 628
10.245 gdcm::QueryFactory Class Reference . . . 629
10.245.1 Detailed Description . . . 629
10.245.2 Member Function Documentation . . . 629
10.245.2.1 GetCharacterFromCurrentLocale() . . . 629
10.245.2.2 ListCharSets(std::ostream &os) . . . 629
10.245.2.3 ProduceCharacterSetDataElement(const std::vector< ECharSet > &inCharSetType) . . . 630
10.245.2.4 ProduceQuery(const std::string &sopInstanceUID, ENQueryType inQueryType) . . . 630
10.245.2.5 ProduceQuery(ERootType inRootType, EQueryType inQueryType, EQueryLevel in↩ QueryLevel) . . . 630
10.246 gdcm::QueryImage Class Reference . . . 630
10.246.1 Detailed Description . . . 631
10.246.2 Member Function Documentation . . . 631
10.246.2.1 GetHierachicalSearchTags(const ERootType &inRootType) const . . . 631
10.246.2.2 GetName() const . . . 631
10.246.2.3 GetOptionalTags(const ERootType &inRootType) const . . . 632
10.246.2.4 GetQueryLevel() const . . . 632
10.246.2.5 GetRequiredTags(const ERootType &inRootType) const . . . 632
10.246.2.6 GetUniqueTags(const ERootType &inRootType) const . . . 632
10.247 gdcm::QueryPatient Class Reference . . . 632
10.247.1 Detailed Description . . . 633
10.247.2 Member Function Documentation . . . 633
10.247.2.1 GetHierachicalSearchTags(const ERootType &inRootType) const . . . 633
10.247.2.2 GetName() const . . . 633

10.247.2.3 GetOptionalTags(const ERootType &inRootType) const . . . 634
10.247.2.4 GetQueryLevel() const . . . 634
10.247.2.5 GetRequiredTags(const ERootType &inRootType) const . . . 634
10.247.2.6 GetUniqueTags(const ERootType &inRootType) const . . . 634
10.248 gdcm::QuerySeries Class Reference . . . 634
10.248.1 Detailed Description . . . 635
10.248.2 Member Function Documentation . . . 635
10.248.2.1 GetHierachicalSearchTags(const ERootType &inRootType) const . . . 635
10.248.2.2 GetName() const . . . 635
10.248.2.3 GetOptionalTags(const ERootType &inRootType) const . . . 636
10.248.2.4 GetQueryLevel() const . . . 636
10.248.2.5 GetRequiredTags(const ERootType &inRootType) const . . . 636
10.248.2.6 GetUniqueTags(const ERootType &inRootType) const . . . 636
10.249 gdcm::QueryStudy Class Reference . . . 636
10.249.1 Detailed Description . . . 637
10.249.2 Member Function Documentation . . . 637
10.249.2.1 GetHierachicalSearchTags(const ERootType &inRootType) const . . . 637
10.249.2.2 GetName() const . . . 637
10.249.2.3 GetOptionalTags(const ERootType &inRootType) const . . . 638
10.249.2.4 GetQueryLevel() const . . . 638
10.249.2.5 GetRequiredTags(const ERootType &inRootType) const . . . 638
10.249.2.6 GetUniqueTags(const ERootType &inRootType) const . . . 638
10.250 gdcm::RAWCodec Class Reference . . . 638
10.250.1 Detailed Description . . . 639
10.250.2 Constructor & Destructor Documentation . . . 639
10.250.2.1 RAWCodec() . . . 639
10.250.2.2 ~RAWCodec() . . . 639
10.250.3 Member Function Documentation . . . 639

10.250.3.1 CanCode(TransferSyntax const &ts) const . . . 639
10.250.3.2 CanDecode(TransferSyntax const &ts) const . . . 640
10.250.3.3 Clone() const . . . 640
10.250.3.4 Code(DataElement const &in, DataElement &out) . . . 640
10.250.3.5 Decode(DataElement const &is, DataElement &os) . . . 640
10.250.3.6 DecodeByStreams(std::istream &is, std::ostream &os) . . . 640
10.250.3.7 DecodeBytes(const char *inBytes, size_t inBufferLength, char *outBytes, size_t inOutBufferLength) . . . 640
10.250.3.8 GetHeaderInfo(std::istream &is, TransferSyntax &ts) . . . 640
10.251 gdcm::Reader Class Reference . . . 641
10.251.1 Detailed Description . . . 643
10.251.2 Constructor & Destructor Documentation . . . 644
10.251.2.1 Reader() . . . 644
10.251.2.2 ~Reader() . . . 644
10.251.3 Member Function Documentation . . . 644
10.251.3.1 CanRead() const . . . 644
10.251.3.2 GetFile() const . . . 644
10.251.3.3 GetFile() . . . 644
10.251.3.4 GetStreamCurrentPosition() const . . . 644
10.251.3.5 GetStreamPtr() const . . . 645
10.251.3.6 Read() . . . 645
10.251.3.7 ReadDataSet() . . . 645
10.251.3.8 ReadMetaInformation() . . . 645
10.251.3.9 ReadPreamble() . . . 645
10.251.3.10 ReadSelectedPrivateTags(std::set< PrivateTag > const &ptags, bool readvalues=true) 645
10.251.3.11 ReadSelectedTags(std::set< Tag > const &tags, bool readvalues=true) . . . 645
10.251.3.12 ReadUpToTag(const Tag &tag, std::set< Tag > const &skiptags=std::set< Tag >()) . 645
10.251.3.13 SetFile(File &file) . . . 646

10.251.3.12SetFileName(const char *filename_native) . . . 646
10.251.3.13SetStream(std::istream &input_stream) . . . 646
10.251.4 Friends And Related Function Documentation . . . 646
10.251.4.1StreamImageReader . . . 646
10.251.5 Member Data Documentation . . . 646
10.251.5.1F . . . 646
10.252gdcm::RealWorldValueMappingContent Struct Reference . . . 647
10.252.1 Member Data Documentation . . . 647
10.252.1.1CodeMeaning . . . 647
10.252.1.2CodeValue . . . 647
10.252.1.3RealWorldValueIntercept . . . 647
10.252.1.4RealWorldValueSlope . . . 647
10.253gdcm::Region Class Reference . . . 648
10.253.1 Detailed Description . . . 648
10.253.2 Constructor & Destructor Documentation . . . 649
10.253.2.1Region() . . . 649
10.253.2.2~Region() . . . 649
10.253.3 Member Function Documentation . . . 649
10.253.3.1Area() const =0 . . . 649
10.253.3.2Clone() const =0 . . . 649
10.253.3.3ComputeBoundingBox()=0 . . . 649
10.253.3.4Empty() const =0 . . . 649
10.253.3.5IsValid() const =0 . . . 649
10.253.3.6Print(std::ostream &os=std::cout) const . . . 649
10.254gdcm::Rescaler Class Reference . . . 650
10.254.1 Detailed Description . . . 651
10.254.2 Constructor & Destructor Documentation . . . 651
10.254.2.1Rescaler() . . . 651

10.254.2.2 ~Rescaler() . . . 651
10.254.3 Member Function Documentation . . . 651
10.254.3.1 ComputeInterceptSlopePixelType() . . . 651
10.254.3.2 ComputePixelTypeFromMinMax() . . . 652
10.254.3.3 GetIntercept() const . . . 652
10.254.3.4 GetSlope() const . . . 652
10.254.3.5 InverseRescale(char *out, const char *in, size_t n) . . . 652
10.254.3.6 InverseRescaleFunctionIntoBestFit(char *out, const TIn *in, size_t n) . . . 652
10.254.3.7 Rescale(char *out, const char *in, size_t n) . . . 652
10.254.3.8 RescaleFunctionIntoBestFit(char *out, const TIn *in, size_t n) . . . 652
10.254.3.9 SetIntercept(double i) . . . 652
10.254.3.10 SetMinMaxForPixelType(double min, double max) . . . 652
10.254.3.11 SetPixelFormat(PixelFormat const &pf) . . . 652
10.254.3.12 SetSlope(double s) . . . 652
10.254.3.13 SetTargetPixelType(PixelFormat const &targetst) . . . 653
10.254.3.14 SetUseTargetPixelType(bool b) . . . 653
10.255 gdcm::RLECodec Class Reference . . . 653
10.255.1 Detailed Description . . . 655
10.255.2 Constructor & Destructor Documentation . . . 655
10.255.2.1 RLECodec() . . . 655
10.255.2.2 ~RLECodec() . . . 655
10.255.3 Member Function Documentation . . . 655
10.255.3.1 AppendFrameEncode(std::ostream &out, const char *data, size_t datalen) . . . 655
10.255.3.2 AppendRowEncode(std::ostream &out, const char *data, size_t datalen) . . . 655
10.255.3.3 CanCode(TransferSyntax const &ts) const . . . 655
10.255.3.4 CanDecode(TransferSyntax const &ts) const . . . 655
10.255.3.5 Clone() const . . . 656
10.255.3.6 Code(DataElement const &in, DataElement &out) . . . 656

10.255.3.7 Decode(DataElement const &is, DataElement &os) . . . 656
10.255.3.8 DecodeByStreams(std::istream &is, std::ostream &os) . . . 656
10.255.3.9 DecodeExtent(char *buffer, unsigned int XMin, unsigned int XMax, unsigned int YMin, unsigned int YMax, unsigned int ZMin, unsigned int ZMax, std::istream &is) . . . 656
10.255.3.10 GetBufferLength() const . . . 656
10.255.3.11 GetHeaderInfo(std::istream &is, TransferSyntax &ts) . . . 656
10.255.3.12 SetFrameEncoder() . . . 656
10.255.3.13 SetRowEncoder() . . . 656
10.255.3.14 SetBufferLength(unsigned long l) . . . 657
10.255.3.15 SetLength(unsigned long l) . . . 657
10.255.3.16 StartEncode(std::ostream &) . . . 657
10.255.3.17 StopEncode(std::ostream &) . . . 657
10.255.4 Friends And Related Function Documentation . . . 657
10.255.4.1 ImageRegionReader . . . 657
10.256 gdcm::network::RoleSelectionSub Class Reference . . . 657
10.256.1 Detailed Description . . . 657
10.256.2 Constructor & Destructor Documentation . . . 658
10.256.2.1 RoleSelectionSub() . . . 658
10.256.3 Member Function Documentation . . . 658
10.256.3.1 Print(std::ostream &os) const . . . 658
10.256.3.2 Read(std::istream &is) . . . 658
10.256.3.3 SetTuple(const char *uid, uint8_t scurole, uint8_t scprole) . . . 658
10.256.3.4 Size() const . . . 658
10.256.3.5 Write(std::ostream &os) const . . . 658
10.257 gdcm::SerieHelper::Rule Struct Reference . . . 658
10.257.1 Member Data Documentation . . . 659
10.257.1.1 elem . . . 659
10.257.1.2 group . . . 659

10.257.1.3 op . . . 659
10.257.1.4 value . . . 659
10.258 gdcm::Scanner Class Reference . . . 659
10.258.1 Detailed Description . . . 661
10.258.2 Member Typedef Documentation . . . 662
10.258.2.1 ConstIterator . . . 662
10.258.2.2 MappingType . . . 662
10.258.2.3 TagToValue . . . 662
10.258.2.4 TagToValueValueType . . . 662
10.258.2.5 ValuesType . . . 662
10.258.3 Constructor & Destructor Documentation . . . 662
10.258.3.1 Scanner() . . . 662
10.258.3.2 ~Scanner() . . . 662
10.258.4 Member Function Documentation . . . 662
10.258.4.1 AddPrivateTag(PrivateTag const &t) . . . 662
10.258.4.2 AddSkipTag(Tag const &t) . . . 662
10.258.4.3 AddTag(Tag const &t) . . . 663
10.258.4.4 Begin() const . . . 663
10.258.4.5 ClearSkipTags() . . . 663
10.258.4.6 ClearTags() . . . 663
10.258.4.7 End() const . . . 663
10.258.4.8 GetAllFilenamesFromTagToValue(Tag const &t, const char *valueref) const . . . 663
10.258.4.9 GetFilenameFromTagToValue(Tag const &t, const char *valueref) const . . . 663
10.258.4.10 GetFilenames() const . . . 663
10.258.4.11 GetKeys() const . . . 663
10.258.4.12 GetMapping(const char *filename) const . . . 663
10.258.4.13 GetMappingFromTagToValue(Tag const &t, const char *value) const . . . 664
10.258.4.14 GetMappings() const . . . 664

10.258.4.16GetOrderedValues(Tag const &t) const . . . 664
10.258.4.17GetValue(const char *filename, Tag const &t) const . . . 664
10.258.4.18GetValues() const . . . 664
10.258.4.19GetValues(Tag const &t) const . . . 664
10.258.4.20IsKey(const char *filename) const . . . 664
10.258.4.21New() . . . 665
10.258.4.22Print(std::ostream &os) const . . . 665
10.258.4.23ProcessPublicTag(StringFilter &sf, const char *filename) . . . 665
10.258.4.24Scan(Directory::FilenamesType const &filenames) . . . 665
10.258.5Friends And Related Function Documentation . . . 665
10.258.5.1operator<< . . . 665
10.259gdcm::Segment Class Reference . . . 666
10.259.1Detailed Description . . . 668
10.259.2Member Typedef Documentation . . . 668
10.259.2.1SurfaceVector . . . 668
10.259.3Member Enumeration Documentation . . . 668
10.259.3.1ALGOType . . . 668
10.259.4Constructor & Destructor Documentation . . . 668
10.259.4.1Segment() . . . 668
10.259.4.2~Segment() . . . 668
10.259.5Member Function Documentation . . . 668
10.259.5.1AddSurface(SmartPointer< Surface > surface) . . . 668
10.259.5.2GetALGOType(const char *type) . . . 668
10.259.5.3GetALGOTypeString(ALGOType type) . . . 668
10.259.5.4GetAnatomicRegion() const . . . 668
10.259.5.5GetAnatomicRegion() . . . 668
10.259.5.6GetPropertyCategory() const . . . 669
10.259.5.7GetPropertyCategory() . . . 669

10.259.5.8 GetPropertyType() const . . . 669
10.259.5.9 GetPropertyType() . . . 669
10.259.5.10 GetSegmentAlgorithmName() const . . . 669
10.259.5.11 GetSegmentAlgorithmType() const . . . 669
10.259.5.12 GetSegmentDescription() const . . . 669
10.259.5.13 GetSegmentLabel() const . . . 669
10.259.5.14 GetSegmentNumber() const . . . 669
10.259.5.15 GetSurface(const unsigned int idx=0) const . . . 669
10.259.5.16 GetSurfaceCount() . . . 669
10.259.5.17 GetSurfaces() const . . . 669
10.259.5.18 GetSurfaces() . . . 669
10.259.5.19 SetAnatomicRegion(SegmentHelper::BasicCodedEntry const &BSE) . . . 669
10.259.5.20 SetPropertyCategory(SegmentHelper::BasicCodedEntry const &BSE) . . . 669
10.259.5.21 SetPropertyType(SegmentHelper::BasicCodedEntry const &BSE) . . . 669
10.259.5.22 SetSegmentAlgorithmName(const char *name) . . . 669
10.259.5.23 SetSegmentAlgorithmType(ALGOType type) . . . 669
10.259.5.24 SetSegmentAlgorithmType(const char *typeStr) . . . 669
10.259.5.25 SetSegmentDescription(const char *description) . . . 669
10.259.5.26 SetSegmentLabel(const char *label) . . . 669
10.259.5.27 SetSegmentNumber(const unsigned short num) . . . 670
10.259.5.28 SetSurfaceCount(const unsigned long nb) . . . 670
10.259.6 Member Data Documentation . . . 670
10.259.6.1 AnatomicRegion . . . 670
10.259.6.2 PropertyCategory . . . 670
10.259.6.3 PropertyType . . . 670
10.259.6.4 SegmentAlgorithmName . . . 670
10.259.6.5 SegmentAlgorithmType . . . 670
10.259.6.6 SegmentDescription . . . 670

10.259.6.7 SegmentLabel . . . 670
10.259.6.8 SegmentNumber . . . 670
10.259.6.9 SurfaceCount . . . 670
10.259.6.10 Surfaces . . . 670
10.260 gdcm::SegmentedPaletteColorLookupTable Class Reference . . . 671
10.260.1 Detailed Description . . . 672
10.260.2 Constructor & Destructor Documentation . . . 672
10.260.2.1 SegmentedPaletteColorLookupTable() . . . 672
10.260.2.2 ~SegmentedPaletteColorLookupTable() . . . 672
10.260.3 Member Function Documentation . . . 672
10.260.3.1 Print(std::ostream &) const . . . 672
10.260.3.2 SetLUT(LookupTableType type, const unsigned char *array, unsigned int length) . . . 672
10.261 gdcm::SegmentReader Class Reference . . . 673
10.261.1 Detailed Description . . . 674
10.261.2 Member Typedef Documentation . . . 675
10.261.2.1 SegmentMap . . . 675
10.261.2.2 SegmentVector . . . 675
10.261.3 Constructor & Destructor Documentation . . . 675
10.261.3.1 SegmentReader() . . . 675
10.261.3.2 ~SegmentReader() . . . 675
10.261.4 Member Function Documentation . . . 675
10.261.4.1 GetSegments() const . . . 675
10.261.4.2 GetSegments() . . . 675
10.261.4.3 Read() . . . 675
10.261.4.4 ReadSegment(const Item &segmentItem, const unsigned int idx) . . . 675
10.261.4.5 ReadSegments() . . . 675
10.261.5 Member Data Documentation . . . 675
10.261.5.1 Segments . . . 675

10.262gdcm::SegmentWriter Class Reference . . . 676
10.262.1Detailed Description . . . 677
10.262.2Member Typedef Documentation . . . 677
10.262.2.1SegmentVector . . . 677
10.262.3Constructor & Destructor Documentation . . . 677
10.262.3.1SegmentWriter() . . . 677
10.262.3.2∼SegmentWriter() . . . 677
10.262.4Member Function Documentation . . . 677
10.262.4.1AddSegment(SmartPointer< Segment > segment) . . . 677
10.262.4.2GetNumberOfSegments() const . . . 677
10.262.4.3GetSegment(const unsigned int idx=0) const . . . 677
10.262.4.4GetSegments() const . . . 677
10.262.4.5GetSegments() . . . 677
10.262.4.6PrepareWrite() . . . 677
10.262.4.7SetNumberOfSegments(const unsigned int size) . . . 677
10.262.4.8SetSegments(SegmentVector &segments) . . . 677
10.262.4.9Write() . . . 677
10.262.5Member Data Documentation . . . 678
10.262.5.1Segments . . . 678
10.263gdcm::SequenceOfFragments Class Reference . . . 678
10.263.1Detailed Description . . . 680
10.263.2Member Typedef Documentation . . . 680
10.263.2.1ConstIterator . . . 680
10.263.2.2FragmentVector . . . 680
10.263.2.3Iterator . . . 680
10.263.2.4SizeType . . . 680
10.263.3Constructor & Destructor Documentation . . . 680
10.263.3.1SequenceOfFragments() . . . 680

10.263.4 Member Function Documentation . . . 681
10.263.4.1 AddFragment(Fragment const &item) . . . 681
10.263.4.2 Begin() . . . 681
10.263.4.3 Begin() const . . . 681
10.263.4.4 Clear() . . . 681
10.263.4.5 ComputeByteLength() const . . . 681
10.263.4.6 ComputeLength() const . . . 681
10.263.4.7 End() . . . 681
10.263.4.8 End() const . . . 681
10.263.4.9 GetBuffer(char *buffer, unsigned long length) const . . . 681
10.263.4.10 GetFragBuffer(unsigned int fragNb, char *buffer, unsigned long &length) const . . . 681
10.263.4.11 GetFragment(SizeType num) const . . . 681
10.263.4.12 GetLength() const . . . 681
10.263.4.13 GetNumberOfFragments() const . . . 682
10.263.4.14 GetTable() const . . . 682
10.263.4.15 GetTable() . . . 682
10.263.4.16 New() . . . 682
10.263.4.17 operator==(const Value &val) const . . . 682
10.263.4.18 Print(std::ostream &os) const . . . 682
10.263.4.19 Read(std::istream &is, bool readvalues=true) . . . 682
10.263.4.20 ReadPreValue(std::istream &is) . . . 682
10.263.4.21 ReadValue(std::istream &is, bool) . . . 682
10.263.4.22 SetLength(VL length) . . . 682
10.263.4.23 Write(std::ostream &os) const . . . 683
10.263.4.24 WriteBuffer(std::ostream &os) const . . . 683
10.264 gdcm::SequenceOfItems Class Reference . . . 683
10.264.1 Detailed Description . . . 685
10.264.2 Member Typedef Documentation . . . 686

10.264.2.1 ConstIterator . . . 686
10.264.2.2 ItemVector . . . 686
10.264.2.3 Iterator . . . 686
10.264.2.4 SizeType . . . 686
10.264.3 Constructor & Destructor Documentation . . . 686
10.264.3.1 SequenceOfItems() . . . 686
10.264.4 Member Function Documentation . . . 686
10.264.4.1 AddItem(Item const &item) . . . 686
10.264.4.2 AddNewUndefinedLengthItem() . . . 686
10.264.4.3 Begin() . . . 687
10.264.4.4 Begin() const . . . 687
10.264.4.5 Clear() . . . 687
10.264.4.6 ComputeLength() const . . . 687
10.264.4.7 End() . . . 687
10.264.4.8 End() const . . . 687
10.264.4.9 FindDataElement(const Tag &t) const . . . 687
10.264.4.10 GetItem(SizeType position) const . . . 687
10.264.4.11 GetItem(SizeType position) . . . 687
10.264.4.12 GetLength() const . . . 687
10.264.4.13 GetNumberOfItems() const . . . 687
10.264.4.14 IsUndefinedLength() const . . . 688
10.264.4.15 New() . . . 688
10.264.4.16 operator=(const SequenceOfItems &val) . . . 688
10.264.4.17 operator==(const Value &val) const . . . 688
10.264.4.18 Print(std::ostream &os) const . . . 688
10.264.4.19 Read(std::istream &is, bool readvalues=true) . . . 688
10.264.4.20 RemoveItemByIndex(const SizeType index) . . . 688
10.264.4.21 SetLength(VL length) . . . 689

10.264.4.22 SetLengthToUndefined() . . . 689
10.264.4.23 SetNumberOfItems(SizeType n) . . . 689
10.264.4.24 Write(std::ostream &os) const . . . 689
10.264.5 Member Data Documentation . . . 689
10.264.5.1 Items . . . 689
10.264.5.2 SequenceLengthField . . . 689
10.265 gdcm::SerieHelper Class Reference . . . 690
10.265.1 Detailed Description . . . 691
10.265.2 Member Typedef Documentation . . . 691
10.265.2.1 SerieRestrictions . . . 691
10.265.2.2 SingleSerieUIDFileSetmap . . . 691
10.265.3 Constructor & Destructor Documentation . . . 691
10.265.3.1 SerieHelper() . . . 691
10.265.3.2 ~SerieHelper() . . . 691
10.265.4 Member Function Documentation . . . 691
10.265.4.1 AddFile(FileWithName &header) . . . 691
10.265.4.2 AddFileName(std::string const &filename) . . . 692
10.265.4.3 AddRestriction(const std::string &tag) . . . 692
10.265.4.4 AddRestriction(uint16_t group, uint16_t elem, std::string const &value, int op) . . . 692
10.265.4.5 AddRestriction(const Tag &tag) . . . 692
10.265.4.6 Clear() . . . 692
10.265.4.7 CreateDefaultUniqueSeriesIdentifier() . . . 692
10.265.4.8 CreateUniqueSeriesIdentifier(File *inFile) . . . 692
10.265.4.9 FileNameOrdering(FileList *fileList) . . . 692
10.265.4.10 GetFirstSingleSerieUIDFileSet() . . . 692
10.265.4.11 GetNextSingleSerieUIDFileSet() . . . 692
10.265.4.12 ImagePositionPatientOrdering(FileList *fileSet) . . . 692
10.265.4.13 OrderFileList(FileList *fileSet) . . . 692

10.265.4.13SetDirectory(std::string const &dir, bool recursive=false) . . . 692
10.265.4.15SetLoadMode(int) . . . 692
10.265.4.16SetUseSeriesDetails(bool useSeriesDetails) . . . 692
10.265.4.17UserOrdering(FileList *fileSet) . . . 692
10.265.5Member Data Documentation . . . 692
10.265.5.1ItFileSetHt . . . 692
10.265.5.2SingleSerieUIDFileSetHT . . . 692
10.266gdcm::Series Class Reference . . . 693
10.266.1Detailed Description . . . 693
10.266.2Constructor & Destructor Documentation . . . 693
10.266.2.1Series() . . . 693
10.267gdcm::network::ServiceClassApplicationInformation Class Reference . . . 693
10.267.1Detailed Description . . . 693
10.267.2Constructor & Destructor Documentation . . . 694
10.267.2.1ServiceClassApplicationInformation() . . . 694
10.267.3Member Function Documentation . . . 694
10.267.3.1Print(std::ostream &os) const . . . 694
10.267.3.2Read(std::istream &is) . . . 694
10.267.3.3SetTuple(uint8_t levelofsupport, uint8_t levelofdigitalsig, uint8_t elementcoercion) . . . 694
10.267.3.4Size() const . . . 694
10.267.3.5Write(std::ostream &os) const . . . 694
10.268gdcm::ServiceClassUser Class Reference . . . 694
10.268.1Detailed Description . . . 696
10.268.2Constructor & Destructor Documentation . . . 696
10.268.2.1ServiceClassUser() . . . 696
10.268.2.2~ServiceClassUser() . . . 697
10.268.3Member Function Documentation . . . 697
10.268.3.1GetAETitle() const . . . 697

10.268.3.2 GetCalledAETitle() const . . . 697
10.268.3.3 GetTimeout() const . . . 697
10.268.3.4 InitializeConnection() . . . 697
10.268.3.5 IsPresentationContextAccepted(const PresentationContext &pc) const . . . 697
10.268.3.6 New() . . . 697
10.268.3.7 SendEcho() . . . 697
10.268.3.8 SendFind(const BaseRootQuery *query, std::vector< DataSet > &retDatasets) . . . 697
10.268.3.9 SendMove(const BaseRootQuery *query, const char *outputdir) . . . 697
10.268.3.10 SendMove(const BaseRootQuery *query, std::vector< DataSet > &retDatasets) . . . 698
10.268.3.11 SendMove(const BaseRootQuery *query, std::vector< File > &retFile) . . . 698
10.268.3.12 SendStore(const char *filename) . . . 698
10.268.3.13 SendStore(File const &file) . . . 698
10.268.3.14 SendStore(DataSet const &ds) . . . 698
10.268.3.15 SetAETitle(const char *aetitle) . . . 698
10.268.3.16 SetCalledAETitle(const char *aetitle) . . . 698
10.268.3.17 SetHostname(const char *hostname) . . . 699
10.268.3.18 SetPort(uint16_t port) . . . 699
10.268.3.19 SetPortSCP(uint16_t portscp) . . . 699
10.268.3.20 SetPresentationContexts(std::vector< PresentationContext > const &pcs) . . . 699
10.268.3.21 SetTimeout(double t) . . . 699
10.268.3.22 StartAssociation() . . . 699
10.268.3.23 StopAssociation() . . . 700
10.269 gdcm::SHA1 Class Reference . . . 700
10.269.1 Detailed Description . . . 700
10.269.2 Constructor & Destructor Documentation . . . 701
10.269.2.1 SHA1() . . . 701
10.269.2.2 ~SHA1() . . . 701
10.269.3 Member Function Documentation . . . 701

10.269.3.1 Compute(const char *buffer, unsigned long buf_len, char digest_str[20 *2+1]) 701
10.269.3.2 ComputeFile(const char *filename, char digest_str[20 *2+1]) 701
10.270 gdcm::SimpleMemberCommand< T > Class Template Reference 701
10.270.1 Detailed Description 703
10.270.2 Member Typedef Documentation 703
10.270.2.1 Self 703
10.270.2.2 TMemberFunctionPointer 703
10.270.3 Constructor & Destructor Documentation 703
10.270.3.1 SimpleMemberCommand() 703
10.270.3.2 ~SimpleMemberCommand() 703
10.270.4 Member Function Documentation 703
10.270.4.1 Execute(Subject *, const Event &) 703
10.270.4.2 Execute(const Subject *, const Event &) 704
10.270.4.3 New() 704
10.270.4.4 SetCallbackFunction(T *object, TMemberFunctionPointer memberFunction) 704
10.270.5 Member Data Documentation 704
10.270.5.1 m_MemberFunction 704
10.270.5.2 m_This 704
10.271 gdcm::SimpleSubjectWatcher Class Reference 704
10.271.1 Detailed Description 705
10.271.2 Constructor & Destructor Documentation 705
10.271.2.1 SimpleSubjectWatcher(Subject *s, const char *comment="") 705
10.271.2.2 ~SimpleSubjectWatcher() 705
10.271.3 Member Function Documentation 705
10.271.3.1 EndFilter() 705
10.271.3.2 ShowAbort() 705
10.271.3.3 ShowAnonymization(Subject *caller, const Event &evt) 705
10.271.3.4 ShowData(Subject *caller, const Event &evt) 705

10.271.3.5 ShowDataSet(Subject *caller, const Event &evt) . . . 705
10.271.3.6 ShowFileName(Subject *caller, const Event &evt) . . . 705
10.271.3.7 ShowIteration() . . . 706
10.271.3.8 ShowProgress(Subject *caller, const Event &evt) . . . 706
10.271.3.9 StartFilter() . . . 706
10.271.3.10 TestAbortOff() . . . 706
10.271.3.11 TestAbortOn() . . . 706
10.272 gdcm::SmartPointer< ObjectType > Class Template Reference . . . 706
10.272.1 Detailed Description . . . 708
10.272.2 Constructor & Destructor Documentation . . . 708
10.272.2.1 SmartPointer() . . . 708
10.272.2.2 SmartPointer(const SmartPointer< ObjectType > &p) . . . 708
10.272.2.3 SmartPointer(ObjectType *p) . . . 708
10.272.2.4 SmartPointer(ObjectType const &p) . . . 708
10.272.2.5 ~SmartPointer() . . . 708
10.272.3 Member Function Documentation . . . 708
10.272.3.1 GetPointer() const . . . 708
10.272.3.2 operator ObjectType *() const . . . 709
10.272.3.3 operator*() const . . . 709
10.272.3.4 operator->() const . . . 709
10.272.3.5 operator=(SmartPointer const &r) . . . 709
10.272.3.6 operator=(ObjectType *r) . . . 709
10.272.3.7 operator=(ObjectType const &r) . . . 709
10.273 gdcm::network::SOPClassExtendedNegociationSub Class Reference . . . 709
10.273.1 Detailed Description . . . 710
10.273.2 Constructor & Destructor Documentation . . . 710
10.273.2.1 SOPClassExtendedNegociationSub() . . . 710
10.273.3 Member Function Documentation . . . 710

10.273.3.1 Print(std::ostream &os) const . . . 710
10.273.3.2 Read(std::istream &is) . . . 710
10.273.3.3 SetTuple(const char *uid, uint8_t levelofsupport=3, uint8_t levelofdigitalsig=0, uint8_t elementcoercion=2) . . . 710
10.273.3.4 Size() const . . . 710
10.273.3.5 Write(std::ostream &os) const . . . 710
10.274 gdcm::SOPClassUIDToIOD Class Reference . . . 710
10.274.1 Detailed Description . . . 711
10.274.2 Member Typedef Documentation . . . 711
10.274.2.1 const . . . 711
10.274.3 Member Function Documentation . . . 711
10.274.3.1 GetIOD(UIDs const &uid) . . . 711
10.274.3.2 GetIODFromSOPClassUID(const char *sopclassuid) . . . 711
10.274.3.3 GetNumberOfSOPClassToIOD() . . . 711
10.274.3.4 GetSOPClassUIDFromIOD(const char *iod) . . . 712
10.274.3.5 GetSOPClassUIDToIOD(unsigned int i) . . . 712
10.274.3.6 GetSOPClassUIDToIODs() . . . 712
10.275 gdcm::Sorter Class Reference . . . 712
10.275.1 Detailed Description . . . 713
10.275.2 Member Typedef Documentation . . . 714
10.275.2.1 SelectionMap . . . 714
10.275.2.2 SortFunction . . . 714
10.275.3 Constructor & Destructor Documentation . . . 714
10.275.3.1 Sorter() . . . 714
10.275.3.2 ~Sorter() . . . 714
10.275.4 Member Function Documentation . . . 714
10.275.4.1 AddSelect(Tag const &tag, const char *value) . . . 714
10.275.4.2 GetFilenames() const . . . 714

10.275.4.3 Print(std::ostream &os) const . . . 714
10.275.4.4 SetSortFunction(SortFunction f) . . . 714
10.275.4.5 Sort(std::vector< std::string > const &filenames) . . . 715
10.275.4.6 StableSort(std::vector< std::string > const &filenames) . . . 715
10.275.5 Friends And Related Function Documentation . . . 715
10.275.5.1 operator<< . . . 715
10.275.6 Member Data Documentation . . . 715
10.275.6.1 Filenames . . . 715
10.275.6.2 Selection . . . 715
10.275.6.3 SortFunc . . . 715
10.276 gdcm::Spacing Class Reference . . . 715
10.276.1 Detailed Description . . . 716
10.276.2 Member Enumeration Documentation . . . 717
10.276.2.1 SpacingType . . . 717
10.276.3 Constructor & Destructor Documentation . . . 717
10.276.3.1 Spacing() . . . 717
10.276.3.2 ~Spacing() . . . 717
10.276.4 Member Function Documentation . . . 717
10.276.4.1 ComputePixelAspectRatioFromPixelSpacing(const Attribute< 0x28, 0x30 > &pixelspacing) . . . 717
10.277 gdcm::Spectroscopy Class Reference . . . 717
10.277.1 Detailed Description . . . 717
10.277.2 Constructor & Destructor Documentation . . . 718
10.277.2.1 Spectroscopy() . . . 718
10.278 gdcm::SplitMosaicFilter Class Reference . . . 718
10.278.1 Detailed Description . . . 718
10.278.2 Constructor & Destructor Documentation . . . 718
10.278.2.1 SplitMosaicFilter() . . . 718

10.278.2.2 ~SplitMosaicFilter() . . . 718
10.278.3 Member Function Documentation . . . 718
10.278.3.1 ComputeMOSAICDimensions(unsigned int dims[3]) . . . 718
10.278.3.2 GetFile() . . . 719
10.278.3.3 GetFile() const . . . 719
10.278.3.4 GetImage() const . . . 719
10.278.3.5 GetImage() . . . 719
10.278.3.6 SetFile(const File &f) . . . 719
10.278.3.7 SetImage(const Image &image) . . . 719
10.278.3.8 Split() . . . 719
10.279 gdcm::StartEvent Class Reference . . . 719
10.280 gdcm::static_assert_test< x > Struct Template Reference . . . 720
10.281 gdcm::STATIC_ASSERTION_FAILURE< x > Struct Template Reference . . . 720
10.282 gdcm::STATIC_ASSERTION_FAILURE< true > Struct Template Reference . . . 721
10.282.1 Member Enumeration Documentation . . . 721
10.282.1.1 anonymous enum . . . 721
10.283 gdcm::StreamImageReader Class Reference . . . 721
10.283.1 Detailed Description . . . 722
10.283.2 Constructor & Destructor Documentation . . . 722
10.283.2.1 StreamImageReader() . . . 722
10.283.2.2 ~StreamImageReader() . . . 722
10.283.3 Member Function Documentation . . . 722
10.283.3.1 CanReadImage() const . . . 722
10.283.3.2 DefinePixelExtent(uint16_t inXMin, uint16_t inXMax, uint16_t inYMin, uint16_t inY↩ Max, uint16_t inZMin=0, uint16_t inZMax=1) . . . 722
10.283.3.3 DefineProperBufferLength() const . . . 723
10.283.3.4 GetDimensionsValueForResolution(unsigned int) . . . 723
10.283.3.5 GetFile() const . . . 723

10.283.3.6 Read(char *inReadBuffer, const std::size_t &inBufferLength) . . . 723
10.283.3.7 ReadImageInformation() . . . 723
10.283.3.8 SetFileName(const char *inFileName) . . . 724
10.283.3.9 SetStream(std::istream &inStream) . . . 724
10.284 gdcm::StreamImageWriter Class Reference . . . 724
10.284.1 Detailed Description . . . 726
10.284.2 Constructor & Destructor Documentation . . . 727
10.284.2.1 StreamImageWriter() . . . 727
10.284.2.2 ~StreamImageWriter() . . . 727
10.284.3 Member Function Documentation . . . 727
10.284.3.1 CanWriteFile() const . . . 727
10.284.3.2 DefinePixelExtent(uint16_t inXMin, uint16_t inXMax, uint16_t inYMin, uint16_t inYMax, uint16_t inZMin=0, uint16_t inZMax=1) . . . 727
10.284.3.3 DefineProperBufferLength() . . . 727
10.284.3.4 SetFile(const File &inFile) . . . 727
10.284.3.5 SetFileName(const char *inFileName) . . . 728
10.284.3.6 SetStream(std::ostream &inStream) . . . 728
10.284.3.7 Write(void *inWriteBuffer, const std::size_t &inBufferLength) . . . 728
10.284.3.8 WriteImageInformation() . . . 728
10.284.3.9 WriteImageSubregionRAW(char *inWriteBuffer, const std::size_t &inBufferLength) . . . 728
10.284.3.10 WriteRawHeader(RAWCodec *inCodec, std::ostream *inStream) . . . 729
10.284.4 Member Data Documentation . . . 729
10.284.4.1 mElementOffsets . . . 729
10.284.4.2 mElementOffsets1 . . . 729
10.284.4.3 mspFile . . . 729
10.284.4.4 mWriter . . . 729
10.284.4.5 mXMax . . . 729
10.284.4.6 mXMin . . . 729

10.284.4.7 mYMax . . . 729
10.284.4.8 mYMin . . . 729
10.284.4.9 mZMax . . . 729
10.284.4.10 mZMin . . . 729
10.285 gdcm::StrictScanner Class Reference . . . 730
10.285.1 Detailed Description . . . 732
10.285.2 Member Typedef Documentation . . . 732
10.285.2.1 ConstIterator . . . 732
10.285.2.2 MappingType . . . 732
10.285.2.3 TagToValue . . . 732
10.285.2.4 TagToValueValueType . . . 733
10.285.2.5 ValuesType . . . 733
10.285.3 Constructor & Destructor Documentation . . . 733
10.285.3.1 StrictScanner() . . . 733
10.285.3.2 ~StrictScanner() . . . 733
10.285.4 Member Function Documentation . . . 733
10.285.4.1 AddPrivateTag(PrivateTag const &t) . . . 733
10.285.4.2 AddSkipTag(Tag const &t) . . . 733
10.285.4.3 AddTag(Tag const &t) . . . 733
10.285.4.4 Begin() const . . . 733
10.285.4.5 ClearSkipTags() . . . 733
10.285.4.6 ClearTags() . . . 733
10.285.4.7 End() const . . . 733
10.285.4.8 GetAllFilenamesFromTagToValue(Tag const &t, const char *valueref) const . . . 733
10.285.4.9 GetFilenameFromTagToValue(Tag const &t, const char *valueref) const . . . 733
10.285.4.10 GetFilenames() const . . . 734
10.285.4.11 GetKeys() const . . . 734
10.285.4.12 GetMapping(const char *filename) const . . . 734

10.285.4.13GetMappingFromTagToValue(Tag const &t, const char ∗value) const 734
10.285.4.14GetMappings() const 734
10.285.4.15GetOrderedValues(Tag const &t) const 734
10.285.4.16GetValue(const char ∗filename, Tag const &t) const 734
10.285.4.17GetValues() const 734
10.285.4.18GetValues(Tag const &t) const 735
10.285.4.19HasKey(const char ∗filename) const 735
10.285.4.20New() 735
10.285.4.21Print(std::ostream &os) const 735
10.285.4.22ProcessPublicTag(StringFilter &sf, const char ∗filename) 735
10.285.4.23Scan(Directory::FilenamesType const &filenames) 735
10.285.5Friends And Related Function Documentation 735
10.285.5.1operator<< 735
10.286gdcm::String< TDelimiter, TMaxLength, TPadChar > Class Template Reference 736
10.286.1Detailed Description 737
10.286.2Member Typedef Documentation 738
10.286.2.1const_iterator 738
10.286.2.2const_reference 738
10.286.2.3const_reverse_iterator 738
10.286.2.4difference_type 738
10.286.2.5iterator 738
10.286.2.6pointer 738
10.286.2.7reference 738
10.286.2.8reverse_iterator 738
10.286.2.9size_type 738
10.286.2.10value_type 738
10.286.3Constructor & Destructor Documentation 738
10.286.3.1String() 738

10.286.3.2 String(const value_type *s) . . . 739
10.286.3.3 String(const value_type *s, size_type n) . . . 739
10.286.3.4 String(const std::string &s, size_type pos=0, size_type n=npos) . . . 739
10.286.4 Member Function Documentation . . . 739
10.286.4.1 IsValid() const . . . 739
10.286.4.2 operator const char *() const . . . 739
10.286.4.3 Trim() const . . . 739
10.286.4.4 Trim(const char *input) . . . 739
10.286.4.5 Truncate() const . . . 739
10.287 gdcm::StringFilter Class Reference . . . 740
10.287.1 Detailed Description . . . 740
10.287.2 Constructor & Destructor Documentation . . . 741
10.287.2.1 StringFilter() . . . 741
10.287.2.2 ~StringFilter() . . . 741
10.287.3 Member Function Documentation . . . 741
10.287.3.1 ExecuteQuery(std::string const &query, std::string &value) const . . . 741
10.287.3.2 ExecuteQuery(std::string const &query, DataSet const &ds, std::string &value) const . . . 741
10.287.3.3 FromString(const Tag &t, const char *value, VL const &vl) . . . 741
10.287.3.4 FromString(const Tag &t, const char *value, size_t len) . . . 741
10.287.3.5 GetFile() . . . 741
10.287.3.6 GetFile() const . . . 741
10.287.3.7 SetDicts(const Dicts &dicts) . . . 741
10.287.3.8 SetFile(const File &f) . . . 741
10.287.3.9 ToString(const DataElement &de) const . . . 742
10.287.3.10 ToString(const Tag &t) const . . . 742
10.287.3.11 ToStringPair(const DataElement &de) const . . . 742
10.287.3.12 ToStringPair(const Tag &t) const . . . 742
10.287.3.13 ToStringPair(const Tag &t, DataSet const &ds) const . . . 742

10.287.3.14 UseDictAlways(bool) . . . 742
10.288 gdcm::Study Class Reference . . . 742
10.288.1 Detailed Description . . . 743
10.288.2 Constructor & Destructor Documentation . . . 743
10.288.2.1 Study() . . . 743
10.289 gdcm::Subject Class Reference . . . 743
10.289.1 Detailed Description . . . 744
10.289.2 Constructor & Destructor Documentation . . . 745
10.289.2.1 Subject() . . . 745
10.289.2.2 ~Subject() . . . 745
10.289.3 Member Function Documentation . . . 745
10.289.3.1 AddObserver(const Event &event, Command *) . . . 745
10.289.3.2 AddObserver(const Event &event, Command *) const . . . 745
10.289.3.3 GetCommand(unsigned long tag) . . . 745
10.289.3.4 HasObserver(const Event &event) const . . . 745
10.289.3.5 InvokeEvent(const Event &) . . . 745
10.289.3.6 InvokeEvent(const Event &) const . . . 745
10.289.3.7 RemoveAllObservers() . . . 745
10.289.3.8 RemoveObserver(unsigned long tag) . . . 746
10.290 gdcm::Surface Class Reference . . . 746
10.290.1 Detailed Description . . . 748
10.290.2 Member Enumeration Documentation . . . 749
10.290.2.1 STATES . . . 749
10.290.2.2 VIEWType . . . 749
10.290.3 Constructor & Destructor Documentation . . . 749
10.290.3.1 Surface() . . . 749
10.290.3.2 ~Surface() . . . 749
10.290.4 Member Function Documentation . . . 749

10.290.4.1 GetAlgorithmFamily() const . . . 749
10.290.4.2 GetAlgorithmFamily() . . . 749
10.290.4.3 GetAlgorithmName() const . . . 749
10.290.4.4 GetAlgorithmVersion() const . . . 749
10.290.4.5 GetAxisOfRotation() const . . . 749
10.290.4.6 GetCenterOfRotation() const . . . 750
10.290.4.7 GetFiniteVolume() const . . . 750
10.290.4.8 GetManifold() const . . . 750
10.290.4.9 GetMaximumPointDistance() const . . . 750
10.290.4.10 GetMeanPointDistance() const . . . 750
10.290.4.11 GetMeshPrimitive() const . . . 750
10.290.4.12 GetMeshPrimitive() . . . 750
10.290.4.13 GetNumberOfSurfacePoints() const . . . 750
10.290.4.14 GetNumberOfVectors() const . . . 750
10.290.4.15 GetPointCoordinatesData() const . . . 750
10.290.4.16 GetPointCoordinatesData() . . . 750
10.290.4.17 GetPointPositionAccuracy() const . . . 750
10.290.4.18 GetPointsBoundingBoxCoordinates() const . . . 750
10.290.4.19 GetProcessingAlgorithm() const . . . 751
10.290.4.20 GetProcessingAlgorithm() . . . 751
10.290.4.21 GetRecommendedDisplayCIELabValue() const . . . 751
10.290.4.22 GetRecommendedDisplayCIELabValue(const unsigned int idx) const . . . 751
10.290.4.23 GetRecommendedDisplayGrayscaleValue() const . . . 751
10.290.4.24 GetRecommendedPresentationOpacity() const . . . 751
10.290.4.25 GetRecommendedPresentationType() const . . . 751
10.290.4.26 GetSTATES(const char *state) . . . 751
10.290.4.27 GetSTATESString(STATES state) . . . 751
10.290.4.28 GetSurfaceComments() const . . . 751

10.290.4.29 GetSurfaceNumber() const . . . 751
10.290.4.30 GetSurfaceProcessing() const . . . 751
10.290.4.31 GetSurfaceProcessingDescription() const . . . 751
10.290.4.32 GetSurfaceProcessingRatio() const . . . 751
10.290.4.33 GetVectorAccuracy() const . . . 751
10.290.4.34 GetVectorCoordinateData() const . . . 751
10.290.4.35 GetVectorCoordinateData() . . . 751
10.290.4.36 GetVectorDimensionality() const . . . 751
10.290.4.37 GetVIEWType(const char *type) . . . 751
10.290.4.38 GetVIEWTypeString(VIEWType type) . . . 751
10.290.4.39 SetAlgorithmFamily(SegmentHelper::BasicCodedEntry const &BSE) . . . 751
10.290.4.40 SetAlgorithmName(const char *str) . . . 752
10.290.4.41 SetAlgorithmVersion(const char *str) . . . 752
10.290.4.42 SetAxisOfRotation(const float *axis) . . . 752
10.290.4.43 SetCenterOfRotation(const float *center) . . . 752
10.290.4.44 SetFiniteVolume(STATES state) . . . 752
10.290.4.45 SetManifold(STATES state) . . . 752
10.290.4.46 SetMaximumPointDistance(float maximum) . . . 752
10.290.4.47 SetMeanPointDistance(float average) . . . 752
10.290.4.48 SetMeshPrimitive(MeshPrimitive &mp) . . . 752
10.290.4.49 SetNumberOfSurfacePoints(const unsigned long nb) . . . 752
10.290.4.50 SetNumberOfVectors(const unsigned long nb) . . . 752
10.290.4.51 SetPointCoordinatesData(DataElement const &de) . . . 752
10.290.4.52 SetPointPositionAccuracy(const float *accuracies) . . . 752
10.290.4.53 SetPointsBoundingBoxCoordinates(const float *coordinates) . . . 752
10.290.4.54 SetProcessingAlgorithm(SegmentHelper::BasicCodedEntry const &BSE) . . . 752
10.290.4.55 SetRecommendedDisplayCIELabValue(const unsigned short vl[3]) . . . 752

10.290.4.56 SetRecommendedDisplayCIELabValue(const unsigned short vl, const unsigned int idx=0) . . . 752
10.290.4.57 SetRecommendedDisplayCIELabValue(const std::vector< unsigned short > &vl) . . . 752
10.290.4.58 SetRecommendedDisplayGrayscaleValue(const unsigned short vl) . . . 752
10.290.4.59 SetRecommendedPresentationOpacity(const float opacity) . . . 752
10.290.4.60 SetRecommendedPresentationType(VIEWType type) . . . 752
10.290.4.61 SetSurfaceComments(const char *comment) . . . 753
10.290.4.62 SetSurfaceNumber(const unsigned long nb) . . . 753
10.290.4.63 SetSurfaceProcessing(bool b) . . . 753
10.290.4.64 SetSurfaceProcessingDescription(const char *description) . . . 753
10.290.4.65 SetSurfaceProcessingRatio(const float ratio) . . . 753
10.290.4.66 SetVectorAccuracy(const float *accuracy) . . . 753
10.290.4.67 SetVectorCoordinateData(DataElement const &de) . . . 753
10.290.4.68 SetVectorDimensionality(const unsigned short dim) . . . 753
10.291 gdcm::SurfaceHelper Class Reference . . . 753
10.291.1 Detailed Description . . . 754
10.291.2 Member Typedef Documentation . . . 754
10.291.2.1 ColorArray . . . 754
10.291.3 Member Function Documentation . . . 754
10.291.3.1 RecommendedDisplayCIELabToRGB(const ColorArray &CIELab, const U rangeMax=255) . . . 754
10.291.3.2 RecommendedDisplayCIELabToRGB(const ColorArray &CIELab, const U rangeMax=255) . . . 754
10.291.3.3 RGBToRecommendedDisplayCIELab(const std::vector< T > &RGB, const U rangeMax=255) . . . 755
10.291.3.4 RGBToRecommendedDisplayGrayscale(const std::vector< T > &RGB, const U rangeMax=255) . . . 755
10.292 gdcm::SurfaceReader Class Reference . . . 756
10.292.1 Detailed Description . . . 757
10.292.2 Constructor & Destructor Documentation . . . 758

10.292.2.1 SurfaceReader() . . . 758
10.292.2.2 ~SurfaceReader() . . . 758
10.292.3 Member Function Documentation . . . 758
10.292.3.1 GetNumberOfSurfaces() const . . . 758
10.292.3.2 Read() . . . 758
10.292.3.3 ReadPointMacro(SmartPointer< Surface > surface, const DataSet &surfaceDS) . . . 758
10.292.3.4 ReadSurface(const Item &surfaceItem, const unsigned long idx) . . . 758
10.292.3.5 ReadSurfaces() . . . 758
10.293 gdcm::SurfaceWriter Class Reference . . . 758
10.293.1 Detailed Description . . . 759
10.293.2 Constructor & Destructor Documentation . . . 760
10.293.2.1 SurfaceWriter() . . . 760
10.293.2.2 ~SurfaceWriter() . . . 760
10.293.3 Member Function Documentation . . . 760
10.293.3.1 ComputeNumberOfSurfaces() . . . 760
10.293.3.2 GetNumberOfSurfaces() . . . 760
10.293.3.3 PrepareWrite() . . . 760
10.293.3.4 PrepareWritePointMacro(SmartPointer< Surface > surface, DataSet &surfaceDS, const TransferSyntax &ts) . . . 760
10.293.3.5 SetNumberOfSurfaces(const unsigned long nb) . . . 760
10.293.3.6 Write() . . . 760
10.293.4 Member Data Documentation . . . 760
10.293.4.1 NumberOfSurfaces . . . 760
10.294 gdcm::SwapCode Class Reference . . . 760
10.294.1 Detailed Description . . . 761
10.294.2 Member Enumeration Documentation . . . 761
10.294.2.1 SwapCodeType . . . 761
10.294.3 Constructor & Destructor Documentation . . . 762

10.294.3.1 SwapCode(SwapCodeType sc=Unknown) . . . 762
10.294.4 Member Function Documentation . . . 762
10.294.4.1 GetIndex(SwapCode const &sc) . . . 762
10.294.4.2 GetSwapCodeString(SwapCode const &sc) . . . 762
10.294.4.3 operator SwapCode::SwapCodeType() const . . . 762
10.294.5 Friends And Related Function Documentation . . . 762
10.294.5.1 operator<< . . . 762
10.295 gdcm::SwapperDoOp Class Reference . . . 762
10.295.1 Member Function Documentation . . . 763
10.295.1.1 Swap(T val) . . . 763
10.295.1.2 SwapArray(T *array, size_t n) . . . 763
10.296 gdcm::SwapperNoOp Class Reference . . . 763
10.296.1 Detailed Description . . . 763
10.296.2 Member Function Documentation . . . 763
10.296.2.1 Swap(T val) . . . 763
10.296.2.2 SwapArray(T *, size_t) . . . 763
10.297 gdcm::System Class Reference . . . 763
10.297.1 Detailed Description . . . 765
10.297.2 Member Function Documentation . . . 765
10.297.2.1 DeleteDirectory(const char *source) . . . 765
10.297.2.2 EncodeBytes(char *out, const unsigned char *data, int size) . . . 765
10.297.2.3 FileExists(const char *filename) . . . 765
10.297.2.4 FileIsDirectory(const char *name) . . . 765
10.297.2.5 FileIsSymlink(const char *name) . . . 765
10.297.2.6 FileSize(const char *filename) . . . 766
10.297.2.7 FileTime(const char *filename) . . . 766
10.297.2.8 FormatDateTime(char date[22], time_t t, long milliseconds=0) . . . 766
10.297.2.9 GetCurrentDateTime(char date[22]) . . . 766

10.297.2.10 GetCurrentModuleFileName() . . . 766
10.297.2.11 GetCurrentProcessFileName() . . . 766
10.297.2.12 GetCurrentResourcesDirectory() . . . 766
10.297.2.13 GetCWD() . . . 767
10.297.2.14 GetHostName(char hostname[255]) . . . 767
10.297.2.15 GetLastSystemError() . . . 767
10.297.2.16 GetLocaleCharset() . . . 767
10.297.2.17 GetPermissions(const char *file, unsigned short &mode) . . . 767
10.297.2.18 GetTimezoneOffsetFromUTC() . . . 767
10.297.2.19 MakeDirectory(const char *path) . . . 767
10.297.2.20 ParseDateTime(time_t &timep, const char date[22]) . . . 767
10.297.2.21 ParseDateTime(time_t &timep, long &milliseconds, const char date[22]) . . . 768
10.297.2.22 RemoveFile(const char *source) . . . 768
10.297.2.23 SetPermissions(const char *file, unsigned short mode) . . . 768
10.297.2.24 StrCaseCmp(const char *s1, const char *s2) . . . 768
10.297.2.25 StrNCaseCmp(const char *s1, const char *s2, size_t n) . . . 768
10.297.2.26 StrSep(char **stringp, const char *delim) . . . 768
10.297.2.27 StrTokR(char *ptr, const char *sep, char **end) . . . 768
10.298 gdcm::Table Class Reference . . . 768
10.298.1 Detailed Description . . . 769
10.298.2 Member Typedef Documentation . . . 769
10.298.2.1 MapTableEntry . . . 769
10.298.3 Constructor & Destructor Documentation . . . 769
10.298.3.1 Table() . . . 769
10.298.3.2 ~Table() . . . 769
10.298.4 Member Function Documentation . . . 769
10.298.4.1 GetTableEntry(const Tag &tag) const . . . 769
10.298.4.2 InsertEntry(Tag const &tag, TableEntry const &te) . . . 770

10.298.5 Friends And Related Function Documentation . . . 770
10.298.5.1 operator<< . . . 770
10.299 gdcm::TableEntry Class Reference . . . 770
10.299.1 Detailed Description . . . 770
10.299.2 Constructor & Destructor Documentation . . . 770
10.299.2.1 TableEntry(const char *attribute=0, Type const &type=Type(), const char *des=0) . . . 770
10.299.2.2 ~TableEntry() . . . 770
10.300 gdcm::TableReader Class Reference . . . 771
10.300.1 Detailed Description . . . 771
10.300.2 Constructor & Destructor Documentation . . . 772
10.300.2.1 TableReader(Defs &defs) . . . 772
10.300.2.2 ~TableReader() . . . 772
10.300.3 Member Function Documentation . . . 772
10.300.3.1 CharacterDataHandler(const char *data, int length) . . . 772
10.300.3.2 EndElement(const char *name) . . . 772
10.300.3.3 GetDefs() const . . . 772
10.300.3.4 GetFilename() . . . 772
10.300.3.5 HandleIOD(const char **atts) . . . 772
10.300.3.6 HandleIODEntry(const char **atts) . . . 772
10.300.3.7 HandleMacro(const char **atts) . . . 772
10.300.3.8 HandleMacroEntry(const char **atts) . . . 772
10.300.3.9 HandleMacroEntryDescription(const char **atts) . . . 772
10.300.3.10 HandleModule(const char **atts) . . . 772
10.300.3.11 HandleModuleEntry(const char **atts) . . . 772
10.300.3.12 HandleModuleEntryDescription(const char **atts) . . . 772
10.300.3.13 HandleModuleInclude(const char **atts) . . . 772
10.300.3.14 Read() . . . 772
10.300.3.15 SetFilename(const char *filename) . . . 772

10.300.3.15 StartElement(const char *name, const char **atts) . . . 772
10.301 gdcm::network::TableRow Class Reference . . . 773
10.301.1 Constructor & Destructor Documentation . . . 773
10.301.1.1 TableRow() . . . 773
10.301.1.2 ~TableRow() . . . 773
10.301.2 Member Data Documentation . . . 774
10.301.2.1 transitions . . . 774
10.302 gdcm::Tag Class Reference . . . 774
10.302.1 Detailed Description . . . 776
10.302.2 Constructor & Destructor Documentation . . . 776
10.302.2.1 Tag(uint16_t group, uint16_t element) . . . 776
10.302.2.2 Tag(uint32_t tag=0) . . . 776
10.302.2.3 Tag(const Tag &_val) . . . 776
10.302.3 Member Function Documentation . . . 776
10.302.3.1 GetElement() const . . . 776
10.302.3.2 GetElementTag() const . . . 777
10.302.3.3 GetGroup() const . . . 777
10.302.3.4 GetLength() const . . . 777
10.302.3.5 GetPrivateCreator() const . . . 777
10.302.3.6 IsGroupLength() const . . . 777
10.302.3.7 IsGroupXX(const Tag &t) const . . . 777
10.302.3.8 IsIllegal() const . . . 777
10.302.3.9 IsPrivate() const . . . 778
10.302.3.10 IsPrivateCreator() const . . . 778
10.302.3.11 IsPublic() const . . . 778
10.302.3.12 operator"!=(const Tag &_val) const . . . 778
10.302.3.13 operator<(const Tag &_val) const . . . 778
10.302.3.14 operator<=(const Tag &t2) const . . . 778

10.302.3.15 operator=(const Tag &_val) . . . 778
10.302.3.16 operator==(const Tag &_val) const . . . 779
10.302.3.17 operator[](const unsigned int &_id) const . . . 779
10.302.3.18 operator[](const unsigned int &_id) . . . 779
10.302.3.19 PrintAsContinuousString() const . . . 779
10.302.3.20 PrintAsContinuousUpperCaseString() const . . . 779
10.302.3.21 PrintAsPipeSeparatedString() const . . . 779
10.302.3.22 Read(std::istream &is) . . . 779
10.302.3.23 ReadFromCommaSeparatedString(const char *str) . . . 779
10.302.3.24 ReadFromContinuousString(const char *str) . . . 780
10.302.3.25 ReadFromPipeSeparatedString(const char *str) . . . 780
10.302.3.26 SetElement(uint16_t element) . . . 780
10.302.3.27 SetElementTag(uint16_t group, uint16_t element) . . . 780
10.302.3.28 SetElementTag(uint32_t tag) . . . 780
10.302.3.29 SetGroup(uint16_t group) . . . 780
10.302.3.30 SetPrivateCreator(Tag const &t) . . . 781
10.302.3.31 Write(std::ostream &os) const . . . 781
10.302.4 Friends And Related Function Documentation . . . 781
10.302.4.1 operator<< . . . 781
10.302.4.2 operator>> . . . 781
10.302.5 Member Data Documentation . . . 781
10.302.5.1 bytes . . . 781
10.302.5.2 tag . . . 781
10.302.5.3 tags . . . 781
10.303 gdcm::TagPath Class Reference . . . 781
10.303.1 Detailed Description . . . 782
10.303.2 Constructor & Destructor Documentation . . . 782
10.303.2.1 TagPath() . . . 782

10.303.2.2 ~TagPath() . . . 782
10.303.3 Member Function Documentation . . . 782
10.303.3.1 ConstructFromString(const char *path) . . . 782
10.303.3.2 ConstructFromTagList(Tag const *l, unsigned int n) . . . 782
10.303.3.3 IsValid(const char *path) . . . 782
10.303.3.4 Print(std::ostream &) const . . . 783
10.303.3.5 Push(Tag const &t) . . . 783
10.303.3.6 Push(unsigned int itemnum) . . . 783
10.304 gdcm::Testing Class Reference . . . 783
10.304.1 Detailed Description . . . 784
10.304.2 Member Typedef Documentation . . . 784
10.304.2.1 MD5DataImagesType . . . 784
10.304.2.2 MediaStorageDataFilesType . . . 784
10.304.3 Constructor & Destructor Documentation . . . 785
10.304.3.1 Testing() . . . 785
10.304.3.2 ~Testing() . . . 785
10.304.4 Member Function Documentation . . . 785
10.304.4.1 ComputeFileMD5(const char *filename, char digest_str[33]) . . . 785
10.304.4.2 ComputeMD5(const char *buffer, unsigned long buf_len, char digest_str[33]) . . . 785
10.304.4.3 GetDataExtraRoot() . . . 785
10.304.4.4 GetDataRoot() . . . 785
10.304.4.5 GetFileName(unsigned int file) . . . 785
10.304.4.6 GetFileNames() . . . 786
10.304.4.7 GetLossyFlagFromFile(const char *filepath) . . . 786
10.304.4.8 GetMD5DataImage(unsigned int file) . . . 786
10.304.4.9 GetMD5DataImages() . . . 786
10.304.4.10 GetMD5FromBrokenFile(const char *filepath) . . . 786
10.304.4.11 GetMD5FromFile(const char *filepath) . . . 786

10.304.4.12 GetMediaStorageDataFile(unsigned int file) . . . 786
10.304.4.13 GetMediaStorageDataFiles() . . . 786
10.304.4.14 GetMediaStorageFromFile(const char *filepath) . . . 786
10.304.4.15 GetNumberOfFileNames() . . . 786
10.304.4.16 GetNumberOfMD5DataImages() . . . 787
10.304.4.17 GetNumberOfMediaStorageDataFiles() . . . 787
10.304.4.18 GetPixelSpacingDataRoot() . . . 787
10.304.4.19 GetSelectedPrivateGroupOffsetFromFile(const char *filepath) . . . 787
10.304.4.20 GetSelectedTagsOffsetFromFile(const char *filepath) . . . 787
10.304.4.21 GetSourceDirectory() . . . 787
10.304.4.22 GetStreamOffsetFromFile(const char *filepath) . . . 787
10.304.4.23 GetTempDirectory(const char *subdir=0) . . . 787
10.304.4.24 GetTempDirectoryW(const wchar_t *subdir=0) . . . 787
10.304.4.25 GetTempFilename(const char *filename, const char *subdir=0) . . . 788
10.304.4.26 GetTempFilenameW(const wchar_t *filename, const wchar_t *subdir=0) . . . 788
10.304.4.27 Print(std::ostream &os=std::cout) . . . 788
10.305 gdcm::Trace Class Reference . . . 788
10.305.1 Detailed Description . . . 789
10.305.2 Constructor & Destructor Documentation . . . 790
10.305.2.1 Trace() . . . 790
10.305.2.2 ~Trace() . . . 790
10.305.3 Member Function Documentation . . . 790
10.305.3.1 DebugOff() . . . 790
10.305.3.2 DebugOn() . . . 790
10.305.3.3 ErrorOff() . . . 790
10.305.3.4 ErrorOn() . . . 790
10.305.3.5 GetDebugFlag() . . . 790
10.305.3.6 GetDebugStream() . . . 790

10.305.3.7 GetErrorFlag() . . . 790
10.305.3.8 GetErrorStream() . . . 790
10.305.3.9 GetStream() . . . 790
10.305.3.10 GetWarningFlag() . . . 790
10.305.3.11 GetWarningStream() . . . 790
10.305.3.12 SetDebug(bool debug) . . . 790
10.305.3.13 SetDebugStream(std::ostream &os) . . . 791
10.305.3.14 SetError(bool debug) . . . 791
10.305.3.15 SetErrorStream(std::ostream &os) . . . 791
10.305.3.16 SetStream(std::ostream &os) . . . 791
10.305.3.17 SetStreamToFile(const char *filename) . . . 791
10.305.3.18 SetWarning(bool debug) . . . 791
10.305.3.19 SetWarningStream(std::ostream &os) . . . 791
10.305.3.20 WarningOff() . . . 792
10.305.3.21 WarningOn() . . . 792
10.306 gdcm::TransferSyntax Class Reference . . . 792
10.306.1 Detailed Description . . . 794
10.306.2 Member Enumeration Documentation . . . 794
10.306.2.1 NegociatedType . . . 794
10.306.2.2 TSType . . . 795
10.306.3 Constructor & Destructor Documentation . . . 795
10.306.3.1 TransferSyntax(TSType type=ImplicitVRLittleEndian) . . . 795
10.306.4 Member Function Documentation . . . 795
10.306.4.1 CanStoreLossy() const . . . 795
10.306.4.2 GetNegociatedType() const . . . 796
10.306.4.3 GetString() const . . . 796
10.306.4.4 GetSwapCode() const . . . 796
10.306.4.5 GetTSString(TSType ts) . . . 796

10.306.4.6 GetTSType(const char *str) . . . 796
10.306.4.7 IsEncapsulated() const . . . 796
10.306.4.8 IsEncoded() const . . . 796
10.306.4.9 IsExplicit() const . . . 796
10.306.4.10 IsImplicit() const . . . 796
10.306.4.11 IsLossless() const . . . 796
10.306.4.12 IsLossy() const . . . 796
10.306.4.13 IsValid() const . . . 797
10.306.4.14 operator TSType() const . . . 797
10.306.5 Friends And Related Function Documentation . . . 797
10.306.5.1 operator<< . . . 797
10.307 gdcm::network::TransferSyntaxSub Class Reference . . . 797
10.307.1 Detailed Description . . . 797
10.307.2 Constructor & Destructor Documentation . . . 798
10.307.2.1 TransferSyntaxSub() . . . 798
10.307.3 Member Function Documentation . . . 798
10.307.3.1 GetName() const . . . 798
10.307.3.2 operator==(const TransferSyntaxSub &ts) const . . . 798
10.307.3.3 Print(std::ostream &os) const . . . 798
10.307.3.4 Read(std::istream &is) . . . 798
10.307.3.5 SetName(const char *name) . . . 798
10.307.3.6 SetNameFromUID(UIDs::TSName tsname) . . . 798
10.307.3.7 Size() const . . . 798
10.307.3.8 Write(std::ostream &os) const . . . 798
10.308 gdcm::network::Transition Struct Reference . . . 799
10.308.1 Constructor & Destructor Documentation . . . 799
10.308.1.1 Transition() . . . 799
10.308.1.2 ~Transition() . . . 800

10.308.1.3 Transition(int inEndState, ULAction *inAction) . . . 800
10.308.2 Member Function Documentation . . . 800
10.308.2.1 MakeNew(int inEndState, ULAction *inAction) . . . 800
10.308.3 Member Data Documentation . . . 800
10.308.3.1 mAction . . . 800
10.308.3.2 mEnd . . . 800
10.309 gdcm::Type Class Reference . . . 800
10.309.1 Detailed Description . . . 801
10.309.2 Member Enumeration Documentation . . . 801
10.309.2.1 TypeType . . . 801
10.309.3 Constructor & Destructor Documentation . . . 802
10.309.3.1 Type(TypeType type=UNKNOWN) . . . 802
10.309.4 Member Function Documentation . . . 802
10.309.4.1 GetTypeString(TypeType type) . . . 802
10.309.4.2 GetTypeType(const char *type) . . . 802
10.309.4.3 operator TypeType() const . . . 802
10.309.5 Friends And Related Function Documentation . . . 802
10.309.5.1 operator<< . . . 802
10.310 gdcm::UI Struct Reference . . . 802
10.310.1 Friends And Related Function Documentation . . . 803
10.310.1.1 operator<< . . . 803
10.310.2 Member Data Documentation . . . 803
10.310.2.1 Internal . . . 803
10.311 gdcm::UIDGenerator Class Reference . . . 803
10.311.1 Detailed Description . . . 804
10.311.2 Constructor & Destructor Documentation . . . 804
10.311.2.1 UIDGenerator() . . . 804
10.311.3 Member Function Documentation . . . 804

10.311.3.1 Generate() . . . 804
10.311.3.2 GenerateUUID(unsigned char *uuid_data) . . . 804
10.311.3.3 GetGDCMUID() . . . 804
10.311.3.4 GetRoot() . . . 804
10.311.3.5 IsValid(const char *uid) . . . 805
10.311.3.6 SetRoot(const char *root) . . . 805
10.312 gdcm::UIDs Class Reference . . . 805
10.312.1 Detailed Description . . . 810
10.312.2 Member Typedef Documentation . . . 810
10.312.2.1 TransferSyntaxStringsType . . . 810
10.312.3 Member Enumeration Documentation . . . 810
10.312.3.1 TSName . . . 810
10.312.3.2 TSType . . . 817
10.312.4 Member Function Documentation . . . 824
10.312.4.1 GetName() const . . . 824
10.312.4.2 GetNumberOfTransferSyntaxStrings() . . . 824
10.312.4.3 GetString() const . . . 824
10.312.4.4 GetTransferSyntaxString(unsigned int ts) . . . 824
10.312.4.5 GetTransferSyntaxStrings() . . . 824
10.312.4.6 GetUIDName(unsigned int ts) . . . 824
10.312.4.7 GetUIDString(unsigned int ts) . . . 824
10.312.4.8 operator TSType() const . . . 824
10.312.4.9 SetFromUID(const char *str) . . . 824
10.313 gdcm::network::ULAction Class Reference . . . 825
10.313.1 Detailed Description . . . 827
10.313.2 Constructor & Destructor Documentation . . . 827
10.313.2.1 ULAction() . . . 827
10.313.2.2 ~ULAction() . . . 827

10.313.3 Member Function Documentation . . . 828
10.313.3.1 PerformAction(Subject *s, ULEvent &inEvent, ULConnection &inConnection, bool &outWaitingForEvent, EEventID &outRaisedEvent)=0 . . . 828
10.314 dcm::network::ULActionAA1 Class Reference . . . 828
10.314.1 Member Function Documentation . . . 829
10.314.1.1 PerformAction(Subject *s, ULEvent &inEvent, ULConnection &inConnection, bool &outWaitingForEvent, EEventID &outRaisedEvent) . . . 829
10.315 dcm::network::ULActionAA2 Class Reference . . . 829
10.315.1 Member Function Documentation . . . 830
10.315.1.1 PerformAction(Subject *s, ULEvent &inEvent, ULConnection &inConnection, bool &outWaitingForEvent, EEventID &outRaisedEvent) . . . 830
10.316 dcm::network::ULActionAA3 Class Reference . . . 830
10.316.1 Member Function Documentation . . . 831
10.316.1.1 PerformAction(Subject *s, ULEvent &inEvent, ULConnection &inConnection, bool &outWaitingForEvent, EEventID &outRaisedEvent) . . . 831
10.317 dcm::network::ULActionAA4 Class Reference . . . 831
10.317.1 Member Function Documentation . . . 832
10.317.1.1 PerformAction(Subject *s, ULEvent &inEvent, ULConnection &inConnection, bool &outWaitingForEvent, EEventID &outRaisedEvent) . . . 832
10.318 dcm::network::ULActionAA5 Class Reference . . . 832
10.318.1 Member Function Documentation . . . 833
10.318.1.1 PerformAction(Subject *s, ULEvent &inEvent, ULConnection &inConnection, bool &outWaitingForEvent, EEventID &outRaisedEvent) . . . 833
10.319 dcm::network::ULActionAA6 Class Reference . . . 833
10.319.1 Member Function Documentation . . . 834
10.319.1.1 PerformAction(Subject *s, ULEvent &inEvent, ULConnection &inConnection, bool &outWaitingForEvent, EEventID &outRaisedEvent) . . . 834
10.320 dcm::network::ULActionAA7 Class Reference . . . 834
10.320.1 Member Function Documentation . . . 835
10.320.1.1 PerformAction(Subject *s, ULEvent &inEvent, ULConnection &inConnection, bool &outWaitingForEvent, EEventID &outRaisedEvent) . . . 835
10.321 dcm::network::ULActionAA8 Class Reference . . . 835

10.321.1 Member Function Documentation . . . 836
10.321.1.1 PerformAction(Subject *s, ULEvent &inEvent, ULConnection &inConnection, bool &outWaitingForEvent, EEventID &outRaisedEvent) . . . 836
10.322 gdcm::network::ULActionAE1 Class Reference . . . 836
10.322.1 Member Function Documentation . . . 837
10.322.1.1 PerformAction(Subject *s, ULEvent &inEvent, ULConnection &inConnection, bool &outWaitingForEvent, EEventID &outRaisedEvent) . . . 837
10.323 gdcm::network::ULActionAE2 Class Reference . . . 837
10.323.1 Member Function Documentation . . . 838
10.323.1.1 PerformAction(Subject *s, ULEvent &inEvent, ULConnection &inConnection, bool &outWaitingForEvent, EEventID &outRaisedEvent) . . . 838
10.324 gdcm::network::ULActionAE3 Class Reference . . . 838
10.324.1 Member Function Documentation . . . 839
10.324.1.1 PerformAction(Subject *s, ULEvent &inEvent, ULConnection &inConnection, bool &outWaitingForEvent, EEventID &outRaisedEvent) . . . 839
10.325 gdcm::network::ULActionAE4 Class Reference . . . 839
10.325.1 Member Function Documentation . . . 840
10.325.1.1 PerformAction(Subject *s, ULEvent &inEvent, ULConnection &inConnection, bool &outWaitingForEvent, EEventID &outRaisedEvent) . . . 840
10.326 gdcm::network::ULActionAE5 Class Reference . . . 840
10.326.1 Member Function Documentation . . . 841
10.326.1.1 PerformAction(Subject *s, ULEvent &inEvent, ULConnection &inConnection, bool &outWaitingForEvent, EEventID &outRaisedEvent) . . . 841
10.327 gdcm::network::ULActionAE6 Class Reference . . . 841
10.327.1 Member Function Documentation . . . 842
10.327.1.1 PerformAction(Subject *s, ULEvent &inEvent, ULConnection &inConnection, bool &outWaitingForEvent, EEventID &outRaisedEvent) . . . 842
10.328 gdcm::network::ULActionAE7 Class Reference . . . 842
10.328.1 Member Function Documentation . . . 843
10.328.1.1 PerformAction(Subject *s, ULEvent &inEvent, ULConnection &inConnection, bool &outWaitingForEvent, EEventID &outRaisedEvent) . . . 843
10.329 gdcm::network::ULActionAE8 Class Reference . . . 843

10.329.1 Member Function Documentation . . . 844
10.329.1.1 PerformAction(Subject *s, ULEvent &inEvent, ULConnection &inConnection, bool &outWaitingForEvent, EEventID &outRaisedEvent) . . . 844
10.330 gdcm::network::ULActionAR1 Class Reference . . . 844
10.330.1 Member Function Documentation . . . 845
10.330.1.1 PerformAction(Subject *s, ULEvent &inEvent, ULConnection &inConnection, bool &outWaitingForEvent, EEventID &outRaisedEvent) . . . 845
10.331 gdcm::network::ULActionAR10 Class Reference . . . 845
10.331.1 Member Function Documentation . . . 846
10.331.1.1 PerformAction(Subject *s, ULEvent &inEvent, ULConnection &inConnection, bool &outWaitingForEvent, EEventID &outRaisedEvent) . . . 846
10.332 gdcm::network::ULActionAR2 Class Reference . . . 846
10.332.1 Member Function Documentation . . . 847
10.332.1.1 PerformAction(Subject *s, ULEvent &inEvent, ULConnection &inConnection, bool &outWaitingForEvent, EEventID &outRaisedEvent) . . . 847
10.333 gdcm::network::ULActionAR3 Class Reference . . . 847
10.333.1 Member Function Documentation . . . 848
10.333.1.1 PerformAction(Subject *s, ULEvent &inEvent, ULConnection &inConnection, bool &outWaitingForEvent, EEventID &outRaisedEvent) . . . 848
10.334 gdcm::network::ULActionAR4 Class Reference . . . 848
10.334.1 Member Function Documentation . . . 849
10.334.1.1 PerformAction(Subject *s, ULEvent &inEvent, ULConnection &inConnection, bool &outWaitingForEvent, EEventID &outRaisedEvent) . . . 849
10.335 gdcm::network::ULActionAR5 Class Reference . . . 849
10.335.1 Member Function Documentation . . . 850
10.335.1.1 PerformAction(Subject *s, ULEvent &inEvent, ULConnection &inConnection, bool &outWaitingForEvent, EEventID &outRaisedEvent) . . . 850
10.336 gdcm::network::ULActionAR6 Class Reference . . . 850
10.336.1 Member Function Documentation . . . 851
10.336.1.1 PerformAction(Subject *s, ULEvent &inEvent, ULConnection &inConnection, bool &outWaitingForEvent, EEventID &outRaisedEvent) . . . 851
10.337 gdcm::network::ULActionAR7 Class Reference . . . 851

10.337.1 Member Function Documentation . . . 852
10.337.1.1 PerformAction(Subject *s, ULEvent &inEvent, ULConnection &inConnection, bool &outWaitingForEvent, EEventID &outRaisedEvent) . . . 852
10.338 gdcm::network::ULActionAR8 Class Reference . . . 852
10.338.1 Member Function Documentation . . . 853
10.338.1.1 PerformAction(Subject *s, ULEvent &inEvent, ULConnection &inConnection, bool &outWaitingForEvent, EEventID &outRaisedEvent) . . . 853
10.339 gdcm::network::ULActionAR9 Class Reference . . . 853
10.339.1 Member Function Documentation . . . 854
10.339.1.1 PerformAction(Subject *s, ULEvent &inEvent, ULConnection &inConnection, bool &outWaitingForEvent, EEventID &outRaisedEvent) . . . 854
10.340 gdcm::network::ULActionDT1 Class Reference . . . 854
10.340.1 Member Function Documentation . . . 855
10.340.1.1 PerformAction(Subject *s, ULEvent &inEvent, ULConnection &inConnection, bool &outWaitingForEvent, EEventID &outRaisedEvent) . . . 855
10.341 gdcm::network::ULActionDT2 Class Reference . . . 855
10.341.1 Member Function Documentation . . . 856
10.341.1.1 PerformAction(Subject *s, ULEvent &inEvent, ULConnection &inConnection, bool &outWaitingForEvent, EEventID &outRaisedEvent) . . . 856
10.342 gdcm::network::ULBasicCallback Class Reference . . . 856
10.342.1 Detailed Description . . . 857
10.342.2 Constructor & Destructor Documentation . . . 858
10.342.2.1 ULBasicCallback() . . . 858
10.342.2.2 ~ULBasicCallback() . . . 858
10.342.3 Member Function Documentation . . . 858
10.342.3.1 GetDataSets() const . . . 858
10.342.3.2 GetResponses() const . . . 858
10.342.3.3 HandleDataSet(const DataSet &inDataSet) . . . 858
10.342.3.4 HandleResponse(const DataSet &inDataSet) . . . 858
10.343 gdcm::network::ULConnection Class Reference . . . 858

10.343.1 Detailed Description . . . 859
10.343.2 Constructor & Destructor Documentation . . . 860
10.343.2.1 ULConnection(const ULConnectionInfo &inUserInformation) . . . 860
10.343.2.2 ~ULConnection() . . . 860
10.343.3 Member Function Documentation . . . 860
10.343.3.1 AddAcceptedPresentationContext(const PresentationContextAC &inPC) . . . 860
10.343.3.2 FindContext(const DataElement &de) const . . . 860
10.343.3.3 GetAcceptedPresentationContexts() const . . . 860
10.343.3.4 GetAcceptedPresentationContexts() . . . 860
10.343.3.5 GetConnectionInfo() const . . . 860
10.343.3.6 GetMaxPDUSize() const . . . 860
10.343.3.7 GetPresentationContextACByID(uint8_t id) const . . . 860
10.343.3.8 GetPresentationContextIDFromPresentationContext(PresentationContextRQ const &pc) const . . . 860
10.343.3.9 GetPresentationContextRQByID(uint8_t id) const . . . 860
10.343.3.10 GetPresentationContexts() const . . . 860
10.343.3.11 GetProtocol() . . . 860
10.343.3.12 GetState() const . . . 860
10.343.3.13 GetTimer() . . . 860
10.343.3.14 InitializeConnection() . . . 860
10.343.3.15 InitializeIncomingConnection() . . . 861
10.343.3.16 SetMaxPDUSize(uint32_t inSize) . . . 861
10.343.3.17 SetPresentationContexts(const std::vector< PresentationContextRQ > &inContexts) . . . 861
10.343.3.18 SetPresentationContexts(const std::vector< PresentationContext > &inContexts) . . . 861
10.343.3.19 SetState(const EStateID &inState) . . . 861
10.343.3.20 StopProtocol() . . . 861
10.343.4 Friends And Related Function Documentation . . . 861
10.343.4.1 ULActionAE6 . . . 861

10.343.4.2 ULConnectionManager . . . 861
10.344 dcm::network::ULConnectionCallback Class Reference . . . 861
10.344.1 Detailed Description . . . 862
10.344.2 Constructor & Destructor Documentation . . . 862
10.344.2.1 ULConnectionCallback() . . . 862
10.344.2.2 ~ULConnectionCallback() . . . 862
10.344.3 Member Function Documentation . . . 862
10.344.3.1 DataSetHandled() . . . 862
10.344.3.2 DataSetHandles() const . . . 862
10.344.3.3 HandleDataSet(const DataSet &inDataSet)=0 . . . 862
10.344.3.4 HandleResponse(const DataSet &inDataSet)=0 . . . 863
10.344.3.5 ResetHandledDataSet() . . . 863
10.344.3.6 SetImplicitFlag(const bool imp) . . . 863
10.344.4 Member Data Documentation . . . 863
10.344.4.1 mImplicit . . . 863
10.345 dcm::network::ULConnectionInfo Class Reference . . . 863
10.345.1 Detailed Description . . . 863
10.345.2 Constructor & Destructor Documentation . . . 864
10.345.2.1 ULConnectionInfo() . . . 864
10.345.3 Member Function Documentation . . . 864
10.345.3.1 GetCalledAETitle() const . . . 864
10.345.3.2 GetCalledComputerName() const . . . 864
10.345.3.3 GetCalledIPAddress() const . . . 864
10.345.3.4 GetCalledIPPort() const . . . 864
10.345.3.5 GetCallingAETitle() const . . . 864
10.345.3.6 GetMaxPDULength() const . . . 864
10.345.3.7 Initialize(UserInformation const &inUserInformation, const char *inCalledAETitle, const char *inCallingAETitle, unsigned long inCalledIPAddress, int inCalledIPPort, std::string inCalledComputerName) . . . 864

10.345.3.8 SetMaxPDULength(unsigned long inMaxPDULength) . . . 864
10.346 dcm::network::ULConnectionManager Class Reference . . . 864
10.346.1 Detailed Description . . . 866
10.346.2 Constructor & Destructor Documentation . . . 867
10.346.2.1 ULConnectionManager(const ULConnectionManager &inCM) . . . 867
10.346.2.2 ULConnectionManager() . . . 867
10.346.2.3 ~ULConnectionManager() . . . 867
10.346.3 Member Function Documentation . . . 867
10.346.3.1 BreakConnection(const double &inTimeout) . . . 867
10.346.3.2 BreakConnectionNow() . . . 867
10.346.3.3 EstablishConnection(const std::string &inAETitle, const std::string &inConnectAE↩ Title, const std::string &inComputerName, long inIPAddress, uint16_t inConnectPort, double inTimeout, std::vector< PresentationContext > const &pcVector) . . . 867
10.346.3.4 EstablishConnectionMove(const std::string &inAETitle, const std::string &inConnect↩ AETitle, const std::string &inComputerName, long inIPAddress, uint16_t inConnect↩ Port, double inTimeout, uint16_t inReturnPort, std::vector< PresentationContext > const &pcVector) . . . 867
10.346.3.5 RunEventLoop(ULEvent &inEvent, ULConnection *inWhichConnection, UL↩ ConnectionCallback *inCallback, const bool &startWaiting) . . . 867
10.346.3.6 RunMoveEventLoop(ULEvent &inEvent, ULConnectionCallback *inCallback) . . . 867
10.346.3.7 SendEcho() . . . 867
10.346.3.8 SendFind(const BaseRootQuery *inRootQuery) . . . 867
10.346.3.9 SendFind(const BaseRootQuery *inRootQuery, ULConnectionCallback *inCallback) . 867
10.346.3.10 SendMove(const BaseRootQuery *inRootQuery) . . . 867
10.346.3.11 SendMove(const BaseRootQuery *inRootQuery, ULConnectionCallback *inCallback) 867
10.346.3.12 SendNAction(const BaseQuery *inQuery) . . . 868
10.346.3.13 SendNAction(const BaseQuery *inQuery, ULConnectionCallback *inCallback) . . . 868
10.346.3.14 SendNCreate(const BaseQuery *inQuery) . . . 868
10.346.3.15 SendNCreate(const BaseQuery *inQuery, ULConnectionCallback *inCallback) . . . 868
10.346.3.16 SendNDelete(const BaseQuery *inQuery) . . . 868
10.346.3.17 SendNDelete(const BaseQuery *inQuery, ULConnectionCallback *inCallback) . . . 868

10.346.3.18 SendNEventReport(const BaseQuery *inQuery) . . . 868
10.346.3.19 SendNEventReport(const BaseQuery *inQuery, ULConnectionCallback *inCallback) . 868
10.346.3.20 SendNGet(const BaseQuery *inQuery) . . . 868
10.346.3.21 SendNGet(const BaseQuery *inQuery, ULConnectionCallback *inCallback) . . . 868
10.346.3.22 SendNSet(const BaseQuery *inQuery) . . . 868
10.346.3.23 SendNSet(const BaseQuery *inQuery, ULConnectionCallback *inCallback) . . . 868
10.346.3.24 SendStore(const File &file, std::istream *pStream=NULL, std::streampos dataSetOffset=0) . . . 868
10.346.3.25 SendStore(const File &file, ULConnectionCallback *inCallback, std::istream *pStream=NULL, std::streampos dataSetOffset=0) . . . 868
10.346.4 Member Data Documentation . . . 868
10.346.4.1 mConnection . . . 868
10.346.4.2 mSecondaryConnection . . . 868
10.346.4.3 mTransitions . . . 868
10.347 dcm::network::ULEvent Class Reference . . . 869
10.347.1 Detailed Description . . . 869
10.347.2 Constructor & Destructor Documentation . . . 869
10.347.2.1 ULEvent(const EEventID &inEventID, std::vector< BasePDU * > inBasePDU, std::istream *iStream=NULL, std::streampos posDataSet=0) . . . 869
10.347.2.2 ULEvent(const EEventID &inEventID, BasePDU *inBasePDU, std::istream *iStream=NULL, std::streampos posDataSet=0) . . . 869
10.347.2.3 ~ULEvent() . . . 869
10.347.3 Member Function Documentation . . . 869
10.347.3.1 GetDataSetPos() const . . . 869
10.347.3.2 GetEvent() const . . . 869
10.347.3.3 GetIStream() const . . . 869
10.347.3.4 GetPDUs() const . . . 869
10.347.3.5 SetEvent(const EEventID &inEvent) . . . 869
10.347.3.6 SetPDU(std::vector< BasePDU * > const &inPDU) . . . 869
10.348 dcm::network::ULTransitionTable Class Reference . . . 870

10.348.1 Detailed Description . . . 870
10.348.2 Constructor & Destructor Documentation . . . 870
10.348.2.1 ULTransitionTable() . . . 870
10.348.3 Member Function Documentation . . . 870
10.348.3.1 HandleEvent(Subject *s, ULEvent &inEvent, ULConnection &inConnection, bool &outWaitingForEvent, EEventID &outRaisedEvent) const . . . 870
10.348.3.2 PrintTable() const . . . 870
10.349 gdcm::network::ULWritingCallback Class Reference . . . 871
10.349.1 Constructor & Destructor Documentation . . . 872
10.349.1.1 ULWritingCallback() . . . 872
10.349.1.2 ~ULWritingCallback() . . . 872
10.349.2 Member Function Documentation . . . 872
10.349.2.1 HandleDataSet(const DataSet &inDataSet) . . . 872
10.349.2.2 HandleResponse(const DataSet &inDataSet) . . . 872
10.349.2.3 SetDirectory(const std::string &inDirectoryName) . . . 872
10.350 gdcm::UNExplicitDataElement Class Reference . . . 872
10.350.1 Detailed Description . . . 873
10.350.2 Member Function Documentation . . . 874
10.350.2.1 GetLength() const . . . 874
10.350.2.2 Read(std::istream &is) . . . 874
10.350.2.3 ReadPreValue(std::istream &is) . . . 874
10.350.2.4 ReadValue(std::istream &is, bool readvalues=true) . . . 874
10.350.2.5 ReadWithLength(std::istream &is, VL &length) . . . 874
10.351 gdcm::UNExplicitImplicitDataElement Class Reference . . . 874
10.351.1 Detailed Description . . . 875
10.351.2 Member Function Documentation . . . 876
10.351.2.1 GetLength() const . . . 876
10.351.2.2 Read(std::istream &is) . . . 876

10.351.2.3 ReadPreValue(std::istream &is) . . . 876
10.351.2.4 ReadValue(std::istream &is) . . . 876
10.352 gdcm::Unpacker12Bits Class Reference . . . 876
10.352.1 Detailed Description . . . 876
10.352.2 Member Function Documentation . . . 877
10.352.2.1 Pack(char *out, const char *in, size_t n) . . . 877
10.352.2.2 Unpack(char *out, const char *in, size_t n) . . . 877
10.353 gdcm::Usage Class Reference . . . 877
10.353.1 Detailed Description . . . 878
10.353.2 Member Enumeration Documentation . . . 878
10.353.2.1 UsageType . . . 878
10.353.3 Constructor & Destructor Documentation . . . 878
10.353.3.1 Usage(UsageType type=Invalid) . . . 878
10.353.4 Member Function Documentation . . . 878
10.353.4.1 GetUsageString(UsageType type) . . . 878
10.353.4.2 GetUsageType(const char *type) . . . 879
10.353.4.3 operator UsageType() const . . . 879
10.353.5 Friends And Related Function Documentation . . . 879
10.353.5.1 operator<< . . . 879
10.354 gdcm::UserEvent Class Reference . . . 879
10.355 gdcm::network::UserInformation Class Reference . . . 880
10.355.1 Detailed Description . . . 881
10.355.2 Constructor & Destructor Documentation . . . 881
10.355.2.1 UserInformation() . . . 881
10.355.2.2 ~UserInformation() . . . 881
10.355.3 Member Function Documentation . . . 881
10.355.3.1 AddRoleSelectionSub(RoleSelectionSub const &r) . . . 881
10.355.3.2 AddSOPClassExtendedNegociationSub(SOPClassExtendedNegociationSub const &s) 881

10.355.3.3 GetMaximumLengthSub() const . . . 881
10.355.3.4 GetMaximumLengthSub() . . . 881
10.355.3.5 operator=(const UserInformation &) . . . 881
10.355.3.6 Print(std::ostream &os) const . . . 882
10.355.3.7 Read(std::istream &is) . . . 882
10.355.3.8 Size() const . . . 882
10.355.3.9 Write(std::ostream &os) const . . . 882
10.356 gdcm::UUIDGenerator Class Reference . . . 882
10.356.1 Detailed Description . . . 882
10.356.2 Member Function Documentation . . . 882
10.356.2.1 Generate() . . . 882
10.356.2.2 IsValid(const char *uid) . . . 883
10.357 gdcm::Validate Class Reference . . . 883
10.357.1 Detailed Description . . . 884
10.357.2 Constructor & Destructor Documentation . . . 884
10.357.2.1 Validate() . . . 884
10.357.2.2 ~Validate() . . . 884
10.357.3 Member Function Documentation . . . 884
10.357.3.1 GetValidatedFile() . . . 884
10.357.3.2 SetFile(File const &f) . . . 884
10.357.3.3 Validation() . . . 884
10.357.4 Member Data Documentation . . . 884
10.357.4.1 F . . . 884
10.357.4.2 V . . . 884
10.358 gdcm::Value Class Reference . . . 884
10.358.1 Detailed Description . . . 885
10.358.2 Constructor & Destructor Documentation . . . 885
10.358.2.1 Value() . . . 885

10.358.2.2 ~Value() . . . 885
10.358.3 Member Function Documentation . . . 886
10.358.3.1 Clear()=0 . . . 886
10.358.3.2 GetLength() const =0 . . . 886
10.358.3.3 operator==(const Value &val) const =0 . . . 886
10.358.3.4 SetLength(VL l)=0 . . . 886
10.358.3.5 SetLengthOnly(VL l) . . . 886
10.358.4 Friends And Related Function Documentation . . . 886
10.358.4.1 DataElement . . . 886
10.359 gdcm::ValueIO< TDE, TSwap, TType > Class Template Reference . . . 886
10.359.1 Detailed Description . . . 887
10.359.2 Member Function Documentation . . . 887
10.359.2.1 Read(std::istream &is, Value &v, bool readvalues) . . . 887
10.359.2.2 Write(std::ostream &os, const Value &v) . . . 887
10.360 gdcm::Version Class Reference . . . 887
10.360.1 Detailed Description . . . 888
10.360.2 Constructor & Destructor Documentation . . . 888
10.360.2.1 Version() . . . 888
10.360.2.2 ~Version() . . . 888
10.360.3 Member Function Documentation . . . 888
10.360.3.1 GetBuildVersion() . . . 888
10.360.3.2 GetMajorVersion() . . . 888
10.360.3.3 GetMinorVersion() . . . 888
10.360.3.4 GetVersion() . . . 888
10.360.3.5 Print(std::ostream &os=std::cout) const . . . 888
10.360.4 Friends And Related Function Documentation . . . 888
10.360.4.1 operator<< . . . 888
10.361 gdcm::VL Class Reference . . . 888

10.361.1 Detailed Description . . . 889
10.361.2 Member Typedef Documentation . . . 890
10.361.2.1 Type . . . 890
10.361.3 Constructor & Destructor Documentation . . . 890
10.361.3.1 VL(uint32_t vl=0) . . . 890
10.361.4 Member Function Documentation . . . 890
10.361.4.1 GetLength() const . . . 890
10.361.4.2 GetVL16Max() . . . 890
10.361.4.3 GetVL32Max() . . . 890
10.361.4.4 IsOdd() const . . . 890
10.361.4.5 IsUndefined() const . . . 890
10.361.4.6 operator uint32_t() const . . . 890
10.361.4.7 operator++() . . . 890
10.361.4.8 operator++(int) . . . 890
10.361.4.9 operator+=(VL const &vl) . . . 890
10.361.4.10 Read(std::istream &is) . . . 890
10.361.4.11 Read16(std::istream &is) . . . 890
10.361.4.12 SetToUndefined() . . . 890
10.361.4.13 Write(std::ostream &os) const . . . 890
10.361.4.14 Write16(std::ostream &os) const . . . 891
10.361.5 Friends And Related Function Documentation . . . 891
10.361.5.1 operator<< . . . 891
10.362 gdcm::VM Class Reference . . . 891
10.362.1 Detailed Description . . . 893
10.362.2 Member Enumeration Documentation . . . 893
10.362.2.1 VMType . . . 893
10.362.3 Constructor & Destructor Documentation . . . 894
10.362.3.1 VM(VMType type=VM0) . . . 894

10.362.4 Member Function Documentation . . . 894
10.362.4.1 Compatible(VM const &vm) const . . . 894
10.362.4.2 GetIndex(VMType vm) . . . 895
10.362.4.3 GetLength() const . . . 895
10.362.4.4 GetNumberOfElementsFromArray(const char *array, unsigned int length) . . . 895
10.362.4.5 GetVMString(VMType vm) . . . 895
10.362.4.6 GetVMType(const char *vm) . . . 895
10.362.4.7 GetVMTypeFromLength(unsigned int length, unsigned int size) . . . 895
10.362.4.8 IsValid(int vm1, VMType vm2) . . . 895
10.362.4.9 operator VMType() const . . . 895
10.362.5 Friends And Related Function Documentation . . . 895
10.362.5.1 operator<< . . . 895
10.363 gdcm::VMToLength< T > Struct Template Reference . . . 895
10.364 gdcm::VR Class Reference . . . 896
10.364.1 Detailed Description . . . 897
10.364.2 Member Enumeration Documentation . . . 898
10.364.2.1 VRType . . . 898
10.364.3 Constructor & Destructor Documentation . . . 899
10.364.3.1 VR(VRType vr=INVALID) . . . 899
10.364.4 Member Function Documentation . . . 899
10.364.4.1 CanDisplay(VRType vr) . . . 899
10.364.4.2 Compatible(VR const &vr) const . . . 899
10.364.4.3 GetLength() const . . . 899
10.364.4.4 GetLength(VRType vr) . . . 899
10.364.4.5 GetSize() const . . . 899
10.364.4.6 GetSizeof() const . . . 899
10.364.4.7 GetVRString(VRType vr) . . . 899
10.364.4.8 GetVRStringFromFile(VRType vr) . . . 899

10.364.4.9 GetVRType(const char *vr) . . . 899
10.364.4.10 GetVRTypeFromFile(const char *vr) . . . 899
10.364.4.11 IsASCII(VRType vr) . . . 899
10.364.4.12 IsASCII2(VRType vr) . . . 899
10.364.4.13 IsBinary(VRType vr) . . . 899
10.364.4.14 IsBinary2(VRType vr) . . . 899
10.364.4.15 IsDual() const . . . 899
10.364.4.16 IsSwap(const char *vr) . . . 899
10.364.4.17 IsValid(const char *vr) . . . 899
10.364.4.18 IsValid(const char *vr1, VRType vr2) . . . 899
10.364.4.19 IsVRFile() const . . . 899
10.364.4.20 operator VRType() const . . . 900
10.364.4.21 Read(std::istream &is) . . . 900
10.364.4.22 Write(std::ostream &os) const . . . 900
10.364.5 Friends And Related Function Documentation . . . 900
10.364.5.1 operator<< . . . 900
10.365 gdcm::VR16ExplicitDataElement Class Reference . . . 900
10.365.1 Detailed Description . . . 901
10.365.2 Member Function Documentation . . . 902
10.365.2.1 GetLength() const . . . 902
10.365.2.2 Read(std::istream &is) . . . 902
10.365.2.3 ReadPreValue(std::istream &is) . . . 902
10.365.2.4 ReadValue(std::istream &is, bool readvalues=true) . . . 902
10.365.2.5 ReadWithLength(std::istream &is, VL &length) . . . 902
10.366 gdcm::VRToEncoding< T > Struct Template Reference . . . 902
10.367 gdcm::VRToType< T > Struct Template Reference . . . 902
10.367.1 Detailed Description . . . 903
10.368 gdcm::VRVLSize< T > Class Template Reference . . . 903

10.368gdcm::VRVLSize< 0 > Class Template Reference . . . 903
10.369.1Member Function Documentation . . . 903
10.369.1.1Read(std::istream &_is) . . . 903
10.369.1.2Write(std::ostream &os) . . . 903
10.370gdcm::VRVLSize< 1 > Class Template Reference . . . 904
10.370.1Member Function Documentation . . . 904
10.370.1.1Read(std::istream &_is) . . . 904
10.370.1.2Write(std::ostream &os) . . . 904
10.371vtkGDCMImageReader Class Reference . . . 904
10.371.1Detailed Description . . . 907
10.371.2Constructor & Destructor Documentation . . . 907
10.371.2.1vtkGDCMImageReader() . . . 907
10.371.2.2~vtkGDCMImageReader() . . . 907
10.371.3Member Function Documentation . . . 907
10.371.3.1CanReadFile(const char *fname) . . . 907
10.371.3.2ExecuteData(vtkDataObject *out) . . . 907
10.371.3.3ExecuteInformation() . . . 907
10.371.3.4FillMedicalImageInformation(const gdcm::ImageReader &reader) . . . 907
10.371.3.5GetDescriptiveName() . . . 907
10.371.3.6GetFileExtensions() . . . 907
10.371.3.7GetIconImage() . . . 907
10.371.3.8GetOverlay(int i) . . . 907
10.371.3.9LoadSingleFile(const char *filename, char *pointer, unsigned long &outlen) . . . 907
10.371.3.10New() . . . 907
10.371.3.11PrintSelf(ostream &os, vtkIndent indent) . . . 908
10.371.3.12RequestDataCompat() . . . 908
10.371.3.13RequestInformationCompat() . . . 908
10.371.3.14SetCurve(vtkPolyData *pd) . . . 908

10.371.3.15 SetFileNames(vtkStringArray *) . . . 908
10.371.3.16 SetFilePattern(const char *) . . . 908
10.371.3.17 SetFilePrefix(const char *) . . . 908
10.371.3.18 SetMedicalImageProperties(vtkMedicalImageProperties *pd) . . . 908
10.371.3.19 vtkBooleanMacro(LoadOverlays, int) . . . 908
10.371.3.20 vtkBooleanMacro(LoadIconImage, int) . . . 908
10.371.3.21 vtkBooleanMacro(LossyFlag, int) . . . 908
10.371.3.22 vtkBooleanMacro(ApplyLookupTable, int) . . . 908
10.371.3.23 vtkBooleanMacro(ApplyYBRToRGB, int) . . . 908
10.371.3.24 vtkGetMacro(LoadOverlays, int) . . . 908
10.371.3.25 vtkGetMacro(LoadIconImage, int) . . . 908
10.371.3.26 vtkGetMacro(LossyFlag, int) . . . 908
10.371.3.27 vtkGetMacro(NumberOfOverlays, int) . . . 908
10.371.3.28 vtkGetMacro(NumberOfIconImages, int) . . . 909
10.371.3.29 vtkGetMacro(ApplyLookupTable, int) . . . 909
10.371.3.30 vtkGetMacro(ApplyYBRToRGB, int) vtkSetMacro(ApplyYBRToRGB . . . 909
10.371.3.31 vtkGetMacro(ImageFormat, int) . . . 909
10.371.3.32 vtkGetMacro(PlanarConfiguration, int) . . . 909
10.371.3.33 vtkGetMacro(Shift, double) . . . 909
10.371.3.34 vtkGetMacro(Scale, double) . . . 909
10.371.3.35 vtkGetObjectMacro(DirectionCosines, vtkMatrix4x4) . . . 909
10.371.3.36 vtkGetObjectMacro(MedicalImageProperties, vtkMedicalImageProperties) . . . 909
10.371.3.37 vtkGetObjectMacro(FileNames, vtkStringArray) . . . 909
10.371.3.38 vtkGetObjectMacro(Curve, vtkPolyData) . . . 909
10.371.3.39 vtkGetStringMacro(FilePrefix) . . . 909
10.371.3.40 vtkGetStringMacro(FilePattern) . . . 909
10.371.3.41 vtkGetVector3Macro(ImagePositionPatient, double) . . . 909
10.371.3.42 vtkGetVector6Macro(ImageOrientationPatient, double) . . . 909

10.371.3.43 vtkSetMacro(LoadOverlays, int) . . . 909
10.371.3.44 vtkSetMacro(LoadIconImage, int) . . . 909
10.371.3.45 vtkSetMacro(LossyFlag, int) . . . 909
10.371.3.46 vtkSetMacro(ApplyLookupTable, int) . . . 909
10.371.3.47 vtkSetVector6Macro(ImageOrientationPatient, double) . . . 909
10.371.3.48 vtkTypeMacro(vtkGDCMImageReader, vtkMedicalImageReader2) . . . 909
10.371.4 Member Data Documentation . . . 910
10.371.4.1 ApplyInverseVideo . . . 910
10.371.4.2 ApplyLookupTable . . . 910
10.371.4.3 ApplyPlanarConfiguration . . . 910
10.371.4.4 ApplyShiftScale . . . 910
10.371.4.5 ApplyYBRToRGB . . . 910
10.371.4.6 Curve . . . 910
10.371.4.7 DirectionCosines . . . 910
10.371.4.8 FileNames . . . 910
10.371.4.9 ForceRescale . . . 910
10.371.4.10 IconDataScalarType . . . 910
10.371.4.11 IconImageDataExtent . . . 910
10.371.4.12 IconNumberOfScalarComponents . . . 910
10.371.4.13 ImageFormat . . . 910
10.371.4.14 ImageOrientationPatient . . . 910
10.371.4.15 ImagePositionPatient . . . 910
10.371.4.16 LoadIconImage . . . 910
10.371.4.17 LoadOverlays . . . 910
10.371.4.18 LossyFlag . . . 910
10.371.4.19 MedicalImageProperties . . . 910
10.371.4.20 NumberOfIconImages . . . 910
10.371.4.21 NumberOfOverlays . . . 910

10.371.4.22 PlanarConfiguration . . . 910
10.371.4.23 Scale . . . 910
10.371.4.24 Shift . . . 910
10.372 vtkGDCMImageReader2 Class Reference . . . 911
10.372.1 Detailed Description . . . 913
10.372.2 Constructor & Destructor Documentation . . . 913
10.372.2.1 vtkGDCMImageReader2() . . . 913
10.372.2.2 ~vtkGDCMImageReader2() . . . 913
10.372.3 Member Function Documentation . . . 913
10.372.3.1 CanReadFile(const char *fname) . . . 913
10.372.3.2 FillMedicalImageInformation(const gdcm::ImageReader &reader) . . . 913
10.372.3.3 GetDescriptiveName() . . . 913
10.372.3.4 GetFileExtensions() . . . 914
10.372.3.5 GetIconImage() . . . 914
10.372.3.6 GetIconImagePort() . . . 914
10.372.3.7 GetOverlay(int i) . . . 914
10.372.3.8 GetOverlayPort(int index) . . . 914
10.372.3.9 LoadSingleFile(const char *filename, char *pointer, unsigned long &outlen) . . . 914
10.372.3.10 New() . . . 914
10.372.3.11 PrintSelf(ostream &os, vtkIndent indent) . . . 914
10.372.3.12 ProcessRequest(vtkInformation *request, vtkInformationVector **inputVector, vtkInformationVector *outputVector) . . . 914
10.372.3.13 RequestData(vtkInformation *request, vtkInformationVector **inputVector, vtkInformationVector *outputVector) . . . 914
10.372.3.14 RequestDataCompat() . . . 914
10.372.3.15 RequestInformation(vtkInformation *request, vtkInformationVector **inputVector, vtkInformationVector *outputVector) . . . 914
10.372.3.16 RequestInformationCompat() . . . 914
10.372.3.17 SetCurve(vtkPolyData *pd) . . . 914

10.372.3.18 SetFilePattern(const char *) . . . 914
10.372.3.19 SetFilePrefix(const char *) . . . 914
10.372.3.20 SetMedicalImageProperties(vtkMedicalImageProperties *pd) . . . 914
10.372.3.21 vtkBooleanMacro(LoadOverlays, int) . . . 914
10.372.3.22 vtkBooleanMacro(LoadIconImage, int) . . . 915
10.372.3.23 vtkBooleanMacro(LossyFlag, int) . . . 915
10.372.3.24 vtkBooleanMacro(ApplyLookupTable, int) . . . 915
10.372.3.25 vtkBooleanMacro(ApplyYBRToRGB, int) . . . 915
10.372.3.26 vtkGetMacro(LoadOverlays, int) . . . 915
10.372.3.27 vtkGetMacro(LoadIconImage, int) . . . 915
10.372.3.28 vtkGetMacro(LossyFlag, int) . . . 915
10.372.3.29 vtkGetMacro(NumberOfOverlays, int) . . . 915
10.372.3.30 vtkGetMacro(NumberOfIconImages, int) . . . 915
10.372.3.31 vtkGetMacro(ApplyLookupTable, int) . . . 915
10.372.3.32 vtkGetMacro(ApplyYBRToRGB, int) vtkSetMacro(ApplyYBRToRGB . . . 915
10.372.3.33 vtkGetMacro(ImageFormat, int) . . . 915
10.372.3.34 vtkGetMacro(PlanarConfiguration, int) . . . 915
10.372.3.35 vtkGetMacro(Shift, double) . . . 915
10.372.3.36 vtkGetMacro(Scale, double) . . . 915
10.372.3.37 vtkGetObjectMacro(DirectionCosines, vtkMatrix4x4) . . . 915
10.372.3.38 vtkGetObjectMacro(Curve, vtkPolyData) . . . 915
10.372.3.39 vtkGetStringMacro(FilePrefix) . . . 915
10.372.3.40 vtkGetStringMacro(FilePattern) . . . 915
10.372.3.41 vtkGetVector3Macro(ImagePositionPatient, double) . . . 915
10.372.3.42 vtkGetVector6Macro(ImageOrientationPatient, double) . . . 915
10.372.3.43 vtkSetMacro(LoadOverlays, int) . . . 916
10.372.3.44 vtkSetMacro(LoadIconImage, int) . . . 916
10.372.3.45 vtkSetMacro(LossyFlag, int) . . . 916

10.372.3.46 vtkSetMacro(ApplyLookupTable, int) . . . 916
10.372.3.47 vtkSetVector6Macro(ImageOrientationPatient, double) . . . 916
10.372.3.48 vtkTypeMacro(vtkGDCMImageReader2, vtkMedicalImageReader2) . . . 916
10.372.4 Member Data Documentation . . . 916
10.372.4.1 ApplyInverseVideo . . . 916
10.372.4.2 ApplyLookupTable . . . 916
10.372.4.3 ApplyPlanarConfiguration . . . 916
10.372.4.4 ApplyShiftScale . . . 916
10.372.4.5 ApplyYBRToRGB . . . 916
10.372.4.6 Curve . . . 916
10.372.4.7 DirectionCosines . . . 916
10.372.4.8 ForceRescale . . . 916
10.372.4.9 IconDataScalarType . . . 916
10.372.4.10 IconImageDataExtent . . . 916
10.372.4.11 IconNumberOfScalarComponents . . . 916
10.372.4.12 ImageFormat . . . 916
10.372.4.13 ImageOrientationPatient . . . 916
10.372.4.14 ImagePositionPatient . . . 916
10.372.4.15 LoadIconImage . . . 917
10.372.4.16 LoadOverlays . . . 917
10.372.4.17 LossyFlag . . . 917
10.372.4.18 NumberOfIconImages . . . 917
10.372.4.19 NumberOfOverlays . . . 917
10.372.4.20 PlanarConfiguration . . . 917
10.372.4.21 Scale . . . 917
10.372.4.22 Shift . . . 917
10.373 vtkGDCMImageWriter Class Reference . . . 917
10.373.1 Detailed Description . . . 919

10.373.2 Member Enumeration Documentation . . . 919
10.373.2.1 CompressionTypes . . . 919
10.373.3 Constructor & Destructor Documentation . . . 920
10.373.3.1 vtkGDCMImageWriter() . . . 920
10.373.3.2 ~vtkGDCMImageWriter() . . . 920
10.373.4 Member Function Documentation . . . 920
10.373.4.1 GetDescriptiveName() . . . 920
10.373.4.2 GetFileExtensions() . . . 920
10.373.4.3 GetFileName() . . . 920
10.373.4.4 New() . . . 920
10.373.4.5 PrintSelf(ostream &os, vtkIndent indent) . . . 920
10.373.4.6 SetDirectionCosines(vtkMatrix4x4 *matrix) . . . 920
10.373.4.7 SetDirectionCosinesFromImageOrientationPatient(const double dircos[6]) . . . 920
10.373.4.8 SetFileNames(vtkStringArray *) . . . 920
10.373.4.9 SetMedicalImageProperties(vtkMedicalImageProperties *) . . . 921
10.373.4.10 vtkBooleanMacro(LossyFlag, int) . . . 921
10.373.4.11 vtkBooleanMacro(FileLowerLeft, int) . . . 921
10.373.4.12 vtkGetMacro(LossyFlag, int) . . . 921
10.373.4.13 vtkGetMacro(Shift, double) . . . 921
10.373.4.14 vtkGetMacro(Scale, double) . . . 921
10.373.4.15 vtkGetMacro(ImageFormat, int) . . . 921
10.373.4.16 vtkGetMacro(FileLowerLeft, int) . . . 921
10.373.4.17 vtkGetMacro(PlanarConfiguration, int) . . . 921
10.373.4.18 vtkGetMacro(CompressionType, int) . . . 921
10.373.4.19 vtkGetObjectMacro(MedicalImageProperties, vtkMedicalImageProperties) . . . 921
10.373.4.20 vtkGetObjectMacro(FileNames, vtkStringArray) . . . 921
10.373.4.21 vtkGetObjectMacro(DirectionCosines, vtkMatrix4x4) . . . 921
10.373.4.22 vtkGetStringMacro(StudyUID) . . . 921

10.373.4.23 vtkGetStringMacro(SeriesUID) . . . 921
10.373.4.24 vtkSetMacro(LossyFlag, int) . . . 921
10.373.4.25 vtkSetMacro(Shift, double) . . . 921
10.373.4.26 vtkSetMacro(Scale, double) . . . 921
10.373.4.27 vtkSetMacro(ImageFormat, int) . . . 921
10.373.4.28 vtkSetMacro(FileLowerLeft, int) . . . 922
10.373.4.29 vtkSetMacro(PlanarConfiguration, int) . . . 922
10.373.4.30 vtkSetMacro(CompressionType, int) . . . 922
10.373.4.31 vtkSetStringMacro(StudyUID) . . . 922
10.373.4.32 vtkSetStringMacro(SeriesUID) . . . 922
10.373.4.33 vtkTypeMacro(vtkGDCMImageWriter, vtkImageWriter) . . . 922
10.373.4.34 Write() . . . 922
10.373.4.35 WriteGDCMData(vtkImageData *data, int timeStep) . . . 922
10.373.4.36 WriteSlice(vtkImageData *data) . . . 922
10.374 vtkGDCMMedicalImageProperties Class Reference . . . 922
10.374.1 Constructor & Destructor Documentation . . . 924
10.374.1.1 vtkGDCMMedicalImageProperties() . . . 924
10.374.1.2 ~vtkGDCMMedicalImageProperties() . . . 924
10.374.2 Member Function Documentation . . . 924
10.374.2.1 Clear() . . . 924
10.374.2.2 GetFile(unsigned int t) . . . 924
10.374.2.3 New() . . . 924
10.374.2.4 PrintSelf(ostream &os, vtkIndent indent) . . . 924
10.374.2.5 PushBackFile(gdcm::File const &f) . . . 924
10.374.2.6 vtkTypeMacro(vtkGDCMMedicalImageProperties, vtkMedicalImageProperties) . . . 924
10.374.3 Friends And Related Function Documentation . . . 924
10.374.3.1 vtkGDCMImageReader . . . 924
10.374.3.2 vtkGDCMImageReader2 . . . 924

10.374.3.3 vtkGDCMImageWriter . . . 924
10.375 vtkGDCMPolyDataReader Class Reference . . . 925
10.375.1 Detailed Description . . . 926
10.375.2 Constructor & Destructor Documentation . . . 926
10.375.2.1 vtkGDCMPolyDataReader() . . . 926
10.375.2.2 ~vtkGDCMPolyDataReader() . . . 926
10.375.3 Member Function Documentation . . . 926
10.375.3.1 FillMedicalImageInformation(const gdcm::Reader &reader) . . . 926
10.375.3.2 New() . . . 926
10.375.3.3 PrintSelf(ostream &os, vtkIndent indent) . . . 927
10.375.3.4 RequestData(vtkInformation *, vtkInformationVector **, vtkInformationVector *) . . . 927
10.375.3.5 RequestData_HemodynamicWaveformStorage(gdcm::Reader const &reader, vtkInformationVector *outputVector) . . . 927
10.375.3.6 RequestData_RTStructureSetStorage(gdcm::Reader const &reader, vtkInformationVector *outputVector) . . . 927
10.375.3.7 RequestInformation(vtkInformation *vtkNotUsed(request), vtkInformationVector **vtkNotUsed(inputVector), vtkInformationVector *outputVector) . . . 927
10.375.3.8 RequestInformation_HemodynamicWaveformStorage(gdcm::Reader const &reader) . . . 927
10.375.3.9 RequestInformation_RTStructureSetStorage(gdcm::Reader const &reader) . . . 927
10.375.3.10 vtkGetObjectMacro(MedicalImageProperties, vtkMedicalImageProperties) . . . 927
10.375.3.11 vtkGetObjectMacro(RTStructSetProperties, vtkRTStructSetProperties) . . . 927
10.375.3.12 vtkGetStringMacro(FileName) . . . 927
10.375.3.13 vtkSetStringMacro(FileName) . . . 927
10.375.3.14 vtkTypeMacro(vtkGDCMPolyDataReader, vtkPolyDataAlgorithm) . . . 927
10.375.4 Member Data Documentation . . . 927
10.375.4.1 FileName . . . 927
10.375.4.2 MedicalImageProperties . . . 927
10.375.4.3 RTStructSetProperties . . . 927
10.376 vtkGDCMPolyDataWriter Class Reference . . . 928

10.376.1 Detailed Description . . . 929
10.376.2 Constructor & Destructor Documentation . . . 929
10.376.2.1 vtkGDCMPolyDataWriter() . . . 929
10.376.2.2 ~vtkGDCMPolyDataWriter() . . . 929
10.376.3 Member Function Documentation . . . 929
10.376.3.1 InitializeRTStructSet(vtkStdString inDirectory, vtkStdString inStructLabel, vtkStdString inStructName, vtkStringArray *inROINames, vtkStringArray *inROIAlgorithmName, vtkStringArray *inROIType) . . . 929
10.376.3.2 New() . . . 929
10.376.3.3 PrintSelf(ostream &os, vtkIndent indent) . . . 930
10.376.3.4 SetMedicalImageProperties(vtkMedicalImageProperties *pd) . . . 930
10.376.3.5 SetNumberOfInputPorts(int n) . . . 930
10.376.3.6 SetRTStructSetProperties(vtkRTStructSetProperties *pd) . . . 930
10.376.3.7 vtkTypeMacro(vtkGDCMPolyDataWriter, vtkPolyDataWriter) . . . 930
10.376.3.8 WriteData() . . . 930
10.376.3.9 WriteRTSTRUCTData(gdcm::File &file, int num) . . . 930
10.376.3.10 WriteRTSTRUCTInfo(gdcm::File &file) . . . 930
10.376.4 Member Data Documentation . . . 930
10.376.4.1 MedicalImageProperties . . . 930
10.376.4.2 RTStructSetProperties . . . 930
10.377 vtkGDCMTesting Class Reference . . . 931
10.377.1 Detailed Description . . . 932
10.377.2 Member Typedef Documentation . . . 932
10.377.2.1 MD5MetaImagesType . . . 932
10.377.3 Constructor & Destructor Documentation . . . 932
10.377.3.1 vtkGDCMTesting() . . . 932
10.377.3.2 ~vtkGDCMTesting() . . . 932
10.377.4 Member Function Documentation . . . 932
10.377.4.1 GetGDCMDataRoot() . . . 932

10.377.4.2 GetMD5MetaImage(unsigned int file) . . . 932
10.377.4.3 GetMHDMD5FromFile(const char *filepath) . . . 932
10.377.4.4 GetNumberOfMD5MetaImages() . . . 933
10.377.4.5 GetRAWMD5FromFile(const char *filepath) . . . 933
10.377.4.6 GetVTKDataRoot() . . . 933
10.377.4.7 New() . . . 933
10.377.4.8 PrintSelf(ostream &os, vtkIndent indent) . . . 933
10.377.4.9 vtkTypeMacro(vtkGDCMTesting, vtkObject) . . . 933
10.378 vtkGDCMThreadedImageReader Class Reference . . . 933
10.378.1 Constructor & Destructor Documentation . . . 935
10.378.1.1 vtkGDCMThreadedImageReader() . . . 935
10.378.1.2 ~vtkGDCMThreadedImageReader() . . . 935
10.378.2 Member Function Documentation . . . 935
10.378.2.1 ExecuteData(vtkDataObject *out) . . . 935
10.378.2.2 ExecuteInformation() . . . 935
10.378.2.3 New() . . . 935
10.378.2.4 PrintSelf(ostream &os, vtkIndent indent) . . . 935
10.378.2.5 ReadFiles(unsigned int nfiles, const char *filenames[]) . . . 935
10.378.2.6 RequestDataCompat() . . . 935
10.378.2.7 vtkBooleanMacro(UseShiftScale, int) . . . 935
10.378.2.8 vtkGetMacro(UseShiftScale, int) . . . 935
10.378.2.9 vtkSetMacro(Shift, double) . . . 935
10.378.2.10 vtkSetMacro(Scale, double) . . . 935
10.378.2.11 vtkSetMacro(UseShiftScale, int) . . . 935
10.378.2.12 vtkTypeMacro(vtkGDCMThreadedImageReader, vtkGDCMImageReader) . . . 935
10.379 vtkGDCMThreadedImageReader2 Class Reference . . . 936
10.379.1 Constructor & Destructor Documentation . . . 937
10.379.1.1 vtkGDCMThreadedImageReader2() . . . 937

10.379.1.2 ~vtkGDCMThreadedImageReader2() . . . 937
10.379.2 Member Function Documentation . . . 937
10.379.2.1 GetFileName(int i=0) . . . 937
10.379.2.2 New() . . . 938
10.379.2.3 PrintSelf(ostream &os, vtkIndent indent) . . . 938
10.379.2.4 RequestInformation(vtkInformation *request, vtkInformationVector **inputVector, vtkInformationVector *outputVector) . . . 938
10.379.2.5 SetFileName(const char *filename) . . . 938
10.379.2.6 SetFileNames(vtkStringArray *) . . . 938
10.379.2.7 SplitExtent(int splitExt[6], int startExt[6], int num, int total) . . . 938
10.379.2.8 ThreadedRequestData(vtkInformation *request, vtkInformationVector **inputVector, vtkInformationVector *outputVector, vtkImageData ***inData, vtkImageData **outData, int outExt[6], int id) . . . 938
10.379.2.9 vtkBooleanMacro(FileLowerLeft, int) . . . 938
10.379.2.10 vtkBooleanMacro(LoadOverlays, int) . . . 938
10.379.2.11 vtkBooleanMacro(UseShiftScale, int) . . . 938
10.379.2.12 vtkGetMacro(FileLowerLeft, int) . . . 938
10.379.2.13 vtkGetMacro(NumberOfOverlays, int) . . . 938
10.379.2.14 vtkGetMacro(DataScalarType, int) . . . 938
10.379.2.15 vtkGetMacro(NumberOfScalarComponents, int) . . . 938
10.379.2.16 vtkGetMacro(LoadOverlays, int) . . . 938
10.379.2.17 vtkGetMacro(Shift, double) . . . 938
10.379.2.18 vtkGetMacro(Scale, double) . . . 938
10.379.2.19 vtkGetMacro(UseShiftScale, int) . . . 938
10.379.2.20 vtkGetObjectMacro(FileNames, vtkStringArray) . . . 938
10.379.2.21 vtkGetVector3Macro(DataOrigin, double) . . . 938
10.379.2.22 vtkGetVector3Macro(DataSpacing, double) . . . 939
10.379.2.23 vtkGetVector6Macro(DataExtent, int) . . . 939
10.379.2.24 vtkSetMacro(FileLowerLeft, int) . . . 939

10.379.2.25 vtkSetMacro(DataScalarType, int) . . . 939
10.379.2.26 vtkSetMacro(NumberOfScalarComponents, int) . . . 939
10.379.2.27 vtkSetMacro(LoadOverlays, int) . . . 939
10.379.2.28 vtkSetMacro(Shift, double) . . . 939
10.379.2.29 vtkSetMacro(Scale, double) . . . 939
10.379.2.30 vtkSetMacro(UseShiftScale, int) . . . 939
10.379.2.31 vtkSetVector3Macro(DataOrigin, double) . . . 939
10.379.2.32 vtkSetVector3Macro(DataSpacing, double) . . . 939
10.379.2.33 vtkSetVector6Macro(DataExtent, int) . . . 939
10.379.2.34 vtkTypeMacro(vtkGDCMThreadedImageReader2, vtkThreadedImageAlgorithm) . . . 939
10.380 vtkImageColorViewer Class Reference . . . 940
10.380.1 Detailed Description . . . 942
10.380.2 Member Enumeration Documentation . . . 942
10.380.2.1 anonymous enum . . . 942
10.380.3 Constructor & Destructor Documentation . . . 943
10.380.3.1 vtkImageColorViewer() . . . 943
10.380.3.2 ~vtkImageColorViewer() . . . 943
10.380.4 Member Function Documentation . . . 943
10.380.4.1 AddInput(vtkImageData *input) . . . 943
10.380.4.2 AddInputConnection(vtkAlgorithmOutput *input) . . . 943
10.380.4.3 GetColorLevel() . . . 943
10.380.4.4 GetColorWindow() . . . 943
10.380.4.5 GetInput() . . . 943
10.380.4.6 GetOffScreenRendering() . . . 943
10.380.4.7 GetOverlayVisibility() . . . 943
10.380.4.8 GetPosition() . . . 943
10.380.4.9 GetSize() . . . 943
10.380.4.10 GetSliceMax() . . . 943

10.380.4.10 GetSliceMin() . . . 943
10.380.4.11 GetSliceRange(int range[2]) . . . 943
10.380.4.12 GetSliceRange(int &min, int &max) . . . 943
10.380.4.13 GetSliceRange() . . . 943
10.380.4.14 GetWindowName() . . . 943
10.380.4.15 InstallPipeline() . . . 943
10.380.4.16 New() . . . 943
10.380.4.17 PrintSelf(ostream &os, vtkIndent indent) . . . 944
10.380.4.18 Render(void) . . . 944
10.380.4.19 SetColorLevel(double s) . . . 944
10.380.4.20 SetColorWindow(double s) . . . 944
10.380.4.21 SetDisplayId(void *a) . . . 944
10.380.4.22 SetInput(vtkImageData *in) . . . 944
10.380.4.23 SetInputConnection(vtkAlgorithmOutput *input) . . . 944
10.380.4.24 SetOffScreenRendering(int) . . . 944
10.380.4.25 SetOverlayVisibility(double vis) . . . 944
10.380.4.26 SetParentId(void *a) . . . 944
10.380.4.27 SetPosition(int a, int b) . . . 944
10.380.4.28 SetPosition(int a[2]) . . . 944
10.380.4.29 SetRenderer(vtkRenderer *arg) . . . 944
10.380.4.30 SetRenderWindow(vtkRenderWindow *arg) . . . 944
10.380.4.31 SetSize(int a, int b) . . . 944
10.380.4.32 SetSize(int a[2]) . . . 945
10.380.4.33 SetSlice(int s) . . . 945
10.380.4.34 SetSliceOrientation(int orientation) . . . 945
10.380.4.35 SetSliceOrientationToXY() . . . 945
10.380.4.36 SetSliceOrientationToXZ() . . . 945
10.380.4.37 SetSliceOrientationToYZ() . . . 945

10.380.4.39 SetupInteractor(vtkRenderWindowInteractor *) . . . 945
10.380.4.40 SetWindowId(void *a) . . . 945
10.380.4.41 UnInstallPipeline() . . . 945
10.380.4.42 UpdateDisplayExtent() . . . 945
10.380.4.43 UpdateOrientation() . . . 945
10.380.4.44 VTK_LEGACY(int GetWholeZMin()) . . . 945
10.380.4.45 VTK_LEGACY(int GetWholeZMax()) . . . 945
10.380.4.46 VTK_LEGACY(int GetZSlice()) . . . 946
10.380.4.47 VTK_LEGACY(void SetZSlice(int)) . . . 946
10.380.4.48 vtkBooleanMacro(OffScreenRendering, int) . . . 946
10.380.4.49 vtkGetMacro(SliceOrientation, int) . . . 946
10.380.4.50 vtkGetMacro(Slice, int) . . . 946
10.380.4.51 vtkGetObjectMacro(RenderWindow, vtkRenderWindow) . . . 946
10.380.4.52 vtkGetObjectMacro(Renderer, vtkRenderer) . . . 946
10.380.4.53 vtkGetObjectMacro(ImageActor, vtkImageActor) . . . 946
10.380.4.54 vtkGetObjectMacro(WindowLevel, vtkImageMapToWindowLevelColors2) . . . 946
10.380.4.55 vtkGetObjectMacro(InteractorStyle, vtkInteractorStyleImage) . . . 946
10.380.4.56 vtkTypeMacro(vtkImageColorViewer, vtkObject) . . . 946
10.380.5 Friends And Related Function Documentation . . . 946
10.380.5.1 vtkImageColorViewerCallback . . . 946
10.380.6 Member Data Documentation . . . 946
10.380.6.1 FirstRender . . . 946
10.380.6.2 ImageActor . . . 946
10.380.6.3 Interactor . . . 946
10.380.6.4 InteractorStyle . . . 946
10.380.6.5 OverlayImageActor . . . 946
10.380.6.6 Renderer . . . 946
10.380.6.7 RenderWindow . . . 946

10.380.6.8 Slice . . . 946
10.380.6.9 SliceOrientation . . . 946
10.380.6.10 WindowLevel . . . 946
10.381 vtkImageMapToColors16 Class Reference . . . 947
10.381.1 Constructor & Destructor Documentation . . . 948
10.381.1.1 vtkImageMapToColors16() . . . 948
10.381.1.2 ~vtkImageMapToColors16() . . . 948
10.381.2 Member Function Documentation . . . 948
10.381.2.1 GetMTime() . . . 948
10.381.2.2 New() . . . 948
10.381.2.3 PrintSelf(ostream &os, vtkIndent indent) . . . 948
10.381.2.4 RequestData(vtkInformation *request, vtkInformationVector **inputVector, vtkInformationVector *outputVector) . . . 949
10.381.2.5 RequestInformation(vtkInformation *, vtkInformationVector **, vtkInformationVector *) 949
10.381.2.6 SetLookupTable(vtkScalarsToColors *) . . . 949
10.381.2.7 SetOutputFormatToLuminance() . . . 949
10.381.2.8 SetOutputFormatToLuminanceAlpha() . . . 949
10.381.2.9 SetOutputFormatToRGB() . . . 949
10.381.2.10 SetOutputFormatToRGBA() . . . 949
10.381.2.11 ThreadedRequestData(vtkInformation *request, vtkInformationVector **inputVector, vtkInformationVector *outputVector, vtkImageData ***inData, vtkImageData **outData, int extent[6], int id) . . . 949
10.381.2.12 vtkBooleanMacro(PassAlphaToOutput, int) . . . 949
10.381.2.13 vtkGetMacro(OutputFormat, int) . . . 949
10.381.2.14 vtkGetMacro(ActiveComponent, int) . . . 949
10.381.2.15 vtkGetMacro(PassAlphaToOutput, int) . . . 949
10.381.2.16 vtkGetObjectMacro(LookupTable, vtkScalarsToColors) . . . 949
10.381.2.17 vtkSetMacro(OutputFormat, int) . . . 949
10.381.2.18 vtkSetMacro(ActiveComponent, int) . . . 949

10.381.2.19 vtkSetMacro(PassAlphaToOutput, int) . . . 949
10.381.2.20 vtkTypeMacro(vtkImageMapToColors16, vtkThreadedImageAlgorithm) . . . 949
10.381.3 Member Data Documentation . . . 949
10.381.3.1 ActiveComponent . . . 949
10.381.3.2 DataWasPassed . . . 949
10.381.3.3 LookupTable . . . 949
10.381.3.4 OutputFormat . . . 949
10.381.3.5 PassAlphaToOutput . . . 949
10.382 vtkImageMapToWindowLevelColors2 Class Reference . . . 950
10.382.1 Constructor & Destructor Documentation . . . 951
10.382.1.1 vtkImageMapToWindowLevelColors2() . . . 951
10.382.1.2 ~vtkImageMapToWindowLevelColors2() . . . 951
10.382.2 Member Function Documentation . . . 951
10.382.2.1 New() . . . 951
10.382.2.2 PrintSelf(ostream &os, vtkIndent indent) . . . 951
10.382.2.3 RequestData(vtkInformation *request, vtkInformationVector **inputVector, vtkInformationVector *outputVector) . . . 951
10.382.2.4 RequestInformation(vtkInformation *, vtkInformationVector **, vtkInformationVector *) 951
10.382.2.5 ThreadedRequestData(vtkInformation *request, vtkInformationVector **inputVector, vtkInformationVector *outputVector, vtkImageData ***inData, vtkImageData **outData, int extent[6], int id) . . . 951
10.382.2.6 vtkGetMacro(Window, double) . . . 951
10.382.2.7 vtkGetMacro(Level, double) . . . 952
10.382.2.8 vtkSetMacro(Window, double) . . . 952
10.382.2.9 vtkSetMacro(Level, double) . . . 952
10.382.2.10 vtkTypeMacro(vtkImageMapToWindowLevelColors2, vtkImageMapToColors) . . . 952
10.382.3 Member Data Documentation . . . 952
10.382.3.1 Level . . . 952
10.382.3.2 Window . . . 952

10.383 vtkImagePlanarComponentsToComponents Class Reference . . . 952
10.383.1 Constructor & Destructor Documentation . . . 953
10.383.1.1 vtkImagePlanarComponentsToComponents() . . . 953
10.383.1.2 ∼vtkImagePlanarComponentsToComponents() . . . 953
10.383.2 Member Function Documentation . . . 953
10.383.2.1 New() . . . 953
10.383.2.2 PrintSelf(ostream &os, vtkIndent indent) . . . 954
10.383.2.3 RequestData(vtkInformation ∗, vtkInformationVector ∗∗, vtkInformationVector ∗) . . . 954
10.383.2.4 vtkTypeMacro(vtkImagePlanarComponentsToComponents, vtkImageAlgorithm) . . . 954
10.384 vtkImageRGBToYBR Class Reference . . . 954
10.384.1 Constructor & Destructor Documentation . . . 955
10.384.1.1 vtkImageRGBToYBR() . . . 955
10.384.1.2 ∼vtkImageRGBToYBR() . . . 955
10.384.2 Member Function Documentation . . . 955
10.384.2.1 New() . . . 955
10.384.2.2 PrintSelf(ostream &os, vtkIndent indent) . . . 955
10.384.2.3 ThreadedExecute(vtkImageData ∗inData, vtkImageData ∗outData, int ext[6], int id) . . . 955
10.384.2.4 vtkTypeMacro(vtkImageRGBToYBR, vtkThreadedImageAlgorithm) . . . 955
10.385 vtkImageYBRToRGB Class Reference . . . 956
10.385.1 Constructor & Destructor Documentation . . . 957
10.385.1.1 vtkImageYBRToRGB() . . . 957
10.385.1.2 ∼vtkImageYBRToRGB() . . . 957
10.385.2 Member Function Documentation . . . 957
10.385.2.1 New() . . . 957
10.385.2.2 PrintSelf(ostream &os, vtkIndent indent) . . . 957
10.385.2.3 ThreadedExecute(vtkImageData ∗inData, vtkImageData ∗outData, int ext[6], int id) . . . 957
10.385.2.4 vtkTypeMacro(vtkImageYBRToRGB, vtkThreadedImageAlgorithm) . . . 957
10.386 vtkLookupTable16 Class Reference . . . 957

10.386.1 Constructor & Destructor Documentation . . . 959
10.386.1.1 vtkLookupTable16(int sze=256, int ext=256) . . . 959
10.386.1.2 ∼vtkLookupTable16() . . . 959
10.386.2 Member Function Documentation . . . 959
10.386.2.1 Build() . . . 959
10.386.2.2 GetPointer(const vtkIdType id) . . . 959
10.386.2.3 MapScalarsThroughTable2(void ∗input, unsigned char ∗output, int inputDataType, int numberOfValues, int inputIncrement, int outputFormat) . . . 959
10.386.2.4 New() . . . 959
10.386.2.5 PrintSelf(ostream &os, vtkIndent indent) . . . 959
10.386.2.6 SetNumberOfTableValues(vtkIdType number) . . . 959
10.386.2.7 vtkTypeMacro(vtkLookupTable16, vtkLookupTable) . . . 959
10.386.2.8 WritePointer(const vtkIdType id, const int number) . . . 959
10.386.3 Member Data Documentation . . . 959
10.386.3.1 Table16 . . . 959
10.387 vtkRTStructSetProperties Class Reference . . . 960
10.387.1 Detailed Description . . . 962
10.387.2 Constructor & Destructor Documentation . . . 962
10.387.2.1 vtkRTStructSetProperties() . . . 962
10.387.2.2 ∼vtkRTStructSetProperties() . . . 962
10.387.3 Member Function Documentation . . . 962
10.387.3.1 AddContourReferencedFrameOfReference(vtkIdType pdnum, const char ∗classuid, const char ∗instanceuid) . . . 962
10.387.3.2 AddReferencedFrameOfReference(const char ∗classuid, const char ∗instanceuid) . . . 962
10.387.3.3 AddStructureSetROI(int roinumber, const char ∗reframerefuid, const char ∗roiname, const char ∗ROIGenerationAlgorithm, const char ∗ROIDescription=0) . . . 962
10.387.3.4 AddStructureSetROIObservation(int refnumber, int observationnumber, const char ∗rtroiinterpretedtype, const char ∗roiinterpreter, const char ∗roiobservationlabel=0) . . . 962
10.387.3.5 Clear() . . . 962
10.387.3.6 DeepCopy(vtkRTStructSetProperties ∗p) . . . 962

10.387.3.7 GetContourReferencedFrameOfReferenceClassUID(vtkIdType pdnum, vtkIdType id) . 962
10.387.3.8 GetContourReferencedFrameOfReferenceInstanceUID(vtkIdType pdnum, vtkIdType id) 963
10.387.3.9 GetNumberOfContourReferencedFrameOfReferences() 963
10.387.3.10 GetNumberOfContourReferencedFrameOfReferences(vtkIdType pdnum) 963
10.387.3.11 GetNumberOfReferencedFrameOfReferences() 963
10.387.3.12 GetNumberOfStructureSetROIs() 963
10.387.3.13 GetReferencedFrameOfReferenceClassUID(vtkIdType id) 963
10.387.3.14 GetReferencedFrameOfReferenceInstanceUID(vtkIdType id) 963
10.387.3.15 GetStructureSetObservationNumber(vtkIdType id) 963
10.387.3.16 GetStructureSetROIDescription(vtkIdType id) 963
10.387.3.17 GetStructureSetROIGenerationAlgorithm(vtkIdType) 963
10.387.3.18 GetStructureSetROIName(vtkIdType) 963
10.387.3.19 GetStructureSetROINumber(vtkIdType id) 963
10.387.3.20 GetStructureSetROIObservationLabel(vtkIdType id) 963
10.387.3.21 GetStructureSetROIRefFrameRefUID(vtkIdType) 963
10.387.3.22 GetStructureSetRTROIInterpretedType(vtkIdType id) 963
10.387.3.23 New() 963
10.387.3.24 PrintSelf(ostream &os, vtkIndent indent) 964
10.387.3.25 vtkGetStringMacro(StructureSetLabel) 964
10.387.3.26 vtkGetStringMacro(StructureSetName) 964
10.387.3.27 vtkGetStringMacro(StructureSetDate) 964
10.387.3.28 vtkGetStringMacro(StructureSetTime) 964
10.387.3.29 vtkGetStringMacro(SOPInstanceUID) 964
10.387.3.30 vtkGetStringMacro(StudyInstanceUID) 964
10.387.3.31 vtkGetStringMacro(SeriesInstanceUID) 964
10.387.3.32 vtkGetStringMacro(ReferenceSeriesInstanceUID) 964
10.387.3.33 vtkGetStringMacro(ReferenceFrameOfReferenceUID) 964
10.387.3.34 vtkSetStringMacro(StructureSetLabel) 964

10.387.3.35 vtkSetStringMacro(StructureSetName) . . . 964
10.387.3.36 vtkSetStringMacro(StructureSetDate) . . . 964
10.387.3.37 vtkSetStringMacro(StructureSetTime) . . . 964
10.387.3.38 vtkSetStringMacro(SOPInstanceUID) . . . 964
10.387.3.39 vtkSetStringMacro(StudyInstanceUID) . . . 964
10.387.3.40 vtkSetStringMacro(SeriesInstanceUID) . . . 964
10.387.3.41 vtkSetStringMacro(ReferenceSeriesInstanceUID) . . . 964
10.387.3.42 vtkSetStringMacro(ReferenceFrameOfReferenceUID) . . . 964
10.387.3.43 vtkTypeMacro(vtkRTStructSetProperties, vtkObject) . . . 964
10.387.4 Member Data Documentation . . . 964
10.387.4.1 Internals . . . 965
10.387.4.2 ReferenceFrameOfReferenceUID . . . 965
10.387.4.3 ReferenceSeriesInstanceUID . . . 965
10.387.4.4 SeriesInstanceUID . . . 965
10.387.4.5 SOPInstanceUID . . . 965
10.387.4.6 StructureSetDate . . . 965
10.387.4.7 StructureSetLabel . . . 965
10.387.4.8 StructureSetName . . . 965
10.387.4.9 StructureSetTime . . . 965
10.387.4.10 StudyInstanceUID . . . 965
10.388 gdcm::Waveform Class Reference . . . 965
10.388.1 Detailed Description . . . 965
10.388.2 Constructor & Destructor Documentation . . . 966
10.388.2.1 Waveform() . . . 966
10.389 gdcm::WLMFindQuery Class Reference . . . 966
10.389.1 Detailed Description . . . 967
10.389.2 Constructor & Destructor Documentation . . . 967
10.389.2.1 WLMFindQuery() . . . 967

10.389.3 Member Function Documentation . . . 967
10.389.3.1 GetAbstractSyntaxUID() const . . . 967
10.389.3.2 GetTagListByLevel(const EQueryLevel &inQueryLevel) . . . 968
10.389.3.3 GetValidDataSet() const . . . 968
10.389.3.4 InitializeDataSet(const EQueryLevel &inQueryLevel) . . . 968
10.389.3.5 ValidateQuery(bool inStrict=true) const . . . 968
10.389.4 Friends And Related Function Documentation . . . 968
10.389.4.1 QueryFactory . . . 968
10.390 gdcm::Writer Class Reference . . . 969
10.390.1 Detailed Description . . . 971
10.390.2 Constructor & Destructor Documentation . . . 972
10.390.2.1 Writer() . . . 972
10.390.2.2 ~Writer() . . . 972
10.390.3 Member Function Documentation . . . 972
10.390.3.1 CheckFileMetaInformationOff() . . . 972
10.390.3.2 CheckFileMetaInformationOn() . . . 972
10.390.3.3 GetFile() . . . 972
10.390.3.4 GetStreamPtr() const . . . 972
10.390.3.5 SetCheckFileMetaInformation(bool b) . . . 972
10.390.3.6 SetFile(const File &f) . . . 972
10.390.3.7 SetFileName(const char *filename_native) . . . 973
10.390.3.8 SetStream(std::ostream &output_stream) . . . 973
10.390.3.9 SetWriteDataSetOnly(bool b) . . . 973
10.390.3.10 Write() . . . 973
10.390.4 Friends And Related Function Documentation . . . 973
10.390.4.1 StreamImageWriter . . . 973
10.390.5 Member Data Documentation . . . 973
10.390.5.1 Ofstream . . . 973

10.390.5.2 Stream . . . 973
10.391 gdcm::XMLDictReader Class Reference . . . 974
10.391.1 Detailed Description . . . 975
10.391.2 Constructor & Destructor Documentation . . . 975
10.391.2.1 XMLDictReader() . . . 975
10.391.2.2 ~XMLDictReader() . . . 975
10.391.3 Member Function Documentation . . . 975
10.391.3.1 CharacterDataHandler(const char *data, int length) . . . 975
10.391.3.2 EndElement(const char *name) . . . 975
10.391.3.3 GetDict() . . . 975
10.391.3.4 HandleDescription(const char **atts) . . . 975
10.391.3.5 HandleEntry(const char **atts) . . . 975
10.391.3.6 StartElement(const char *name, const char **atts) . . . 975
10.392 gdcm::XMLPrinter Class Reference . . . 976
10.392.1 Member Enumeration Documentation . . . 977
10.392.1.1 PrintStyles . . . 977
10.392.2 Constructor & Destructor Documentation . . . 977
10.392.2.1 XMLPrinter() . . . 977
10.392.2.2 ~XMLPrinter() . . . 977
10.392.3 Member Function Documentation . . . 977
10.392.3.1 GetPrintStyle() const . . . 977
10.392.3.2 HandleBulkData(const char *uuid, const TransferSyntax &ts, const char *bulkdata, size_t bulklen) . . . 977
10.392.3.3 Print(std::ostream &os) . . . 977
10.392.3.4 PrintDataElement(std::ostream &os, const Dicts &dicts, const DataSet &ds, const DataElement &de, const TransferSyntax &ts) . . . 977
10.392.3.5 PrintDataSet(const DataSet &ds, const TransferSyntax &ts, std::ostream &os) . . . 977
10.392.3.6 PrintSQ(const SequenceOfItems *sqi, const TransferSyntax &ts, std::ostream &os) . . . 977
10.392.3.7 SetFile(File const &f) . . . 977

10.392.3.8 SetStyle(PrintStyles ps) . . . 977
10.392.4 Member Data Documentation . . . 977
10.392.4.1 F . . . 977
10.392.4.2 PrintStyle . . . 977
10.393 gdcm::XMLPrivateDictReader Class Reference . . . 978
10.393.1 Detailed Description . . . 979
10.393.2 Constructor & Destructor Documentation . . . 979
10.393.2.1 XMLPrivateDictReader() . . . 979
10.393.2.2 ~XMLPrivateDictReader() . . . 979
10.393.3 Member Function Documentation . . . 979
10.393.3.1 CharacterDataHandler(const char *data, int length) . . . 979
10.393.3.2 EndElement(const char *name) . . . 979
10.393.3.3 GetPrivateDict() . . . 979
10.393.3.4 HandleDescription(const char **atts) . . . 979
10.393.3.5 HandleEntry(const char **atts) . . . 979
10.393.3.6 StartElement(const char *name, const char **atts) . . . 979

11 File Documentation **981**

11.1 gdcmAAbortPDU.h File Reference . . . 981
11.2 gdcmAAssociateACPDU.h File Reference . . . 982
11.3 gdcmAAssociateRJPDU.h File Reference . . . 982
11.4 gdcmAAssociateRQPDU.h File Reference . . . 983
11.5 gdcmAbstractSyntax.h File Reference . . . 984
11.6 gdcmAnonymizeEvent.h File Reference . . . 985
11.7 gdcmAnonymizer.h File Reference . . . 986
11.8 gdcmApplicationContext.h File Reference . . . 987
11.9 gdcmApplicationEntity.h File Reference . . . 988
11.10 gdcmAReleaseRPPDU.h File Reference . . . 988

11.11gdcmAReleaseRQPDU.h File Reference 989
11.12gdcmARTIMTimer.h File Reference 990
11.13gdcmASN1.h File Reference 991
11.14gdcmAsynchronousOperationsWindowSub.h File Reference 992
11.15gdcmAttribute.h File Reference 992
11.16gdcmAudioCodec.h File Reference 994
11.17gdcmBase64.h File Reference 994
11.18gdcmBaseCompositeMessage.h File Reference 995
11.19gdcmBaseNormalizedMessage.h File Reference 996
11.20gdcmBasePDU.h File Reference 997
11.21gdcmBaseQuery.h File Reference 998
11.22gdcmBaseRootQuery.h File Reference 999
11.23gdcmBasicOffsetTable.h File Reference 1000
11.24gdcmBitmap.h File Reference 1001
11.25gdcmBitmapToBitmapFilter.h File Reference 1002
11.26gdcmBoxRegion.h File Reference 1003
11.27gdcmByteBuffer.h File Reference 1003
11.28gdcmByteSwap.h File Reference 1005
11.29gdcmByteSwapFilter.h File Reference 1005
11.30gdcmByteValue.h File Reference 1006
11.31gdcmCAPICryptoFactory.h File Reference 1007
11.32gdcmCAPICryptographicMessageSyntax.h File Reference 1008
11.33gdcmCEchoMessages.h File Reference 1008
11.34gdcmCFindMessages.h File Reference 1009
11.35gdcmCMoveMessages.h File Reference 1010
11.36gdcmCodec.h File Reference 1011
11.37gdcmCoder.h File Reference 1012
11.38gdcmCodeString.h File Reference 1014

11.39gdcmCommand.h File Reference 1015
11.40gdcmCommandDataSet.h File Reference 1016
11.41gdcmCompositeMessageFactory.h File Reference 1017
11.42gdcmCompositeNetworkFunctions.h File Reference 1017
11.43gdcmConstCharWrapper.h File Reference 1018
11.44gdcmCP246ExplicitDataElement.h File Reference 1019
11.45gdcmCryptoFactory.h File Reference 1019
11.46gdcmCryptographicMessageSyntax.h File Reference 1020
11.47gdcmCSAElement.h File Reference 1021
11.48gdcmCSAHeader.h File Reference 1022
11.49gdcmCSAHeaderDict.h File Reference 1023
11.50gdcmCSAHeaderDictEntry.h File Reference 1024
11.51gdcmCStoreMessages.h File Reference 1026
11.52gdcmCurve.h File Reference 1027
11.53gdcmDataElement.h File Reference 1028
11.54gdcmDataEvent.h File Reference 1029
11.55gdcmDataSet.h File Reference 1030
11.56gdcmDataSetEvent.h File Reference 1031
11.57gdcmDataSetHelper.h File Reference 1031
11.58gdcmDecoder.h File Reference 1032
11.59gdcmDefinedTerms.h File Reference 1034
11.60gdcmDeflateStream.h File Reference 1034
11.61gdcmDefs.h File Reference 1035
11.62gdcmDeltaEncodingCodec.h File Reference 1036
11.63gdcmDICOMDIR.h File Reference 1036
11.64gdcmDICOMDIRGenerator.h File Reference 1037
11.65gdcmDict.h File Reference 1038
11.66gdcmDictConverter.h File Reference 1039

11.67gdcmDictEntry.h File Reference . . . 1040
11.68gdcmDictPrinter.h File Reference . . . 1041
11.69gdcmDicts.h File Reference . . . 1042
11.70gdcmDIMSE.h File Reference . . . 1043
11.71gdcmDirectionCosines.h File Reference . . . 1043
11.72gdcmDirectory.h File Reference . . . 1044
11.73gdcmDirectoryHelper.h File Reference . . . 1045
11.74gdcmDummyValueGenerator.h File Reference . . . 1046
11.75gdcmDumper.h File Reference . . . 1047
11.76gdcmElement.h File Reference . . . 1047
11.76.1 Macro Definition Documentation . . . 1049
11.76.1.1 VRDS16ILLEGAL . . . 1049
11.77gdcmEncapsulatedDocument.h File Reference . . . 1049
11.78gdcmEnumeratedValues.h File Reference . . . 1050
11.79gdcmEvent.h File Reference . . . 1050
11.79.1 Macro Definition Documentation . . . 1052
11.79.1.1 gdcmEventMacro . . . 1052
11.80gdcmException.h File Reference . . . 1052
11.81gdcmExplicitDataElement.h File Reference . . . 1053
11.82gdcmExplicitImplicitDataElement.h File Reference . . . 1054
11.83gdcmFiducials.h File Reference . . . 1054
11.84gdcmFile.h File Reference . . . 1055
11.85gdcmFileAnonymizer.h File Reference . . . 1056
11.86gdcmFileChangeTransferSyntax.h File Reference . . . 1057
11.87gdcmFileDecompressLookupTable.h File Reference . . . 1057
11.88gdcmFileDerivation.h File Reference . . . 1058
11.89gdcmFileExplicitFilter.h File Reference . . . 1059
11.90gdcmFileMetaInformation.h File Reference . . . 1060

11.91 gdcmFilename.h File Reference . . . 1061
11.92 gdcmFileNameEvent.h File Reference . . . 1061
11.93 gdcmFilenameGenerator.h File Reference . . . 1062
11.94 gdcmFileSet.h File Reference . . . 1063
11.95 gdcmFileStreamer.h File Reference . . . 1064
11.96 gdcmFindPatientRootQuery.h File Reference . . . 1065
11.97 gdcmFindStudyRootQuery.h File Reference . . . 1066
11.98 gdcmFragment.h File Reference . . . 1066
11.99 gdcmGlobal.h File Reference . . . 1068
11.100 gdcmGroupDict.h File Reference . . . 1069
11.101 gdcmIconImage.h File Reference . . . 1069
11.102 gdcmIconImageFilter.h File Reference . . . 1070
11.103 gdcmIconImageGenerator.h File Reference . . . 1071
11.104 gdcmImage.h File Reference . . . 1072
11.105 gdcmImageApplyLookupTable.h File Reference . . . 1073
11.106 gdcmImageChangePhotometricInterpretation.h File Reference . . . 1074
11.107 gdcmImageChangePlanarConfiguration.h File Reference . . . 1074
11.108 gdcmImageChangeTransferSyntax.h File Reference . . . 1075
11.109 gdcmImageCodec.h File Reference . . . 1076
11.110 gdcmImageConverter.h File Reference . . . 1077
11.111 gdcmImageFragmentSplitter.h File Reference . . . 1078
11.112 gdcmImageHelper.h File Reference . . . 1079
11.113 gdcmImageReader.h File Reference . . . 1080
11.114 gdcmImageRegionReader.h File Reference . . . 1081
11.115 gdcmImageToImageFilter.h File Reference . . . 1081
11.116 gdcmImageWriter.h File Reference . . . 1082
11.117 gdcmImplementationClassUIDSub.h File Reference . . . 1083
11.118 gdcmImplementationUIDSub.h File Reference . . . 1084

11.119gdcmImplementationVersionNameSub.h File Reference . . . 1085
11.120gdcmImplicitDataElement.h File Reference . . . 1086
11.121gdcmIOD.h File Reference . . . 1087
11.122gdcmIODEntry.h File Reference . . . 1089
11.123gdcmIODs.h File Reference . . . 1091
11.124gdcmIPPSorter.h File Reference . . . 1092
11.125gdcmItem.h File Reference . . . 1093
11.126gdcmJPEG12Codec.h File Reference . . . 1094
11.127gdcmJPEG16Codec.h File Reference . . . 1095
11.128gdcmJPEG2000Codec.h File Reference . . . 1095
11.129gdcmJPEG8Codec.h File Reference . . . 1096
11.130gdcmJPEGCodec.h File Reference . . . 1097
11.131gdcmJPEGLSCodec.h File Reference . . . 1099
11.132gdcmJSON.h File Reference . . . 1099
11.133gdcmKAKADUCodec.h File Reference . . . 1100
11.134gdcmLegacyMacro.h File Reference . . . 1101
11.134.1Macro Definition Documentation . . . 1102
11.134.1.1GDCM_LEGACY . . . 1102
11.134.1.2GDCM_LEGACY_BODY . . . 1102
11.134.1.3GDCM_LEGACY_REPLACED_BODY . . . 1102
11.135gdcmLO.h File Reference . . . 1102
11.136gdcmLookupTable.h File Reference . . . 1103
11.137gdcmMacro.h File Reference . . . 1104
11.138gdcmMacroEntry.h File Reference . . . 1106
11.138.1Macro Definition Documentation . . . 1107
11.138.1.1GDCMMACROENTRY_H . . . 1107
11.139gdcmMacros.h File Reference . . . 1107
11.140gdcmMaximumLengthSub.h File Reference . . . 1109

11.141gdcmMD5.h File Reference 1110
11.142gdcmMediaStorage.h File Reference 1111
11.143gdcmMeshPrimitive.h File Reference 1112
11.144gdcmModalityPerformedProcedureStepCreateQuery.h File Reference 1113
11.145gdcmModalityPerformedProcedureStepSetQuery.h File Reference 1114
11.146gdcmModule.h File Reference 1114
11.147gdcmModuleEntry.h File Reference 1116
11.148gdcmModules.h File Reference 1118
11.149gdcmMovePatientRootQuery.h File Reference 1119
11.150gdcmMoveStudyRootQuery.h File Reference 1120
11.151gdcmNActionMessages.h File Reference 1120
11.152gdcmNCreateMessages.h File Reference 1121
11.153gdcmNDeleteMessages.h File Reference 1122
11.154gdcmNestedModuleEntries.h File Reference 1122
11.155gdcmNetworkEvents.h File Reference 1124
11.156gdcmNetworkStateID.h File Reference 1125
11.157gdcmNEventReportMessages.h File Reference 1126
11.158gdcmNGetMessages.h File Reference 1126
11.159gdcmNormalizedMessageFactory.h File Reference 1127
11.160gdcmNormalizedNetworkFunctions.h File Reference 1128
11.161gdcmNSetMessages.h File Reference 1129
11.162gdcmObject.h File Reference 1129
11.163gdcmOpenSSLCryptoFactory.h File Reference 1131
11.164gdcmOpenSSLCryptographicMessageSyntax.h File Reference 1131
11.165gdcmOpenSSLP7CryptoFactory.h File Reference 1133
11.166gdcmOpenSSLP7CryptographicMessageSyntax.h File Reference 1133
11.167gdcmOrientation.h File Reference 1135
11.168gdcmOverlay.h File Reference 1135

11.168gdcmParseException.h File Reference . . . 1136
11.169gdcmParser.h File Reference . . . 1138
11.170gdcmPatient.h File Reference . . . 1138
11.171gdcmPDataTFPDU.h File Reference . . . 1139
11.172gdcmPDBElement.h File Reference . . . 1140
11.173gdcmPDBHeader.h File Reference . . . 1141
11.174gdcmPDFCodec.h File Reference . . . 1142
11.175gdcmPDUFactory.h File Reference . . . 1143
11.176gdcmPersonName.h File Reference . . . 1143
11.177gdcmPGXCodec.h File Reference . . . 1144
11.178gdcmPhotometricInterpretation.h File Reference . . . 1145
11.179gdcmPixelFormat.h File Reference . . . 1146
11.180gdcmPixmap.h File Reference . . . 1147
11.181gdcmPixmapReader.h File Reference . . . 1148
11.182gdcmPixmapToPixmapFilter.h File Reference . . . 1149
11.183gdcmPixmapWriter.h File Reference . . . 1150
11.184gdcmPNMCodec.h File Reference . . . 1151
11.185gdcmPreamble.h File Reference . . . 1151
11.186gdcmPresentationContext.h File Reference . . . 1153
11.187gdcmPresentationContextAC.h File Reference . . . 1154
11.188gdcmPresentationContextGenerator.h File Reference . . . 1155
11.189gdcmPresentationContextRQ.h File Reference . . . 1155
11.190gdcmPresentationDataValue.h File Reference . . . 1156
11.191gdcmPrinter.h File Reference . . . 1157
11.192gdcmPrivateTag.h File Reference . . . 1159
11.193gdcmProgressEvent.h File Reference . . . 1160
11.194gdcmPVRGCodec.h File Reference . . . 1161
11.195gdcmPythonFilter.h File Reference . . . 1161

11.197gdcmQueryBase.h File Reference 1162
11.198gdcmQueryFactory.h File Reference 1163
11.199gdcmQueryImage.h File Reference 1164
11.200gdcmQueryPatient.h File Reference 1165
11.201gdcmQuerySeries.h File Reference 1166
11.202gdcmQueryStudy.h File Reference 1167
11.203gdcmRAWCodec.h File Reference 1168
11.204gdcmReader.h File Reference 1168
11.205gdcmRegion.h File Reference 1170
11.206gdcmRescaler.h File Reference 1171
11.207gdcmRLECodec.h File Reference 1172
11.208gdcmRoleSelectionSub.h File Reference 1172
11.209gdcmScanner.h File Reference 1173
11.210gdcmSegment.h File Reference 1174
11.211gdcmSegmentedPaletteColorLookupTable.h File Reference 1175
11.212gdcmSegmentHelper.h File Reference 1176
11.213gdcmSegmentReader.h File Reference 1177
11.214gdcmSegmentWriter.h File Reference 1179
11.215gdcmSequenceOfFragments.h File Reference 1180
11.216gdcmSequenceOfItems.h File Reference 1180
11.217gdcmSerieHelper.h File Reference 1181
11.218gdcmSeries.h File Reference 1183
11.219gdcmServiceClassApplicationInformation.h File Reference 1184
11.220gdcmServiceClassUser.h File Reference 1185
11.221gdcmSHA1.h File Reference 1185
11.222gdcmSimpleSubjectWatcher.h File Reference 1186
11.223gdcmSmartPointer.h File Reference 1187
11.224gdcmSOPClassExtendedNegociationSub.h File Reference 1188

11.225 gdcmSOPClassUIDToIOD.h File Reference . . . 1189
11.226 gdcmSorter.h File Reference . . . 1190
11.227 gdcmSpacing.h File Reference . . . 1192
11.228 gdcmSpectroscopy.h File Reference . . . 1192
11.229 gdcmSplitMosaicFilter.h File Reference . . . 1193
11.230 gdcmStaticAssert.h File Reference . . . 1193
11.230.1 Macro Definition Documentation . . . 1194
11.230.1.1 GDCM_DO_JOIN . . . 1194
11.230.1.2 GDCM_DO_JOIN2 . . . 1194
11.230.1.3 GDCM_JOIN . . . 1194
11.230.1.4 GDCM_STATIC_ASSERT . . . 1194
11.231 gdcmStreamImageReader.h File Reference . . . 1195
11.232 gdcmStreamImageWriter.h File Reference . . . 1195
11.233 gdcmStrictScanner.h File Reference . . . 1196
11.234 gdcmString.h File Reference . . . 1197
11.235 gdcmStringFilter.h File Reference . . . 1199
11.236 gdcmStudy.h File Reference . . . 1199
11.237 gdcmSubject.h File Reference . . . 1201
11.238 gdcmSurface.h File Reference . . . 1202
11.239 gdcmSurfaceHelper.h File Reference . . . 1203
11.240 gdcmSurfaceReader.h File Reference . . . 1203
11.241 gdcmSurfaceWriter.h File Reference . . . 1204
11.242 gdcmSwapCode.h File Reference . . . 1205
11.243 gdcmSwapper.h File Reference . . . 1206
11.244 gdcmSystem.h File Reference . . . 1207
11.245 gdcmTable.h File Reference . . . 1208
11.246 gdcmTableEntry.h File Reference . . . 1209
11.247 gdcmTableReader.h File Reference . . . 1210

11.248gdcmTag.h File Reference . . . 1211
11.249gdcmTagPath.h File Reference . . . 1212
11.250gdcmTagToVR.h File Reference . . . 1212
11.251gdcmTerminal.h File Reference . . . 1213
11.252gdcmTestDriver.h File Reference . . . 1214
11.253gdcmTesting.h File Reference . . . 1215
11.254gdcmTrace.h File Reference . . . 1216
11.254.1Macro Definition Documentation . . . 1217
11.254.1.1GDCM_FUNCTION . . . 1217
11.254.1.2gdcmAssertAlwaysMacro . . . 1217
11.254.1.3gdcmAssertMacro . . . 1217
11.254.1.4gdcmDebugMacro . . . 1218
11.254.1.5gdcmErrorMacro . . . 1218
11.254.1.6gdcmWarningMacro . . . 1219
11.255gdcmTransferSyntax.h File Reference . . . 1219
11.256gdcmTransferSyntaxSub.h File Reference . . . 1221
11.257gdcmType.h File Reference . . . 1222
11.258gdcmTypes.h File Reference . . . 1223
11.259gdcmUIDGenerator.h File Reference . . . 1223
11.260gdcmUIDs.h File Reference . . . 1224
11.261gdcmULAction.h File Reference . . . 1225
11.262gdcmULActionAA.h File Reference . . . 1226
11.263gdcmULActionAE.h File Reference . . . 1226
11.264gdcmULActionAR.h File Reference . . . 1227
11.265gdcmULActionDT.h File Reference . . . 1228
11.266gdcmULBasicCallback.h File Reference . . . 1229
11.267gdcmULConnection.h File Reference . . . 1229
11.268gdcmULConnectionCallback.h File Reference . . . 1230

11.268gdcmULConnectionInfo.h File Reference . . . 1231
11.269gdcmULConnectionManager.h File Reference . . . 1233
11.270gdcmULEvent.h File Reference . . . 1233
11.271gdcmULTransitionTable.h File Reference . . . 1235
11.272gdcmULWritingCallback.h File Reference . . . 1236
11.273gdcmUNExplicitDataElement.h File Reference . . . 1236
11.274gdcmUNExplicitImplicitDataElement.h File Reference . . . 1237
11.275gdcmUnpacker12Bits.h File Reference . . . 1238
11.276gdcmUsage.h File Reference . . . 1238
11.277gdcmUserInformation.h File Reference . . . 1240
11.278gdcmUUIDGenerator.h File Reference . . . 1241
11.279gdcmValidate.h File Reference . . . 1241
11.280gdcmValue.h File Reference . . . 1242
11.281gdcmValueIO.h File Reference . . . 1243
11.282gdcmVersion.h File Reference . . . 1243
11.283gdcmVL.h File Reference . . . 1244
11.284gdcmVM.h File Reference . . . 1245
11.284.1Macro Definition Documentation . . . 1246
11.284.1.1TYPETOLENGTH . . . 1246
11.285gdcmVR.h File Reference . . . 1246
11.285.1Macro Definition Documentation . . . 1248
11.285.1.1TYPETOENCODING . . . 1248
11.285.1.2VRTypeTemplateCase . . . 1248
11.286gdcmVR16ExplicitDataElement.h File Reference . . . 1249
11.287gdcmWaveform.h File Reference . . . 1249
11.288gdcmWin32.h File Reference . . . 1250
11.288.1Macro Definition Documentation . . . 1250
11.288.1.1GDCM_EXPORT . . . 1250

11.290gdcmWLMFindQuery.h File Reference . . . 1250
11.291gdcmWriter.h File Reference . . . 1251
11.292gdcmXMLDictReader.h File Reference . . . 1252
11.293gdcmXMLPrinter.h File Reference . . . 1253
11.294gdcmXMLPrivateDictReader.h File Reference . . . 1253
11.295README.txt File Reference . . . 1254
11.296TestsList.txt File Reference . . . 1254
11.297vtkGDCMImageReader.h File Reference . . . 1254
11.297.1Macro Definition Documentation . . . 1255
11.297.1.1VTK_CMYK . . . 1255
11.297.1.2VTK_INVERSE_LUMINANCE . . . 1255
11.297.1.3VTK_LOOKUP_TABLE . . . 1255
11.297.1.4VTK_YBR . . . 1255
11.298vtkGDCMImageReader2.h File Reference . . . 1255
11.298.1Macro Definition Documentation . . . 1256
11.298.1.1VTK_CMYK . . . 1256
11.298.1.2VTK_INVERSE_LUMINANCE . . . 1256
11.298.1.3VTK_LOOKUP_TABLE . . . 1256
11.298.1.4VTK_YBR . . . 1256
11.299vtkGDCMImageWriter.h File Reference . . . 1256
11.300vtkGDCMMedicalImageProperties.h File Reference . . . 1257
11.301vtkGDCMPolyDataReader.h File Reference . . . 1257
11.302vtkGDCMPolyDataWriter.h File Reference . . . 1258
11.303vtkGDCMTesting.h File Reference . . . 1259
11.304vtkGDCMThreadedImageReader.h File Reference . . . 1259
11.305vtkGDCMThreadedImageReader2.h File Reference . . . 1260
11.306vtkImageColorViewer.h File Reference . . . 1260
11.307vtkImageMapToColors16.h File Reference . . . 1261
11.308vtkImageMapToWindowLevelColors2.h File Reference . . . 1261
11.309vtkImagePlanarComponentsToComponents.h File Reference . . . 1262
11.310vtkImageRGBToYBR.h File Reference . . . 1262
11.311vtkImageYBRToRGB.h File Reference . . . 1263
11.312vtkLookupTable16.h File Reference . . . 1263
11.313vtkRTStructSetProperties.h File Reference . . . 1264

12 Example Documentation 1265

12.1 AWTMedical3.java . . . 1265

12.2 BasicAnonymizer.cs . . . 1269

12.3 BasicImageAnonymizer.cs . . . 1270

12.4 CastConvertPhilips.py . . . 1272

12.5 ChangePrivateTags.cxx . . . 1274

12.6 ChangeSequenceUltrasound.cxx . . . 1275

12.7 CheckBigEndianBug.cxx . . . 1276

12.8 ClinicalTrialAnnotate.cxx . . . 1278

12.9 ClinicalTrialIdentificationWorkflow.cs . . . 1279

12.10 CompressImage.cxx . . . 1282

12.11 CompressLossyJPEG.cs . . . 1283

12.12 Compute3DSpacing.cxx . . . 1284

12.13 Convert16BitsTo8Bits.cxx . . . 1286

12.14 ConvertMPL.py . . . 1286

12.15 ConvertMultiFrameToSingleFrame.cxx . . . 1288

12.16 ConvertNumpy.py . . . 1289

12.17 ConvertPIL.py . . . 1290

12.18 ConvertRGBToLuminance.cxx . . . 1291

12.19 ConvertSingleBitTo8Bits.cxx . . . 1292

12.20 ConvertToQImage.cxx . . . 1293

12.21 CreateARGBImage.cxx . . . 1295

12.22 CreateCMYKImage.cxx . . . 1296

12.23 CreateFakePET.cxx . . . 1297

12.24 CreateFakeRTDOSE.cxx . . . 1299

12.25 CreateJPIPDataSet.cxx . . . 1300

12.26 CreateRAWStorage.py . . . 1301

12.27 csa2img.cxx . . . 1304

12.28CStoreQtProgress.cxx . . . 1306
12.29DecompressImage.cs . . . 1308
12.30DecompressImage.java . . . 1309
12.31DecompressImage.py . . . 1310
12.32DecompressImageMultiframe.cs . . . 1311
12.33DecompressJPEGFile.cs . . . 1313
12.34DecompressPixmap.java . . . 1314
12.35DiffFile.cxx . . . 1315
12.36DiscriminateVolume.cxx . . . 1316
12.37DumbAnonymizer.py . . . 1320
12.38DumpADAC.cxx . . . 1321
12.39DumpExamCard.cxx . . . 1326
12.40DumpGEMSMovieGroup.cxx . . . 1333
12.41DumpImageHeaderInfo.cxx . . . 1339
12.42DumpPhilipsECHO.cxx . . . 1341
12.43DumpToshibaDTI.cxx . . . 1347
12.44DumpToSQLITE3.cxx . . . 1348
12.45DuplicatePCDE.cxx . . . 1350
12.46ELSCINT1WaveToText.cxx . . . 1353
12.47EncapsulateFileInRawData.cxx . . . 1354
12.48ExtractEncapsulatedFile.cs . . . 1355
12.49ExtractEncryptedContent.cxx . . . 1357
12.50ExtractIconFromFile.cxx . . . 1358
12.51ExtractImageRegion.cs . . . 1359
12.52ExtractImageRegion.java . . . 1360
12.53ExtractImageRegionWithLUT.cs . . . 1361
12.54Extracting_All_Resolution.cxx . . . 1363
12.55ExtractOneFrame.cs . . . 1369

12.56Fake_Image_Using_Stream_Image_Writer.cxx . . . 1370
12.57FileAnonymize.cs . . . 1373
12.58FileAnonymize.java . . . 1374
12.59FileChangeTS.cs . . . 1374
12.60FileChangeTSLossy.cs . . . 1377
12.61FileStreaming.cs . . . 1379
12.62FindAllPatientName.py . . . 1380
12.63FixBrokenJ2K.cxx . . . 1381
12.64FixCommaBug.py . . . 1383
12.65FixJAIBugJPEGLS.cxx . . . 1384
12.66FixOrientation.cxx . . . 1387
12.67gdcmorthoplanes.cxx . . . 1388
12.68gdcmreslice.cxx . . . 1394
12.69gdcmrtionplan.cxx . . . 1396
12.70gdcmrtplan.cxx . . . 1400
12.71gdcmscene.cxx . . . 1404
12.72gdcmtexture.cxx . . . 1406
12.73gdcmvolume.cxx . . . 1408
12.74GenAllVR.cxx . . . 1409
12.75GenerateDICOMDIR.cs . . . 1412
12.76GenerateRTSTRUCT.cxx . . . 1412
12.77GenerateStandardSOPClasses.cxx . . . 1415
12.78GenFakeIdentifyFile.cxx . . . 1416
12.79GenFakeImage.cxx . . . 1419
12.80GenLongSeqs.cxx . . . 1420
12.81GenSeqs.cxx . . . 1422
12.82GetArray.cs . . . 1423
12.83GetJPEGSamplePrecision.cxx . . . 1424

12.84GetPortionCSAHeader.py . . . 1426
12.85GetSequenceUltrasound.cxx . . . 1427
12.86GetSubSequenceData.cxx . . . 1429
12.87headsq2dcm.py . . . 1432
12.88HelloActiviz.cs . . . 1432
12.89HelloActiviz2.cs . . . 1434
12.90HelloActiviz3.cs . . . 1435
12.91HelloActiviz4.cs . . . 1436
12.92HelloActiviz5.cs . . . 1436
12.93HelloSimple.java . . . 1438
12.94HelloVizWorld.cxx . . . 1438
12.95HelloVTKWorld.cs . . . 1439
12.96HelloVTKWorld.java . . . 1440
12.97HelloVTKWorld2.cs . . . 1442
12.98HelloWorld.cxx . . . 1442
12.99HelloWorld.py . . . 1443
12.100J22tomultisc.cxx . . . 1444
12.101LargeVRDSExplicit.cxx . . . 1446
12.102MagnifyFile.cxx . . . 1448
12.103MakeTemplate.cxx . . . 1449
12.104ManipulateFile.cs . . . 1450
12.105ManipulateFile.py . . . 1451
12.106ManipulateSequence.py . . . 1452
12.107MergeFile.py . . . 1453
12.108MergeTwoFiles.cxx . . . 1454
12.109MetaImageMD5Activiz.cs . . . 1455
12.110MIPViewer.java . . . 1457
12.111MpegVideoInfo.cs . . . 1459

12.112MPRViewer.java . . . 1464
12.113MPRViewer2.java . . . 1466
12.114MrProtocol.cxx . . . 1470
12.115NewSequence.cs . . . 1477
12.116NewSequence.py . . . 1478
12.117offscreenimage.cxx . . . 1479
12.118PatchFile.cxx . . . 1480
12.119PhilipsPrivateRescaleInterceptSlope.py . . . 1482
12.120PlaySound.py . . . 1483
12.121pmsct_rgb1.cxx . . . 1484
12.122PrivateDict.py . . . 1487
12.123PublicDict.cxx . . . 1488
12.124QIDO-RS.cxx . . . 1489
12.125ReadAndDumpDICOMDIR.cxx . . . 1490
12.126ReadAndDumpDICOMDIR.py . . . 1493
12.127ReadAndPrintAttributes.cxx . . . 1495
12.128ReadExplicitLengthSQIVR.cxx . . . 1497
12.129ReadFiles.java . . . 1497
12.130ReadGEMSSDO.cxx . . . 1499
12.131ReadMultiTimesException.cxx . . . 1501
12.132ReadSeriesIntoVTK.java . . . 1502
12.133ReadUTF8QtDir.cxx . . . 1503
12.134RefCounting.cs . . . 1505
12.135ReformatFile.cs . . . 1505
12.136RemovePrivateTags.py . . . 1507
12.137RescaleImage.cs . . . 1507
12.138reslicesphere.cxx . . . 1508
12.139ReWriteSCAsMR.py . . . 1516

12.140de2img.cxx . . . 1517
12.141ltstructapp.cxx . . . 1520
12.142ScanDirectory.cs . . . 1522
12.143ScanDirectory.java . . . 1523
12.144ScanDirectory.py . . . 1527
12.145SendFileSCU.cs . . . 1527
12.146SimplePrint.cs . . . 1528
12.147SimplePrintPatientName.cs . . . 1529
12.148SimpleScanner.cxx . . . 1530
12.149SortImage.cxx . . . 1532
12.150SortImage.py . . . 1533
12.151SortImage2.cs . . . 1534
12.152StandardizeFiles.cs . . . 1535
12.153StreamImageReaderTest.cxx . . . 1536
12.154TestByteSwap.cxx . . . 1540
12.155TestReader.cxx . . . 1542
12.156TestReader.py . . . 1543
12.157threadgdcm.cxx . . . 1544
12.158TraverseModules.cxx . . . 1547
12.159uid_unique.cxx . . . 1549
12.160VolumeSorter.cxx . . . 1549
12.161WriteBuffer.py . . . 1552

Index **1555**

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-2.6.3.pdf`

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

`http://gdcm.sourceforge.net/2.6/gdcm-2.6.3-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 45
gdcm::network 69
gdcm::SegmentHelper 76
gdcm::terminal
Class for Terminal Allow one to print in color in a shell 76

Chapter 6

Hierarchical Index

6.1 Class Hierarchy

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

gdcm::network::AbstractSyntax . . . 91
gdcm::network::ApplicationContext . . . 102
gdcm::ApplicationEntity . . . 104
gdcm::network::ARTIMTimer . . . 109
gdcm::ASN1 . . . 111
gdcm::network::AsynchronousOperationsWindowSub . . . 112
gdcm::Attribute< Group, Element, TVR, TVM > . . . 113
gdcm::Attribute< Group, Element, TVR, VM::VM1 > . . . 118
gdcm::Attribute< Group, Element, TVR, VM::VM1_n > . . . 125
gdcm::Attribute< Group, Element, TVR, VM::VM1_3 > . . . 123
gdcm::Attribute< Group, Element, TVR, VM::VM1_8 > . . . 124
gdcm::Attribute< Group, Element, TVR, VM::VM2_n > . . . 130
gdcm::Attribute< Group, Element, TVR, VM::VM2_2n > . . . 129
gdcm::Attribute< Group, Element, TVR, VM::VM3_n > . . . 134
gdcm::Attribute< Group, Element, TVR, VM::VM3_3n > . . . 132
gdcm::Base64 . . . 137
gdcm::network::BaseCompositeMessage . . . 139
gdcm::network::CEchoRQ . . . 182
gdcm::network::CEchoRSP . . . 184
gdcm::network::CFindCancelRQ . . . 185
gdcm::network::CFindRQ . . . 186
gdcm::network::CFindRSP . . . 188
gdcm::network::CMoveCancelRq . . . 189
gdcm::network::CMoveRQ . . . 190
gdcm::network::CMoveRSP . . . 192
gdcm::network::CStoreRQ . . . 227
gdcm::network::CStoreRSP . . . 229
gdcm::network::BaseNormalizedMessage . . . 140
gdcm::network::NActionRQ . . . 517
gdcm::network::NActionRSP . . . 518

gdcm::network::NCreateRQ . . . 519
gdcm::network::NCreateRSP . . . 521
gdcm::network::NDeleteRQ . . . 522
gdcm::network::NDeleteRSP . . . 524
gdcm::network::NEventReportRQ . . . 528
gdcm::network::NEventReportRSP . . . 529
gdcm::network::NGetRQ . . . 530
gdcm::network::NGetRSP . . . 532
gdcm::network::NSetRQ . . . 536
gdcm::network::NSetRSP . . . 538
gdcm::network::BasePDU . . . 143
gdcm::network::AAbortPDU . . . 79
gdcm::network::AAssociateACPDU . . . 81
gdcm::network::AAssociateRJPDU . . . 84
gdcm::network::AAssociateRQPDU . . . 86
gdcm::network::AReleaseRPPDU . . . 106
gdcm::network::AReleaseRQPDU . . . 108
gdcm::network::PDataTFPDU . . . 563
std::basic_string< Char >
std::string
gdcm::String< TDelimiter, TMaxLength, TPadChar > . . . 736
gdcm::SegmentHelper::BasicCodedEntry . . . 152
gdcm::BitmapToBitmapFilter . . . 165
gdcm::PixmapToPixmapFilter . . . 592
gdcm::ImageToImageFilter . . . 419
gdcm::ImageApplyLookupTable . . . 385
gdcm::ImageChangePhotometricInterpretation . . . 387
gdcm::ImageChangePlanarConfiguration . . . 390
gdcm::ImageChangeTransferSyntax . . . 393
gdcm::ImageFragmentSplitter . . . 405
gdcm::ByteBuffer . . . 170
gdcm::ByteSwap< T > . . . 171
gdcm::ByteSwapFilter . . . 173
gdcm::network::CFind . . . 185
gdcm::Coder . . . 194
gdcm::Codec . . . 193
gdcm::AudioCodec . . . 135
gdcm::ImageCodec . . . 397
gdcm::DeltaEncodingCodec . . . 262
gdcm::JPEG2000Codec . . . 448
gdcm::JPEGCodec . . . 455
gdcm::JPEG12Codec . . . 444
gdcm::JPEG16Codec . . . 446
gdcm::JPEG8Codec . . . 453
gdcm::JPEGLSCodec . . . 461
gdcm::KAKADUCodec . . . 466
gdcm::PGXCodec . . . 575
gdcm::PNMCodec . . . 598
gdcm::PVRGCodec . . . 623
gdcm::RAWCodec . . . 638
gdcm::RLECodec . . . 653
gdcm::PDFCodec . . . 570
gdcm::CodeString . . . 195

gdcm::network::CompositeMessageFactory . . . 202
gdcm::CompositeNetworkFunctions . . . 203
gdcm::ConstCharWrapper . . . 206
gdcm::CryptoFactory . . . 209
gdcm::CAPICryptoFactory . . . 179
gdcm::OpenSSLCryptoFactory . . . 541
gdcm::OpenSSLP7CryptoFactory . . . 545
gdcm::CryptographicMessageSyntax . . . 211
gdcm::CAPICryptographicMessageSyntax . . . 180
gdcm::OpenSSLCryptographicMessageSyntax . . . 543
gdcm::OpenSSLP7CryptographicMessageSyntax . . . 546
gdcm::CSAElement . . . 213
gdcm::CSAHeader . . . 218
gdcm::CSAHeaderDict . . . 223
gdcm::CSAHeaderDictEntry . . . 225
gdcm::DataElement . . . 233
gdcm::CP246ExplicitDataElement . . . 207
gdcm::ExplicitDataElement . . . 320
gdcm::ExplicitImplicitDataElement . . . 322
gdcm::Fragment . . . 367
gdcm::BasicOffsetTable . . . 154
gdcm::ImplicitDataElement . . . 426
gdcm::Item . . . 439
gdcm::UNExplicitDataElement . . . 872
gdcm::UNExplicitImplicitDataElement . . . 874
gdcm::VR16ExplicitDataElement . . . 900
gdcm::DataSet . . . 246
gdcm::CommandDataSet . . . 200
gdcm::FileMetaInformation . . . 342
gdcm::DataSetHelper . . . 256
gdcm::Decoder . . . 257
gdcm::Codec . . . 193
gdcm::DefinedTerms . . . 258
gdcm::Defs . . . 259
gdcm::DICOMDIR . . . 263
gdcm::DICOMDIRGenerator . . . 264
gdcm::Dict . . . 267
gdcm::DictConverter . . . 269
gdcm::DictEntry . . . 272
gdcm::Dicts . . . 276
gdcm::network::DIMSE . . . 279
gdcm::DirectionCosines . . . 281
gdcm::Directory . . . 283
gdcm::DirectoryHelper . . . 286
gdcm::DummyValueGenerator . . . 288
gdcm::Element< TVR, TVM > . . . 290
gdcm::Element< TVR, VM::VM1_n > . . . 295
gdcm::Element< TVR, VM::VM1_2 > . . . 294
gdcm::Element< TVR, VM::VM2_n > . . . 300
gdcm::Element< TVR, VM::VM2_2n > . . . 298
gdcm::Element< TVR, VM::VM3_n > . . . 303
gdcm::Element< TVR, VM::VM3_3n > . . . 301

gdcm::Element< VR::AS, VM::VM5 > . . . 304
gdcm::Element< VR::OB, VM::VM1_n > . . . 290
gdcm::Element< VR::OB, VM::VM1 > . . . 305
gdcm::Element< VR::OW, VM::VM1_n > . . . 290
gdcm::Element< VR::OW, VM::VM1 > . . . 306
gdcm::ElementDisableCombinations< TVR, TVM > . . . 308
gdcm::ElementDisableCombinations< VR::OB, VM::VM1_n > . . . 309
gdcm::ElementDisableCombinations< VR::OW, VM::VM1_n > . . . 309
gdcm::EncapsulatedDocument . . . 309
gdcm::EncodingImplementation< T > . . . 310
gdcm::EncodingImplementation< VR::VRASCII > . . . 310
gdcm::EncodingImplementation< VR::VRBINARY > . . . 312
gdcm::EnumeratedValues . . . 314
gdcm::Event . . . 314
gdcm::AnyEvent . . . 100
gdcm::AbortEvent . . . 90
gdcm::AnonymizeEvent . . . 93
gdcm::DataEvent . . . 244
gdcm::DataSetEvent . . . 254
gdcm::EndEvent . . . 312
gdcm::ExitEvent . . . 319
gdcm::FileNameEvent . . . 350
gdcm::InitializeEvent . . . 428
gdcm::IterationEvent . . . 443
gdcm::ModifiedEvent . . . 504
gdcm::ProgressEvent . . . 620
gdcm::StartEvent . . . 719
gdcm::UserEvent . . . 879
gdcm::NoEvent . . . 533
std::exception
gdcm::CSAHeaderDictException . . . 227
gdcm::DataElementException . . . 243
gdcm::Exception . . . 317
gdcm::ParseException . . . 558
gdcm::Fiducials . . . 324
gdcm::FileDerivation . . . 337
gdcm::FileExplicitFilter . . . 339
gdcm::Filename . . . 347
gdcm::FilenameGenerator . . . 353
gdcm::FileSet . . . 355
gdcm::Global . . . 370
gdcm::GroupDict . . . 373
gdcm::IconImageFilter . . . 375
gdcm::IconImageGenerator . . . 377
gdcm::ignore_char . . . 380
gdcm::ImageConverter . . . 404
gdcm::ImageHelper . . . 407
gdcm::network::ImplementationClassUIDSub . . . 424
gdcm::network::ImplementationUIDSub . . . 425
gdcm::network::ImplementationVersionNameSub . . . 425
gdcm::IOD . . . 429
gdcm::IODEntry . . . 431
gdcm::IODs . . . 433

gdcm::JSON . . . 465
gdcm::Scanner::ltstr . . . 475
gdcm::StrictScanner::ltstr . . . 476
gdcm::Macro . . . 476
gdcm::Macros . . . 478
gdcm::network::MaximumLengthSub . . . 480
gdcm::MD5 . . . 481
gdcm::MediaStorage . . . 482
gdcm::Module . . . 505
gdcm::ModuleEntry . . . 507
  gdcm::NestedModuleEntries . . . 525
gdcm::Modules . . . 510
gdcm::network::NormalizedMessageFactory . . . 534
gdcm::NormalizedNetworkFunctions . . . 535
gdcm::Object . . . 539
  gdcm::BaseQuery . . . 145
    gdcm::BaseRootQuery . . . 149
      gdcm::FindPatientRootQuery . . . 363
      gdcm::FindStudyRootQuery . . . 365
      gdcm::MovePatientRootQuery . . . 512
      gdcm::MoveStudyRootQuery . . . 515
      gdcm::WLMFindQuery . . . 966
    gdcm::ModalityPerformedProcedureStepCreateQuery . . . 499
    gdcm::ModalityPerformedProcedureStepSetQuery . . . 501
  gdcm::Bitmap . . . 156
    gdcm::Pixmap . . . 586
      gdcm::Image . . . 381
  gdcm::Curve . . . 230
  gdcm::File . . . 325
    gdcm::FileWithName . . . 361
  gdcm::LookupTable . . . 471
    gdcm::SegmentedPaletteColorLookupTable . . . 671
  gdcm::MeshPrimitive . . . 495
  gdcm::Overlay . . . 551
  gdcm::Segment . . . 666
  gdcm::Subject . . . 743
    gdcm::Anonymizer . . . 95
    gdcm::Command . . . 198
      gdcm::MemberCommand< T > . . . 491
      gdcm::SimpleMemberCommand< T > . . . 701
    gdcm::FileAnonymizer . . . 328
    gdcm::FileChangeTransferSyntax . . . 332
    gdcm::FileDecompressLookupTable . . . 334
    gdcm::FileStreamer . . . 357
    gdcm::network::ULConnectionManager . . . 864
    gdcm::Scanner . . . 659
    gdcm::ServiceClassUser . . . 694
    gdcm::StrictScanner . . . 730
  gdcm::Surface . . . 746
  gdcm::Value . . . 884
    gdcm::ByteValue . . . 174
    gdcm::SequenceOfFragments . . . 678
    gdcm::SequenceOfItems . . . 683

gdcm::Orientation . . . 549
gdcm::Parser . . . 560
gdcm::Patient . . . 562
gdcm::PDBElement . . . 565
gdcm::PDBHeader . . . 567
gdcm::network::PDUFactory . . . 572
gdcm::PersonName . . . 574
gdcm::PhotometricInterpretation . . . 577
gdcm::PixelFormat . . . 580
gdcm::Preamble . . . 600
gdcm::PresentationContext . . . 602
gdcm::network::PresentationContextAC . . . 604
gdcm::PresentationContextGenerator . . . 605
gdcm::network::PresentationContextRQ . . . 608
gdcm::network::PresentationDataValue . . . 610
gdcm::Printer . . . 612
  gdcm::DictPrinter . . . 275
  gdcm::Dumper . . . 288
gdcm::PrivateDict . . . 616
gdcm::PythonFilter . . . 625
gdcm::QueryBase . . . 626
  gdcm::QueryImage . . . 630
  gdcm::QueryPatient . . . 632
  gdcm::QuerySeries . . . 634
  gdcm::QueryStudy . . . 636
gdcm::QueryFactory . . . 629
gdcm::Reader . . . 641
  gdcm::PixmapReader . . . 589
    gdcm::ImageReader . . . 411
      gdcm::ImageRegionReader . . . 416
  gdcm::SegmentReader . . . 673
    gdcm::SurfaceReader . . . 756
gdcm::RealWorldValueMappingContent . . . 647
gdcm::Region . . . 648
  gdcm::BoxRegion . . . 167
gdcm::Rescaler . . . 650
gdcm::network::RoleSelectionSub . . . 657
gdcm::SerieHelper::Rule . . . 658
gdcm::SerieHelper . . . 690
gdcm::Series . . . 693
gdcm::network::ServiceClassApplicationInformation . . . 693
gdcm::SHA1 . . . 700
gdcm::SimpleSubjectWatcher . . . 704
gdcm::SmartPointer< ObjectType > . . . 706
gdcm::SmartPointer< gdcm::Bitmap > . . . 706
gdcm::SmartPointer< gdcm::File > . . . 706
gdcm::SmartPointer< gdcm::Image > . . . 706
gdcm::SmartPointer< gdcm::MemberCommand > . . . 706
gdcm::SmartPointer< gdcm::MeshPrimitive > . . . 706
gdcm::SmartPointer< gdcm::Pixmap > . . . 706
gdcm::SmartPointer< gdcm::SimpleMemberCommand > . . . 706
gdcm::SmartPointer< gdcm::Subject > . . . 706
gdcm::SmartPointer< LookupTable > . . . 706

gdcm::SmartPointer< Segment > . . . 706
gdcm::SmartPointer< Surface > . . . 706
gdcm::SmartPointer< Value > . . . 706
gdcm::network::SOPClassExtendedNegociationSub . . . 709
gdcm::SOPClassUIDToIOD . . . 710
gdcm::Sorter . . . 712
  gdcm::IPPSorter . . . 435
gdcm::Spacing . . . 715
gdcm::Spectroscopy . . . 717
gdcm::SplitMosaicFilter . . . 718
gdcm::static_assert_test< x > . . . 720
gdcm::STATIC_ASSERTION_FAILURE< x > . . . 720
gdcm::STATIC_ASSERTION_FAILURE< true > . . . 721
gdcm::StreamImageReader . . . 721
gdcm::StreamImageWriter . . . 724
String<'\\', 64 >
  gdcm::LO . . . 468
gdcm::StringFilter . . . 740
gdcm::Study . . . 742
gdcm::SurfaceHelper . . . 753
gdcm::SwapCode . . . 760
gdcm::SwapperDoOp . . . 762
gdcm::SwapperNoOp . . . 763
gdcm::System . . . 763
gdcm::Table . . . 768
gdcm::TableEntry . . . 770
gdcm::TableReader . . . 771
  gdcm::XMLDictReader . . . 974
  gdcm::XMLPrivateDictReader . . . 978
gdcm::network::TableRow . . . 773
gdcm::Tag . . . 774
  gdcm::PrivateTag . . . 618
gdcm::TagPath . . . 781
gdcm::Testing . . . 783
gdcm::Trace . . . 788
gdcm::TransferSyntax . . . 792
gdcm::network::TransferSyntaxSub . . . 797
gdcm::network::Transition . . . 799
gdcm::Type . . . 800
gdcm::UI . . . 802
gdcm::UIDGenerator . . . 803
gdcm::UIDs . . . 805
gdcm::network::ULAction . . . 825
  gdcm::network::ULActionAA1 . . . 828
  gdcm::network::ULActionAA2 . . . 829
  gdcm::network::ULActionAA3 . . . 830
  gdcm::network::ULActionAA4 . . . 831
  gdcm::network::ULActionAA5 . . . 832
  gdcm::network::ULActionAA6 . . . 833
  gdcm::network::ULActionAA7 . . . 834
  gdcm::network::ULActionAA8 . . . 835
  gdcm::network::ULActionAE1 . . . 836
  gdcm::network::ULActionAE2 . . . 837

gdcm::network::ULActionAE3 . . . 838
gdcm::network::ULActionAE4 . . . 839
gdcm::network::ULActionAE5 . . . 840
gdcm::network::ULActionAE6 . . . 841
gdcm::network::ULActionAE7 . . . 842
gdcm::network::ULActionAE8 . . . 843
gdcm::network::ULActionAR1 . . . 844
gdcm::network::ULActionAR10 . . . 845
gdcm::network::ULActionAR2 . . . 846
gdcm::network::ULActionAR3 . . . 847
gdcm::network::ULActionAR4 . . . 848
gdcm::network::ULActionAR5 . . . 849
gdcm::network::ULActionAR6 . . . 850
gdcm::network::ULActionAR7 . . . 851
gdcm::network::ULActionAR8 . . . 852
gdcm::network::ULActionAR9 . . . 853
gdcm::network::ULActionDT1 . . . 854
gdcm::network::ULActionDT2 . . . 855
gdcm::network::ULConnection . . . 858
gdcm::network::ULConnectionCallback . . . 861
gdcm::network::ULBasicCallback . . . 856
gdcm::network::ULWritingCallback . . . 871
gdcm::network::ULConnectionInfo . . . 863
gdcm::network::ULEvent . . . 869
gdcm::network::ULTransitionTable . . . 870
gdcm::Unpacker12Bits . . . 876
gdcm::Usage . . . 877
gdcm::network::UserInformation . . . 880
gdcm::UUIDGenerator . . . 882
gdcm::Validate . . . 883
gdcm::ValueIO< TDE, TSwap, TType > . . . 886
gdcm::Version . . . 887
gdcm::VL . . . 888
gdcm::VM . . . 891
gdcm::VMToLength< T > . . . 895
gdcm::VR . . . 896
gdcm::VRToEncoding< T > . . . 902
gdcm::VRToType< T > . . . 902
gdcm::VRToType< TVR > . . . 902
gdcm::VRVLSize< T > . . . 903
gdcm::VRVLSize< 0 > . . . 903
gdcm::VRVLSize< 1 > . . . 904
vtkImageAlgorithm
vtkImagePlanarComponentsToComponents . . . 952
vtkImageMapToColors
vtkImageMapToWindowLevelColors2 . . . 950
vtkImageWriter
vtkGDCMImageWriter . . . 917
vtkLookupTable
vtkLookupTable16 . . . 957
vtkMedicalImageProperties
vtkGDCMMedicalImageProperties . . . 922
vtkMedicalImageReader2
vtkGDCMImageReader . . . 904

vtkGDCMThreadedImageReader . . . 933
vtkGDCMImageReader2 . . . 911
vtkObject
vtkGDCMTesting . . . 931
vtkImageColorViewer . . . 940
vtkRTStructSetProperties . . . 960
vtkPolyDataAlgorithm
vtkGDCMPolyDataReader . . . 925
vtkPolyDataWriter
vtkGDCMPolyDataWriter . . . 928
vtkThreadedImageAlgorithm
vtkGDCMThreadedImageReader2 . . . 936
vtkImageMapToColors16 . . . 947
vtkImageRGBToYBR . . . 954
vtkImageYBRToRGB . . . 956
gdcm::Waveform . . . 965
gdcm::Writer . . . 969
gdcm::PixmapWriter . . . 594
gdcm::ImageWriter . . . 421
gdcm::SegmentWriter . . . 676
gdcm::SurfaceWriter . . . 758
gdcm::XMLPrinter . . . 976

Chapter 7

Class Index

7.1 Class List

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

gdcm::network::AAbortPDU
AAbortPDU Table 9-26 A-ABORT PDU FIELDS . . . 79
gdcm::network::AAssociateACPDU
AAssociateACPDU Table 9-17 ASSOCIATE-AC PDU fields . . . 81
gdcm::network::AAssociateRJPDU
AAssociateRJPDU Table 9-21 ASSOCIATE-RJ PDU FIELDS . . . 84
gdcm::network::AAssociateRQPDU
AAssociateRQPDU Table 9-11 ASSOCIATE-RQ PDU fields . . . 86
gdcm::AbortEvent . . . 90
gdcm::network::AbstractSyntax
AbstractSyntax Table 9-14 ABSTRACT SYNTAX SUB-ITEM FIELDS . . . 91
gdcm::AnonymizeEvent
AnonymizeEvent Special type of event triggered during the Anonymization process . . . 93
gdcm::Anonymizer
Anonymizer This class is a multi purpose anonymizer. It can work in 2 mode: . . . 95
gdcm::AnyEvent . . . 100
gdcm::network::ApplicationContext
ApplicationContext Table 9-12 APPLICATION CONTEXT ITEM FIELDS . . . 102
gdcm::ApplicationEntity
ApplicationEntity . . . 104
gdcm::network::AReleaseRPPDU
AReleaseRPPDU Table 9-25 A-RELEASE-RP PDU fields . . . 106
gdcm::network::AReleaseRQPDU
AReleaseRQPDU Table 9-24 A-RELEASE-RQ PDU FIELDS . . . 108
gdcm::network::ARTIMTimer
ARTIMTimer This file contains the code for the ARTIM timer . . . 109
gdcm::ASN1
Class for ASN1 . . . 111
gdcm::network::AsynchronousOperationsWindowSub
AsynchronousOperationsWindowSub PS 3.7 Table D.3-7 ASYNCHRONOUS OPERATIONS WIN↩
DOW SUB-ITEM FIELDS (A-ASSOCIATE-RQ) . . . 112

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 . . . 113
gdcm::Attribute< Group, Element, TVR, VM::VM1 > . . . 118
gdcm::Attribute< Group, Element, TVR, VM::VM1_3 > . . . 123
gdcm::Attribute< Group, Element, TVR, VM::VM1_8 > . . . 124
gdcm::Attribute< Group, Element, TVR, VM::VM1_n > . . . 125
gdcm::Attribute< Group, Element, TVR, VM::VM2_2n > . . . 129
gdcm::Attribute< Group, Element, TVR, VM::VM2_n > . . . 130
gdcm::Attribute< Group, Element, TVR, VM::VM3_3n > . . . 132
gdcm::Attribute< Group, Element, TVR, VM::VM3_n > . . . 134
gdcm::AudioCodec
AudioCodec . . . 135
gdcm::Base64
Class for Base64 . . . 137
gdcm::network::BaseCompositeMessage
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 . . . 139
gdcm::network::BaseNormalizedMessage
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 . . . 140
gdcm::network::BasePDU
BasePDU base class for PDUs . . . 143
gdcm::BaseQuery
BaseQuery contains: a baseclass which will produce a dataset for all dimse messages . . . 145
gdcm::BaseRootQuery
BaseRootQuery contains: a baseclass which will produce a dataset for c-find and c-move with patient/study root . . . 149
gdcm::SegmentHelper::BasicCodedEntry
This structure defines a basic coded entry with all of its attributes . . . 152
gdcm::BasicOffsetTable
Class to represent a BasicOffsetTable . . . 154
gdcm::Bitmap
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) . . . 156
gdcm::BitmapToBitmapFilter
BitmapToBitmapFilter class Super class for all filter taking an image and producing an output image . 165
gdcm::BoxRegion
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) . . . 167
gdcm::ByteBuffer
ByteBuffer . . . 170
gdcm::ByteSwap< T >
ByteSwap . . . 171
gdcm::ByteSwapFilter
ByteSwapFilter In place byte-swapping of a dataset FIXME: FL status ?? . . . 173
gdcm::ByteValue
Class to represent binary value (array of bytes) . . . 174
gdcm::CAPICryptoFactory . . . 179
gdcm::CAPICryptographicMessageSyntax . . . 180
gdcm::network::CEchoRQ
CEchoRQ this file defines the messages for the cecho action . . . 182

gdcm::network::CEchoRSP
CEchoRSP this file defines the messages for the cecho action . . . 184
gdcm::network::CFind . . . 185
gdcm::network::CFindCancelRQ
CFindCancelRQ this file defines the messages for the cfind action . . . 185
gdcm::network::CFindRQ
CFindRQ this file defines the messages for the cfind action . . . 186
gdcm::network::CFindRSP
CFindRSP this file defines the messages for the cfind action . . . 188
gdcm::network::CMoveCancelRq . . . 189
gdcm::network::CMoveRQ
CMoveRQ this file defines the messages for the cmove action . . . 190
gdcm::network::CMoveRSP
CMoveRSP this file defines the messages for the cmove action . . . 192
gdcm::Codec
Codec class . . . 193
gdcm::Coder
Coder . . . 194
gdcm::CodeString
CodeString This is an implementation of DICOM VR: CS The cstor will properly Trim so that operator== is correct . . . 195
gdcm::Command
Command superclass for callback/observer methods . . . 198
gdcm::CommandDataSet
Class to represent a Command DataSet . . . 200
gdcm::network::CompositeMessageFactory
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) . . . 202
gdcm::CompositeNetworkFunctions
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: . . . 203
gdcm::ConstCharWrapper
Do not use me . . . 206
gdcm::CP246ExplicitDataElement
Class to read/write a DataElement as CP246Explicit Data Element . . . 207
gdcm::CryptoFactory
Class to do handle the crypto factory . . . 209
gdcm::CryptographicMessageSyntax . . . 211
gdcm::CSAElement
Class to represent a CSA Element . . . 213
gdcm::CSAHeader
Class for CSAHeader . . . 218
gdcm::CSAHeaderDict
Class to represent a map of CSAHeaderDictEntry . . . 223

gdcm::CSAHeaderDictEntry
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 . . . 225
gdcm::CSAHeaderDictException . . . 227
gdcm::network::CStoreRQ
CStoreRQ this file defines the messages for the cecho action . . . 227
gdcm::network::CStoreRSP
CStoreRSP this file defines the messages for the cecho action . . . 229
gdcm::Curve
Curve class to handle element 50xx,3000 Curve Data WARNING: This is deprecated and lastly defined in PS 3.3 - 2004 . . . 230
gdcm::DataElement
Class to represent a Data Element either Implicit or Explicit . . . 233
gdcm::DataElementException . . . 243
gdcm::DataEvent
DataEvent . . . 244
gdcm::DataSet
Class to represent a Data Set (which contains Data Elements) A Data Set represents an instance of a real world Information Object . . . 246
gdcm::DataSetEvent
DataSetEvent Special type of event triggered during the DataSet store/move process . . . 254
gdcm::DataSetHelper
DataSetHelper (internal class, not intended for user level) . . . 256
gdcm::Decoder
Decoder . . . 257
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 . . . 258
gdcm::Defs
FIXME I do not like the name 'Defs' . . . 259
gdcm::DeltaEncodingCodec
DeltaEncodingCodec compression used by some private vendor . . . 262
gdcm::DICOMDIR
DICOMDIR class . . . 263
gdcm::DICOMDIRGenerator
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 . . . 264
gdcm::Dict
Class to represent a map of DictEntry . . . 267
gdcm::DictConverter
Class to convert a .dic file into something else: . . . 269
gdcm::DictEntry
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 . . . 272
gdcm::DictPrinter
DictPrinter class . . . 275

gdcm::Dicts
Class to manipulate the sum of knowledge (all the dict user load) . . . 276
gdcm::network::DIMSE
DIMSE PS 3.7 - 2009 Annex E Command Dictionary (Normative) E.1 REGISTRY OF DICOM CO↩
MMAND ELEMENTS Table E.1-1 COMMAND FIELDS (PART 1) . . . 279
gdcm::DirectionCosines
Class to handle DirectionCosines . . . 281
gdcm::Directory
Class for manipulation directories . . . 283
gdcm::DirectoryHelper
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 . . . 286
gdcm::DummyValueGenerator
Class for generating dummy value . . . 288
gdcm::Dumper
Codec class . . . 288
gdcm::Element< TVR, TVM >
Element class . . . 290
gdcm::Element< TVR, VM::VM1_2 > . . . 294
gdcm::Element< TVR, VM::VM1_n > . . . 295
gdcm::Element< TVR, VM::VM2_2n > . . . 298
gdcm::Element< TVR, VM::VM2_n > . . . 300
gdcm::Element< TVR, VM::VM3_3n > . . . 301
gdcm::Element< TVR, VM::VM3_n > . . . 303
gdcm::Element< VR::AS, VM::VM5 > . . . 304
gdcm::Element< VR::OB, VM::VM1 > . . . 305
gdcm::Element< VR::OW, VM::VM1 > . . . 306
gdcm::ElementDisableCombinations< TVR, TVM >
A class which is used to produce compile errors for an invalid combination of template parameters . 308
gdcm::ElementDisableCombinations< VR::OB, VM::VM1_n > . . . 309
gdcm::ElementDisableCombinations< VR::OW, VM::VM1_n > . . . 309
gdcm::EncapsulatedDocument
EncapsulatedDocument . . . 309
gdcm::EncodingImplementation< T >
EncodingImplementation . . . 310
gdcm::EncodingImplementation< VR::VRASCII > . . . 310
gdcm::EncodingImplementation< VR::VRBINARY > . . . 312
gdcm::EndEvent . . . 312
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: . . . 314
gdcm::Event
Superclass for callback/observer methods . . . 314
gdcm::Exception
Exception . . . 317
gdcm::ExitEvent . . . 319
gdcm::ExplicitDataElement
Class to read/write a DataElement as Explicit Data Element . . . 320

gdcm::ExplicitImplicitDataElement
Class to read/write a DataElement as ExplicitImplicit Data Element . . . 322
gdcm::Fiducials
Fiducials . . . 324
gdcm::File
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 . . . 325
gdcm::FileAnonymizer
FileAnonymizer . . . 328
gdcm::FileChangeTransferSyntax
FileChangeTransferSyntax . . . 332
gdcm::FileDecompressLookupTable
FileDecompressLookupTable class It decompress the segmented LUT into linearized one (only P↩ ALETTE_COLOR images) Output will be a PhotometricInterpretation=RGB image . . . 334
gdcm::FileDerivation
FileDerivation class See PS 3.16 - 2008 For the list of Code Value that can be used for in Derivation Code Sequence . . . 337
gdcm::FileExplicitFilter
FileExplicitFilter class After changing a file from Implicit to Explicit representation (see Image↩ ChangeTransferSyntax) 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 . . . 339
gdcm::FileMetaInformation
Class to represent a File Meta Information . . . 342
gdcm::Filename
Class to manipulate file name's . . . 347
gdcm::FileNameEvent
FileNameEvent Special type of event triggered during processing of FileSet . . . 350
gdcm::FilenameGenerator
FilenameGenerator . . . 353
gdcm::FileSet
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 . . . 355
gdcm::FileStreamer
FileStreamer This class let a user create a massive DICOM DataSet from a template DICOM file, by appending chunks of data . . . 357
gdcm::FileWithName
FileWithName . . . 361
gdcm::FindPatientRootQuery
PatientRootQuery contains: the class which will produce a dataset for c-find with patient root . . . 363
gdcm::FindStudyRootQuery
FindStudyRootQuery contains: the class which will produce a dataset for C-FIND with study root . . . 365
gdcm::Fragment
Class to represent a Fragment . . . 367
gdcm::Global
Global . . . 370
gdcm::GroupDict
Class to represent the mapping from group number to its abbreviation and name . . . 373
gdcm::IconImageFilter
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 375

gdcm::IconImageGenerator
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: . . . 377
gdcm::ignore_char . . . 380
gdcm::Image
Image This is the container for an Image in the general sense. From this container you should be able to request information like: . . . 381
gdcm::ImageApplyLookupTable
ImageApplyLookupTable class It applies the LUT the PixelData (only PALETTE_COLOR images) Output will be a PhotometricInterpretation=RGB image . . . 385
gdcm::ImageChangePhotometricInterpretation
ImageChangePhotometricInterpretation class Class to change the Photometric Interpetation of an input DICOM . . . 387
gdcm::ImageChangePlanarConfiguration
ImageChangePlanarConfiguration class Class to change the Planar configuration of an input DICOM By default it will change into the more usual reprensentation: PlanarConfiguration = 0 . . . 390
gdcm::ImageChangeTransferSyntax
ImageChangeTransferSyntax class Class to change the transfer syntax of an input DICOM . . . 393
gdcm::ImageCodec
ImageCodec . . . 397
gdcm::ImageConverter
Image Converter . . . 404
gdcm::ImageFragmentSplitter
ImageFragmentSplitter class For single frame image, DICOM standard allow splitting the frame into multiple fragments . . . 405
gdcm::ImageHelper
ImageHelper (internal class, not intended for user level) . . . 407
gdcm::ImageReader
ImageReader . . . 411
gdcm::ImageRegionReader
ImageRegionReader . . . 416
gdcm::ImageToImageFilter
ImageToImageFilter class Super class for all filter taking an image and producing an output image . 419
gdcm::ImageWriter
ImageWriter . . . 421
gdcm::network::ImplementationClassUIDSub
ImplementationClassUIDSub PS 3.7 Table D.3-1 IMPLEMENTATION CLASS UID SUB-ITEM FIE↩
LDS (A-ASSOCIATE-RQ) . . . 424
gdcm::network::ImplementationUIDSub
ImplementationUIDSub Table D.3-2 IMPLEMENTATION UID SUB-ITEM FIELDS (A-ASSOCIATE-↩
AC) . . . 425
gdcm::network::ImplementationVersionNameSub
ImplementationVersionNameSub Table D.3-3 IMPLEMENTATION VERSION NAME SUB-ITEM F↩
IELDS (A-ASSOCIATE-RQ) . . . 425
gdcm::ImplicitDataElement
Class to represent an *Implicit VR* Data Element . . . 426
gdcm::InitializeEvent . . . 428
gdcm::IOD
Class for representing a IOD . . . 429
gdcm::IODEntry
Class for representing a IODEntry . . . 431
gdcm::IODs
Class for representing a IODs . . . 433

gdcm::IPPSorter
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 435
gdcm::Item
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 439
gdcm::IterationEvent 443
gdcm::JPEG12Codec
Class to do JPEG 12bits (lossy & lossless) 444
gdcm::JPEG16Codec
Class to do JPEG 16bits (lossless) 446
gdcm::JPEG2000Codec
Class to do JPEG 2000 448
gdcm::JPEG8Codec
Class to do JPEG 8bits (lossy & lossless) 453
gdcm::JPEGCodec
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 455
gdcm::JPEGLSCodec
JPEG-LS 461
gdcm::JSON 465
gdcm::KAKADUCodec
KAKADUCodec 466
gdcm::LO
LO 468
gdcm::LookupTable
LookupTable class 471
gdcm::Scanner::ltstr 475
gdcm::StrictScanner::ltstr 476
gdcm::Macro
Class for representing a Macro 476
gdcm::Macros
Class for representing a Modules 478
gdcm::network::MaximumLengthSub
MaximumLengthSub Annex D Table D.1-1 MAXIMUM LENGTH SUB-ITEM FIELDS (A-ASSOCIATE-RQ) 480
gdcm::MD5
Class for MD5 481
gdcm::MediaStorage
MediaStorage 482
gdcm::MemberCommand< T >
Command subclass that calls a pointer to a member function 491
gdcm::MeshPrimitive
This class defines surface mesh primitives. It is designed from surface mesh primitives macro 495
gdcm::ModalityPerformedProcedureStepCreateQuery
ModalityPerformedProcedureStepCreateQuery contains: the class which will produce a dataset for n-create for Modality Performed Procedure Step sop class 499

gdcm::ModalityPerformedProcedureStepSetQuery
ModalityPerformedProcedureStepSetQuery contains: the class which will produce a dataset for n-set for Modality Performed Procedure Step sop class . . . 501
gdcm::ModifiedEvent . . . 504
gdcm::Module
Class for representing a Module . . . 505
gdcm::ModuleEntry
Class for representing a ModuleEntry . . . 507
gdcm::Modules
Class for representing a Modules . . . 510
gdcm::MovePatientRootQuery
MovePatientRootQuery contains: the class which will produce a dataset for c-move with patient root 512
gdcm::MoveStudyRootQuery
MoveStudyRootQuery contains: the class which will produce a dataset for C-MOVE with study root . 515
gdcm::network::NActionRQ
NActionRQ this file defines the messages for the NAction action . . . 517
gdcm::network::NActionRSP
NActionRSP this file defines the messages for the NAction action . . . 518
gdcm::network::NCreateRQ
NCreateRQ this file defines the messages for the ncreate action . . . 519
gdcm::network::NCreateRSP
NCreateRSP this file defines the messages for the ncreate action . . . 521
gdcm::network::NDeleteRQ
NDeleteRQ this file defines the messages for the ndelete action . . . 522
gdcm::network::NDeleteRSP
NDeleteRSP this file defines the messages for the ndelete action . . . 524
gdcm::NestedModuleEntries
Class for representing a NestedModuleEntries . . . 525
gdcm::network::NEventReportRQ
NEventReportRQ this file defines the messages for the neventreport action . . . 528
gdcm::network::NEventReportRSP
NEventReportRSP this file defines the messages for the neventreport action . . . 529
gdcm::network::NGetRQ
NGetRQ this file defines the messages for the nget action . . . 530
gdcm::network::NGetRSP
NGetRSP this file defines the messages for the nget action . . . 532
gdcm::NoEvent . . . 533
gdcm::network::NormalizedMessageFactory . . . 534
gdcm::NormalizedNetworkFunctions
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: . . . 535
gdcm::network::NSetRQ
NSetRQ this file defines the messages for the nset action . . . 536
gdcm::network::NSetRSP
NSetRSP this file defines the messages for the nset action . . . 538
gdcm::Object
Object . . . 539

gdcm::OpenSSLCryptoFactory . . . 541
gdcm::OpenSSLCryptographicMessageSyntax . . . 543
gdcm::OpenSSLP7CryptoFactory . . . 545
gdcm::OpenSSLP7CryptographicMessageSyntax
Class for CryptographicMessageSyntax encryption. This is just a simple wrapper around openssl PKCS7_encrypt functionalities . . . 546
gdcm::Orientation
Class to handle Orientation . . . 549
gdcm::Overlay
Overlay class . . . 551
gdcm::ParseException
ParseException Standard exception handling object . . . 558
gdcm::Parser
Parser ala XML_Parser from expat (SAX) . . . 560
gdcm::Patient
See PS 3.3 - 2007 DICOM MODEL OF THE REAL-WORLD, p 54 . . . 562
gdcm::network::PDataTFPDU
PDataTFPDU Table 9-22 P-DATA-TF PDU FIELDS . . . 563
gdcm::PDBElement
Class to represent a PDB Element . . . 565
gdcm::PDBHeader
Class for PDBHeader . . . 567
gdcm::PDFCodec
PDFCodec class . . . 570
gdcm::network::PDUFactory
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 . . . 572
gdcm::PersonName
PersonName class . . . 574
gdcm::PGXCodec
Class to do PGX See PGX as used in JPEG 2000 implementation and reference images . . . 575
gdcm::PhotometricInterpretation
Class to represent an PhotometricInterpretation . . . 577
gdcm::PixelFormat
PixelFormat . . . 580
gdcm::Pixmap
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) . . . 586
gdcm::PixmapReader
PixmapReader . . . 589
gdcm::PixmapToPixmapFilter
PixmapToPixmapFilter class Super class for all filter taking an image and producing an output image 592
gdcm::PixmapWriter
PixmapWriter This class will takes two inputs: . . . 594
gdcm::PNMCodec
Class to do PNM PNM is the Portable anymap file format. The main web page can be found at: `http://netpbm.sourceforge.net/` . . . 598
gdcm::Preamble
DICOM Preamble (Part 10) . . . 600
gdcm::PresentationContext
PresentationContext . . . 602
gdcm::network::PresentationContextAC
PresentationContextAC Table 9-18 PRESENTATION CONTEXT ITEM FIELDS . . . 604

gdcm::PresentationContextGenerator
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 . . . 605
gdcm::network::PresentationContextRQ
PresentationContextRQ Table 9-13 PRESENTATION CONTEXT ITEM FIELDS . . . 608
gdcm::network::PresentationDataValue
PresentationDataValue Table 9-23 PRESENTATION-DATA-VALUE ITEM FIELDS . . . 610
gdcm::Printer
Printer class . . . 612
gdcm::PrivateDict
Private Dict . . . 616
gdcm::PrivateTag
Class to represent a Private DICOM Data Element (Attribute) Tag (Group, Element, Owner) . . . 618
gdcm::ProgressEvent
ProgressEvent Special type of event triggered during . . . 620
gdcm::PVRGCodec
PVRGCodec . . . 623
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 . . . 625
gdcm::QueryBase
QueryBase contains: the base class for constructing a query dataset for a C-FIND and a C-MOVE . 626
gdcm::QueryFactory
QueryFactory.h . . . 629
gdcm::QueryImage
QueryImage contains: class to construct an image-based query for C-FIND and C-MOVE . . . 630
gdcm::QueryPatient
QueryPatient contains: class to construct a patient-based query for c-find and c-move . . . 632
gdcm::QuerySeries
QuerySeries contains: class to construct a series-based query for c-find and c-move . . . 634
gdcm::QueryStudy
QueryStudy.h contains: class to construct a study-based query for C-FIND and C-MOVE . . . 636
gdcm::RAWCodec
RAWCodec class . . . 638
gdcm::Reader
Reader ala DOM (Document Object Model) . . . 641
gdcm::RealWorldValueMappingContent . . . 647
gdcm::Region
Class for manipulation region . . . 648
gdcm::Rescaler
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 . . . 650
gdcm::RLECodec
Class to do RLE . . . 653

gdcm::network::RoleSelectionSub
RoleSelectionSub PS 3.7 Table D.3-9 SCP/SCU ROLE SELECTION SUB-ITEM FIELDS (A-ASS↩ OCIATE-RQ) . . . 657
gdcm::SerieHelper::Rule . . . 658
gdcm::Scanner
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 . . . 659
gdcm::Segment
This class defines a segment. It mainly contains attributes of group 0x0062. In addition, it can be associated with surface . . . 666
gdcm::SegmentedPaletteColorLookupTable
SegmentedPaletteColorLookupTable class . . . 671
gdcm::SegmentReader
This class defines a segment reader. It reads attributes of group 0x0062 . . . 673
gdcm::SegmentWriter
This class defines a segment writer. It writes attributes of group 0x0062 . . . 676
gdcm::SequenceOfFragments
Class to represent a Sequence Of Fragments . . . 678
gdcm::SequenceOfItems
Class to represent a Sequence Of Items (value representation : SQ) . . . 683
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 . . . 690
gdcm::Series
Series . . . 693
gdcm::network::ServiceClassApplicationInformation . . . 693
gdcm::ServiceClassUser
ServiceClassUser . . . 694
gdcm::SHA1
Class for SHA1 . . . 700
gdcm::SimpleMemberCommand< T >
Command subclass that calls a pointer to a member function . . . 701
gdcm::SimpleSubjectWatcher
SimpleSubjectWatcher This is a typical Subject Watcher class. It will observe all events . . . 704
gdcm::SmartPointer< ObjectType >
Class for Smart Pointer . . . 706
gdcm::network::SOPClassExtendedNegociationSub
SOPClassExtendedNegociationSub PS 3.7 Table D.3-11 SOP CLASS EXTENDED NEGOTIATION SUB-ITEM FIELDS (A-ASSOCIATE-RQ and A-ASSOCIATE-AC) . . . 709
gdcm::SOPClassUIDToIOD
Class convert a class SOP Class UID into IOD . . . 710
gdcm::Sorter
Sorter General class to do sorting using a custom function You simply need to provide a function of type: Sorter::SortFunction . . . 712
gdcm::Spacing
Class for Spacing . . . 715
gdcm::Spectroscopy
Spectroscopy class . . . 717
gdcm::SplitMosaicFilter
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 . . . 718
gdcm::StartEvent . . . 719

gdcm::static_assert_test< x > . . . 720
gdcm::STATIC_ASSERTION_FAILURE< x > . . . 720
gdcm::STATIC_ASSERTION_FAILURE< true > . . . 721
gdcm::StreamImageReader
StreamImageReader . . . 721
gdcm::StreamImageWriter
StreamImageReader . . . 724
gdcm::StrictScanner
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 . . . 730
gdcm::String< TDelimiter, TMaxLength, TPadChar >
String . . . 736
gdcm::StringFilter
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 . . . 740
gdcm::Study
Study . . . 742
gdcm::Subject
Subject . . . 743
gdcm::Surface
This class defines a SURFACE IE. This members are taken from required surface mesh module attributes . . . 746
gdcm::SurfaceHelper
SurfaceHelper Helper class for Surface object . . . 753
gdcm::SurfaceReader
This class defines a SURFACE IE reader. It reads surface mesh module attributes . . . 756
gdcm::SurfaceWriter
This class defines a SURFACE IE writer. It writes surface mesh module attributes . . . 758
gdcm::SwapCode
SwapCode representation . . . 760
gdcm::SwapperDoOp . . . 762
gdcm::SwapperNoOp . . . 763
gdcm::System
Class to do system operation . . . 763
gdcm::Table
Table . . . 768
gdcm::TableEntry
TableEntry . . . 770
gdcm::TableReader
Class for representing a TableReader . . . 771
gdcm::network::TableRow . . . 773
gdcm::Tag
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) . . . 774
gdcm::TagPath
Class to handle a path of tag . . . 781
gdcm::Testing
Class for testing . . . 783
gdcm::Trace
Trace . . . 788
gdcm::TransferSyntax
Class to manipulate Transfer Syntax . . . 792

gdcm::network::TransferSyntaxSub
TransferSyntaxSub Table 9-15 TRANSFER SYNTAX SUB-ITEM FIELDS . . . 797
gdcm::network::Transition . . . 799
gdcm::Type
Type . . . 800
gdcm::UI . . . 802
gdcm::UIDGenerator
Class for generating unique UID . . . 803
gdcm::UIDs
All known uids . . . 805
gdcm::network::ULAction
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 . . . 825
gdcm::network::ULActionAA1 . . . 828
gdcm::network::ULActionAA2 . . . 829
gdcm::network::ULActionAA3 . . . 830
gdcm::network::ULActionAA4 . . . 831
gdcm::network::ULActionAA5 . . . 832
gdcm::network::ULActionAA6 . . . 833
gdcm::network::ULActionAA7 . . . 834
gdcm::network::ULActionAA8 . . . 835
gdcm::network::ULActionAE1 . . . 836
gdcm::network::ULActionAE2 . . . 837
gdcm::network::ULActionAE3 . . . 838
gdcm::network::ULActionAE4 . . . 839
gdcm::network::ULActionAE5 . . . 840
gdcm::network::ULActionAE6 . . . 841
gdcm::network::ULActionAE7 . . . 842
gdcm::network::ULActionAE8 . . . 843
gdcm::network::ULActionAR1 . . . 844
gdcm::network::ULActionAR10 . . . 845
gdcm::network::ULActionAR2 . . . 846
gdcm::network::ULActionAR3 . . . 847
gdcm::network::ULActionAR4 . . . 848
gdcm::network::ULActionAR5 . . . 849
gdcm::network::ULActionAR6 . . . 850
gdcm::network::ULActionAR7 . . . 851
gdcm::network::ULActionAR8 . . . 852
gdcm::network::ULActionAR9 . . . 853
gdcm::network::ULActionDT1 . . . 854
gdcm::network::ULActionDT2 . . . 855
gdcm::network::ULBasicCallback
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 . . . 856
gdcm::network::ULConnection
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 . . . 858
gdcm::network::ULConnectionCallback . . . 861

gdcm::network::ULConnectionInfo
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 . . . 863
gdcm::network::ULConnectionManager
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) . . . 864
gdcm::network::ULEvent
ULEvent base class for network events . . . 869
gdcm::network::ULTransitionTable
ULTransitionTable The transition table of all the ULEvents, new ULActions, and ULStates . . . 870
gdcm::network::ULWritingCallback . . . 871
gdcm::UNExplicitDataElement
Class to read/write a DataElement as UNExplicit Data Element . . . 872
gdcm::UNExplicitImplicitDataElement
Class to read/write a DataElement as ExplicitImplicit Data Element This class gather two known bugs: . . . 874
gdcm::Unpacker12Bits
Pack/Unpack 12 bits pixel into 16bits . . . 876
gdcm::Usage
Usage . . . 877
gdcm::UserEvent . . . 879
gdcm::network::UserInformation
UserInformation Table 9-16 USER INFORMATION ITEM FIELDS . . . 880
gdcm::UUIDGenerator
Class for generating unique UUID generate DCE 1.1 uid . . . 882
gdcm::Validate
Validate class . . . 883
gdcm::Value
Class to represent the value of a Data Element . . . 884
gdcm::ValueIO< TDE, TSwap, TType >
Class to dispatch template calls . . . 886
gdcm::Version
Major/minor and build version . . . 887
gdcm::VL
Value Length . . . 888
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 . . . 891
gdcm::VMToLength< T > . . . 895
gdcm::VR
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 . . . 896
gdcm::VR16ExplicitDataElement
Class to read/write a DataElement as Explicit Data Element . . . 900
gdcm::VRToEncoding< T > . . . 902
gdcm::VRToType< T > . . . 902
gdcm::VRVLSize< T > . . . 903
gdcm::VRVLSize< 0 > . . . 903
gdcm::VRVLSize< 1 > . . . 904
vtkGDCMImageReader . . . 904

vtkGDCMImageReader2 . . . 911
vtkGDCMImageWriter . . . 917
vtkGDCMMedicalImageProperties . . . 922
vtkGDCMPolyDataReader . . . 925
vtkGDCMPolyDataWriter . . . 928
vtkGDCMTesting . . . 931
vtkGDCMThreadedImageReader . . . 933
vtkGDCMThreadedImageReader2 . . . 936
vtkImageColorViewer . . . 940
vtkImageMapToColors16 . . . 947
vtkImageMapToWindowLevelColors2 . . . 950
vtkImagePlanarComponentsToComponents . . . 952
vtkImageRGBToYBR . . . 954
vtkImageYBRToRGB . . . 956
vtkLookupTable16 . . . 957
vtkRTStructSetProperties . . . 960
gdcm::Waveform
Waveform class . . . 965
gdcm::WLMFindQuery
PatientRootQuery contains: the class which will produce a dataset for c-find with patient root . . . 966
gdcm::Writer
Writer ala DOM (Document Object Model) This class is a non-validating writer, it will only performs well- formedness check only . . . 969
gdcm::XMLDictReader
Class for representing a XMLDictReader . . . 974
gdcm::XMLPrinter . . . 976
gdcm::XMLPrivateDictReader
Class for representing a XMLPrivateDictReader . . . 978

Chapter 8

File Index

8.1 File List

Here is a list of all files with brief descriptions:

gdcmAAbortPDU.h . . . 981
gdcmAAssociateACPDU.h . . . 982
gdcmAAssociateRJPDU.h . . . 982
gdcmAAssociateRQPDU.h . . . 983
gdcmAbstractSyntax.h . . . 984
gdcmAnonymizeEvent.h . . . 985
gdcmAnonymizer.h . . . 986
gdcmApplicationContext.h . . . 987
gdcmApplicationEntity.h . . . 988
gdcmAReleaseRPPDU.h . . . 988
gdcmAReleaseRQPDU.h . . . 989
gdcmARTIMTimer.h . . . 990
gdcmASN1.h . . . 991
gdcmAsynchronousOperationsWindowSub.h . . . 992
gdcmAttribute.h . . . 992
gdcmAudioCodec.h . . . 994
gdcmBase64.h . . . 994
gdcmBaseCompositeMessage.h . . . 995
gdcmBaseNormalizedMessage.h . . . 996
gdcmBasePDU.h . . . 997
gdcmBaseQuery.h . . . 998
gdcmBaseRootQuery.h . . . 999
gdcmBasicOffsetTable.h . . . 1000
gdcmBitmap.h . . . 1001
gdcmBitmapToBitmapFilter.h . . . 1002
gdcmBoxRegion.h . . . 1003
gdcmByteBuffer.h . . . 1003
gdcmByteSwap.h . . . 1005
gdcmByteSwapFilter.h . . . 1005
gdcmByteValue.h . . . 1006
gdcmCAPICryptoFactory.h . . . 1007

gdcmCAPICryptographicMessageSyntax.h . . . 1008
gdcmCEchoMessages.h . . . 1008
gdcmCFindMessages.h . . . 1009
gdcmCMoveMessages.h . . . 1010
gdcmCodec.h . . . 1011
gdcmCoder.h . . . 1012
gdcmCodeString.h . . . 1014
gdcmCommand.h . . . 1015
gdcmCommandDataSet.h . . . 1016
gdcmCompositeMessageFactory.h . . . 1017
gdcmCompositeNetworkFunctions.h . . . 1017
gdcmConstCharWrapper.h . . . 1018
gdcmCP246ExplicitDataElement.h . . . 1019
gdcmCryptoFactory.h . . . 1019
gdcmCryptographicMessageSyntax.h . . . 1020
gdcmCSAElement.h . . . 1021
gdcmCSAHeader.h . . . 1022
gdcmCSAHeaderDict.h . . . 1023
gdcmCSAHeaderDictEntry.h . . . 1024
gdcmCStoreMessages.h . . . 1026
gdcmCurve.h . . . 1027
gdcmDataElement.h . . . 1028
gdcmDataEvent.h . . . 1029
gdcmDataSet.h . . . 1030
gdcmDataSetEvent.h . . . 1031
gdcmDataSetHelper.h . . . 1031
gdcmDecoder.h . . . 1032
gdcmDefinedTerms.h . . . 1034
gdcmDeflateStream.h . . . 1034
gdcmDefs.h . . . 1035
gdcmDeltaEncodingCodec.h . . . 1036
gdcmDICOMDIR.h . . . 1036
gdcmDICOMDIRGenerator.h . . . 1037
gdcmDict.h . . . 1038
gdcmDictConverter.h . . . 1039
gdcmDictEntry.h . . . 1040
gdcmDictPrinter.h . . . 1041
gdcmDicts.h . . . 1042
gdcmDIMSE.h . . . 1043
gdcmDirectionCosines.h . . . 1043
gdcmDirectory.h . . . 1044
gdcmDirectoryHelper.h . . . 1045
gdcmDummyValueGenerator.h . . . 1046
gdcmDumper.h . . . 1047
gdcmElement.h . . . 1047
gdcmEncapsulatedDocument.h . . . 1049
gdcmEnumeratedValues.h . . . 1050
gdcmEvent.h . . . 1050
gdcmException.h . . . 1052
gdcmExplicitDataElement.h . . . 1053
gdcmExplicitImplicitDataElement.h . . . 1054
gdcmFiducials.h . . . 1054
gdcmFile.h . . . 1055
gdcmFileAnonymizer.h . . . 1056

gdcmFileChangeTransferSyntax.h . . . 1057
gdcmFileDecompressLookupTable.h . . . 1057
gdcmFileDerivation.h . . . 1058
gdcmFileExplicitFilter.h . . . 1059
gdcmFileMetaInformation.h . . . 1060
gdcmFilename.h . . . 1061
gdcmFileNameEvent.h . . . 1061
gdcmFilenameGenerator.h . . . 1062
gdcmFileSet.h . . . 1063
gdcmFileStreamer.h . . . 1064
gdcmFindPatientRootQuery.h . . . 1065
gdcmFindStudyRootQuery.h . . . 1066
gdcmFragment.h . . . 1066
gdcmGlobal.h . . . 1068
gdcmGroupDict.h . . . 1069
gdcmIconImage.h . . . 1069
gdcmIconImageFilter.h . . . 1070
gdcmIconImageGenerator.h . . . 1071
gdcmImage.h . . . 1072
gdcmImageApplyLookupTable.h . . . 1073
gdcmImageChangePhotometricInterpretation.h . . . 1074
gdcmImageChangePlanarConfiguration.h . . . 1074
gdcmImageChangeTransferSyntax.h . . . 1075
gdcmImageCodec.h . . . 1076
gdcmImageConverter.h . . . 1077
gdcmImageFragmentSplitter.h . . . 1078
gdcmImageHelper.h . . . 1079
gdcmImageReader.h . . . 1080
gdcmImageRegionReader.h . . . 1081
gdcmImageToImageFilter.h . . . 1081
gdcmImageWriter.h . . . 1082
gdcmImplementationClassUIDSub.h . . . 1083
gdcmImplementationUIDSub.h . . . 1084
gdcmImplementationVersionNameSub.h . . . 1085
gdcmImplicitDataElement.h . . . 1086
gdcmIOD.h . . . 1087
gdcmIODEntry.h . . . 1089
gdcmIODs.h . . . 1091
gdcmIPPSorter.h . . . 1092
gdcmItem.h . . . 1093
gdcmJPEG12Codec.h . . . 1094
gdcmJPEG16Codec.h . . . 1095
gdcmJPEG2000Codec.h . . . 1095
gdcmJPEG8Codec.h . . . 1096
gdcmJPEGCodec.h . . . 1097
gdcmJPEGLSCodec.h . . . 1099
gdcmJSON.h . . . 1099
gdcmKAKADUCodec.h . . . 1100
gdcmLegacyMacro.h . . . 1101
gdcmLO.h . . . 1102
gdcmLookupTable.h . . . 1103
gdcmMacro.h . . . 1104
gdcmMacroEntry.h . . . 1106
gdcmMacros.h . . . 1107

gdcmMaximumLengthSub.h . . . 1109
gdcmMD5.h . . . 1110
gdcmMediaStorage.h . . . 1111
gdcmMeshPrimitive.h . . . 1112
gdcmModalityPerformedProcedureStepCreateQuery.h . . . 1113
gdcmModalityPerformedProcedureStepSetQuery.h . . . 1114
gdcmModule.h . . . 1114
gdcmModuleEntry.h . . . 1116
gdcmModules.h . . . 1118
gdcmMovePatientRootQuery.h . . . 1119
gdcmMoveStudyRootQuery.h . . . 1120
gdcmNActionMessages.h . . . 1120
gdcmNCreateMessages.h . . . 1121
gdcmNDeleteMessages.h . . . 1122
gdcmNestedModuleEntries.h . . . 1122
gdcmNetworkEvents.h . . . 1124
gdcmNetworkStateID.h . . . 1125
gdcmNEventReportMessages.h . . . 1126
gdcmNGetMessages.h . . . 1126
gdcmNormalizedMessageFactory.h . . . 1127
gdcmNormalizedNetworkFunctions.h . . . 1128
gdcmNSetMessages.h . . . 1129
gdcmObject.h . . . 1129
gdcmOpenSSLCryptoFactory.h . . . 1131
gdcmOpenSSLCryptographicMessageSyntax.h . . . 1131
gdcmOpenSSLP7CryptoFactory.h . . . 1133
gdcmOpenSSLP7CryptographicMessageSyntax.h . . . 1133
gdcmOrientation.h . . . 1135
gdcmOverlay.h . . . 1135
gdcmParseException.h . . . 1136
gdcmParser.h . . . 1138
gdcmPatient.h . . . 1138
gdcmPDataTFPDU.h . . . 1139
gdcmPDBElement.h . . . 1140
gdcmPDBHeader.h . . . 1141
gdcmPDFCodec.h . . . 1142
gdcmPDUFactory.h . . . 1143
gdcmPersonName.h . . . 1143
gdcmPGXCodec.h . . . 1144
gdcmPhotometricInterpretation.h . . . 1145
gdcmPixelFormat.h . . . 1146
gdcmPixmap.h . . . 1147
gdcmPixmapReader.h . . . 1148
gdcmPixmapToPixmapFilter.h . . . 1149
gdcmPixmapWriter.h . . . 1150
gdcmPNMCodec.h . . . 1151
gdcmPreamble.h . . . 1151
gdcmPresentationContext.h . . . 1153
gdcmPresentationContextAC.h . . . 1154
gdcmPresentationContextGenerator.h . . . 1155
gdcmPresentationContextRQ.h . . . 1155
gdcmPresentationDataValue.h . . . 1156
gdcmPrinter.h . . . 1157
gdcmPrivateTag.h . . . 1159

gdcmProgressEvent.h . . . 1160
gdcmPVRGCodec.h . . . 1161
gdcmPythonFilter.h . . . 1161
gdcmQueryBase.h . . . 1162
gdcmQueryFactory.h . . . 1163
gdcmQueryImage.h . . . 1164
gdcmQueryPatient.h . . . 1165
gdcmQuerySeries.h . . . 1166
gdcmQueryStudy.h . . . 1167
gdcmRAWCodec.h . . . 1168
gdcmReader.h . . . 1168
gdcmRegion.h . . . 1170
gdcmRescaler.h . . . 1171
gdcmRLECodec.h . . . 1172
gdcmRoleSelectionSub.h . . . 1172
gdcmScanner.h . . . 1173
gdcmSegment.h . . . 1174
gdcmSegmentedPaletteColorLookupTable.h . . . 1175
gdcmSegmentHelper.h . . . 1176
gdcmSegmentReader.h . . . 1177
gdcmSegmentWriter.h . . . 1179
gdcmSequenceOfFragments.h . . . 1180
gdcmSequenceOfItems.h . . . 1180
gdcmSerieHelper.h . . . 1181
gdcmSeries.h . . . 1183
gdcmServiceClassApplicationInformation.h . . . 1184
gdcmServiceClassUser.h . . . 1185
gdcmSHA1.h . . . 1185
gdcmSimpleSubjectWatcher.h . . . 1186
gdcmSmartPointer.h . . . 1187
gdcmSOPClassExtendedNegociationSub.h . . . 1188
gdcmSOPClassUIDToIOD.h . . . 1189
gdcmSorter.h . . . 1190
gdcmSpacing.h . . . 1192
gdcmSpectroscopy.h . . . 1192
gdcmSplitMosaicFilter.h . . . 1193
gdcmStaticAssert.h . . . 1193
gdcmStreamImageReader.h . . . 1195
gdcmStreamImageWriter.h . . . 1195
gdcmStrictScanner.h . . . 1196
gdcmString.h . . . 1197
gdcmStringFilter.h . . . 1199
gdcmStudy.h . . . 1199
gdcmSubject.h . . . 1201
gdcmSurface.h . . . 1202
gdcmSurfaceHelper.h . . . 1203
gdcmSurfaceReader.h . . . 1203
gdcmSurfaceWriter.h . . . 1204
gdcmSwapCode.h . . . 1205
gdcmSwapper.h . . . 1206
gdcmSystem.h . . . 1207
gdcmTable.h . . . 1208
gdcmTableEntry.h . . . 1209
gdcmTableReader.h . . . 1210

gdcmTag.h . . . 1211
gdcmTagPath.h . . . 1212
gdcmTagToVR.h . . . 1212
gdcmTerminal.h . . . 1213
gdcmTestDriver.h . . . 1214
gdcmTesting.h . . . 1215
gdcmTrace.h . . . 1216
gdcmTransferSyntax.h . . . 1219
gdcmTransferSyntaxSub.h . . . 1221
gdcmType.h . . . 1222
gdcmTypes.h . . . 1223
gdcmUIDGenerator.h . . . 1223
gdcmUIDs.h . . . 1224
gdcmULAction.h . . . 1225
gdcmULActionAA.h . . . 1226
gdcmULActionAE.h . . . 1226
gdcmULActionAR.h . . . 1227
gdcmULActionDT.h . . . 1228
gdcmULBasicCallback.h . . . 1229
gdcmULConnection.h . . . 1229
gdcmULConnectionCallback.h . . . 1230
gdcmULConnectionInfo.h . . . 1231
gdcmULConnectionManager.h . . . 1233
gdcmULEvent.h . . . 1233
gdcmULTransitionTable.h . . . 1235
gdcmULWritingCallback.h . . . 1236
gdcmUNExplicitDataElement.h . . . 1236
gdcmUNExplicitImplicitDataElement.h . . . 1237
gdcmUnpacker12Bits.h . . . 1238
gdcmUsage.h . . . 1238
gdcmUserInformation.h . . . 1240
gdcmUUIDGenerator.h . . . 1241
gdcmValidate.h . . . 1241
gdcmValue.h . . . 1242
gdcmValueIO.h . . . 1243
gdcmVersion.h . . . 1243
gdcmVL.h . . . 1244
gdcmVM.h . . . 1245
gdcmVR.h . . . 1246
gdcmVR16ExplicitDataElement.h . . . 1249
gdcmWaveform.h . . . 1249
gdcmWin32.h . . . 1250
gdcmWLMFindQuery.h . . . 1250
gdcmWriter.h . . . 1251
gdcmXMLDictReader.h . . . 1252
gdcmXMLPrinter.h . . . 1253
gdcmXMLPrivateDictReader.h . . . 1253
vtkGDCMImageReader.h . . . 1254
vtkGDCMImageReader2.h . . . 1255
vtkGDCMImageWriter.h . . . 1256
vtkGDCMMedicalImageProperties.h . . . 1257
vtkGDCMPolyDataReader.h . . . 1257
vtkGDCMPolyDataWriter.h . . . 1258
vtkGDCMTesting.h . . . 1259

vtkGDCMThreadedImageReader.h 1259
vtkGDCMThreadedImageReader2.h 1260
vtkImageColorViewer.h 1260
vtkImageMapToColors16.h 1261
vtkImageMapToWindowLevelColors2.h 1261
vtkImagePlanarComponentsToComponents.h 1262
vtkImageRGBToYBR.h 1262
vtkImageYBRToRGB.h 1263
vtkLookupTable16.h 1263
vtkRTStructSetProperties.h 1264

Chapter 9

Namespace Documentation

9.1 gdcm Namespace Reference

Namespaces

- network
- SegmentHelper
- terminal

  Class for Terminal Allow one to print in color in a shell.

Classes

- class AbortEvent
- class AnonymizeEvent

  AnonymizeEvent Special type of event triggered during the Anonymization process.
- class Anonymizer

  Anonymizer This class is a multi purpose anonymizer. It can work in 2 mode:
- 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 contains: a baseclass which will produce a dataset for all dimse messages.
- class BaseRootQuery

  BaseRootQuery contains: a baseclass which will produce a dataset for c-find and c-move with patient/study root.
- class BasicOffsetTable

  Class to represent a *BasicOffsetTable*.
- class Bitmap

  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)
- class BitmapToBitmapFilter

  BitmapToBitmapFilter class Super class for all filter taking an image and producing an output image.
- class BoxRegion

  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)
- class ByteBuffer

  ByteBuffer.
- class ByteSwap

  ByteSwap.
- class ByteSwapFilter

  ByteSwapFilter In place byte-swapping of a dataset FIXME: FL status ??
- class ByteValue

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

  Codec class.
- class Coder

  Coder.
- class CodeString

  CodeString This is an implementation of DICOM *VR*: CS The cstor will properly Trim so that operator== is correct.
- class Command

  Command superclass for callback/observer methods.
- class CommandDataSet

  Class to represent a *Command DataSet*.
- class CompositeNetworkFunctions

  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:
- 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 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.
- class CSAHeaderDictException
- class Curve

  Curve class to handle element 50xx,3000 Curve Data WARNING: This is deprecated and lastly defined in PS 3.3 - 2004.
- 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) A Data Set represents an instance of a real world Information Object.
- class DataSetEvent

  DataSetEvent Special type of event triggered during the DataSet store/move process.
- 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 This is a STD-GEN-CD DICOMDIR generator. ref: PS 3.11-2008 Annex D (Normative) - General Purpose CD-R and DVD Interchange Profiles.
- 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 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.

- 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 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.

- 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 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.
- class FileAnonymizer

  FileAnonymizer.
- class FileChangeTransferSyntax

  FileChangeTransferSyntax.
- class FileDecompressLookupTable

  FileDecompressLookupTable class It decompress the segmented LUT into linearized one (only PALETTE_COLOR images) Output will be a PhotometricInterpretation=RGB image.
- class FileDerivation

  FileDerivation class See PS 3.16 - 2008 For the list of Code Value that can be used for in Derivation Code Sequence.
- class FileExplicitFilter

  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.
- class FileMetaInformation

  Class to represent a File Meta Information.
- class Filename

  Class to manipulate file name's.
- class FileNameEvent

  FileNameEvent Special type of event triggered during processing of FileSet.
- class FilenameGenerator

  FilenameGenerator.
- class FileSet

  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.
- class FileStreamer

  FileStreamer This class let a user create a massive DICOM DataSet from a template DICOM file, by appending chunks of data.
- class FileWithName

  FileWithName.
- class FindPatientRootQuery

  PatientRootQuery contains: the class which will produce a dataset for c-find with patient root.
- class FindStudyRootQuery

  FindStudyRootQuery contains: the class which will produce a dataset for C-FIND with study root.

- 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 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.
- class IconImageGenerator

  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:
- struct ignore_char
- class Image

  Image This is the container for an Image in the general sense. From this container you should be able to request information like:
- class ImageApplyLookupTable

  ImageApplyLookupTable class It applies the LUT the PixelData (only PALETTE_COLOR images) Output will be a PhotometricInterpretation=RGB image.
- class ImageChangePhotometricInterpretation

  ImageChangePhotometricInterpretation class Class to change the Photometric Interpetation of an input DICOM.
- class ImageChangePlanarConfiguration

  ImageChangePlanarConfiguration class Class to change the Planar configuration of an input DICOM By default it will change into the more usual reprensentation: PlanarConfiguration = 0.
- class ImageChangeTransferSyntax

  ImageChangeTransferSyntax class Class to change the transfer syntax of an input DICOM.
- class ImageCodec

  ImageCodec.
- class ImageConverter

  Image Converter.
- class ImageFragmentSplitter

  ImageFragmentSplitter class For single frame image, DICOM standard allow splitting the frame into multiple fragments.
- class ImageHelper

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

  ImageReader.
- class ImageRegionReader

  ImageRegionReader.
- class ImageToImageFilter

  ImageToImageFilter class Super class for all filter taking an image and producing an output image.
- 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 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.

- class Item

  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.

- 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 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.

- 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. It is designed from surface mesh primitives macro.

- class ModalityPerformedProcedureStepCreateQuery

ModalityPerformedProcedureStepCreateQuery contains: the class which will produce a dataset for n-create for Modality Performed Procedure Step sop class.

- class ModalityPerformedProcedureStepSetQuery

  ModalityPerformedProcedureStepSetQuery contains: the class which will produce a dataset for n-set for Modality Performed Procedure Step sop class.
- 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 contains: the class which will produce a dataset for c-move with patient root.
- class MoveStudyRootQuery

  MoveStudyRootQuery contains: the class which will produce a dataset for C-MOVE with study root.
- class NestedModuleEntries

  Class for representing a NestedModuleEntries.
- class NoEvent
- class NormalizedNetworkFunctions

  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:
- class Object

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

  Class for CryptographicMessageSyntax encryption. This is just a simple wrapper around openssl PKCS7_encrypt functionalities.
- 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 See PGX as used in JPEG 2000 implementation and reference images.
- class PhotometricInterpretation

  Class to represent an PhotometricInterpretation.
- class PixelFormat

  PixelFormat.
- class Pixmap

  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)
- class PixmapReader

  PixmapReader.
- class PixmapToPixmapFilter

  PixmapToPixmapFilter class Super class for all filter taking an image and producing an output image.
- class PixmapWriter

  PixmapWriter This class will takes two inputs:
- class PNMCodec

  Class to do PNM PNM is the Portable anymap file format. The main web page can be found at: `http://netpbm.↩sourceforge.net/`.
- class Preamble

  DICOM Preamble (Part 10)
- class PresentationContext

  PresentationContext.
- class PresentationContextGenerator

  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.
- 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 Special type of event triggered during.
- 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 contains: the base class for constructing a query dataset for a C-FIND and a C-MOVE.
- class QueryFactory

  QueryFactory.h.
- class QueryImage

QueryImage contains: class to construct an image-based query for C-FIND and C-MOVE.

- class QueryPatient

  QueryPatient contains: class to construct a patient-based query for c-find and c-move.

- class QuerySeries

  QuerySeries contains: class to construct a series-based query for c-find and c-move.

- class QueryStudy

  QueryStudy.h contains: class to construct a study-based query for C-FIND and C-MOVE.

- class RAWCodec

  RAWCodec class.

- class Reader

  Reader ala DOM (Document Object Model)

- struct RealWorldValueMappingContent
- class Region

  Class for manipulation region.

- class Rescaler

  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.

- class RLECodec

  Class to do RLE.

- class Scanner

  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.

- class Segment

  This class defines a segment. It mainly contains attributes of group 0x0062. In addition, it can be associated with surface.

- class SegmentedPaletteColorLookupTable

  SegmentedPaletteColorLookupTable class.

- class SegmentReader

  This class defines a segment reader. It reads attributes of group 0x0062.

- class SegmentWriter

  This class defines a segment writer. It writes attributes of group 0x0062.

- class SequenceOfFragments

  Class to represent a Sequence Of Fragments.

- class SequenceOfItems

  Class to represent a Sequence Of Items (value representation : SQ)

- 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 This is a typical Subject Watcher class. It will observe all events.
- class SmartPointer

  Class for Smart Pointer.
- class SOPClassUIDToIOD

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

  Sorter General class to do sorting using a custom function You simply need to provide a function of type: Sorter::SortFunction.
- class Spacing

  Class for Spacing.
- class Spectroscopy

  Spectroscopy class.
- class SplitMosaicFilter

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

  StreamImageReader.
- class StreamImageWriter

  StreamImageReader.
- class StrictScanner

  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.
- class String

  String.
- class StringFilter

  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.
- class Study

  Study.
- class Subject

  Subject.
- class Surface

  This class defines a SURFACE IE. This members are taken from required surface mesh module attributes.
- class SurfaceHelper

  SurfaceHelper Helper class for Surface object.
- class SurfaceReader

  This class defines a SURFACE IE reader. It reads surface mesh module attributes.
- class SurfaceWriter

  This class defines a SURFACE IE writer. It writes surface mesh module attributes.
- 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). Basically an uint32_t which can also be expressed as two uint16_t (group and 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 This class gather two known bugs:
- class Unpacker12Bits

  Pack/Unpack 12 bits pixel into 16bits.
- class Usage

  Usage.
- class UserEvent
- class UUIDGenerator

  Class for generating unique UUID generate DCE 1.1 uid.
- 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 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.
- 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 contains: the class which will produce a dataset for c-find with patient root.
- class Writer

  Writer ala DOM (Document Object Model) This class is a non-validating writer, it will only performs well- formedness check only.
- 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 typedef String<'\\',16> gdcm::AEComp

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

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

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

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

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

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

9.1.2.8 typedef Bitmap gdcm::IconImage

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

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

9.1.2.11 typedef ModuleEntry gdcm::MacroEntry

9.1.2.12 typedef NestedModuleEntries gdcm::NestedMacroEntries

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

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

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

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

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

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

9.1.3 Enumeration Type Documentation

9.1.3.1 enum gdcm::CompOperators

Enumerator

GDCM_EQUAL
GDCM_DIFFERENT
GDCM_GREATER
GDCM_GREATEROREQUAL
GDCM_LESS
GDCM_LESSOREQUAL

9.1.3.2 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 enum gdcm::ENQueryType

Enumerator

eCreateMMPS

eSetMMPS

9.1.3.4 enum gdcm::EQueryLevel

Enumerator

ePatient

eStudy

eSeries

eImage

9.1.3.5 enum gdcm::EQueryType

Enumerator

eFind

eMove

eWLMFind

9.1.3.6 enum gdcm::ERootType

Enumerator

ePatientRootType

eStudyRootType

9.1.3.7 enum gdcm::LodModeType

Enumerator

LD_ALL

LD_NOSEQ

LD_NOSHADOW

LD_NOSHADOWSEQ

9.1.4 Function Documentation

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

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

9.1.4.2 VR::VRType gdcm::GetVRFromTag (Tag const & *tag*)

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

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

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

References gdcm::Version::Print().

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

References gdcm::SwapCode::GetSwapCodeString().

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

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

References gdcm::Region::Print().

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

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

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

References gdcm::DataSet::Print().

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

References gdcm::Orientation::Print().

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

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

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

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

References gdcm::Type::GetTypeString().

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

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

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

References gdcm::File::GetHeader().

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

References gdcm::Usage::GetUsageString().

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

References gdcm::Sorter::Print().

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

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

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

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

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

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

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

References gdcm::PDBHeader::Print().

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

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

References gdcm::Directory::Print().

Referenced by gdcm::CSAElement::CSAElement(), gdcm::CSAHeaderDict::CSAHeaderDict(), gdcm::CSAHeader↩ DictEntry::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::Photometric↩ Interpretation::PhotometricInterpretation(), gdcm::SwapCode::SwapCode(), gdcm::Tag::Tag(), gdcm::VR::Write(), gdcm::CommandDataSet::∼CommandDataSet(), gdcm::GroupDict::∼GroupDict(), and gdcm::ModuleEntry::∼↩ ModuleEntry().

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

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

References gdcm::PhotometricInterpretation::GetPIString().

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

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

References gdcm::Object::Print().

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

References gdcm::DataElement::GetByteValue(), gdcm::DataElement::ValueField, and gdcm::DataElement::Value↩ LengthField.

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

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

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

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

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

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

References gdcm::CSAHeader::Print().

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

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

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

References gdcm::TransferSyntax::GetTSString().

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

References gdcm::VM::GetVMString().

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

References gdcm::StrictScanner::Print().

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

References gdcm::Scanner::Print().

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

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

References gdcm::MediaStorage::GetMSString().

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

References gdcm::VR::GetVRString().

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

References gdcm::PixelFormat::Print().

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

References gdcm::UI::Internal.

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

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

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

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

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

Examples:

DumpPhilipsECHO.cxx.

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

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

References gdcm::ignore_char::m_char.

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

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

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

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

9.1.4.65 gdcm::TYPETOENCODING (SQ , VRBINARY , unsigned *char*)

9.1.5 Variable Documentation

9.1.5.1 Global gdcm::GlobalInstance [static]

9.1.5.2 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 Table 9-26 A-ABORT PDU FIELDS.
- class AAssociateACPDU

  AAssociateACPDU Table 9-17 ASSOCIATE-AC PDU fields.
- class AAssociateRJPDU

  AAssociateRJPDU Table 9-21 ASSOCIATE-RJ PDU FIELDS.
- class AAssociateRQPDU

  AAssociateRQPDU Table 9-11 ASSOCIATE-RQ PDU fields.
- class AbstractSyntax

AbstractSyntax Table 9-14 ABSTRACT SYNTAX SUB-ITEM FIELDS.

- class ApplicationContext

  ApplicationContext Table 9-12 APPLICATION CONTEXT ITEM FIELDS.
- class AReleaseRPPDU

  AReleaseRPPDU Table 9-25 A-RELEASE-RP PDU fields.
- class AReleaseRQPDU

  AReleaseRQPDU Table 9-24 A-RELEASE-RQ PDU FIELDS.
- class ARTIMTimer

  ARTIMTimer This file contains the code for the ARTIM timer.
- class AsynchronousOperationsWindowSub

  AsynchronousOperationsWindowSub PS 3.7 Table D.3-7 ASYNCHRONOUS OPERATIONS WINDOW SUB-ITEM FIE↩ LDS (A-ASSOCIATE-RQ)
- class BaseCompositeMessage

  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.
- class BaseNormalizedMessage

  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.
- class BasePDU

  BasePDU base class for PDUs.
- class CEchoRQ

  CEchoRQ this file defines the messages for the cecho action.
- 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 this file defines the messages for the cfind action.
- class CFindRSP

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

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

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

  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).
- class CStoreRQ

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

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

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)

- class ImplementationClassUIDSub

  ImplementationClassUIDSub PS 3.7 Table D.3-1 IMPLEMENTATION CLASS UID SUB-ITEM FIELDS (A-ASSOCIAT↩ E-RQ)
- class ImplementationUIDSub

  ImplementationUIDSub Table D.3-2 IMPLEMENTATION UID SUB-ITEM FIELDS (A-ASSOCIATE-AC)
- class ImplementationVersionNameSub

  ImplementationVersionNameSub Table D.3-3 IMPLEMENTATION VERSION NAME SUB-ITEM FIELDS (A-ASSOCIATE-RQ)
- class MaximumLengthSub

  MaximumLengthSub Annex D Table D.1-1 MAXIMUM LENGTH SUB-ITEM FIELDS (A-ASSOCIATE-RQ)
- class NActionRQ

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

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

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

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

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

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

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

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

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

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

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

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

  PDataTFPDU Table 9-22 P-DATA-TF PDU FIELDS.
- class PDUFactory

  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.
- class PresentationContextAC

  PresentationContextAC Table 9-18 PRESENTATION CONTEXT ITEM FIELDS.
- class PresentationContextRQ

  PresentationContextRQ Table 9-13 PRESENTATION CONTEXT ITEM FIELDS.
- class PresentationDataValue

  PresentationDataValue Table 9-23 PRESENTATION-DATA-VALUE ITEM FIELDS.

- class RoleSelectionSub

  RoleSelectionSub PS 3.7 Table D.3-9 SCP/SCU ROLE SELECTION SUB-ITEM FIELDS (A-ASSOCIATE-RQ)
- class ServiceClassApplicationInformation
- class SOPClassExtendedNegociationSub

  SOPClassExtendedNegociationSub PS 3.7 Table D.3-11 SOP CLASS EXTENDED NEGOTIATION SUB-ITEM FIELDS (A-ASSOCIATE-RQ and A-ASSOCIATE-AC)
- class TableRow
- class TransferSyntaxSub

  TransferSyntaxSub Table 9-15 TRANSFER SYNTAX SUB-ITEM FIELDS.
- struct Transition
- class ULAction

  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.
- 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 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.
- class ULConnection

  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.
- class ULConnectionCallback
- class ULConnectionInfo

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.

- class ULConnectionManager

  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).
- class ULEvent

  ULEvent base class for network events.
- class ULTransitionTable

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

  UserInformation Table 9-16 USER INFORMATION ITEM FIELDS.

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 enum gdcm::network::EEventID

Enumerator

eAASSOCIATERequestLocalUser

eTransportConnConfirmLocal

eASSOCIATE_ACPDUreceived

eASSOCIATE_RJPDUreceived

eTransportConnIndicLocal

eAASSOCIATE_RQPDUreceived

eAASSOCIATEresponseAccept

eAASSOCIATEresponseReject

ePDATArequest

ePDATATFPDU

eARELEASERequest

eARELEASE_RQPDUReceivedOpen

eARELEASE_RPPDUReceived

eARELEASEResponse

eAABORTRequest

eAABORTPDUReceivedOpen

eTransportConnectionClosed

eARTIMTimerExpired

eUnrecognizedPDUReceived

eEventDoesNotExist

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

References eSta10ReleaseCollisionAc, eSta11ReleaseCollisionRq, eSta12ReleaseCollisionAcLocal, eSta13↩ AwaitingClose, eSta1Idle, eSta2Open, eSta3WaitLocalAssoc, eSta4LocalAssocDone, eSta5WaitRemoteAssoc, e↩ Sta6TransferReady, eSta7WaitRelease, eSta8WaitLocalRelease, eSta9ReleaseCollisionRqLocal, and eStaDoesNot↩ Exist.

9.2.3 Variable Documentation

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

9.2.3.2 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 Allow one to print in color in a shell.

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 enum gdcm::terminal::Attribute

Enumerator

reset
bright
dim
underline
blink
reverse
hidden

9.4.2.2 enum gdcm::terminal::Color

Enumerator

black
red
green
yellow
blue
magenta
cyan
white

9.4.2.3 enum gdcm::terminal::Mode

Enumerator

CONSOLE
VT100

9.4.3 Function Documentation

9.4.3.1 GDCM_EXPORT std::string gdcm::terminal::setattribute (Attribute *att*)

9.4.3.2 GDCM_EXPORT std::string gdcm::terminal::setbgcolor (Color *c*)

9.4.3.3 GDCM_EXPORT std::string gdcm::terminal::setfgcolor (Color *c*)

9.4.3.4 GDCM_EXPORT void gdcm::terminal::setmode (Mode *m*)

Chapter 10

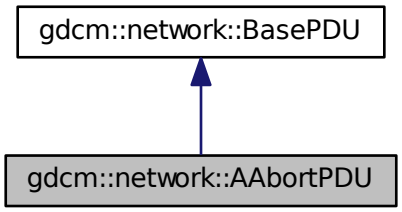
Class Documentation

10.1 gdcm::network::AAbortPDU Class Reference

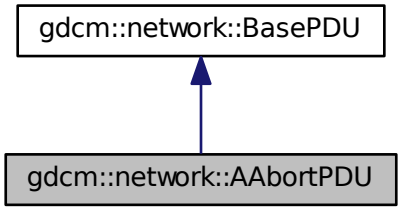
AAbortPDU Table 9-26 A-ABORT PDU FIELDS.

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

10.1.3 Member Function Documentation

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

Implements gdcm::network::BasePDU.

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

Implements gdcm::network::BasePDU.

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

Implements gdcm::network::BasePDU.

10.1.3.4 void gdcm::network::AAbortPDU::SetReason (const uint8_t *r*)

10.1.3.5 void gdcm::network::AAbortPDU::SetSource (const uint8_t *s*)

10.1.3.6 size_t gdcm::network::AAbortPDU::Size ( ) const [virtual]

Implements gdcm::network::BasePDU.

10.1.3.7 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:

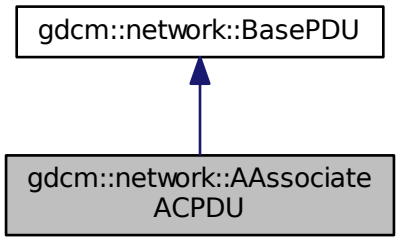
- gdcmAAbortPDU.h

10.2 gdcm::network::AAssociateACPDU Class Reference

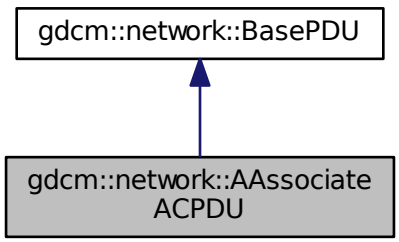
AAssociateACPDU Table 9-17 ASSOCIATE-AC PDU fields.

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

10.2.3 Constructor & Destructor Documentation

10.2.3.1 gdcm::network::AAssociateACPDU::AAssociateACPDU ( )

10.2.4 Member Function Documentation

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

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

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

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

References Print(), and Size().

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

Referenced by IsLastFragment().

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

Implements gdcm::network::BasePDU.

References InitFromRQ().

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

Implements gdcm::network::BasePDU.

Referenced by GetUserInformation().

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

Implements gdcm::network::BasePDU.

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

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

10.2.4.11 SizeType gdcm::network::AAssociateACPDU::Size ( ) const [virtual]

Implements gdcm::network::BasePDU.

Referenced by GetUserInformation().

10.2.4.12 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 friend class AAssociateRQPDU [friend]

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

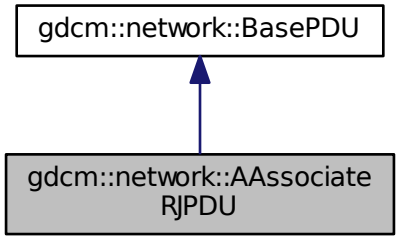
- gdcmAAssociateACPDU.h

10.3 gdcm::network::AAssociateRJPDU Class Reference

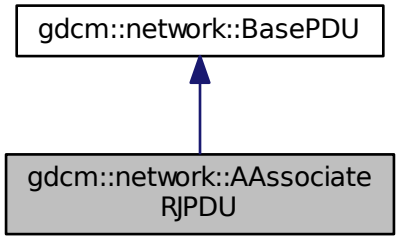
AAssociateRJPDU Table 9-21 ASSOCIATE-RJ PDU FIELDS.

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

10.3.3 Member Function Documentation

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

Implements gdcm::network::BasePDU.

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

Implements gdcm::network::BasePDU.

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

Implements gdcm::network::BasePDU.

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

Implements gdcm::network::BasePDU.

10.3.3.5 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:

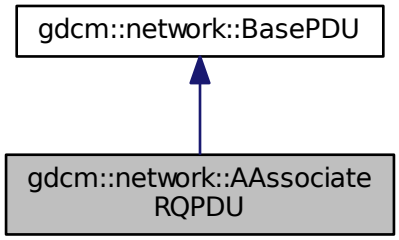
- gdcmAAssociateRJPDU.h

10.4 gdcm::network::AAssociateRQPDU Class Reference

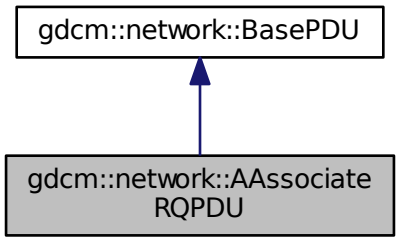
AAssociateRQPDU Table 9-11 ASSOCIATE-RQ PDU fields.

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

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

10.4.3 Constructor & Destructor Documentation

10.4.3.1 gdcm::network::AAssociateRQPDU::AAssociateRQPDU ( )

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

10.4.4 Member Function Documentation

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

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

References SetCallingAETitle().

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

References IsAETitleValid(), and Print().

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

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

10.4.4.6 const PresentationContextRQ* gdcm::network::AAssociateRQPDU::GetPresentationContextByAbstractSyntax (AbstractSyntax const & *absyn*) const

Referenced by GetPresentationContexts().

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

Referenced by GetPresentationContexts().

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

References GetPresentationContextByAbstractSyntax(), and GetPresentationContextByID().

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

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

References SetUserInformation().

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

Implements gdcm::network::BasePDU.

10.4.4.13 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 std::istream& gdcm::network::AAssociateRQPDU::Read (std::istream & *is*) [virtual]

Implements gdcm::network::BasePDU.

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

Set the Called AE Title.

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

Set the Calling AE Title.

Referenced by GetCalledAETitle().

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

Referenced by GetUserInformation().

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

Implements gdcm::network::BasePDU.

10.4.4.19 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 friend class AAssociateACPDU [friend]

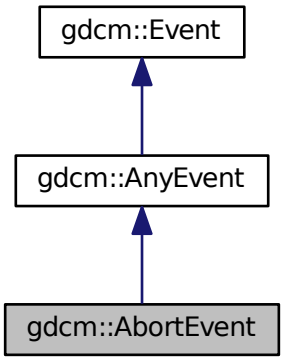
The documentation for this class was generated from the following file:

- gdcmAAssociateRQPDU.h

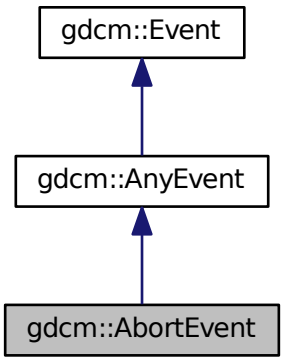
10.5 gdcm::AbortEvent Class Reference

`#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 Table 9-14 ABSTRACT SYNTAX SUB-ITEM FIELDS.

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

10.6.3 Member Function Documentation

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

Referenced by operator==().

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

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

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

References GetAsDataElement().

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

Referenced by GetName().

10.6.3.5 std::istream& gdcm::network::AbstractSyntax::Read (std::istream & *is*)

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

10.6.3.7 void gdcm::network::AbstractSyntax::SetNameFromUID (UIDs::TSName *tsname*)

Referenced by GetName().

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

Referenced by GetName().

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

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

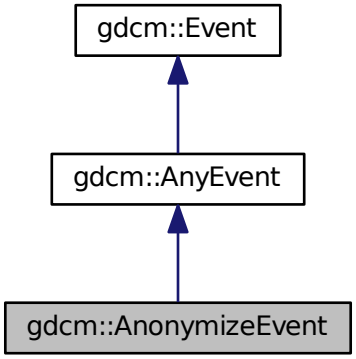
- gdcmAbstractSyntax.h

10.7 gdcm::AnonymizeEvent Class Reference

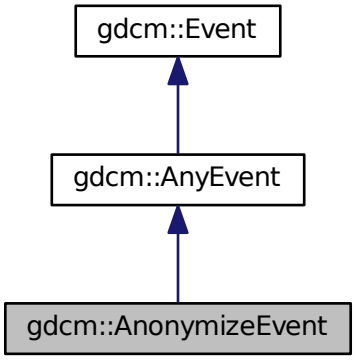
AnonymizeEvent Special type of event triggered during the Anonymization process.

`#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 typedef AnonymizeEvent gdcm::AnonymizeEvent::Self

10.7.2.2 typedef AnyEvent gdcm::AnonymizeEvent::Superclass

10.7.3 Constructor & Destructor Documentation

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

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

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

10.7.4 Member Function Documentation

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

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

Return the StringName associated with the event.

Implements gdcm::Event.

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

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

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

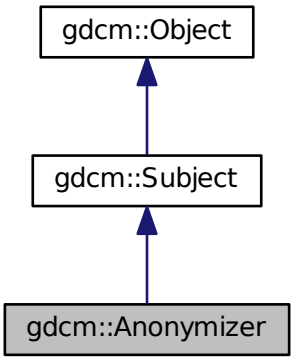
- gdcmAnonymizeEvent.h

10.8 gdcm::Anonymizer Class Reference

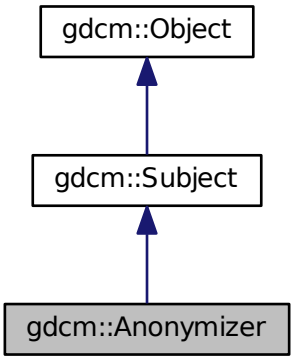
Anonymizer This class is a multi purpose anonymizer. It can work in 2 mode:

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

10.8.2.2 gdcm::Anonymizer::~Anonymizer ( )

10.8.3 Member Function Documentation

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

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

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

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

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

for wrapped language: instantiate a reference counted object

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

10.8.3.11 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 bool gdcm::Anonymizer::RemoveGroupLength ( )

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

Examples:

ClinicalTrialAnnotate.cxx.

10.8.3.13 bool gdcm::Anonymizer::RemovePrivateTags ( )

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

Examples:

ClinicalTrialAnnotate.cxx.

10.8.3.14 bool gdcm::Anonymizer::RemoveRetired ( )

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

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

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

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

Set/Get File.

Examples:

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

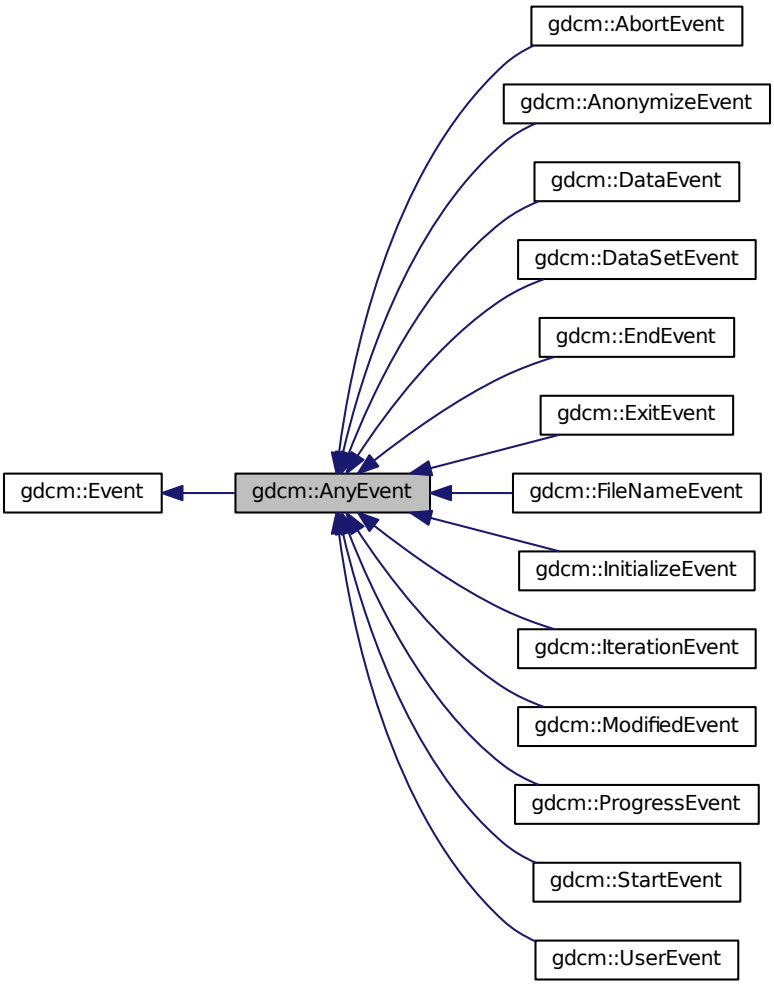
The documentation for this class was generated from the following file:

- gdcmAnonymizer.h

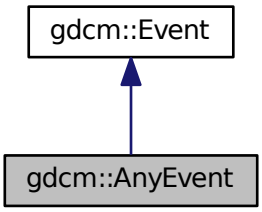
10.9 gdcm::AnyEvent Class Reference

```
#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 Table 9-12 APPLICATION CONTEXT ITEM FIELDS.

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

10.10.3 Member Function Documentation

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

References Print(), and Size().

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

Referenced by GetName().

10.10.3.3 std::istream& gdcm::network::ApplicationContext::Read (std::istream & *is*)

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

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

Referenced by GetName().

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

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

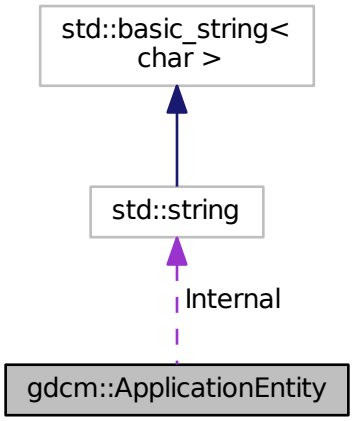
- gdcmApplicationContext.h

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

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

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

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

10.11.3 Member Data Documentation

10.11.3.1 std::string gdcm::ApplicationEntity::Internal

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

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

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

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

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

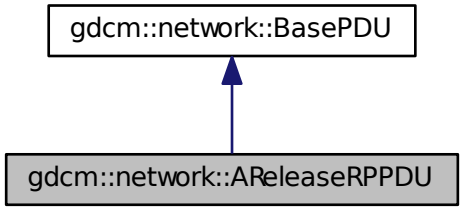
- gdcmApplicationEntity.h

10.12 gdcm::network::AReleaseRPPDU Class Reference

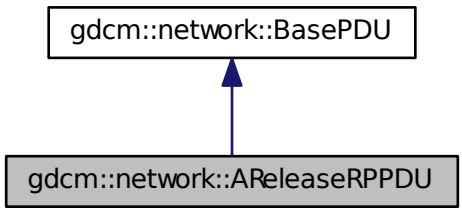
AReleaseRPPDU Table 9-25 A-RELEASE-RP PDU fields.

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

10.12.3 Member Function Documentation

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

Implements gdcm::network::BasePDU.

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

Implements gdcm::network::BasePDU.

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

Implements gdcm::network::BasePDU.

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

Implements gdcm::network::BasePDU.

10.12.3.5 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:

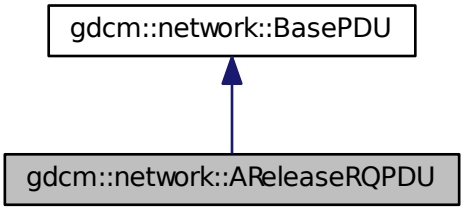
- gdcmAReleaseRPPDU.h

10.13 gdcm::network::AReleaseRQPDU Class Reference

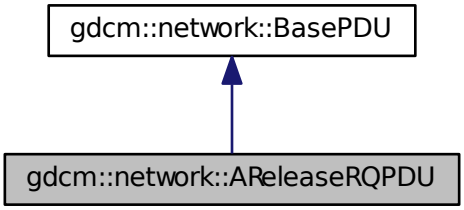
AReleaseRQPDU Table 9-24 A-RELEASE-RQ PDU FIELDS.

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

10.13.3 Member Function Documentation

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

Implements gdcm::network::BasePDU.

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

Implements gdcm::network::BasePDU.

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

Implements gdcm::network::BasePDU.

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

Implements gdcm::network::BasePDU.

10.13.3.5 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 This file contains the code for the ARTIM timer.

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

10.14.3 Member Function Documentation

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

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

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

10.14.3.4 void gdcm::network::ARTIMTimer::SetTimeout (double *inTimeout*)

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

10.14.3.6 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 gdcm::ASN1::ASN1 ( )

10.15.2.2 gdcm::ASN1::∼ASN1 ( )

10.15.3 Member Function Documentation

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

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

10.15.3.3 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 PS 3.7 Table D.3-7 ASYNCHRONOUS OPERATIONS WINDOW SUB-ITEM F↩ IELDS (A-ASSOCIATE-RQ)

`#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 F↩ IELDS (A-ASSOCIATE-RQ)

10.16.2 Constructor & Destructor Documentation

10.16.2.1 gdcm::network::AsynchronousOperationsWindowSub::AsynchronousOperationsWindowSub ( )

10.16.3 Member Function Documentation

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

10.16.3.2 std::istream& gdcm::network::AsynchronousOperationsWindowSub::Read (std::istream & *is*)

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

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

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

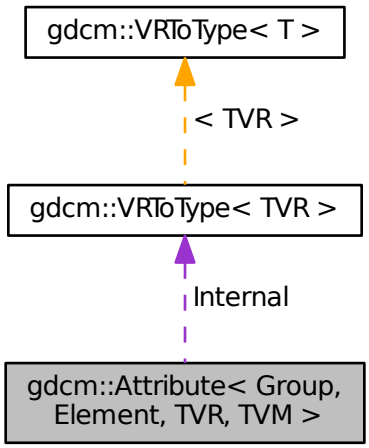
- gdcmAsynchronousOperationsWindowSub.h

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 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 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 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 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 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 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 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 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 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 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 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 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 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 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 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 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 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 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 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 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 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 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 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 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 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 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::IsEmpty().

10.17.4.25 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 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 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().

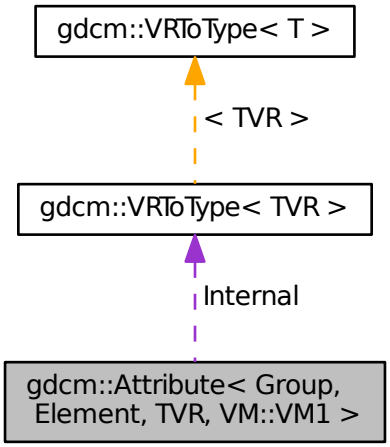
The documentation for this class was generated from the following file:

- 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 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 template<uint16_t Group, uint16_t Element, int TVR> anonymous enum

Enumerator

VMType

10.18.3 Member Function Documentation

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

10.18.3.11 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 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 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 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 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 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 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 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 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 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 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 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 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::IsEmpty().

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

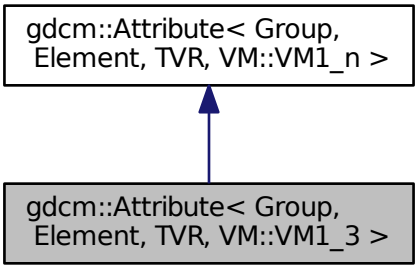
The documentation for this class was generated from the following file:

- gdcmAttribute.h

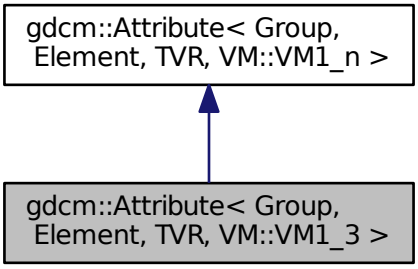
10.19 gdcm::Attribute< Group, Element, TVR, VM::VM1_3 > Class Template Reference

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

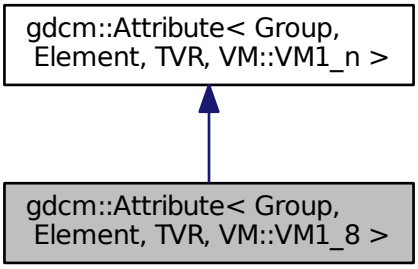
The documentation for this class was generated from the following file:

- gdcmAttribute.h

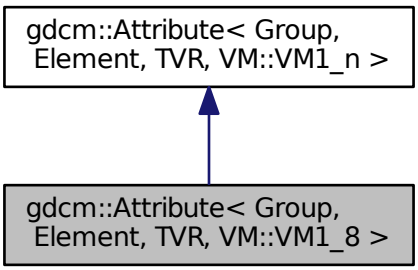
10.20 gdcm::Attribute< Group, Element, TVR, VM::VM1_8 > Class Template Reference

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

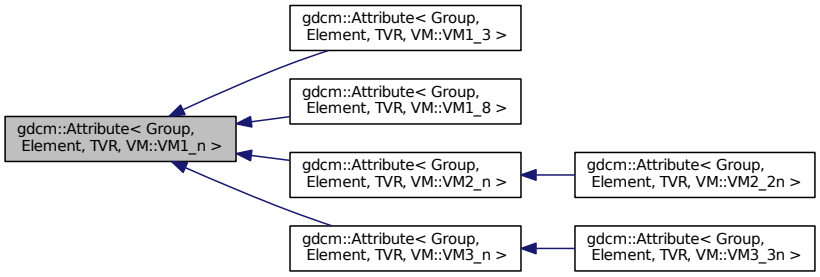
The documentation for this class was generated from the following file:

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

10.21.2.2 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 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 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 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 >::VMType==VM::VM1))||!((VR::VRType) TVR &VR::VR_VM1)))

10.21.3.4 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 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 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 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 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 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 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 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 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 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 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 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 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 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 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 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 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 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 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 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 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]

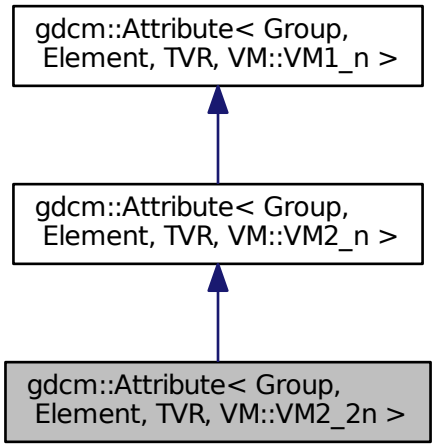
The documentation for this class was generated from the following file:

- gdcmAttribute.h

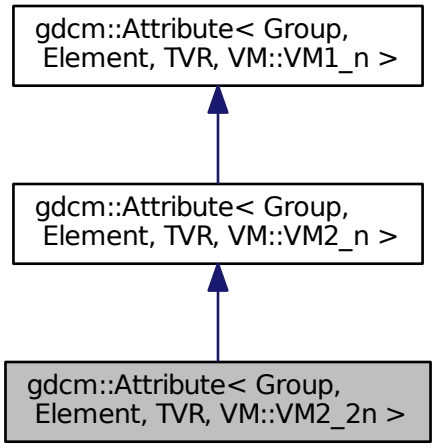
10.22 gdcm::Attribute< Group, Element, TVR, VM::VM2_2n > Class Template Reference

```
#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 template<uint16_t Group, uint16_t Element, int TVR> static VM gdcm::Attribute< Group, Element, TVR, VM::VM2_2n >::GetVM ( ) [inline], [static]

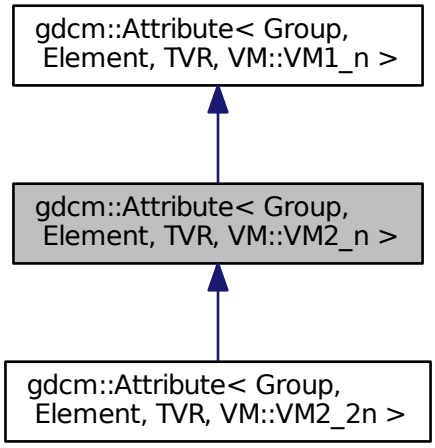
The documentation for this class was generated from the following file:

- gdcmAttribute.h

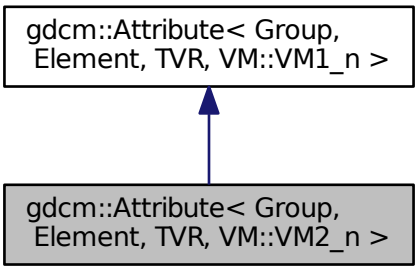
10.23 gdcm::Attribute< Group, Element, TVR, VM::VM2_n > Class Template Reference

`#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 template<uint16_t Group, uint16_t Element, int TVR> VM gdcm::Attribute< Group, Element, TVR, VM::VM2_n >::GetVM ( ) const [inline]

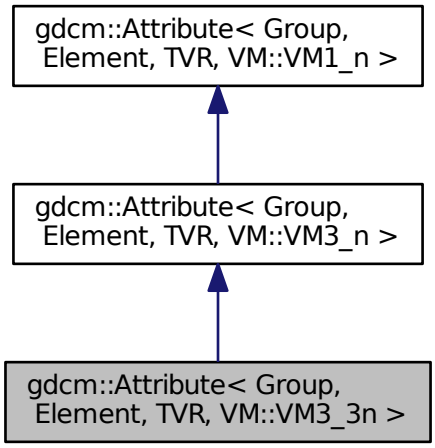
The documentation for this class was generated from the following file:

- gdcmAttribute.h

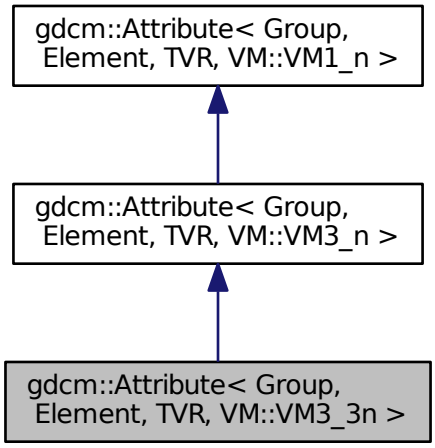
10.24 gdcm::Attribute< Group, Element, TVR, VM::VM3_3n > Class Template Reference

`#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 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.

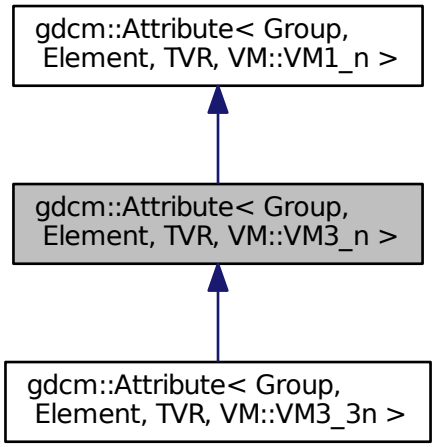
The documentation for this class was generated from the following file:

- gdcmAttribute.h

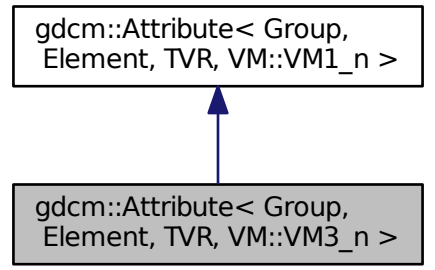
10.25 gdcm::Attribute< Group, Element, TVR, VM::VM3_n > Class Template Reference

`#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 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:

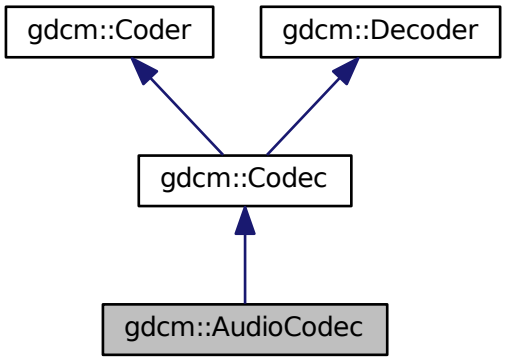
- gdcmAttribute.h

10.26 gdcm::AudioCodec Class Reference

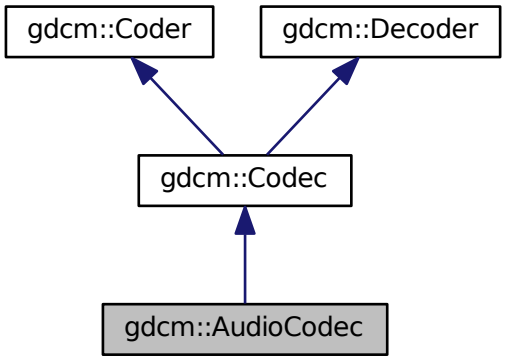
AudioCodec.

```
#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 gdcm::AudioCodec::AudioCodec ( )

10.26.2.2 gdcm::AudioCodec::~AudioCodec ( )

10.26.3 Member Function Documentation

10.26.3.1 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 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 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 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 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 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 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:

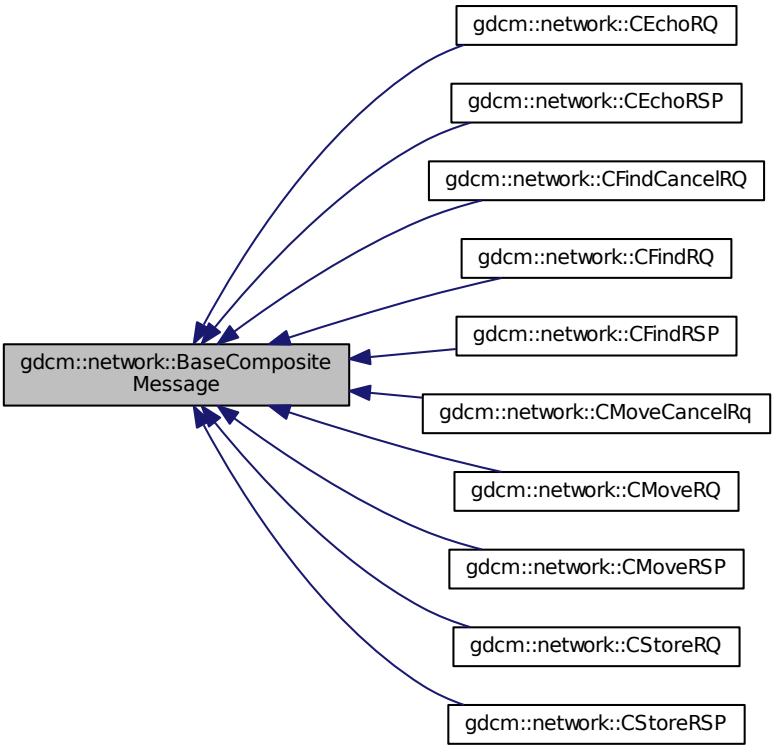
- gdcmBase64.h

10.28 gdcm::network::BaseCompositeMessage Class Reference

BaseCompositeMessage 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.

`#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 virtual gdcm::network::BaseCompositeMessage::~BaseCompositeMessage ( ) [inline], [virtual]

References ConstructPDV().

10.28.3 Member Function Documentation

10.28.3.1 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:

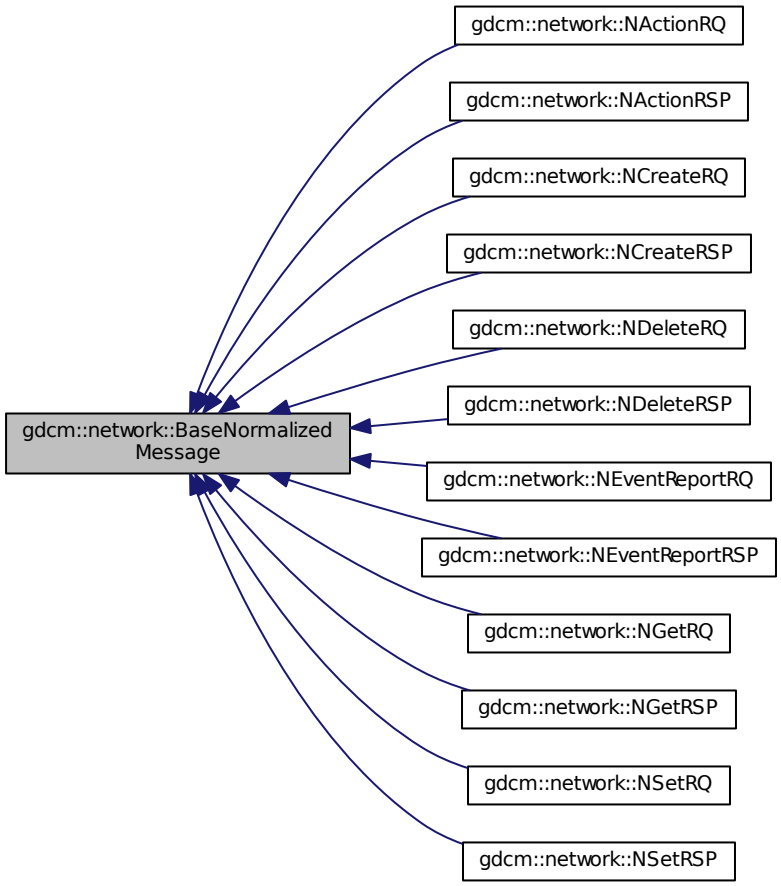
- gdcmBaseCompositeMessage.h

10.29 gdcm::network::BaseNormalizedMessage Class Reference

BaseNormalizedMessage 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.

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

Inheritance diagram for gdcm::network::BaseNormalizedMessage:



Public Member Functions

- virtual ~BaseNormalizedMessage ()
- virtual std::vector< PresentationDataValue > ConstructPDV (const ULConnection &inConnection, const BaseQuery *inQuery)=0

10.29.1 Detailed Description

BaseNormalizedMessage The Normalized events described in section 3.7-2011 of the DICOM standard all use their own messages. These messages are constructed using Presentation Data Values, from section 3.8-2011 of the standard, and then fill in appropriate values in their datasets.

So, for the five normalized:

- N-ACTION
- N-CREATE
- N-DELETE
- N-EVENT
- N-GET
- N-SET there are a series of messages. However, all of these messages are obtained as part of a 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 virtual gdcm::network::BaseNormalizedMessage::~BaseNormalizedMessage ( ) [inline], [virtual]

References ConstructPDV().

10.29.3 Member Function Documentation

10.29.3.1 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:

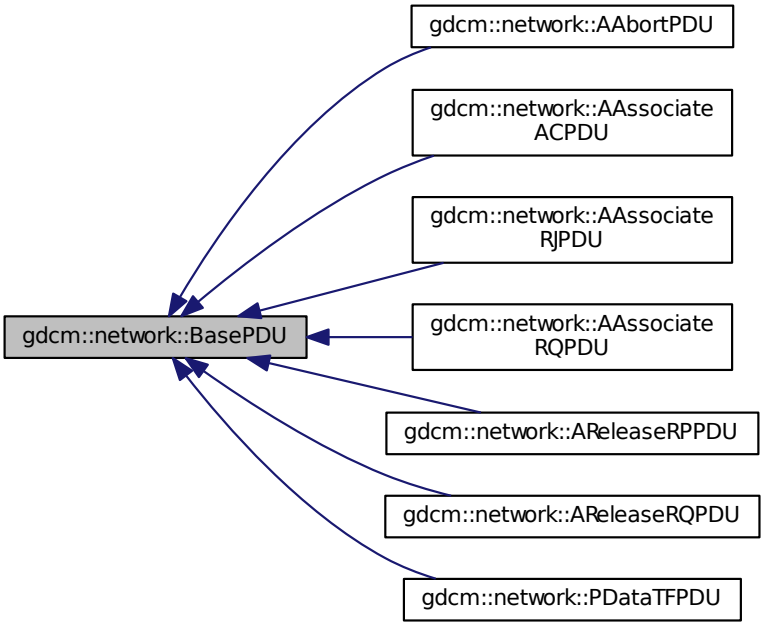
- gdcmBaseNormalizedMessage.h

10.30 gdcm::network::BasePDU Class Reference

BasePDU base class for PDUs.

`#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 virtual gdcm::network::BasePDU::~BasePDU ( ) [inline], [virtual]

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

10.30.3 Member Function Documentation

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

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

Referenced by ~BasePDU().

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

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

Referenced by ~BasePDU().

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

Implemented in gdcm::network::AAssociateACPDU, gdcm::network::AAssociateRQPDU, gdcm::network::PDataTFPDU, 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:

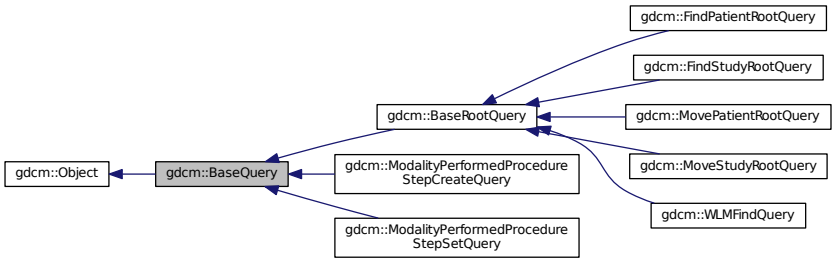
- gdcmBasePDU.h

10.31 gdcm::BaseQuery Class Reference

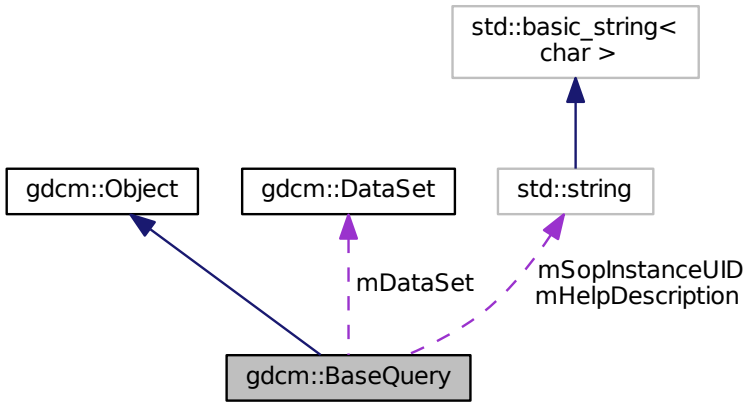
BaseQuery contains: a baseclass which will produce a dataset for all dimse messages.

```
#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 gdcm::BaseQuery::BaseQuery ( ) [protected]

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

10.31.3 Member Function Documentation

10.31.3.1 void gdcm::BaseQuery::AddQueryDataSet (const DataSet & *ds*)

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

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

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

10.31.3.5 std::string gdcm::BaseQuery::GetSOPInstanceUID ( ) const [inline]

10.31.3.6 void gdcm::BaseQuery::Print (std::ostream & *os*) const [virtual]

Reimplemented from gdcm::Object.

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

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

10.31.3.9 void gdcm::BaseQuery::SetSearchParameter (const std::string & *inKeyword,* const std::string & *inValue*)

10.31.3.10 void gdcm::BaseQuery::SetSOPInstanceUID (const std::string & *iSopInstanceUID*) [inline]

10.31.3.11 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 bool gdcm::BaseQuery::ValidDataSet (const DataSet & *dataSetToValid,* const DataSet & *dataSetReference*) const [protected]

10.31.3.13 const std::ostream& gdcm::BaseQuery::WriteHelpFile (std::ostream & *os*)

10.31.3.14 bool gdcm::BaseQuery::WriteQuery (const std::string & *inFileName*)

10.31.4 Friends And Related Function Documentation

10.31.4.1 friend class QueryFactory [friend]

10.31.5 Member Data Documentation

10.31.5.1 DataSet gdcm::BaseQuery::mDataSet [protected]

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

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

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

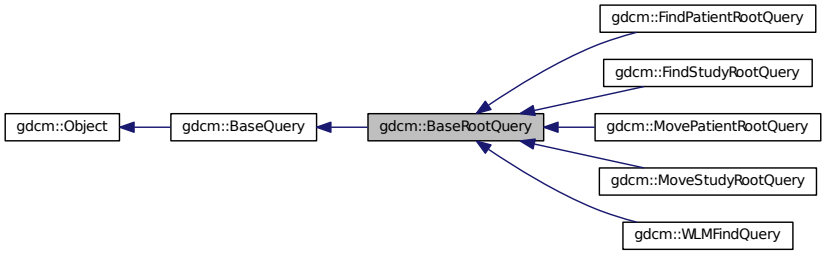
- gdcmBaseQuery.h

10.32 gdcm::BaseRootQuery Class Reference

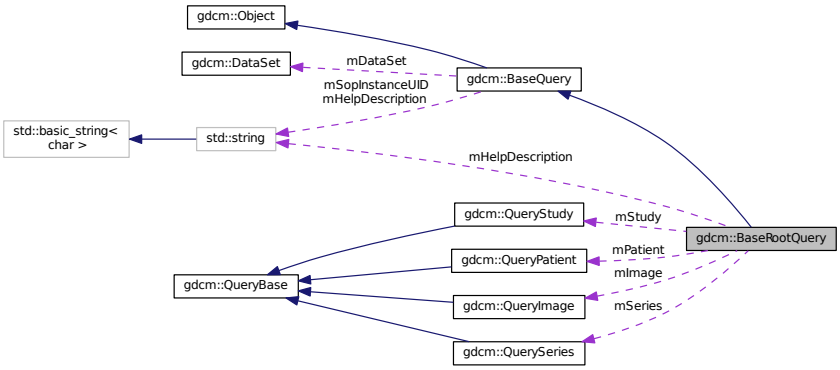
BaseRootQuery contains: a baseclass which will produce a dataset for c-find and c-move with patient/study root.

`#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 **gdcm::BaseRootQuery::BaseRootQuery ( ) [protected]**

10.32.2.2 **virtual gdcm::BaseRootQuery::~BaseRootQuery ( ) [virtual]**

10.32.3 Member Function Documentation

10.32.3.1 **static QueryBase* gdcm::BaseRootQuery::Construct (ERootType *inRootType,* EQueryLevel *qlevel*) [static]**

10.32.3.2 **EQueryLevel gdcm::BaseRootQuery::GetQueryLevelFromQueryRoot (ERootType *roottype*)**

10.32.3.3 **static int gdcm::BaseRootQuery::GetQueryLevelFromString (const char * *str*) [static]**

10.32.3.4 **static const char* gdcm::BaseRootQuery::GetQueryLevelString (EQueryLevel *ql*) [static]**

10.32.3.5 **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 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 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 friend class QueryFactory [friend]

10.32.5 Member Data Documentation

10.32.5.1 std::string gdcm::BaseRootQuery::mHelpDescription [protected]

10.32.5.2 QueryImage gdcm::BaseRootQuery::mImage [protected]

10.32.5.3 QueryPatient gdcm::BaseRootQuery::mPatient [protected]

10.32.5.4 ERootType gdcm::BaseRootQuery::mRootType [protected]

10.32.5.5 QuerySeries gdcm::BaseRootQuery::mSeries [protected]

10.32.5.6 QueryStudy gdcm::BaseRootQuery::mStudy [protected]

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

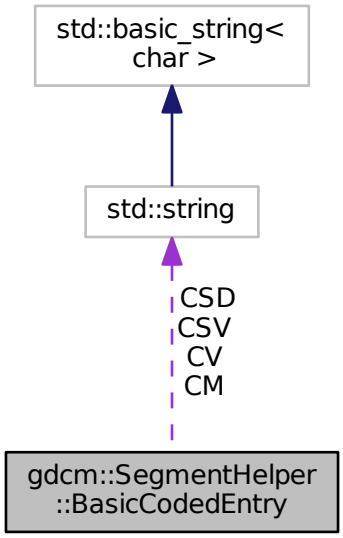
- gdcmBaseRootQuery.h

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 gdcm::SegmentHelper::BasicCodedEntry::BasicCodedEntry ( ) [inline]

Constructor.

10.33.2.2 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 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 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 std::string gdcm::SegmentHelper::BasicCodedEntry::CM

Coding Scheme Version attribute.

10.33.4.2 std::string gdcm::SegmentHelper::BasicCodedEntry::CSD

Code Value attribute.

10.33.4.3 std::string gdcm::SegmentHelper::BasicCodedEntry::CSV

Coding Scheme Designator attribute.

10.33.4.4 std::string gdcm::SegmentHelper::BasicCodedEntry::CV

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

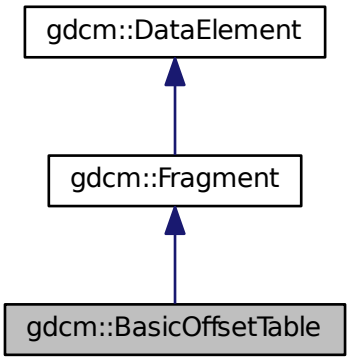
- gdcmSegmentHelper.h

10.34 gdcm::BasicOffsetTable Class Reference

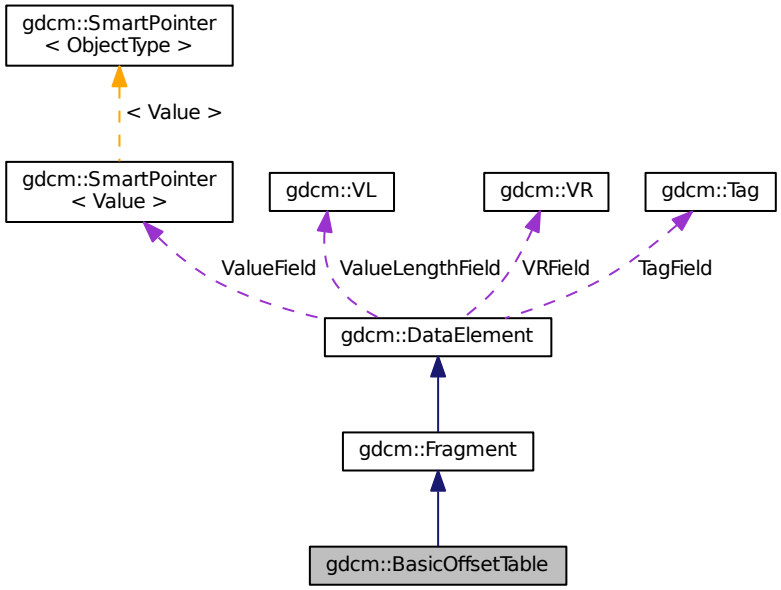
Class to represent a BasicOffsetTable.

`#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 gdcm::BasicOffsetTable::BasicOffsetTable ( ) [inline]

References gdcm::operator<<().

10.34.3 Member Function Documentation

10.34.3.1 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 std::ostream& operator<< (std::ostream & *os,* const BasicOffsetTable & *val*) [friend]

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

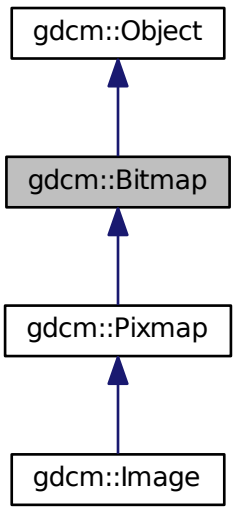
- gdcmBasicOffsetTable.h

10.35 gdcm::Bitmap Class Reference

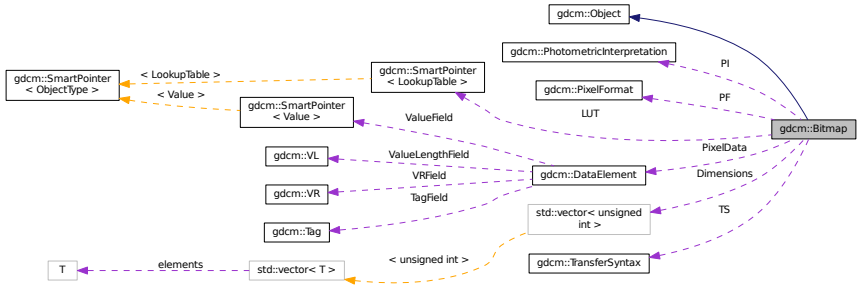
Bitmap class A bitmap based image. Used as parent for both IconImage and the main Pixel Data Image It does not contains any World Space information (IPP, IOP)

`#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 typedef SmartPointer<LookupTable> gdcm::Bitmap::LUTPtr [protected]

10.35.3 Constructor & Destructor Documentation

10.35.3.1 gdcm::Bitmap::Bitmap ( )

10.35.3.2 gdcm::Bitmap::~Bitmap ( )

10.35.4 Member Function Documentation

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

Reimplemented in gdcm::Pixmap.

References gdcm::terminal::dim.

10.35.4.2 void gdcm::Bitmap::Clear ( )

10.35.4.3 bool gdcm::Bitmap::ComputeLossyFlag ( ) [protected]

10.35.4.4 bool gdcm::Bitmap::GetBuffer (char * *buffer*) const

Acces the raw data.

Examples:

ConvertToQImage.cxx, ReadMultiTimesException.cxx, and threadgdcm.cxx.

10.35.4.5 bool gdcm::Bitmap::GetBuffer2 (std::ostream & *os*) const [protected]

10.35.4.6 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, ReadMultiTimesException.cxx, and threadgdcm.cxx.

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

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

Examples:

ExtractIconFromFile.cxx.

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

10.35.4.10 unsigned int gdcm::Bitmap::GetDimension (unsigned int *idx*) const

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

Examples:

ExtractIconFromFile.cxx.

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

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

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

return the photometric interpretation

Examples:

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

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

10.35.4.19 unsigned int gdcm::Bitmap::GetPlanarConfiguration ( ) const

return the planar configuration

10.35.4.20 unsigned int gdcm::Bitmap::GetRows ( ) const [inline]

10.35.4.21 const TransferSyntax& gdcm::Bitmap::GetTransferSyntax ( ) const [inline]

Examples:

ExtractIconFromFile.cxx.

10.35.4.22 bool gdcm::Bitmap::IsEmpty ( ) const [inline]

10.35.4.23 bool gdcm::Bitmap::IsLossy ( ) const

Return whether or not the image was compressed using a lossy compressor or not.

10.35.4.24 bool gdcm::Bitmap::IsTransferSyntaxCompatible (TransferSyntax const & *ts*) const

10.35.4.25 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 void gdcm::Bitmap::SetColumns (unsigned int *col*) [inline]

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

Examples:

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

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

Examples:

CreateARGBImage.cxx, and CreateCMYKImage.cxx.

10.35.4.30 void gdcm::Bitmap::SetLossyFlag (bool *f*) [inline]

Specifically set that the image was compressed using a lossy compression mechanism.

10.35.4.31 void gdcm::Bitmap::SetLUT (LookupTable const & *lut*) [inline]

Set/Get LUT.

10.35.4.32 void gdcm::Bitmap::SetNeedByteSwap (bool *b*) [inline]

10.35.4.33 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 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 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 void gdcm::Bitmap::SetPlanarConfiguration (unsigned int *pc*)

Warning

you need to call SetPixelFormat first (before SetPlanarConfiguration) for consistency checking

10.35.4.37 void gdcm::Bitmap::SetRows (unsigned int *rows*) [inline]

10.35.4.38 void gdcm::Bitmap::SetTransferSyntax (TransferSyntax const & *ts*) [inline]

Transfer syntax.

Examples:

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

10.35.4.39 bool gdcm::Bitmap::TryJPEG2000Codec (char * *buffer,* bool & *lossyflag*) const [protected]

10.35.4.40 bool gdcm::Bitmap::TryJPEG2000Codec2 (std::ostream & *os*) const [protected]

10.35.4.41 bool gdcm::Bitmap::TryJPEGCodec (char * *buffer,* bool & *lossyflag*) const [protected]

10.35.4.42 bool gdcm::Bitmap::TryJPEGCodec2 (std::ostream & *os*) const [protected]

10.35.4.43 bool gdcm::Bitmap::TryJPEGLSCodec (char * *buffer,* bool & *lossyflag*) const [protected]

10.35.4.44 bool gdcm::Bitmap::TryKAKADUCodec (char * *buffer,* bool & *lossyflag*) const [protected]

10.35.4.45 bool gdcm::Bitmap::TryPVRGCodec (char * *buffer,* bool & *lossyflag*) const [protected]

10.35.4.46 bool gdcm::Bitmap::TryRAWCodec (char * *buffer,* bool & *lossyflag*) const [protected]

10.35.4.47 bool gdcm::Bitmap::TryRLECodec (char * *buffer,* bool & *lossyflag*) const [protected]

10.35.5 Friends And Related Function Documentation

10.35.5.1 friend class **ImageChangeTransferSyntax** [friend]

10.35.5.2 friend class **PixmapReader** [friend]

10.35.6 Member Data Documentation

10.35.6.1 std::vector<unsigned int> gdcm::Bitmap::Dimensions [protected]

10.35.6.2 bool gdcm::Bitmap::LossyFlag [protected]

10.35.6.3 **LUTPtr** gdcm::Bitmap::LUT [protected]

10.35.6.4 bool gdcm::Bitmap::NeedByteSwap [protected]

10.35.6.5 unsigned int gdcm::Bitmap::NumberOfDimensions [protected]

10.35.6.6 **PixelFormat** gdcm::Bitmap::PF [protected]

10.35.6.7 **PhotometricInterpretation** gdcm::Bitmap::PI [protected]

10.35.6.8 **DataElement** gdcm::Bitmap::PixelData [protected]

10.35.6.9 unsigned int gdcm::Bitmap::PlanarConfiguration [protected]

10.35.6.10 **TransferSyntax** gdcm::Bitmap::TS [protected]

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

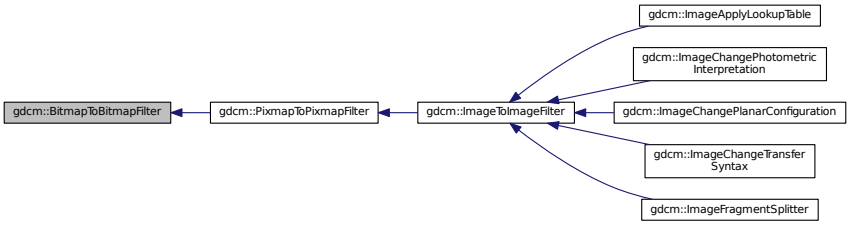
- gdcmBitmap.h

10.36 gdcm::BitmapToBitmapFilter Class Reference

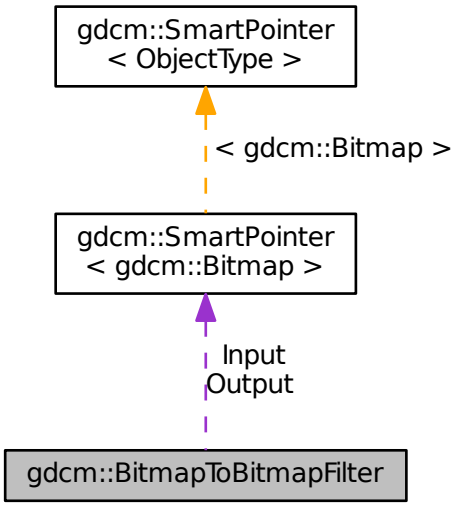
BitmapToBitmapFilter class Super class for all filter taking an image and producing an output image.

`#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 gdcm::BitmapToBitmapFilter::BitmapToBitmapFilter ( )

10.36.2.2 gdcm::BitmapToBitmapFilter::~BitmapToBitmapFilter ( ) [inline]

10.36.3 Member Function Documentation

10.36.3.1 const Bitmap& gdcm::BitmapToBitmapFilter::GetOutput ( ) const [inline]

Get Output image.

10.36.3.2 const Bitmap& gdcm::BitmapToBitmapFilter::GetOutputAsBitmap ( ) const

10.36.3.3 void gdcm::BitmapToBitmapFilter::SetInput (const Bitmap & *image*)

Set input image.

Examples:

CompressImage.cxx.

10.36.4 Member Data Documentation

10.36.4.1 SmartPointer<Bitmap> gdcm::BitmapToBitmapFilter::Input [protected]

10.36.4.2 SmartPointer<Bitmap> gdcm::BitmapToBitmapFilter::Output [protected]

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

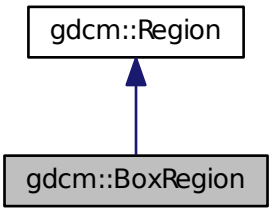
- gdcmBitmapToBitmapFilter.h

10.37 gdcm::BoxRegion Class Reference

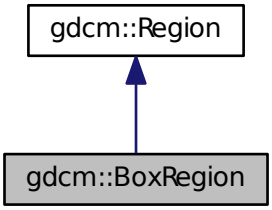
Class for manipulation box region This is a very simple implementation of the Region class. It only support 3D box type region. It assumes the 3D Box does not have a tilt Origin is as (0,0,0)

`#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 gdcm::BoxRegion::BoxRegion ( )

10.37.2.2 gdcm::BoxRegion::~BoxRegion ( )

10.37.2.3 gdcm::BoxRegion::BoxRegion (const BoxRegion &)

copy/cstor and al.

10.37.3 Member Function Documentation

10.37.3.1 size_t gdcm::BoxRegion::Area ( ) const [virtual]

compute the area

Implements gdcm::Region.

10.37.3.2 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 Region* gdcm::BoxRegion::Clone ( ) const [virtual]

Implements gdcm::Region.

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

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

Implements gdcm::Region.

10.37.3.5 bool gdcm::BoxRegion::Empty ( ) const [virtual]

return whether this domain is empty:

Implements gdcm::Region.

10.37.3.6 unsigned int gdcm::BoxRegion::GetXMax ( ) const

10.37.3.7 unsigned int gdcm::BoxRegion::GetXMin ( ) const

Get domain.

10.37.3.8 unsigned int gdcm::BoxRegion::GetYMax ( ) const

10.37.3.9 unsigned int gdcm::BoxRegion::GetYMin ( ) const

10.37.3.10 unsigned int gdcm::BoxRegion::GetZMax ( ) const

10.37.3.11 unsigned int gdcm::BoxRegion::GetZMin ( ) const

10.37.3.12 bool gdcm::BoxRegion::IsValid ( ) const [virtual]

return whether this is valid domain

Implements gdcm::Region.

10.37.3.13 void gdcm::BoxRegion::operator= (const BoxRegion &)

10.37.3.14 void gdcm::BoxRegion::Print (std::ostream & *os* = `std::cout`) const [virtual]

Print.

Reimplemented from gdcm::Region.

10.37.3.15 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 gdcm::ByteBuffer::ByteBuffer ( ) [inline]

10.38.3 Member Function Documentation

10.38.3.1 char* gdcm::ByteBuffer::Get (int *len*) [inline]

10.38.3.2 const char* gdcm::ByteBuffer::GetStart ( ) const [inline]

10.38.3.3 void gdcm::ByteBuffer::ShiftEnd (int *len*) [inline]

10.38.3.4 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 template<class T > static void gdcm::ByteSwap< T >::Swap (T & *p*) [static]

10.39.2.2 template<class T > static void gdcm::ByteSwap< T >::SwapFromSwapCodeIntoSystem (T & *p*, SwapCode const & *sc*) [static]

Examples:

TestByteSwap.cxx.

10.39.2.3 template<class T > static void gdcm::ByteSwap< T >::SwapRange (T * *p*, unsigned int *num*) [static]

10.39.2.4 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 template<class T > static bool gdcm::ByteSwap< T >::SystemIsBigEndian ( ) [static]

Query the machine Endian-ness.

10.39.2.6 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 In place byte-swapping of a dataset FIXME: FL status ??

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

10.40.2.2 gdcm::ByteSwapFilter::~ByteSwapFilter ( )

10.40.3 Member Function Documentation

10.40.3.1 bool gdcm::ByteSwapFilter::ByteSwap ( )

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

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

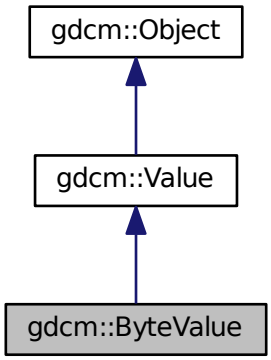
- gdcmByteSwapFilter.h

10.41 gdcm::ByteValue Class Reference

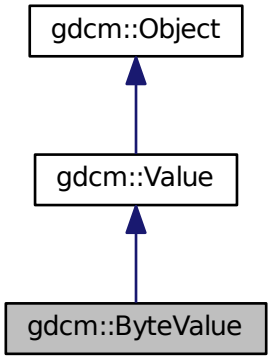
Class to represent binary value (array of bytes)

`#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 gdcm::ByteValue::ByteValue (const char * *array* = 0, VL const & *vl* = 0) [inline]

References gdcmDebugMacro.

10.41.2.2 gdcm::ByteValue::ByteValue (std::vector< char > & *v*) [inline]

Warning

casting to uint32_t

10.41.2.3 gdcm::ByteValue::~ByteValue ( ) [inline]

10.41.3 Member Function Documentation

10.41.3.1 void gdcm::ByteValue::Append (ByteValue const & *bv*)

10.41.3.2 void gdcm::ByteValue::Clear ( ) [inline], [virtual]

Implements gdcm::Value.

10.41.3.3 VL gdcm::ByteValue::ComputeLength ( ) const [inline]

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

10.41.3.4 void gdcm::ByteValue::Fill (char *c*) [inline]

Examples:

DuplicatePCDE.cxx.

10.41.3.5 bool gdcm::ByteValue::GetBuffer (char * *buffer*, unsigned long *length*) const

Examples:

FixJAIBugJPEGLS.cxx.

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

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

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

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

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

Implements gdcm::Value.

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

Reimplemented from gdcm::Object.

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

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

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

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

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

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

To Print Values in Native DICOM format

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

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

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

Implements gdcm::Value.

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

Reimplemented from gdcm::Value.

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

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

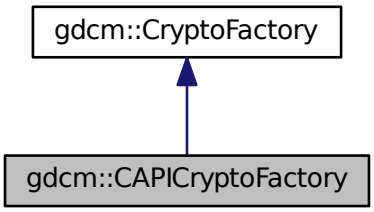
The documentation for this class was generated from the following file:

- gdcmByteValue.h

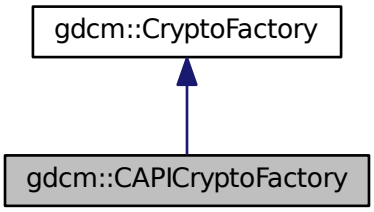
10.42 gdcm::CAPICryptoFactory Class Reference

`#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 gdcm::CAPICryptoFactory::CAPICryptoFactory (CryptoLib *id*)

10.42.2 Member Function Documentation

10.42.2.1 CryptographicMessageSyntax* gdcm::CAPICryptoFactory::CreateCMSProvider ( ) [virtual]

Implements gdcm::CryptoFactory.

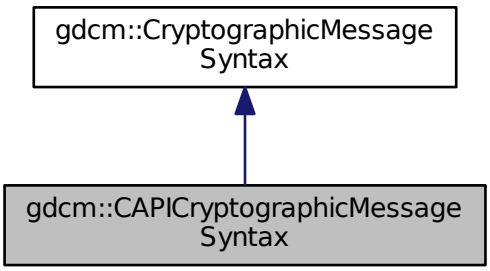
The documentation for this class was generated from the following file:

- gdcmCAPICryptoFactory.h

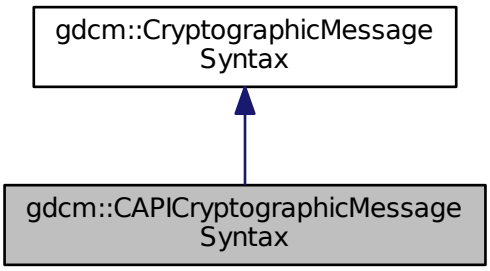
10.43 gdcm::CAPICryptographicMessageSyntax Class Reference

`#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 gdcm::CAPICryptographicMessageSyntax::CAPICryptographicMessageSyntax ( )

10.43.1.2 gdcm::CAPICryptographicMessageSyntax::~CAPICryptographicMessageSyntax ( )

10.43.2 Member Function Documentation

10.43.2.1 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 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 CipherTypes gdcm::CAPICryptographicMessageSyntax::GetCipherType ( ) const [virtual]

Implements gdcm::CryptographicMessageSyntax.

10.43.2.4 bool gdcm::CAPICryptographicMessageSyntax::GetInitialized ( ) const [inline]

10.43.2.5 bool gdcm::CAPICryptographicMessageSyntax::ParseCertificateFile (const char * *filename*) [virtual]

Implements gdcm::CryptographicMessageSyntax.

10.43.2.6 bool gdcm::CAPICryptographicMessageSyntax::ParseKeyFile (const char * *filename*) [virtual]

Implements gdcm::CryptographicMessageSyntax.

10.43.2.7 void gdcm::CAPICryptographicMessageSyntax::SetCipherType (CipherTypes *type*)

10.43.2.8 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:

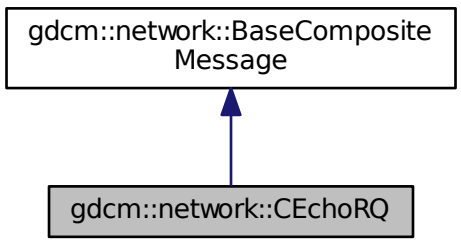
- gdcmCAPICryptographicMessageSyntax.h

10.44 gdcm::network::CEchoRQ Class Reference

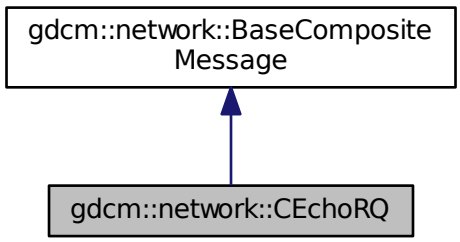
CEchoRQ this file defines the messages for the cecho action.

```
#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 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 UIComp gdcm::network::CEchoRQ::AffectedSOPClassUID

10.44.3.2 uint16_t gdcm::network::CEchoRQ::MessageID

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

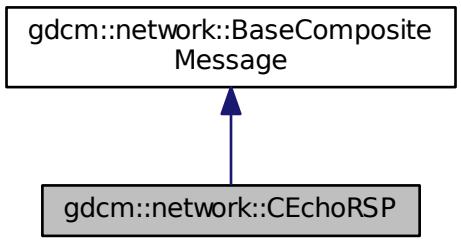
- gdcmCEchoMessages.h
- gdcmDIMSE.h

10.45 gdcm::network::CEchoRSP Class Reference

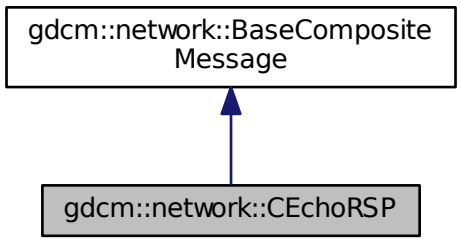
CEchoRSP this file defines the messages for the cecho action.

`#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 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:

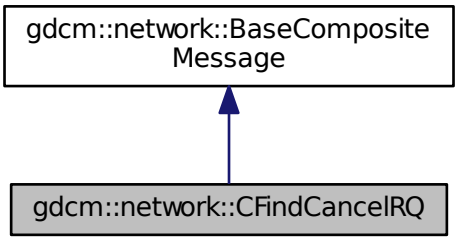
- gdcmDIMSE.h

10.47 gdcm::network::CFindCancelRQ Class Reference

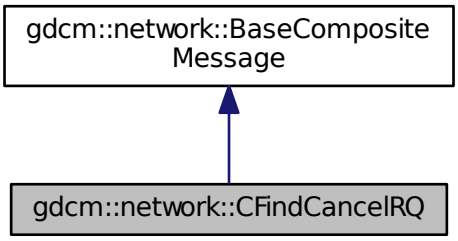
CFindCancelRQ this file defines the messages for the cfind action.

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

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

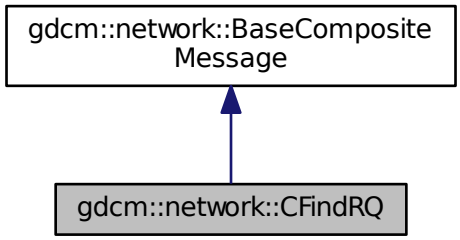
- gdcmCFindMessages.h

10.48 gdcm::network::CFindRQ Class Reference

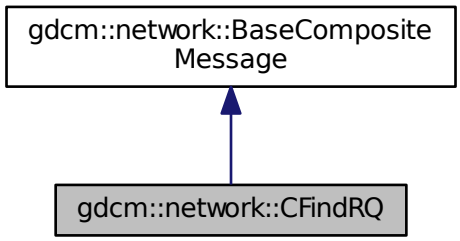
CFindRQ this file defines the messages for the cfind action.

```
#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 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:

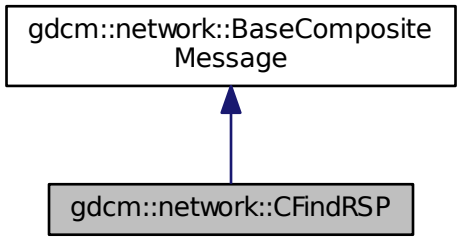
- gdcmCFindMessages.h

10.49 gdcm::network::CFindRSP Class Reference

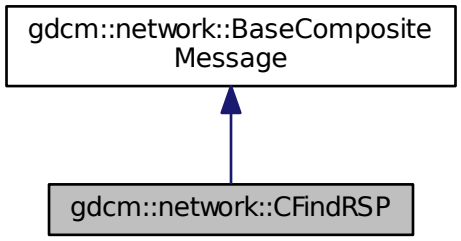
CFindRSP this file defines the messages for the cfind action.

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

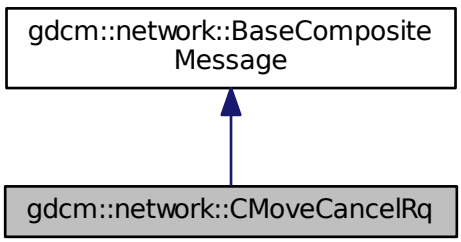
The documentation for this class was generated from the following file:

- gdcmCFindMessages.h

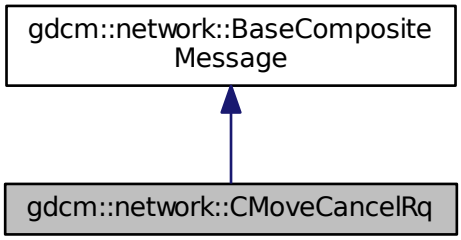
10.50 gdcm::network::CMoveCancelRq Class Reference

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

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

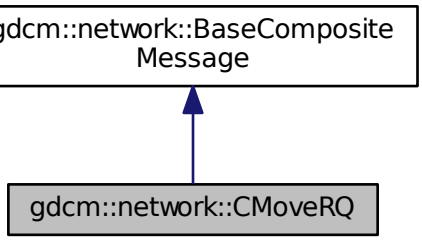
- gdcmCMoveMessages.h

10.51 gdcm::network::CMoveRQ Class Reference

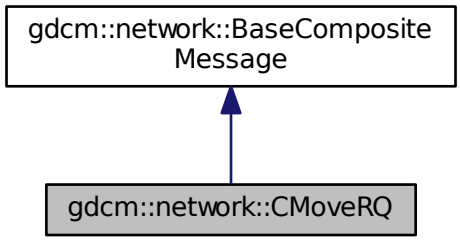
CMoveRQ this file defines the messages for the cmove action.

`#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 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:

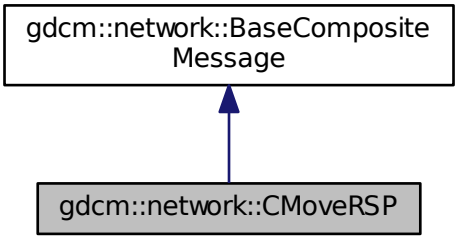
- gdcmCMoveMessages.h

10.52 gdcm::network::CMoveRSP Class Reference

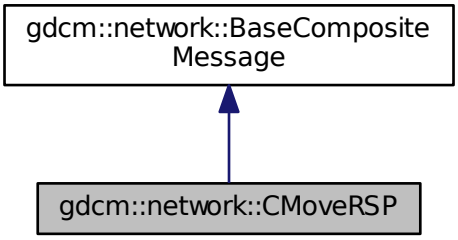
CMoveRSP this file defines the messages for the cmove action.

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

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

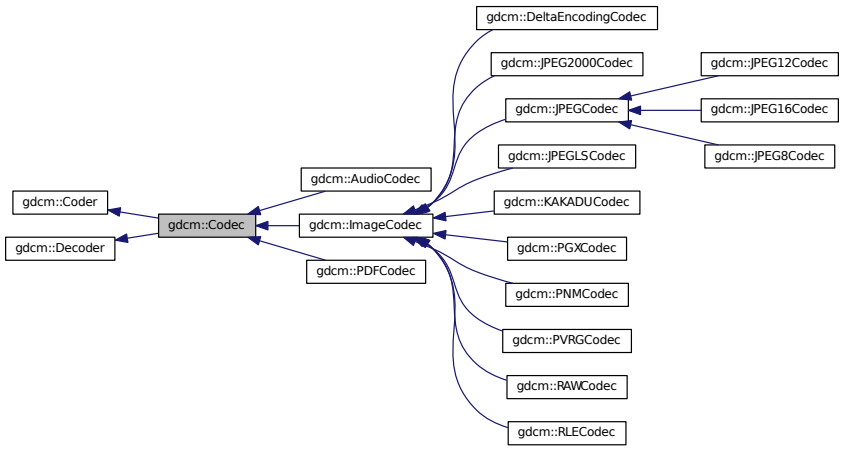
- gdcmCMoveMessages.h

10.53 gdcm::Codec Class Reference

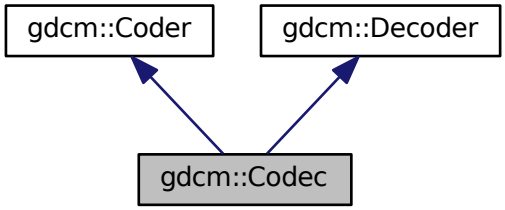
Codec class.

`#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:

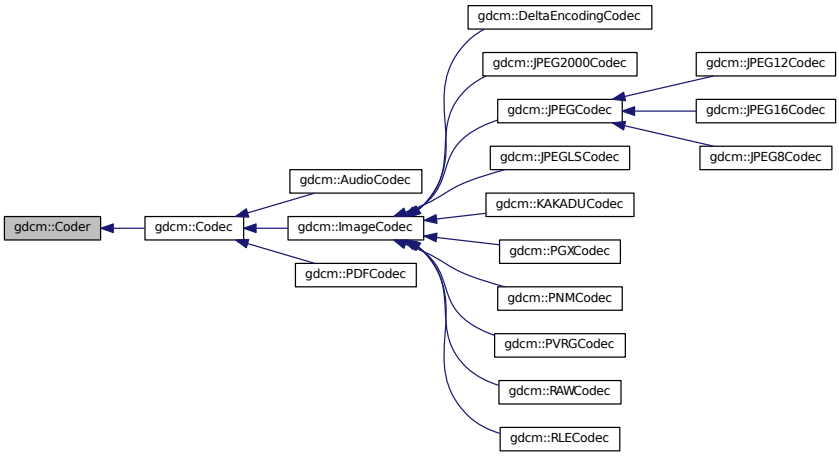
- gdcmCodec.h

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

10.54.3 Member Function Documentation

10.54.3.1 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 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 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 This is an implementation of DICOM VR: CS The cstor will properly Trim so that operator== is correct.

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

10.55.2.2 typedef InternalClass::const_reference gdcm::CodeString::const_reference

10.55.2.3 typedef InternalClass::const_reverse_iterator gdcm::CodeString::const_reverse_iterator

10.55.2.4 typedef InternalClass::difference_type gdcm::CodeString::difference_type

10.55.2.5 typedef InternalClass::iterator gdcm::CodeString::iterator

10.55.2.6 typedef InternalClass::pointer gdcm::CodeString::pointer

10.55.2.7 typedef InternalClass::reference gdcm::CodeString::reference

10.55.2.8 typedef InternalClass::reverse_iterator gdcm::CodeString::reverse_iterator

10.55.2.9 typedef InternalClass::size_type gdcm::CodeString::size_type

10.55.2.10 typedef InternalClass::value_type gdcm::CodeString::value_type

10.55.3 Constructor & Destructor Documentation

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

CodeString constructors.

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

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

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

Return the full code string as std::string.

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

Check if CodeString obj is correct..

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

Return the size of the string.

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

10.55.5 Friends And Related Function Documentation

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

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

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

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

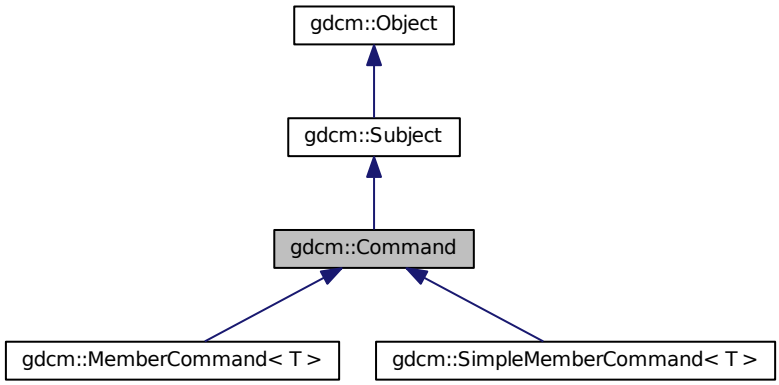
- gdcmCodeString.h

10.56 gdcm::Command Class Reference

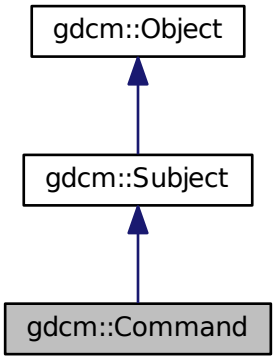
Command superclass for callback/observer methods.

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

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

10.56.3 Member Function Documentation

10.56.3.1 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 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:

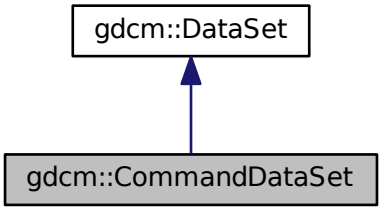
- gdcmCommand.h

10.57 gdcm::CommandDataSet Class Reference

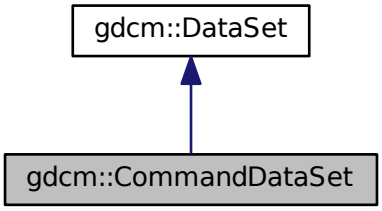
Class to represent a Command DataSet.

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

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

References gdcm::operator<<().

10.57.3 Member Function Documentation

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

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

10.57.3.2 std::istream& gdcm::CommandDataSet::Read (std::istream & *is*)

Read.

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

References gdcm::DataElement::GetTag().

10.57.3.4 std::ostream& gdcm::CommandDataSet::Write (std::ostream & *os*) const

Write.

10.57.4 Friends And Related Function Documentation

10.57.4.1 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 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).

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

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

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

10.58.2.4 static std::vector<PresentationDataValue> gdcm::network::CompositeMessageFactory::ConstructCStoreRQ (const ULConnection & *inConnection,* const File & *file,* bool *writeDataSet* = true) [static]

10.58.2.5 static std::vector<PresentationDataValue> gdcm::network::CompositeMessageFactory::ConstructCStoreRSP (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 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:

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

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

10.59.3 Member Function Documentation

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

Deprecated

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

10.60.3 Member Function Documentation

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

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

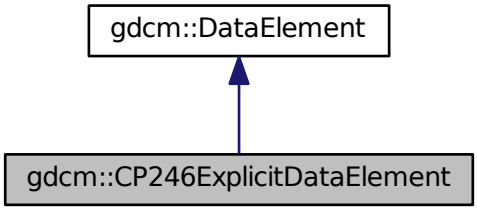
- gdcmConstCharWrapper.h

10.61 gdcm::CP246ExplicitDataElement Class Reference

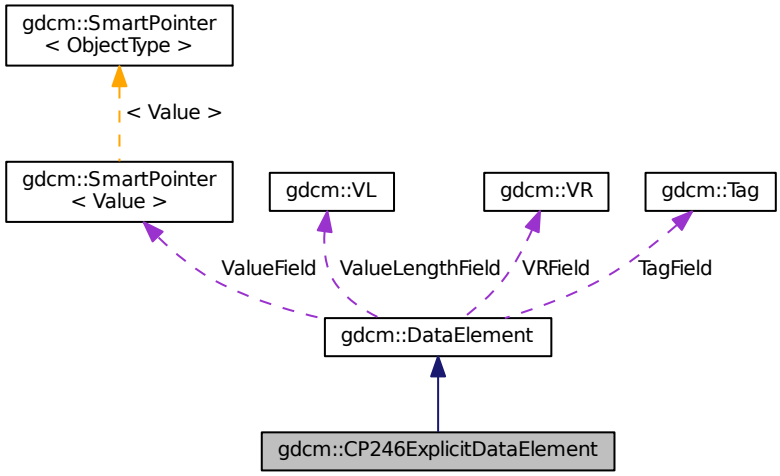
Class to read/write a DataElement as CP246Explicit Data Element.

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

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

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

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

10.61.2.5 template<typename TSwap > std::istream& gdcm::CP246ExplicitDataElement::ReadWithLength (std::istream & *is*, VL & *length*)

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

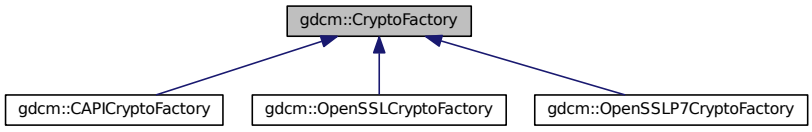
- gdcmCP246ExplicitDataElement.h

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 enum gdcm::CryptoFactory::CryptoLib

Enumerator

DEFAULT

OPENSSL

CAPI

OPENSSLP7

10.62.3 Constructor & Destructor Documentation

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

References gdcmErrorMacro.

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

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

10.62.4 Member Function Documentation

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

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

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

Examples:

BasicAnonymizer.cs, and ClinicalTrialIdentificationWorkflow.cs.

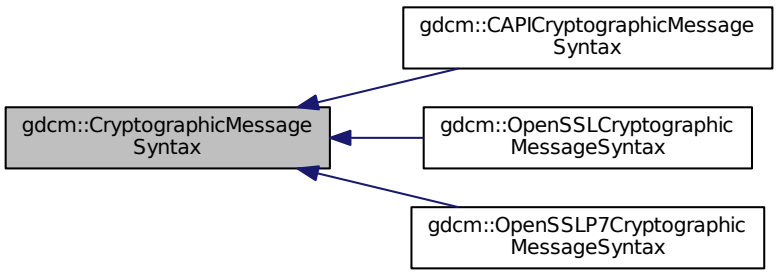
The documentation for this class was generated from the following file:

- 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 enum gdcm::CryptographicMessageSyntax::CipherTypes

Enumerator

DES3_CIPHER

AES128_CIPHER

AES192_CIPHER

AES256_CIPHER

10.63.2 Constructor & Destructor Documentation

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

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

10.63.3 Member Function Documentation

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

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

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

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

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

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

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

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

10.63.3.7 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:

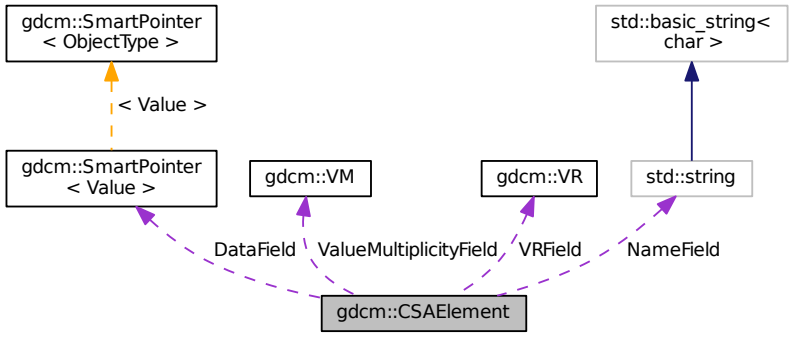
- gdcmCryptographicMessageSyntax.h

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

10.64.3 Constructor & Destructor Documentation

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

References gdcm::operator<<().

10.64.3.2 gdcm::CSAElement::CSAElement (const CSAElement & *_val*) [inline]

10.64.4 Member Function Documentation

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

Set/Get Key.

Referenced by operator<().

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

Set/Get Name.

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

Set/Get NoOfItems.

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

Set/Get SyngoDT.

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

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

Set/Get VM.

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

Set/Get VR.

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

Check if CSA Element is empty.

Examples:

csa2img.cxx.

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

References GetKey().

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

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

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

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

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

Set.

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

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

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

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

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

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

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

10.64.5 Friends And Related Function Documentation

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

10.64.6 Member Data Documentation

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

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

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

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

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

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

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

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

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

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

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

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

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

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

10.65.4 Member Function Documentation

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

10.65.4.4 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 static const PrivateTag& gdcm::CSAHeader::GetCSAImageHeaderInfoTag ( ) [static]

Return the private tag used by SIEMENS to store the CSA Image Header This is: PrivateTag(0x0029,0x0010,"SIEMENS CSA HEADER");

Examples:

csa2img.cxx, and PublicDict.cxx.

10.65.4.6 static const PrivateTag& gdcm::CSAHeader::GetCSASeriesHeaderInfoTag ( ) [static]

Return the private tag used by SIEMENS to store the CSA Series Header This is: PrivateTag(0x0029,0x0020,"SIEMENS CSA HEADER");

Examples:

MrProtocol.cxx.

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

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

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

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

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

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

10.65.4.10 bool gdcm::CSAHeader::LoadFromDataElement (DataElement const & *de*)

Decode the CSAHeader from element 'de'.

Examples:

csa2img.cxx, and MrProtocol.cxx.

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

10.65.4.13 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 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 typedef MapCSAHeaderDictEntry::const_iterator gdcm::CSAHeaderDict::ConstIterator

10.66.2.2 typedef MapCSAHeaderDictEntry::iterator gdcm::CSAHeaderDict::Iterator

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

10.66.3 Constructor & Destructor Documentation

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

References gdcm::operator<<().

10.66.4 Member Function Documentation

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

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

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

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

Examples:

MrProtocol.cxx.

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

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

10.66.5 Friends And Related Function Documentation

10.66.5.1 friend class Dicts [friend]

10.66.5.2 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 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.

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

Set/Get Description.

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

Set/Get Name.

Referenced by operator<().

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

Set/Get VM.

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

Set/Get VR.

10.67.3.5 bool gdcm::CSAHeaderDictEntry::operator< (const CSAHeaderDictEntry & *entry*) const [inline]

References GetName().

10.67.3.6 void gdcm::CSAHeaderDictEntry::SetDescription (const char * *desc*) [inline]

10.67.3.7 void gdcm::CSAHeaderDictEntry::SetName (const char * *name*) [inline]

10.67.3.8 void gdcm::CSAHeaderDictEntry::SetVM (VM const & *vm*) [inline]

10.67.3.9 void gdcm::CSAHeaderDictEntry::SetVR (const VR & *vr*) [inline]

10.67.4 Friends And Related Function Documentation

10.67.4.1 std::ostream& operator<< (std::ostream & *_os*, const CSAHeaderDictEntry & *_val*) [friend]

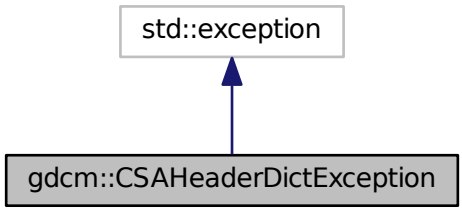
The documentation for this class was generated from the following file:

- gdcmCSAHeaderDictEntry.h

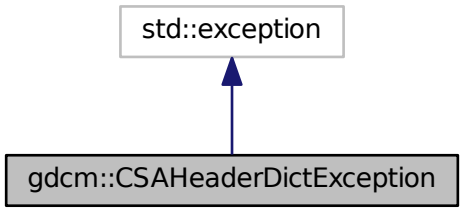
10.68 gdcm::CSAHeaderDictException Class Reference

`#include <gdcmCSAHeaderDict.h>`

Inheritance diagram for gdcm::CSAHeaderDictException:



Collaboration diagram for gdcm::CSAHeaderDictException:



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

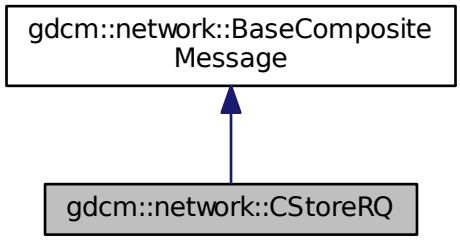
- gdcmCSAHeaderDict.h

10.69 gdcm::network::CStoreRQ Class Reference

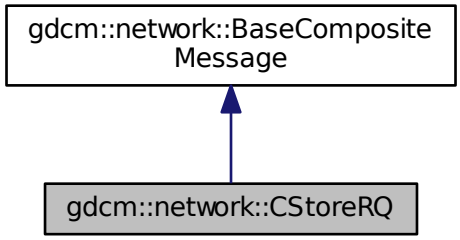
CStoreRQ this file defines the messages for the cecho action.

`#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 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:

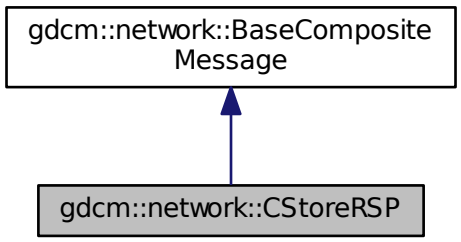
- gdcmCStoreMessages.h

10.70 gdcm::network::CStoreRSP Class Reference

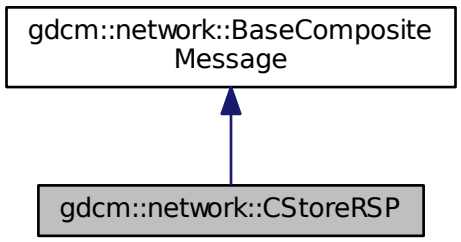
CStoreRSP this file defines the messages for the cecho action.

`#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 std::vector<PresentationDataValue> gdcm::network::CStoreRSP::ConstructPDV (const DataSet * *inDataSet,* const BasePDU * *inPC*)

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

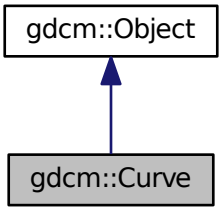
- gdcmCStoreMessages.h

10.71 gdcm::Curve Class Reference

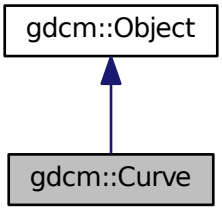
Curve class to handle element 50xx,3000 Curve Data WARNING: This is deprecated and lastly defined in PS 3.3 - 2004.

`#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 gdcm::Curve::Curve ( )

10.71.2.2 gdcm::Curve::~Curve ( )

10.71.2.3 gdcm::Curve::Curve (Curve const & *ov*)

10.71.3 Member Function Documentation

10.71.3.1 void gdcm::Curve::Decode (std::istream & *is*, std::ostream & *os*)

10.71.3.2 void gdcm::Curve::GetAsPoints (float * *array*) const

10.71.3.3 std::vector<unsigned short> const& gdcm::Curve::GetCurveDataDescriptor ( ) const

10.71.3.4 unsigned short gdcm::Curve::GetDataValueRepresentation ( ) const

10.71.3.5 unsigned short gdcm::Curve::GetDimensions ( ) const

10.71.3.6 unsigned short gdcm::Curve::GetGroup ( ) const

10.71.3.7 static unsigned int gdcm::Curve::GetNumberOfCurves (DataSet const & *ds*) [static]

10.71.3.8 unsigned short gdcm::Curve::GetNumberOfPoints ( ) const

10.71.3.9 const char* gdcm::Curve::GetTypeOfData ( ) const

10.71.3.10 const char* gdcm::Curve::GetTypeOfDataDescription ( ) const

10.71.3.11 bool gdcm::Curve::IsEmpty ( ) const

10.71.3.12 void gdcm::Curve::Print (std::ostream &) const [virtual]

Reimplemented from gdcm::Object.

10.71.3.13 void gdcm::Curve::SetCoordinateStartValue (unsigned short *v*)

10.71.3.14 void gdcm::Curve::SetCoordinateStepValue (unsigned short *v*)

10.71.3.15 void gdcm::Curve::SetCurve (const char * *array,* unsigned int *length*)

10.71.3.16 void gdcm::Curve::SetCurveDataDescriptor (const uint16_t * *values,* size_t *num*)

10.71.3.17 void gdcm::Curve::SetCurveDescription (const char * *curvedescription*)

10.71.3.18 void gdcm::Curve::SetDataValueRepresentation (unsigned short *datavaluerepresentation*)

10.71.3.19 void gdcm::Curve::SetDimensions (unsigned short *dimensions*)

10.71.3.20 void gdcm::Curve::SetGroup (unsigned short *group*)

10.71.3.21 void gdcm::Curve::SetNumberOfPoints (unsigned short *numberofpoints*)

10.71.3.22 void gdcm::Curve::SetTypeOfData (const char * *typeofdata*)

10.71.3.23 void gdcm::Curve::Update (const **DataElement** & *de*)

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

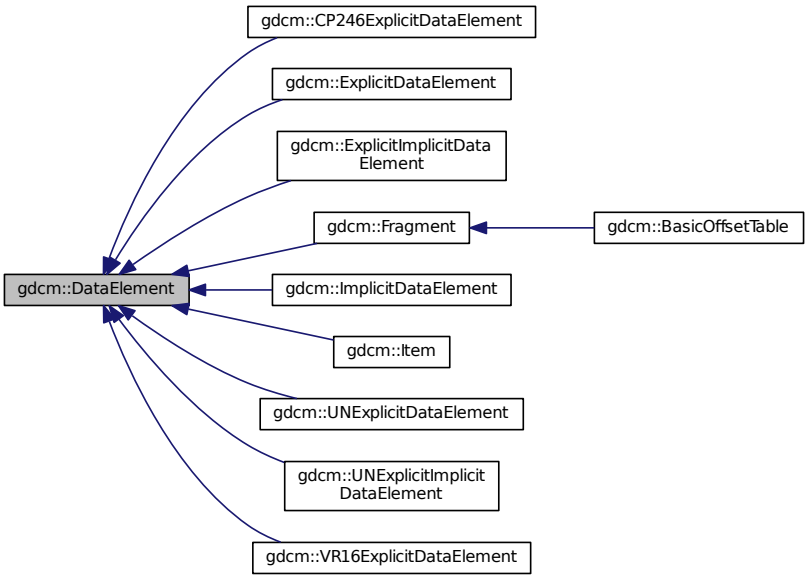
- gdcmCurve.h

10.72 gdcm::DataElement Class Reference

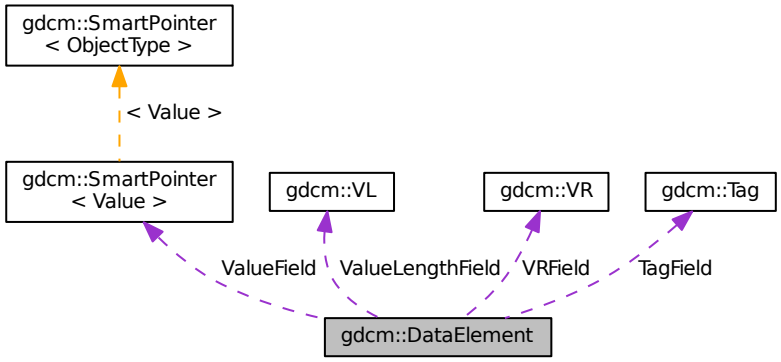
Class to represent a Data Element either Implicit or Explicit.

```
#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 typedef SmartPointer<Value> gdcm::DataElement::ValuePtr [protected]

10.72.3 Constructor & Destructor Documentation

10.72.3.1 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 gdcm::DataElement::DataElement (const DataElement & *_val*) [inline]

10.72.4 Member Function Documentation

10.72.4.1 void gdcm::DataElement::Clear ( ) [inline]

Clear Data Element (make Value empty and invalidate Tag & VR)

10.72.4.2 void gdcm::DataElement::Empty ( ) [inline]

Make Data Element empty (no Value)

10.72.4.3 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 template<typename TDE > VL gdcm::DataElement::GetLength ( ) const [inline]

10.72.4.5 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 SequenceOfFragments* gdcm::DataElement::GetSequenceOfFragments ( )

10.72.4.7 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::CommandDataSet::Replace(), gdcm::FileMetaInformation::Replace(), gdcm::Attribute< Group, Element, TVR, TVM >::SetFromDataElement(), gdcm::Attribute< Group, Element, TVR, VM::VM1 >::SetFromDataElement(), and gdcm::Attribute< Group, Element, TVR, VM::VM1_n >::SetFromDataElement().

10.72.4.8 Tag& gdcm::DataElement::GetTag ( ) [inline]

10.72.4.9 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 Value& gdcm::DataElement::GetValue ( ) [inline]

10.72.4.11 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 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 VL& gdcm::DataElement::GetVL ( ) [inline]

10.72.4.14 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 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 bool gdcm::DataElement::IsUndefinedLength ( ) const [inline]

return if Value Length if of undefined length

10.72.4.17 bool gdcm::DataElement::operator< (const DataElement & *de*) const [inline]

References GetTag().

10.72.4.18 DataElement& gdcm::DataElement::operator= (const DataElement & *de*) [inline]

References TagField, ValueField, ValueLengthField, and VRField.

10.72.4.19 bool gdcm::DataElement::operator== (const DataElement & *de*) const [inline]

References TagField, ValueField, ValueLengthField, and VRField.

10.72.4.20 template<typename TDE , typename TSwap > std::istream& gdcm::DataElement::Read (std::istream & *is*) [inline]

10.72.4.21 template<typename TDE , typename TSwap > std::istream& gdcm::DataElement::ReadOrSkip (std::istream & *is,* std::set< Tag > const & *skiptags*) [inline]

10.72.4.22 template<typename TDE , typename TSwap > std::istream& gdcm::DataElement::ReadPreValue (std::istream & *is,* std::set< Tag > const & *skiptags*) [inline]

10.72.4.23 template<typename TDE , typename TSwap > std::istream& gdcm::DataElement::ReadValue (std::istream & *is,* std::set< Tag > const & *skiptags*) [inline]

10.72.4.24 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 template<typename TDE , typename TSwap > std::istream& gdcm::DataElement::ReadWithLength (std::istream & *is,* VL & *length*) [inline]

10.72.4.26 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 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 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 void gdcm::DataElement::SetValueFieldLength (VL *vl,* bool *readvalues*) [protected]

10.72.4.30 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 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 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 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 std::ostream& operator<< (std::ostream & *_os,* const DataElement & *_val*) [friend]

10.72.6 Member Data Documentation

10.72.6.1 Tag gdcm::DataElement::TagField [protected]

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

10.72.6.2 ValuePtr gdcm::DataElement::ValueField [protected]

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

10.72.6.3 VL gdcm::DataElement::ValueLengthField [protected]

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

10.72.6.4 VR gdcm::DataElement::VRField [protected]

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

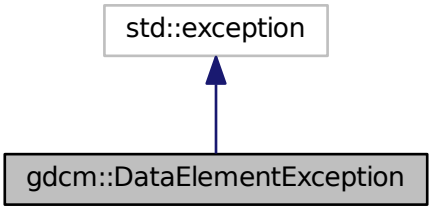
The documentation for this class was generated from the following file:

- gdcmDataElement.h

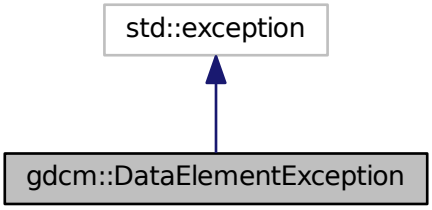
10.73 gdcm::DataElementException Class Reference

`#include <gdcmDataSet.h>`

Inheritance diagram for gdcm::DataElementException:



Collaboration diagram for gdcm::DataElementException:



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

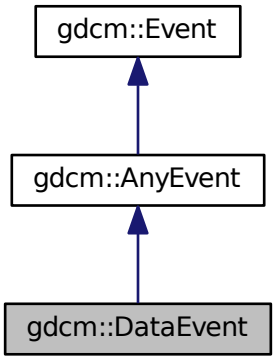
- gdcmDataSet.h

10.74 gdcm::DataEvent Class Reference

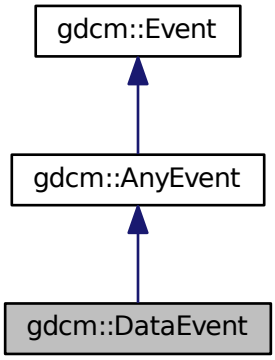
DataEvent.

`#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 typedef DataEvent gdcm::DataEvent::Self

10.74.2.2 typedef AnyEvent gdcm::DataEvent::Superclass

10.74.3 Constructor & Destructor Documentation

10.74.3.1 gdcm::DataEvent::DataEvent (const char * *bytes* = `0`, size_t *len* = `0`) [inline]

10.74.3.2 virtual gdcm::DataEvent::~DataEvent ( ) [inline], [virtual]

10.74.3.3 gdcm::DataEvent::DataEvent (const Self & *s*) [inline]

10.74.4 Member Function Documentation

10.74.4.1 virtual bool gdcm::DataEvent::CheckEvent (const ::gdcm::Event * *e*) const [inline], [virtual]

10.74.4.2 const char* gdcm::DataEvent::GetData ( ) const [inline]

10.74.4.3 size_t gdcm::DataEvent::GetDataLength ( ) const [inline]

10.74.4.4 virtual const char* gdcm::DataEvent::GetEventName ( ) const [inline], [virtual]

Return the StringName associated with the event.

Implements gdcm::Event.

10.74.4.5 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 void gdcm::DataEvent::SetData (const char * *bytes*, size_t *len*) [inline]

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

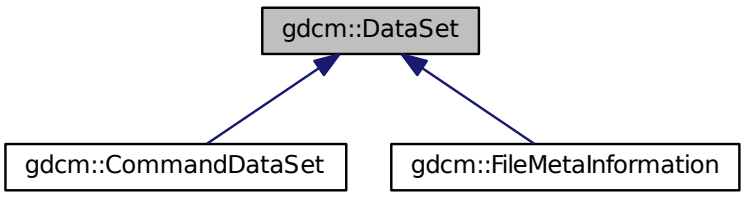
- gdcmDataEvent.h

10.75 gdcm::DataSet Class Reference

Class to represent a Data Set (which contains Data Elements) A Data Set represents an instance of a real world Information Object.

`#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 typedef DataElementSet::const_iterator gdcm::DataSet::ConstIterator

10.75.2.2 typedef std::set<DataElement> gdcm::DataSet::DataElementSet

10.75.2.3 typedef DataElementSet::iterator gdcm::DataSet::Iterator

10.75.2.4 typedef DataElementSet::size_type gdcm::DataSet::SizeType

10.75.3 Member Function Documentation

10.75.3.1 ConstIterator gdcm::DataSet::Begin ( ) const [inline]

Examples:

DiffFile.cxx, DumpGEMSMovieGroup.cxx, and DuplicatePCDE.cxx.

Referenced by gdcm::operator<<().

10.75.3.2 Iterator gdcm::DataSet::Begin ( ) [inline]

10.75.3.3 void gdcm::DataSet::Clear ( ) [inline]

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

10.75.3.4 Tag gdcm::DataSet::ComputeDataElement (const PrivateTag & *t*) const [protected]

10.75.3.5 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 ConstIterator gdcm::DataSet::End ( ) const [inline]

Examples:

DiffFile.cxx, DumpGEMSMovieGroup.cxx, and DuplicatePCDE.cxx.

10.75.3.7 Iterator gdcm::DataSet::End ( ) [inline]

10.75.3.8 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 bool gdcm::DataSet::FindDataElement (const Tag & *t*) const [inline]

10.75.3.10 const DataElement& gdcm::DataSet::FindNextDataElement (const Tag & *t*) const [inline]

Examples:

DuplicatePCDE.cxx.

10.75.3.11 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 const DataElement& gdcm::DataSet::GetDataElement (const PrivateTag & *t*) const

Return the dataelement.

10.75.3.13 const DataElement& gdcm::DataSet::GetDEEnd ( ) const [protected]

10.75.3.14 const DataElementSet& gdcm::DataSet::GetDES ( ) const [inline]

Examples:

ReadAndDumpDICOMDIR.cxx.

10.75.3.15 DataElementSet& gdcm::DataSet::GetDES ( ) [inline]

10.75.3.16 template<typename TDE > VL gdcm::DataSet::GetLength ( ) const [inline]

10.75.3.17 MediaStorage gdcm::DataSet::GetMediaStorage ( ) const

10.75.3.18 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 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 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 bool gdcm::DataSet::IsEmpty ( ) const [inline]

Returns if the dataset is empty.

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

10.75.3.22 const DataElement& gdcm::DataSet::operator() (uint16_t *group*, uint16_t *element*) const [inline]

10.75.3.23 DataSet& gdcm::DataSet::operator= (DataSet const & *val*) [inline]

10.75.3.24 const DataElement& gdcm::DataSet::operator[] (const Tag & *t*) const [inline]

10.75.3.25 void gdcm::DataSet::Print (std::ostream & *os*, std::string const & *indent* = " ") const [inline]

Referenced by gdcm::operator<<().

10.75.3.26 template<typename TDE , typename TSwap > std::istream& gdcm::DataSet::Read (std::istream & *is*)

10.75.3.27 template<typename TDE , typename TSwap > std::istream& gdcm::DataSet::ReadNested (std::istream & *is*)

10.75.3.28 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 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 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 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 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 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 template<typename TDE , typename TSwap > std::istream& gdcm::DataSet::ReadWithLength (std::istream & *is*, VL & *length*)

10.75.3.35 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 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 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 SizeType gdcm::DataSet::Size ( ) const [inline]

Examples:

DumpGEMSMovieGroup.cxx.

Referenced by gdcm::SequenceOfItems::Read().

10.75.3.39 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 friend class CSAHeader [friend]

10.75.4.2 std::ostream& operator<< (std::ostream & *_os,* const DataSet & *val*) [friend]

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

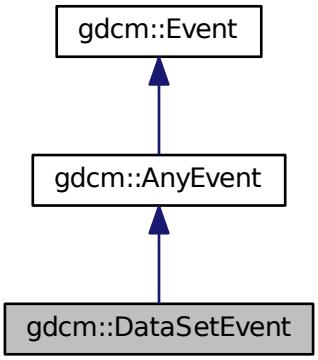
- gdcmDataSet.h

10.76 gdcm::DataSetEvent Class Reference

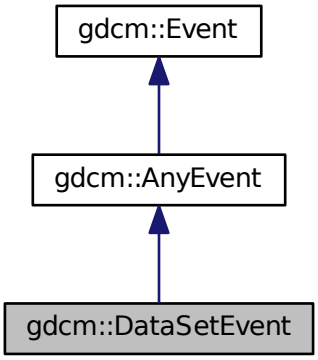
DataSetEvent Special type of event triggered during the DataSet store/move process.

`#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 typedef DataSetEvent gdcm::DataSetEvent::Self

10.76.2.2 typedef AnyEvent gdcm::DataSetEvent::Superclass

10.76.3 Constructor & Destructor Documentation

10.76.3.1 gdcm::DataSetEvent::DataSetEvent (DataSet const * *ds* = NULL) [inline]

10.76.3.2 virtual gdcm::DataSetEvent::~DataSetEvent ( ) [inline], [virtual]

10.76.3.3 gdcm::DataSetEvent::DataSetEvent (const Self & *s*) [inline]

10.76.4 Member Function Documentation

10.76.4.1 virtual bool gdcm::DataSetEvent::CheckEvent (const ::gdcm::Event * *e*) const [inline], [virtual]

10.76.4.2 DataSet const& gdcm::DataSetEvent::GetDataSet ( ) const [inline]

10.76.4.3 virtual const char* gdcm::DataSetEvent::GetEventName ( ) const [inline], [virtual]

Return the StringName associated with the event.

Implements gdcm::Event.

10.76.4.4 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 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:

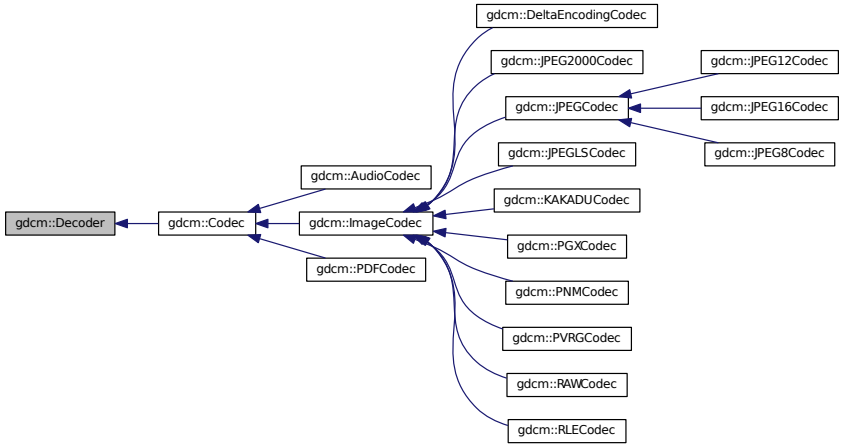
- gdcmDataSetHelper.h

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 virtual gdcm::Decoder::~Decoder ( ) [inline], [virtual]

10.78.3 Member Function Documentation

10.78.3.1 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 virtual bool gdcm::Decoder::Decode (DataElement const & , DataElement &) [inline], [virtual]

Decode.

Reimplemented in gdcm::JPEGCodec, gdcm::RLECodec, gdcm::JPEGLSCodec, gdcm::PVRGCodec, gdcm::JPEG2000Codec, gdcm::ImageCodec, gdcm::DeltaEncodingCodec, gdcm::KAKADUCodec, gdcm::RAWCodec, gdcm::AudioCodec, and gdcm::PDFCodec.

10.78.3.3 virtual bool gdcm::Decoder::DecodeByStreams (std::istream & , std::ostream &) [inline], [protected], [virtual]

Reimplemented in gdcm::JPEGCodec, gdcm::JPEG2000Codec, gdcm::RLECodec, gdcm::ImageCodec, gdcm::RAWCodec, 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 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 gdcm::Defs::Defs ( )

10.80.2.2 gdcm::Defs::~Defs ( )

10.80.3 Member Function Documentation

10.80.3.1 const IOD& gdcm::Defs::GetIODFromFile (const File & *file*) const

10.80.3.2 static const char* gdcm::Defs::GetIODNameFromMediaStorage (MediaStorage const & *ms*) [static]

Examples:

GenerateStandardSOPClasses.cxx.

10.80.3.3 const IODs& gdcm::Defs::GetIODs ( ) const [inline]

Examples:

TraverseModules.cxx.

10.80.3.4 IODs& gdcm::Defs::GetIODs ( ) [inline]

10.80.3.5 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 Macros& gdcm::Defs::GetMacros ( ) [inline]

10.80.3.7 const Modules& gdcm::Defs::GetModules ( ) const [inline]

Examples:

TraverseModules.cxx.

10.80.3.8 Modules& gdcm::Defs::GetModules ( ) [inline]

10.80.3.9 Type gdcm::Defs::GetTypeFromTag (const File & *file,* const Tag & *tag*) const

10.80.3.10 bool gdcm::Defs::IsEmpty ( ) const [inline]

10.80.3.11 void gdcm::Defs::LoadDefaults ( ) [protected]

10.80.3.12 void gdcm::Defs::LoadFromFile (const char * *filename*) [protected]

10.80.3.13 bool gdcm::Defs::Verify (const File & *file*) const

10.80.3.14 bool gdcm::Defs::Verify (const DataSet & *ds*) const

10.80.4 Friends And Related Function Documentation

10.80.4.1 friend class Global [friend]

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

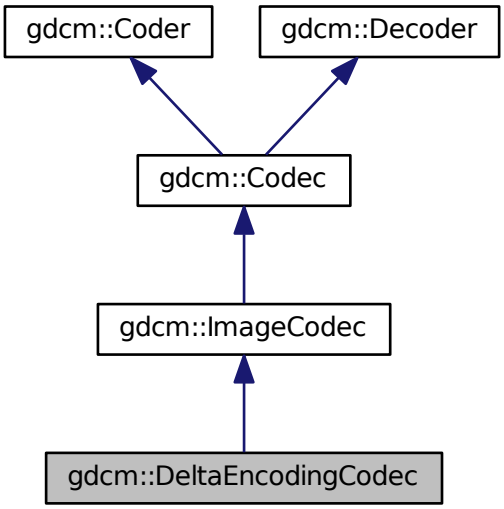
- gdcmDefs.h

10.81 gdcm::DeltaEncodingCodec Class Reference

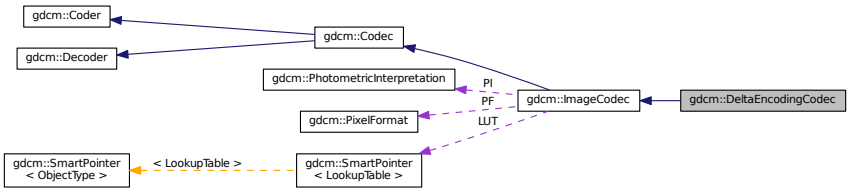
DeltaEncodingCodec compression used by some private vendor.

`#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 gdcm::DeltaEncodingCodec::DeltaEncodingCodec ( )

10.81.2.2 gdcm::DeltaEncodingCodec::~DeltaEncodingCodec ( )

10.81.3 Member Function Documentation

10.81.3.1 bool gdcm::DeltaEncodingCodec::CanDecode (TransferSyntax const & *ts*)

10.81.3.2 bool gdcm::DeltaEncodingCodec::Decode (DataElement const &, DataElement &) [virtual]

Decode.

Reimplemented from gdcm::Decoder.

10.81.3.3 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 gdcm::DICOMDIR::DICOMDIR ( ) [inline]

10.82.2.2 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 This is a STD-GEN-CD DICOMDIR generator. ref: PS 3.11-2008 Annex D (Normative) - General Purpose CD-R and DVD Interchange Profiles.

```
#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 typedef Directory::FilenamesType gdcm::DICOMDIRGenerator::FilenamesType

10.83.2.2 typedef Directory::FilenameType gdcm::DICOMDIRGenerator::FilenameType

10.83.3 Constructor & Destructor Documentation

10.83.3.1 gdcm::DICOMDIRGenerator::DICOMDIRGenerator ( )

10.83.3.2 gdcm::DICOMDIRGenerator::∼DICOMDIRGenerator ( )

10.83.4 Member Function Documentation

10.83.4.1 bool gdcm::DICOMDIRGenerator::AddImageDirectoryRecord ( ) [protected]

10.83.4.2 bool gdcm::DICOMDIRGenerator::AddPatientDirectoryRecord ( ) [protected]

10.83.4.3 bool gdcm::DICOMDIRGenerator::AddSeriesDirectoryRecord ( ) [protected]

10.83.4.4 bool gdcm::DICOMDIRGenerator::AddStudyDirectoryRecord ( ) [protected]

10.83.4.5 bool gdcm::DICOMDIRGenerator::Generate ( )

Main function to generate the DICOMDIR.

10.83.4.6 File& gdcm::DICOMDIRGenerator::GetFile ( )

10.83.4.7 Scanner& gdcm::DICOMDIRGenerator::GetScanner ( ) [protected]

10.83.4.8 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 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 void gdcm::DICOMDIRGenerator::SetFilenames (FilenamesType const & *fns*)

Set the list of filenames from which the DICOMDIR should be generated from.

10.83.4.11 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 typedef MapDictEntry::const_iterator gdcm::Dict::ConstIterator

10.84.2.2 typedef MapDictEntry::iterator gdcm::Dict::Iterator

10.84.2.3 typedef std::map<Tag, DictEntry> gdcm::Dict::MapDictEntry

10.84.3 Constructor & Destructor Documentation

10.84.3.1 gdcm::Dict::Dict ( ) [inline]

References gdcm::operator<<().

10.84.4 Member Function Documentation

10.84.4.1 void gdcm::Dict::AddDictEntry (const Tag & *tag,* const DictEntry & *de*) [inline]

10.84.4.2 ConstIterator gdcm::Dict::Begin ( ) const [inline]

Examples:

GenAllVR.cxx, and GenFakeIdentifyFile.cxx.

10.84.4.3 ConstIterator gdcm::Dict::End ( ) const [inline]

Examples:

GenAllVR.cxx, and GenFakeIdentifyFile.cxx.

10.84.4.4 const DictEntry& gdcm::Dict::GetDictEntry (const Tag & *tag*) const [inline]

Examples:

GenFakeIdentifyFile.cxx, and PublicDict.cxx.

10.84.4.5 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 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 const char* gdcm::Dict::GetKeywordFromTag (Tag const & *tag*) const [inline]

Function to return the Keyword from a Tag.

10.84.4.8 bool gdcm::Dict::IsEmpty ( ) const [inline]

Referenced by gdcm::Dicts::IsEmpty().

10.84.4.9 void gdcm::Dict::LoadDefault ( ) [protected]

10.84.5 Friends And Related Function Documentation

10.84.5.1 friend class Dicts [friend]

10.84.5.2 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 enum gdcm::DictConverter::OutputTypes

Enumerator

DICT_DEFAULT

DICT_DEBUG

DICT_XML

10.85.3 Constructor & Destructor Documentation

10.85.3.1 gdcm::DictConverter::DictConverter ( )

10.85.3.2 gdcm::DictConverter::~DictConverter ( )

10.85.4 Member Function Documentation

10.85.4.1 void gdcm::DictConverter::AddGroupLength ( ) [protected]

10.85.4.2 void gdcm::DictConverter::Convert ( )

10.85.4.3 bool gdcm::DictConverter::ConvertToCXX (const char * *raw*, std::string & *cxx*) [protected]

10.85.4.4 bool gdcm::DictConverter::ConvertToXML (const char * *raw*, std::string & *cxx*) [protected]

10.85.4.5 const std::string& gdcm::DictConverter::GetDictName ( ) const

10.85.4.6 const std::string& gdcm::DictConverter::GetInputFilename ( ) const

10.85.4.7 const std::string& gdcm::DictConverter::GetOutputFilename ( ) const

10.85.4.8 int gdcm::DictConverter::GetOutputType ( ) const [inline]

10.85.4.9 static bool gdcm::DictConverter::Readuint16 (const char * *raw*, uint16_t & *ov*) [static]

10.85.4.10 static bool gdcm::DictConverter::ReadVM (const char * *raw*, VM::VMType & *type*) [static]

10.85.4.11 static bool gdcm::DictConverter::ReadVR (const char * *raw*, VR::VRType & *type*) [static]

10.85.4.12 void gdcm::DictConverter::SetDictName (const char * *name*)

10.85.4.13 void gdcm::DictConverter::SetInputFileName (const char * *filename*)

10.85.4.14 void gdcm::DictConverter::SetOutputFileName (const char * *filename*)

10.85.4.15 void gdcm::DictConverter::SetOutputType (int *type*) [inline]

10.85.4.16 void gdcm::DictConverter::WriteFooter ( ) [protected]

10.85.4.17 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 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.

`#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 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 const char* gdcm::DictEntry::GetKeyword ( ) const [inline]

same as GetName but without spaces...

10.86.3.2 const char* gdcm::DictEntry::GetName ( ) const [inline]

Set/Get Name.

Referenced by gdcm::PrivateDict::PrintXML().

10.86.3.3 bool gdcm::DictEntry::GetRetired ( ) const [inline]

Set/Get Retired flag.

Examples:

GenAllVR.cxx.

10.86.3.4 const VM& gdcm::DictEntry::GetVM ( ) const [inline]

Set/Get VM.

Referenced by gdcm::PrivateDict::AddDictEntry(), and gdcm::PrivateDict::PrintXML().

10.86.3.5 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 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 void gdcm::DictEntry::SetElementXX (bool *v*) [inline]

Set whether element is shared in multiple elements (Source Image IDs typically)

10.86.3.8 void gdcm::DictEntry::SetGroupXX (bool *v*) [inline]

Set whether element is shared in multiple groups (Curve/Overlay typically)

10.86.3.9 void gdcm::DictEntry::SetKeyword (const char * *keyword*) [inline]

10.86.3.10 void gdcm::DictEntry::SetName (const char * *name*) [inline]

10.86.3.11 void gdcm::DictEntry::SetRetired (bool *retired*) [inline]

10.86.3.12 void gdcm::DictEntry::SetVM (VM const & *vm*) [inline]

10.86.3.13 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 friend class Dict [friend]

10.86.4.2 std::ostream& operator<< (std::ostream & *_os,* const DictEntry & *_val*) [friend]

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

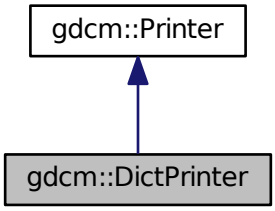
- gdcmDictEntry.h

10.87 gdcm::DictPrinter Class Reference

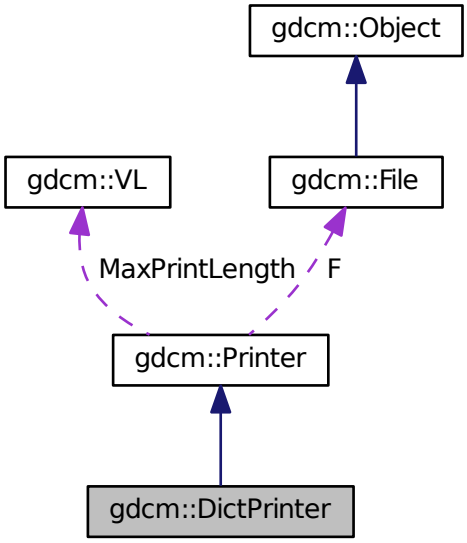
DictPrinter class.

```
#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 gdcm::DictPrinter::DictPrinter ( )

10.87.2.2 gdcm::DictPrinter::~DictPrinter ( )

10.87.3 Member Function Documentation

10.87.3.1 void gdcm::DictPrinter::Print (std::ostream & *os*)

10.87.3.2 void gdcm::DictPrinter::PrintDataElement2 (std::ostream & *os*, const DataSet & *ds*, const DataElement & *ide*) [protected]

10.87.3.3 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 enum gdcm::Dicts::ConstructorType [protected]

Enumerator

PHILIPS

GEMS

SIEMENS

10.88.3 Constructor & Destructor Documentation

10.88.3.1 gdcm::Dicts::Dicts ( )

10.88.3.2 gdcm::Dicts::~Dicts ( )

10.88.4 Member Function Documentation

10.88.4.1 static const char* gdcm::Dicts::GetConstructorString (ConstructorType *type*) [static], [protected]

10.88.4.2 const CSAHeaderDict& gdcm::Dicts::GetCSAHeaderDict ( ) const

Examples:

MrProtocol.cxx.

10.88.4.3 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 const DictEntry& gdcm::Dicts::GetDictEntry (const PrivateTag & *tag*) const

10.88.4.5 const PrivateDict& gdcm::Dicts::GetPrivateDict ( ) const

10.88.4.6 PrivateDict& gdcm::Dicts::GetPrivateDict ( )

10.88.4.7 const Dict& gdcm::Dicts::GetPublicDict ( ) const

Examples:

GenAllVR.cxx, GenFakeIdentifyFile.cxx, PublicDict.cxx, and ReadAndPrintAttributes.cxx.

10.88.4.8 bool gdcm::Dicts::IsEmpty ( ) const [inline]

References gdcm::Dict::IsEmpty().

10.88.4.9 void gdcm::Dicts::LoadDefaults ( ) [protected]

10.88.5 Friends And Related Function Documentation

10.88.5.1 friend class Global [friend]

10.88.5.2 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 PS 3.7 - 2009 Annex E Command Dictionary (Normative) E.1 REGISTRY OF DICOM COMMAND ELEMENTS Table E.1-1 COMMAND FIELDS (PART 1)

```
#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 enum gdcm::network::DIMSE::CommandTypes

Enumerator

C_STORE_RQ

C_STORE_RSP

C_GET_RQ

C_GET_RSP

C_FIND_RQ

C_FIND_RSP

C_MOVE_RQ

C_MOVE_RSP

C_ECHO_RQ

C_ECHO_RSP

N_EVENT_REPORT_RQ
N_EVENT_REPORT_RSP
N_GET_RQ
N_GET_RSP
N_SET_RQ
N_SET_RSP
N_ACTION_RQ
N_ACTION_RSP
N_CREATE_RQ
N_CREATE_RSP
N_DELETE_RQ
N_DELETE_RSP
C_CANCEL_RQ

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

- gdcmDIMSE.h

10.90 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 gdcm::DirectionCosines::DirectionCosines ( )

10.90.2.2 gdcm::DirectionCosines::DirectionCosines (const double *dircos[6]*)

10.90.2.3 gdcm::DirectionCosines::~DirectionCosines ( )

10.90.3 Member Function Documentation

10.90.3.1 double gdcm::DirectionCosines::ComputeDistAlongNormal (const double *ipp[3]*) const

Compute the distance along the normal.

10.90.3.2 void gdcm::DirectionCosines::Cross (double *z[3]*) const

Compute Cross product.

10.90.3.3 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 double gdcm::DirectionCosines::Dot ( ) const

Compute Dot.

10.90.3.5 bool gdcm::DirectionCosines::IsValid ( ) const

Return whether or not this is a valid direction cosines.

10.90.3.6 void gdcm::DirectionCosines::Normalize ( )

Normalize in-place.

10.90.3.7 gdcm::DirectionCosines::operator const double * ( ) const [inline]

Make the class behave like a const double *.

10.90.3.8 void gdcm::DirectionCosines::Print (std::ostream &) const

Print.

10.90.3.9 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 typedef std::vector<FilenameType> gdcm::Directory::FilenamesType

Examples:

DiscriminateVolume.cxx.

10.91.2.2 typedef std::string gdcm::Directory::FilenameType

10.91.3 Constructor & Destructor Documentation

10.91.3.1 gdcm::Directory::Directory ( ) [inline]

10.91.3.2 gdcm::Directory::~Directory ( ) [inline]

10.91.4 Member Function Documentation

10.91.4.1 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 FilenamesType const& gdcm::Directory::GetDirectories ( ) const [inline]

Return the Directories traversed.

10.91.4.3 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 FilenameType const& gdcm::Directory::GetToplevel ( ) const [inline]

Get the name of the toplevel directory.

10.91.4.5 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 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 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 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.

`#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 static **Directory::FilenamesType** gdcm::DirectoryHelper::GetCTImageSeriesUIDs (const std::string & *inDirectory*) [static]

10.92.2.2 static **Directory::FilenamesType** gdcm::DirectoryHelper::GetFilenamesFromSeriesUIDs (const std::string & *inDirectory,* const std::string & *inSeriesUID*) [static]

Examples:

GenerateRTSTRUCT.cxx.

10.92.2.3 static std::string gdcm::DirectoryHelper::GetFrameOfReference (const std::vector< **DataSet** > & *inDS*) [static]

10.92.2.4 static **Directory::FilenamesType** gdcm::DirectoryHelper::GetMRImageSeriesUIDs (const std::string & *inDirectory*) [static]

10.92.2.5 static **Directory::FilenamesType** gdcm::DirectoryHelper::GetRTStructSeriesUIDs (const std::string & *inDirectory*) [static]

Examples:

GenerateRTSTRUCT.cxx.

10.92.2.6 static **Directory::FilenamesType** gdcm::DirectoryHelper::GetSeriesUIDsBySOPClassUID (const std::string & *inDirectory,* const std::string & *inSOPClassUID*) [static]

10.92.2.7 static std::string gdcm::DirectoryHelper::GetSOPClassUID (const std::vector< **DataSet** > & *inDS*) [static]

10.92.2.8 static std::string gdcm::DirectoryHelper::GetStringValueFromTag (const **Tag** & *t,* const **DataSet** & *ds*) [static]

10.92.2.9 static std::vector<**DataSet**> gdcm::DirectoryHelper::LoadImageFromFiles (const std::string & *inDirectory,* const std::string & *inSeriesUID*) [static]

10.92.2.10 static std::string gdcm::DirectoryHelper::RetrieveSOPInstanceUIDFromIndex (int *inIndex,* const std::vector< **DataSet** > & *inDS*) [static]

10.92.2.11 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 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:

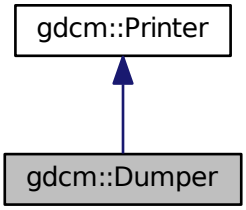
- gdcmDummyValueGenerator.h

10.94 gdcm::Dumper Class Reference

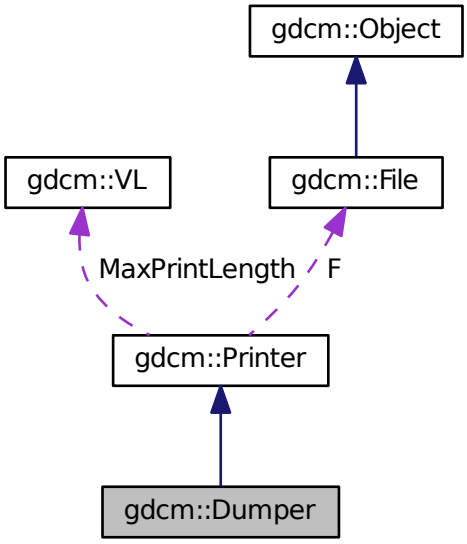
Codec class.

`#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 gdcm::Dumper::Dumper ( ) [inline]

10.94.2.2 gdcm::Dumper::~Dumper ( ) [inline]

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

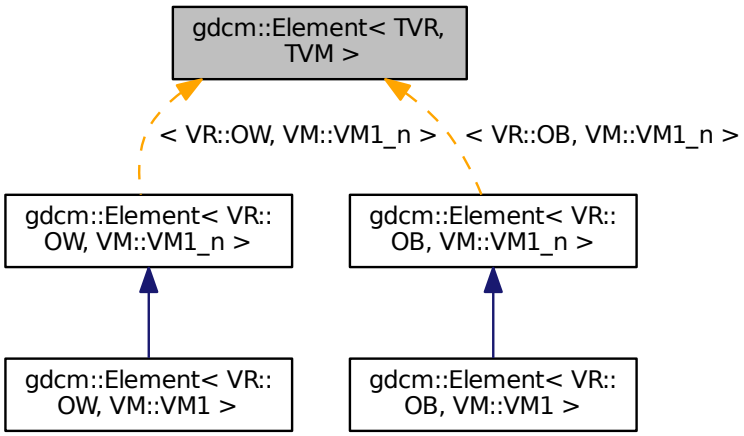
- gdcmDumper.h

10.95 gdcm::Element< TVR, TVM > Class Template Reference

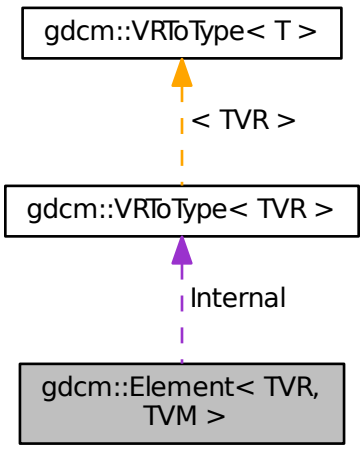
Element class.

`#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 template<int TVR, int TVM> typedef VRToType<TVR>::Type gdcm::Element< TVR, TVM >::Type

10.95.3 Member Function Documentation

10.95.3.1 template<int TVR, int TVM> DataElement gdcm::Element< TVR, TVM >::GetAsDataElement ( ) const [inline]

10.95.3.2 template<int TVR, int TVM> unsigned long gdcm::Element< TVR, TVM >::GetLength ( ) const [inline]

10.95.3.3 template<int TVR, int TVM> const VRToType<TVR>::Type& gdcm::Element< TVR, TVM >::GetValue (unsigned int *idx* = 0) const [inline]

10.95.3.4 template<int TVR, int TVM> VRToType<TVR>::Type& gdcm::Element< TVR, TVM >::GetValue (unsigned int *idx* = 0) [inline]

10.95.3.5 template<int TVR, int TVM> const VRToType<TVR>::Type* gdcm::Element< TVR, TVM >::GetValues ( ) const [inline]

10.95.3.6 template<int TVR, int TVM> static VM gdcm::Element< TVR, TVM >::GetVM ( ) [inline], [static]

10.95.3.7 template<int TVR, int TVM> static VR gdcm::Element< TVR, TVM >::GetVR ( ) [inline], [static]

10.95.3.8 template<int TVR, int TVM> VRToType<TVR>::Type gdcm::Element< TVR, TVM >::operator[ ] (unsigned int *idx*) const [inline]

10.95.3.9 template<int TVR, int TVM> void gdcm::Element< TVR, TVM >::Print (std::ostream & *_os*) const [inline]

10.95.3.10 template<int TVR, int TVM> void gdcm::Element< TVR, TVM >::Read (std::istream & *_is*) [inline]

10.95.3.11 template<int TVR, int TVM> void gdcm::Element< TVR, TVM >::Set (Value const & *v*) [inline]

10.95.3.12 template<int TVR, int TVM> void gdcm::Element< TVR, TVM >::SetFromDataElement (DataElement< TVR, TVM > const & *de*) [inline]

10.95.3.13 template<int TVR, int TVM> void gdcm::Element< TVR, TVM >::SetNoSwap (Value const & *v*) [inline], [protected]

10.95.3.14 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 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 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=().

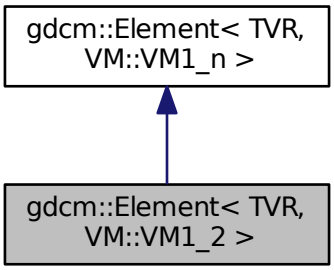
The documentation for this class was generated from the following file:

- gdcmElement.h

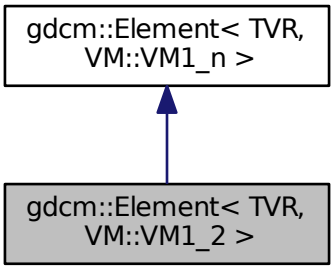
10.96 gdcm::Element< TVR, VM::VM1_2 > Class Template Reference

`#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 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 template<int TVR> void gdcm::Element< TVR, VM::VM1_2 >::SetLength (int *len*) [inline]

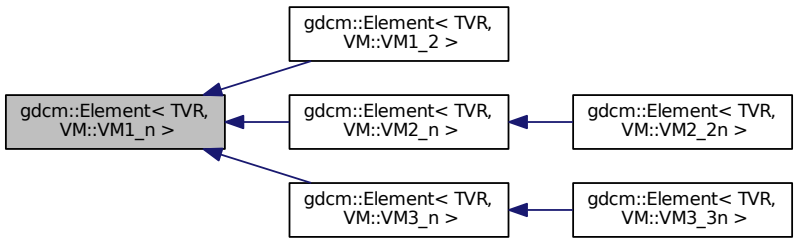
The documentation for this class was generated from the following file:

- 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 template<int TVR> typedef VRToType<TVR>::Type gdcm::Element< TVR, VM::VM1_n >::Type

10.97.2 Constructor & Destructor Documentation

10.97.2.1 template<int TVR> gdcm::Element< TVR, VM::VM1_n >::Element ( ) [inline], [explicit]

10.97.2.2 template<int TVR> gdcm::Element< TVR, VM::VM1_n >::~Element ( ) [inline]

10.97.2.3 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 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 template<int TVR> unsigned long gdcm::Element< TVR, VM::VM1_n >::GetLength ( ) const [inline]

10.97.3.3 template<int TVR> const VRToType<TVR>::Type& gdcm::Element< TVR, VM::VM1_n >::GetValue (unsigned int *idx* = 0) const [inline]

10.97.3.4 template<int TVR> VRToType<TVR>::Type& gdcm::Element< TVR, VM::VM1_n >::GetValue (unsigned int *idx* = 0) [inline]

10.97.3.5 template<int TVR> static VM gdcm::Element< TVR, VM::VM1_n >::GetVM ( ) [inline], [static]

10.97.3.6 template<int TVR> static VR gdcm::Element< TVR, VM::VM1_n >::GetVR ( ) [inline], [static]

10.97.3.7 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 template<int TVR> VRToType<TVR>::Type gdcm::Element< TVR, VM::VM1_n >::operator[ ] (unsigned int *idx*) const [inline]

10.97.3.9 template<int TVR> void gdcm::Element< TVR, VM::VM1_n >::Print (std::ostream & *_os*) const [inline]

10.97.3.10 template<int TVR> void gdcm::Element< TVR, VM::VM1_n >::Read (std::istream & *_is*) [inline]

10.97.3.11 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 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 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 template<int TVR> void gdcm::Element< TVR, VM::VM1_n >::SetLength (unsigned long *len*) [inline]

10.97.3.15 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 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 template<int TVR> void gdcm::Element< TVR, VM::VM1_n >::Write (std::ostream & *_os*) const [inline]

10.97.3.18 template<int TVR> void gdcm::Element< TVR, VM::VM1_n >::WriteASCII (std::ostream & *os*) const [inline]

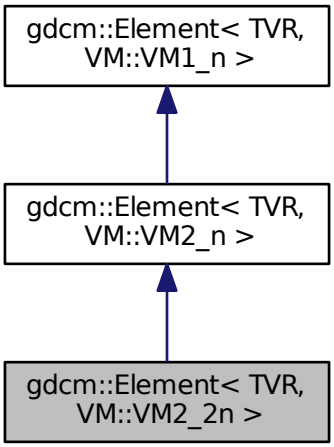
The documentation for this class was generated from the following file:

- gdcmElement.h

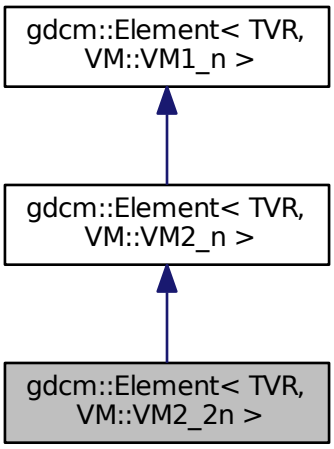
10.98 gdcm::Element< TVR, VM::VM2_2n > Class Template Reference

`#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 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 template<int TVR> void gdcm::Element< TVR, VM::VM2_2n >::SetLength (int *len*) [inline]

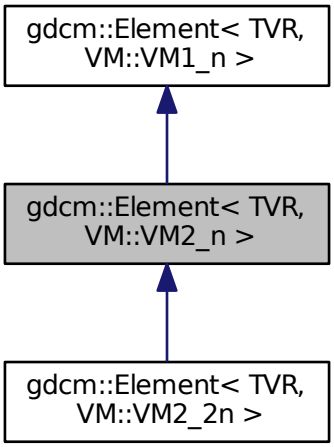
The documentation for this class was generated from the following file:

- gdcmElement.h

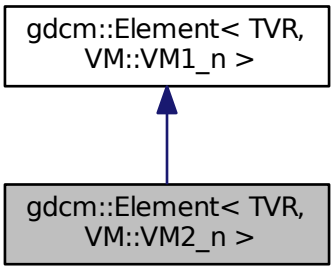
10.99 gdcm::Element< TVR, VM::VM2_n > Class Template Reference

`#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 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 template<int TVR> void gdcm::Element< TVR, VM::VM2_n >::SetLength (int *len*) [inline]

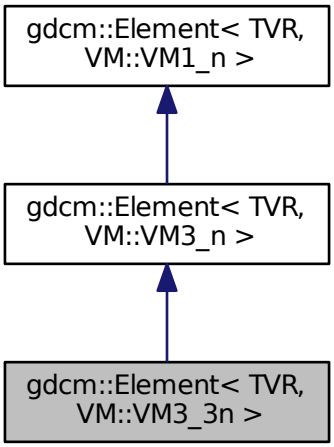
The documentation for this class was generated from the following file:

- gdcmElement.h

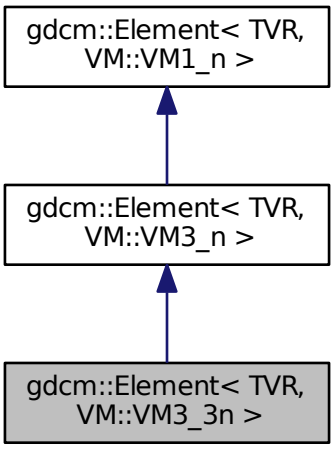
10.100 gdcm::Element< TVR, VM::VM3_3n > Class Template Reference

`#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 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 template<int TVR> void gdcm::Element< TVR, VM::VM3_3n >::SetLength (int *len*) [inline]

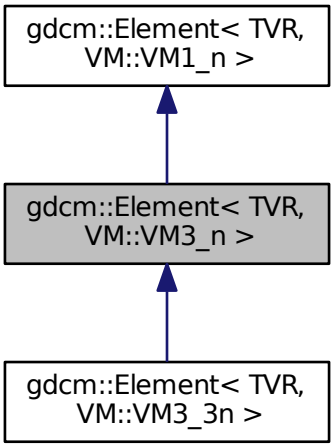
The documentation for this class was generated from the following file:

- gdcmElement.h

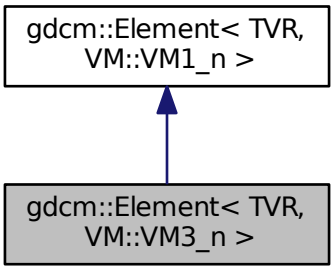
10.101 gdcm::Element< TVR, VM::VM3_n > Class Template Reference

`#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 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 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 unsigned long gdcm::Element< VR::AS, VM::VM5 >::GetLength ( ) const [inline]

10.102.1.2 void gdcm::Element< VR::AS, VM::VM5 >::Print (std::ostream & *_os*) const [inline]

10.102.2 Member Data Documentation

10.102.2.1 char gdcm::Element< VR::AS, VM::VM5 >::Internal[VMToLength< VM::VM5 >::Length *sizeof(VRToType< VR::AS >::Type)]

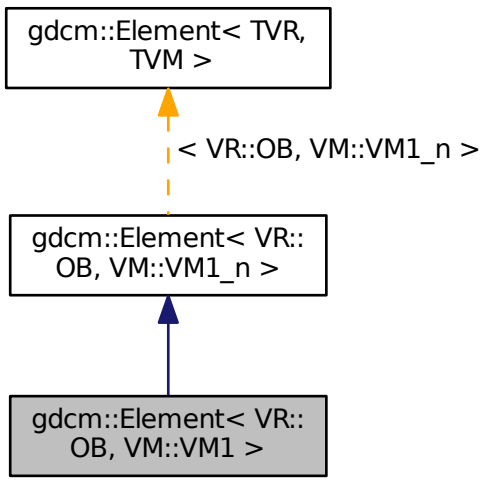
The documentation for this class was generated from the following file:

- gdcmElement.h

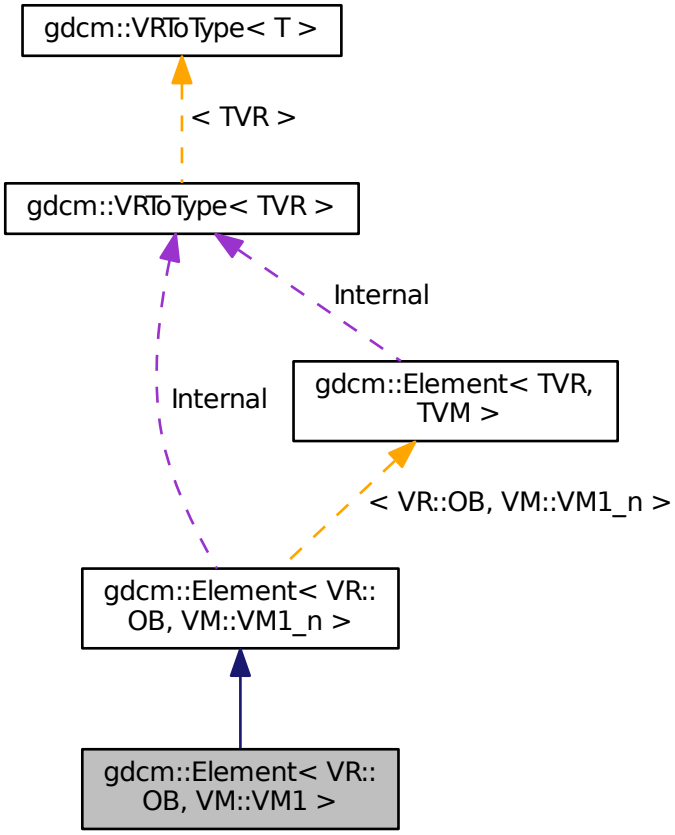
10.103 gdcm::Element< VR::OB, VM::VM1 > Class Template Reference

```
#include <gdcmElement.h>
```

Inheritance diagram for gdcm::Element< VR::OB, VM::VM1 >:



Collaboration diagram for gdcm::Element< VR::OB, VM::VM1 >:



Additional Inherited Members

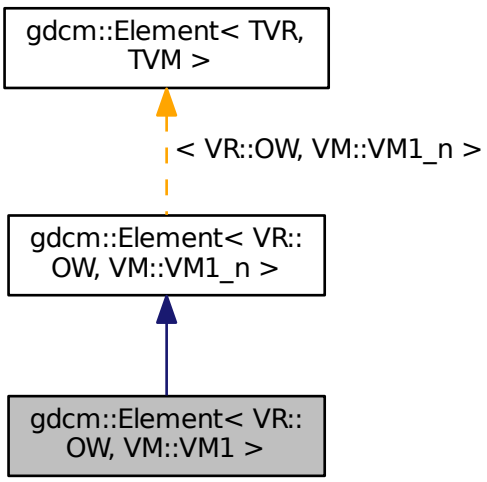
The documentation for this class was generated from the following file:

- gdcmElement.h

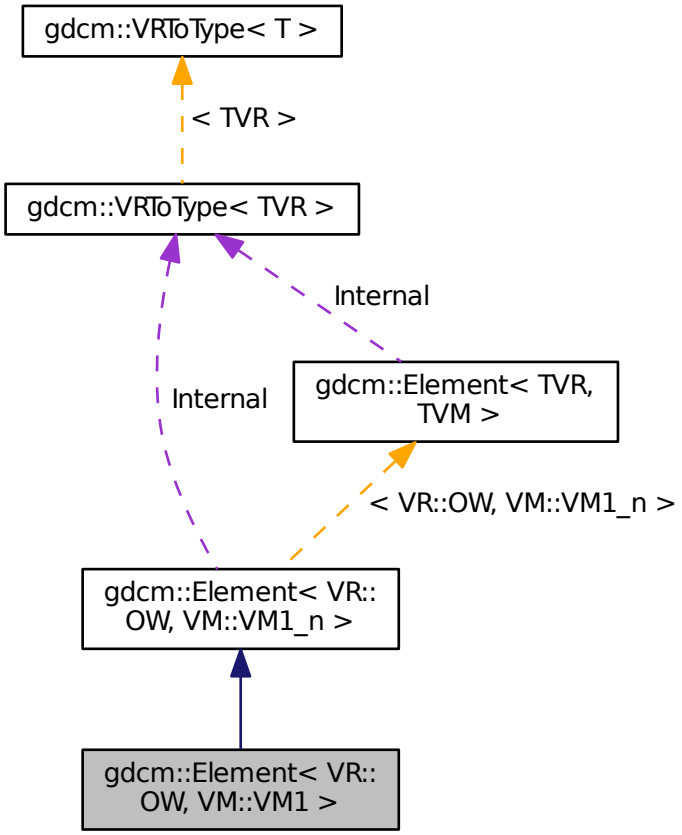
10.104 gdcm::Element< VR::OW, VM::VM1 > Class Template Reference

`#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 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 template<typename T > static void gdcm::EncodingImplementation< VR::VRASCII >::Read (T * *data,* unsigned long *length,* std::istream & *_is*) [inline], [static]

10.110.1.2 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 template<typename T > static void gdcm::EncodingImplementation< VR::VRASCII >::ReadNoSwap (T * *data,* unsigned long *length,* std::istream & *_is*) [inline], [static]

10.110.1.4 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 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 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 template<typename T > static void gdcm::EncodingImplementation< VR::VRBINARY >::Read (T * *data*, unsigned long *length*, std::istream & *_is*) [inline], [static]

10.111.1.2 template<typename T > static void gdcm::EncodingImplementation< VR::VRBINARY >::ReadComputeLength (T * *data*, unsigned int & *length*, std::istream & *_is*) [inline], [static]

10.111.1.3 template<typename T > static void gdcm::EncodingImplementation< VR::VRBINARY >::ReadNoSwap (T * *data*, unsigned long *length*, std::istream & *_is*) [inline], [static]

10.111.1.4 template<typename T > static void gdcm::EncodingImplementation< VR::VRBINARY >::Write (const T * *data*, unsigned long *length*, std::ostream & *_os*) [inline], [static]

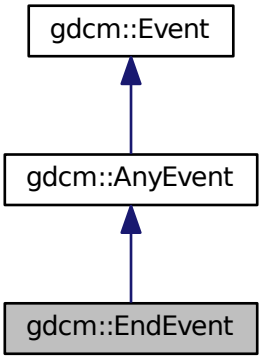
The documentation for this class was generated from the following file:

- gdcmElement.h

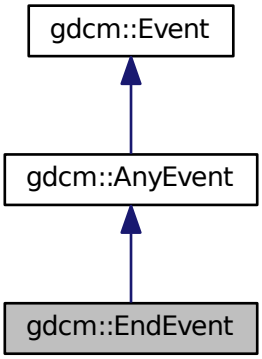
10.112 gdcm::EndEvent Class Reference

`#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 gdcm::EnumeratedValues::EnumeratedValues ( ) [inline]

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

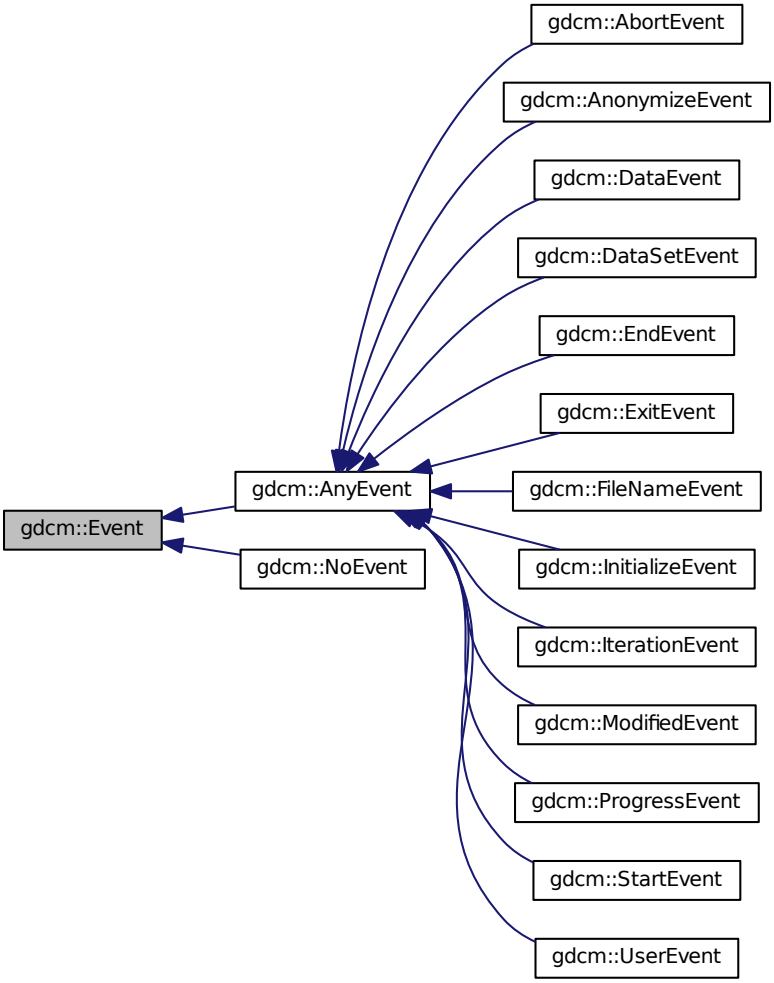
- gdcmEnumeratedValues.h

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 gdcm::Event::Event ( )

10.114.2.2 gdcm::Event::Event (const Event &)

10.114.2.3 virtual gdcm::Event::~Event ( ) [virtual]

10.114.3 Member Function Documentation

10.114.3.1 virtual bool gdcm::Event::CheckEvent (const Event *) const [pure virtual]

Check if given event matches or derives from this event.

10.114.3.2 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 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 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:

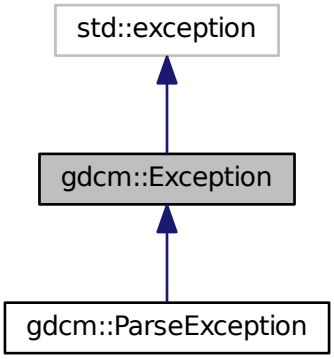
- gdcmEvent.h

10.115 gdcm::Exception Class Reference

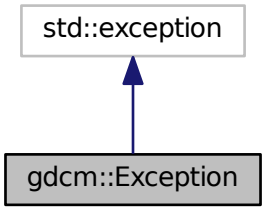
Exception.

`#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 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 virtual gdcm::Exception::~Exception ( ) throw) [inline], [virtual]

10.115.3 Member Function Documentation

10.115.3.1 const char* gdcm::Exception::GetDescription ( ) const [inline]

Return the Description.

Referenced by gdcm::SequenceOfItems::Read().

10.115.3.2 const char* gdcm::Exception::what ( ) const throw) [inline]

what implementation

Referenced by gdcm::SequenceOfFragments::ReadValue().

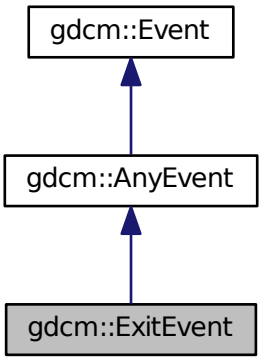
The documentation for this class was generated from the following file:

- gdcmException.h

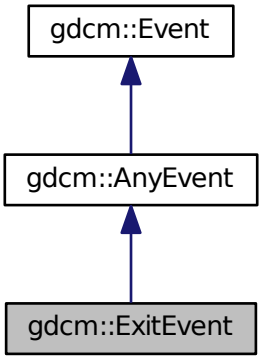
10.116 gdcm::ExitEvent Class Reference

```
#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:

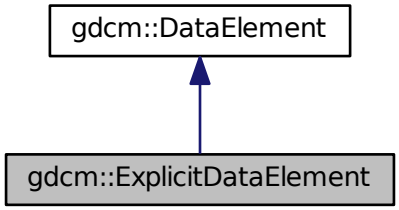
- gdcmEvent.h

10.117 gdcm::ExplicitDataElement Class Reference

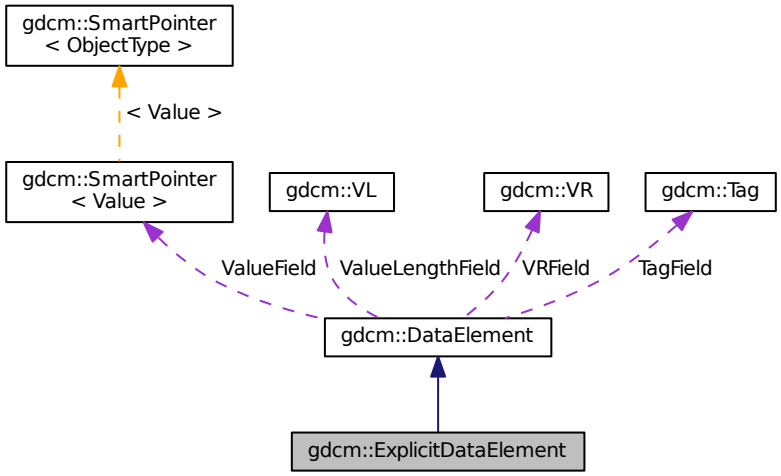
Class to read/write a DataElement as Explicit Data Element.

`#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 VL gdcm::ExplicitDataElement::GetLength ( ) const

10.117.2.2 template<typename TSwap > std::istream& gdcm::ExplicitDataElement::Read (std::istream & *is*)

10.117.2.3 template<typename TSwap > std::istream& gdcm::ExplicitDataElement::ReadPreValue (std::istream & *is*)

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

10.117.2.5 template<typename TSwap > std::istream& gdcm::ExplicitDataElement::ReadWithLength (std::istream & *is*, VL & *length*)

10.117.2.6 template<typename TSwap > const std::ostream& gdcm::ExplicitDataElement::Write (std::ostream & *os*) const

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

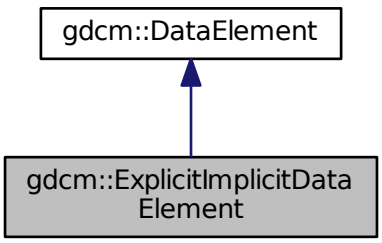
- gdcmExplicitDataElement.h

10.118 gdcm::ExplicitImplicitDataElement Class Reference

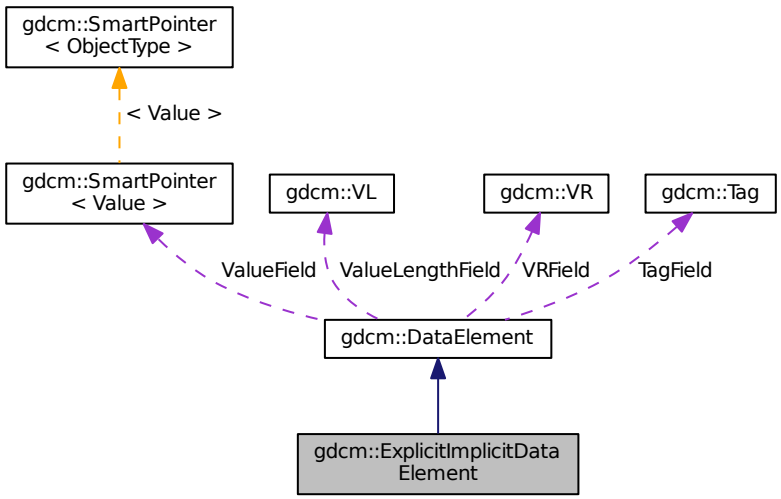
Class to read/write a DataElement as ExplicitImplicit Data Element.

`#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 VL gdcm::ExplicitImplicitDataElement::GetLength ( ) const

10.118.2.2 template<typename TSwap > std::istream& gdcm::ExplicitImplicitDataElement::Read (std::istream & *is*)

10.118.2.3 template<typename TSwap > std::istream& gdcm::ExplicitImplicitDataElement::ReadPreValue (std::istream & *is*)

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

10.118.2.5 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 gdcm::Fiducials::Fiducials ( ) [inline]

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

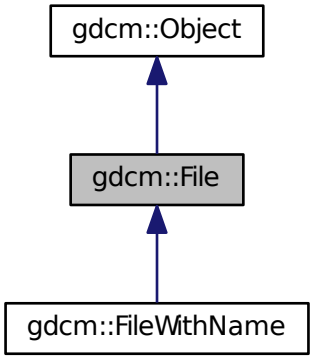
- gdcmFiducials.h

10.120 gdcm::File Class Reference

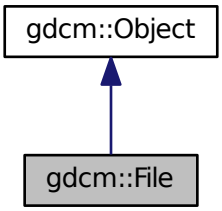
a DICOM File See PS 3.10 File: A File is an ordered string of zero or more bytes, where the first byte is at the beginning of the file and the last byte at the end of the File. Files are identified by a unique File ID and may by written, read and/or deleted.

`#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 gdcm::File::File ( )

10.120.2.2 gdcm::File::~File ( )

10.120.3 Member Function Documentation

10.120.3.1 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 DataSet& gdcm::File::GetDataSet ( ) [inline]

Get Data Set.

10.120.3.3 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 FileMetaInformation& gdcm::File::GetHeader ( ) [inline]

Get File Meta Information.

10.120.3.5 std::istream& gdcm::File::Read (std::istream & *is*)

Read.

10.120.3.6 void gdcm::File::SetDataSet (const DataSet & *ds*) [inline]

Set Data Set.

10.120.3.7 void gdcm::File::SetHeader (const FileMetaInformation & *fmi*) [inline]

Set File Meta Information.

10.120.3.8 std::ostream const& gdcm::File::Write (std::ostream & *os*) const

Write.

10.120.4 Friends And Related Function Documentation

10.120.4.1 std::ostream& operator<< (std::ostream & *os,* const File & *val*) [friend]

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

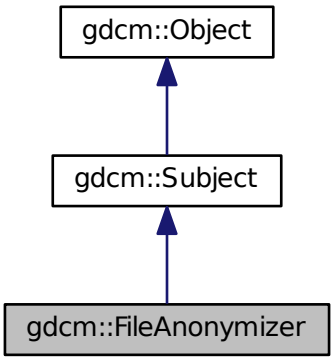
- gdcmFile.h

10.121 gdcm::FileAnonymizer Class Reference

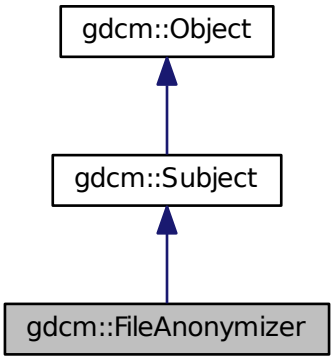
FileAnonymizer.

`#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 gdcm::FileAnonymizer::FileAnonymizer ( )

10.121.2.2 gdcm::FileAnonymizer::~FileAnonymizer ( )

10.121.3 Member Function Documentation

10.121.3.1 void gdcm::FileAnonymizer::Empty (Tag const & *t*)

Make Tag t empty Warning: does not handle SQ element

Examples:

MakeTemplate.cxx.

10.121.3.2 void gdcm::FileAnonymizer::Remove (Tag const & *t*)

remove a tag (even a SQ can be removed)

10.121.3.3 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 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 void gdcm::FileAnonymizer::SetInputFileName (const char * *filename_native*)

Set input filename.

Examples:

FileAnonymize.cs, and MakeTemplate.cxx.

10.121.3.6 void gdcm::FileAnonymizer::SetOutputFileName (const char * *filename_native*)

Set output filename.

Examples:

MakeTemplate.cxx.

10.121.3.7 bool gdcm::FileAnonymizer::Write ( )

Write the output file.

Examples:

MakeTemplate.cxx.

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

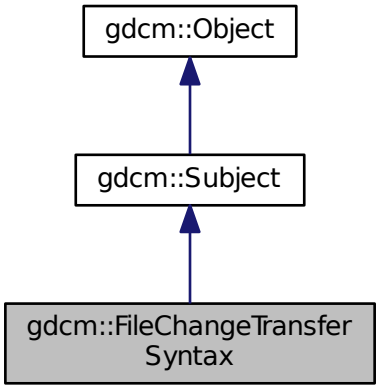
- gdcmFileAnonymizer.h

10.122 gdcm::FileChangeTransferSyntax Class Reference

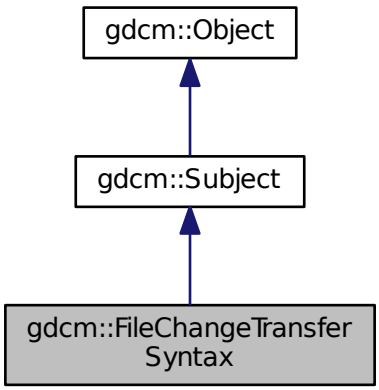
FileChangeTransferSyntax.

```
#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 gdcm::FileChangeTransferSyntax::FileChangeTransferSyntax ( )

10.122.2.2 gdcm::FileChangeTransferSyntax::~FileChangeTransferSyntax ( )

10.122.3 Member Function Documentation

10.122.3.1 bool gdcm::FileChangeTransferSyntax::Change ( )

Change the transfer syntax.

10.122.3.2 ImageCodec* gdcm::FileChangeTransferSyntax::GetCodec ( )

Retrieve the actual codec (valid after calling SetTransferSyntax) Only advanced users should call this function.

10.122.3.3 static SmartPointer<FileChangeTransferSyntax> gdcm::FileChangeTransferSyntax::New ( ) [inline], [static]

for wrapped language: instantiate a reference counted object

10.122.3.4 void gdcm::FileChangeTransferSyntax::SetInputFileName (const char * *filename_native*)

Set input filename (raw DICOM)

10.122.3.5 void gdcm::FileChangeTransferSyntax::SetOutputFileName (const char * *filename_native*)

Set output filename (target compressed DICOM)

10.122.3.6 void gdcm::FileChangeTransferSyntax::SetTransferSyntax (TransferSyntax const & *ts*)

Specify the Target Transfer Syntax.

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

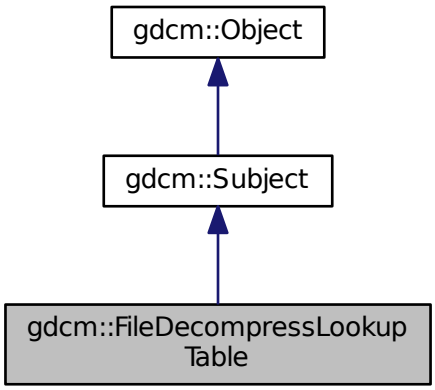
- gdcmFileChangeTransferSyntax.h

10.123 gdcm::FileDecompressLookupTable Class Reference

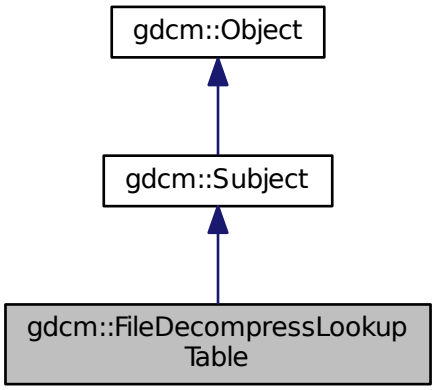
FileDecompressLookupTable class It decompress the segmented LUT into linearized one (only PALETTE_COLOR images) Output will be a PhotometricInterpretation=RGB image.

```
#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 gdcm::FileDecompressLookupTable::FileDecompressLookupTable ( ) [inline]

10.123.2.2 gdcm::FileDecompressLookupTable::~FileDecompressLookupTable ( ) [inline]

10.123.3 Member Function Documentation

10.123.3.1 bool gdcm::FileDecompressLookupTable::Change ( )

Decompress.

10.123.3.2 File& gdcm::FileDecompressLookupTable::GetFile ( ) [inline]

10.123.3.3 const Pixmap& gdcm::FileDecompressLookupTable::GetPixmap ( ) const [inline]

10.123.3.4 Pixmap& gdcm::FileDecompressLookupTable::GetPixmap ( ) [inline]

10.123.3.5 void gdcm::FileDecompressLookupTable::SetFile (const File & *f*) [inline]

Set/Get File.

10.123.3.6 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 See PS 3.16 - 2008 For the list of Code Value that can be used for in Derivation Code Sequence.

`#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 gdcm::FileDerivation::FileDerivation ( )

10.124.2.2 gdcm::FileDerivation::∼FileDerivation ( )

10.124.3 Member Function Documentation

10.124.3.1 bool gdcm::FileDerivation::AddDerivationDescription ( ) [protected]

10.124.3.2 bool gdcm::FileDerivation::AddPurposeOfReferenceCodeSequence (DataSet & *ds*) [protected]

10.124.3.3 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 bool gdcm::FileDerivation::AddSourceImageSequence ( ) [protected]

10.124.3.5 bool gdcm::FileDerivation::Derive ( )

Change.

Examples:

GenFakeImage.cxx.

10.124.3.6 File& gdcm::FileDerivation::GetFile ( ) [inline]

Examples:

GenFakeImage.cxx.

10.124.3.7 const File& gdcm::FileDerivation::GetFile ( ) const [inline]

10.124.3.8 void gdcm::FileDerivation::SetDerivationCodeSequenceCodeValue (unsigned int *codevalue*)

Specify the Derivation Code Sequence Code Value. Eg 113040.

Examples:

GenFakeImage.cxx.

10.124.3.9 void gdcm::FileDerivation::SetDerivationDescription (const char * *dd*)

Specify the Derivation Description. Eg "lossy conversion".

10.124.3.10 void gdcm::FileDerivation::SetFile (const File & *f*) [inline]

Set/Get File.

Examples:

GenFakeImage.cxx.

10.124.3.11 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 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.

```
#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 gdcm::FileExplicitFilter::FileExplicitFilter ( ) [inline]

10.125.2.2 gdcm::FileExplicitFilter::~FileExplicitFilter ( ) [inline]

10.125.3 Member Function Documentation

10.125.3.1 bool gdcm::FileExplicitFilter::Change ( )

Set FMI Transfer Syntax.

Change

Examples:

GenAllVR.cxx, and LargeVRDSExplicit.cxx.

10.125.3.2 bool gdcm::FileExplicitFilter::ChangeFMI ( ) [protected]

10.125.3.3 File& gdcm::FileExplicitFilter::GetFile ( ) [inline]

10.125.3.4 bool gdcm::FileExplicitFilter::ProcessDataSet (DataSet & *ds,* Dicts const & *dicts*) [protected]

10.125.3.5 void gdcm::FileExplicitFilter::SetChangePrivateTags (bool *b*) [inline]

Decide whether or not to VR'ify private tags.

10.125.3.6 void gdcm::FileExplicitFilter::SetFile (const File & *f*) [inline]

Set/Get File.

Examples:

GenAllVR.cxx, and LargeVRDSExplicit.cxx.

10.125.3.7 void gdcm::FileExplicitFilter::SetRecomputeItemLength (bool *b*)

By default set Sequence & Item length to Undefined to avoid recomputing length:

10.125.3.8 void gdcm::FileExplicitFilter::SetRecomputeSequenceLength (bool *b*)

10.125.3.9 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:

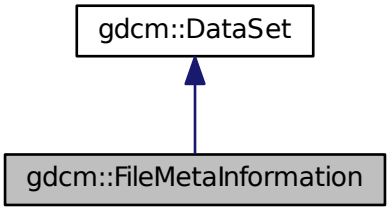
- gdcmFileExplicitFilter.h

10.126 gdcm::FileMetaInformation Class Reference

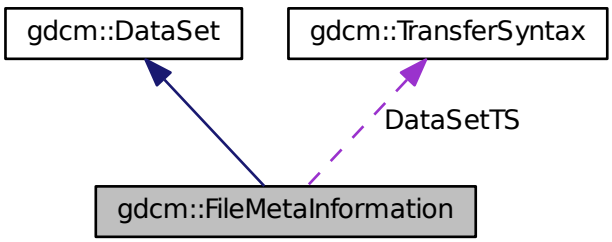
Class to represent a File Meta Information.

`#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 gdcm::FileMetaInformation::FileMetaInformation ( )

10.126.2.2 gdcm::FileMetaInformation::~FileMetaInformation ( )

10.126.2.3 gdcm::FileMetaInformation::FileMetaInformation (FileMetaInformation const & *fmi*) [inline]

References DataSetMS, DataSetTS, and MetaInformationTS.

10.126.3 Member Function Documentation

10.126.3.1 static void gdcm::FileMetaInformation::AppendImplementationClassUID (const char * *imp*) [static]

10.126.3.2 void gdcm::FileMetaInformation::ComputeDataSetMediaStorageSOPClass ( ) [protected]

10.126.3.3 void gdcm::FileMetaInformation::ComputeDataSetTransferSyntax ( ) [protected]

10.126.3.4 void gdcm::FileMetaInformation::Default ( ) [protected]

10.126.3.5 void gdcm::FileMetaInformation::FillFromDataSet (DataSet const & *ds*)

Construct a FileMetaInformation from an already existing DataSet:

10.126.3.6 const TransferSyntax& gdcm::FileMetaInformation::GetDataSetTransferSyntax ( ) const [inline]

Examples:

GetJPEGSamplePrecision.cxx, and MergeTwoFiles.cxx.

10.126.3.7 static const char* gdcm::FileMetaInformation::GetFileMetaInformationVersion ( ) [static], [protected]

10.126.3.8 VL gdcm::FileMetaInformation::GetFullLength ( ) const [inline]

References gdcm::VL::GetLength().

10.126.3.9 static const char* gdcm::FileMetaInformation::GetGDCMImplementationClassUID ( ) [static], [protected]

10.126.3.10 static const char* gdcm::FileMetaInformation::GetGDCMImplementationVersionName ( ) [static], [protected]

10.126.3.11 static const char* gdcm::FileMetaInformation::GetGDCMSourceApplicationEntityTitle ( ) [static], [protected]

10.126.3.12 static const char* gdcm::FileMetaInformation::GetImplementationClassUID ( ) [static]

10.126.3.13 static const char* gdcm::FileMetaInformation::GetImplementationVersionName ( ) [static]

10.126.3.14 MediaStorage gdcm::FileMetaInformation::GetMediaStorage ( ) const

10.126.3.15 std::string gdcm::FileMetaInformation::GetMediaStorageAsString ( ) const

10.126.3.16 TransferSyntax::NegociatedType gdcm::FileMetaInformation::GetMetaInformationTS ( ) const [inline]

10.126.3.17 const Preamble& gdcm::FileMetaInformation::GetPreamble ( ) const [inline]

Get Preamble.

Referenced by gdcm::operator<<().

10.126.3.18 **Preamble& gdcm::FileMetaInformation::GetPreamble ( ) [inline]**

10.126.3.19 **static const char* gdcm::FileMetaInformation::GetSourceApplicationEntityTitle ( ) [static]**

10.126.3.20 **void gdcm::FileMetaInformation::Insert (const DataElement & *de*) [inline]**

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

10.126.3.21 **bool gdcm::FileMetaInformation::IsValid ( ) const [inline]**

10.126.3.22 **std::istream& gdcm::FileMetaInformation::Read (std::istream & *is*)**

Read.

10.126.3.23 **std::istream& gdcm::FileMetaInformation::ReadCompat (std::istream & *is*)**

10.126.3.24 **template<typename TSwap > std::istream& gdcm::FileMetaInformation::ReadCompatInternal (std::istream & *is*) [protected]**

10.126.3.25 **void gdcm::FileMetaInformation::Replace (const DataElement & *de*) [inline]**

Examples:

LargeVRDSExplicit.cxx.

References gdcm::DataElement::GetTag().

10.126.3.26 **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, LargeVRDS_Explicit.cxx, MakeTemplate.cxx, MpegVideoInfo.cs, pmsct_rgb1.cxx, QIDO-RS.cxx, rle2img.cxx, and StreamImageReaderTest.cxx.

10.126.3.27 **static void gdcm::FileMetaInformation::SetImplementationClassUID (const char * *imp*) [static]**

Override the GDCM default values:

10.126.3.28 static void gdcm::FileMetaInformation::SetImplementationVersionName (const char * *version*) [static]

10.126.3.29 void gdcm::FileMetaInformation::SetPreamble (const **Preamble** & *p*) [inline]

10.126.3.30 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 std::ostream& gdcm::FileMetaInformation::Write (std::ostream & *os*) const

Write.

10.126.4 Friends And Related Function Documentation

10.126.4.1 std::ostream& operator<< (std::ostream & *_os,* const **FileMetaInformation** & *_val*) [friend]

10.126.5 Member Data Documentation

10.126.5.1 MediaStorage::MSType gdcm::FileMetaInformation::DataSetMS [protected]

Referenced by FileMetaInformation().

10.126.5.2 TransferSyntax gdcm::FileMetaInformation::DataSetTS [protected]

Referenced by FileMetaInformation().

10.126.5.3 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 gdcm::Filename::Filename (const char * *filename* = " ") [inline]

10.127.3 Member Function Documentation

10.127.3.1 bool gdcm::Filename::EndWith (const char *ending[ ]*) const

Does the filename ends with a particular string ?

10.127.3.2 const char* gdcm::Filename::GetExtension ( )

return only the extension part of a filename

10.127.3.3 const char* gdcm::Filename::GetFileName ( ) const [inline]

Return the full filename.

10.127.3.4 const char* gdcm::Filename::GetName ( )

return only the name part of a filename

10.127.3.5 const char* gdcm::Filename::GetPath ( )

Return only the path component of a filename.

Examples:

ClinicalTrialIdentificationWorkflow.cs.

10.127.3.6 bool gdcm::Filename::IsEmpty ( ) const [inline]

return whether the filename is empty

10.127.3.7 bool gdcm::Filename::IsIdentical (Filename const & *fn*) const

10.127.3.8 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 gdcm::Filename::operator const char * ( ) const [inline]

Simple operator to allow Filename myfilename("..."); const char * s = myfilename;

10.127.3.10 const char* gdcm::Filename::ToUnixSlashes ( )

Convert backslash (windows style) to UNIX style slash.

10.127.3.11 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:

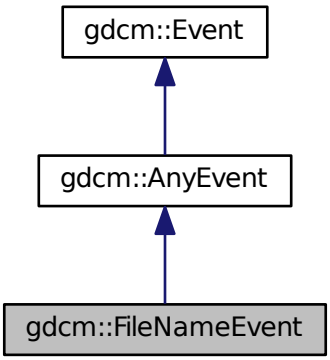
- gdcmFilename.h

10.128 gdcm::FileNameEvent Class Reference

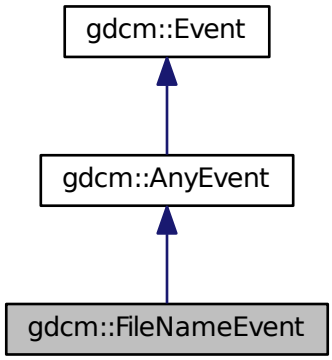
FileNameEvent Special type of event triggered during processing of FileSet.

```
#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 typedef FileNameEvent gdcm::FileNameEvent::Self

10.128.2.2 typedef AnyEvent gdcm::FileNameEvent::Superclass

10.128.3 Constructor & Destructor Documentation

10.128.3.1 gdcm::FileNameEvent::FileNameEvent (const char * *s* = " ") [inline]

10.128.3.2 virtual gdcm::FileNameEvent::~FileNameEvent ( ) [inline], [virtual]

10.128.3.3 gdcm::FileNameEvent::FileNameEvent (const Self & *s*) [inline]

10.128.4 Member Function Documentation

10.128.4.1 virtual bool gdcm::FileNameEvent::CheckEvent (const ::gdcm::Event * *e*) const [inline], [virtual]

10.128.4.2 virtual const char* gdcm::FileNameEvent::GetEventName ( ) const [inline], [virtual]

Return the StringName associated with the event.

Implements gdcm::Event.

10.128.4.3 const char* gdcm::FileNameEvent::GetFileName ( ) const [inline]

Examples:

SimpleScanner.cxx.

10.128.4.4 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 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 typedef std::vector<FilenameType> gdcm::FilenameGenerator::FilenamesType

10.129.2.2 typedef std::string gdcm::FilenameGenerator::FilenameType

10.129.2.3 typedef FilenamesType::size_type gdcm::FilenameGenerator::SizeType

10.129.3 Constructor & Destructor Documentation

10.129.3.1 gdcm::FilenameGenerator::FilenameGenerator ( ) [inline]

10.129.3.2 gdcm::FilenameGenerator::~FilenameGenerator ( ) [inline]

10.129.4 Member Function Documentation

10.129.4.1 bool gdcm::FilenameGenerator::Generate ( )

Generate (return success)

Examples:

ConvertMultiFrameToSingleFrame.cxx, and CreateFakePET.cxx.

10.129.4.2 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 FilenamesType const& gdcm::FilenameGenerator::GetFilenames ( ) const [inline]

Return all filenames.

10.129.4.4 SizeType gdcm::FilenameGenerator::GetNumberOfFilenames ( ) const

Examples:

ConvertMultiFrameToSingleFrame.cxx, and CreateFakePET.cxx.

10.129.4.5 const char* gdcm::FilenameGenerator::GetPattern ( ) const [inline]

10.129.4.6 const char* gdcm::FilenameGenerator::GetPrefix ( ) const [inline]

10.129.4.7 void gdcm::FilenameGenerator::SetNumberOfFilenames (SizeType *nfiles*)

Set/Get the number of filenames to generate.

Examples:

ConvertMultiFrameToSingleFrame.cxx, and CreateFakePET.cxx.

10.129.4.8 void gdcm::FilenameGenerator::SetPattern (const char * *pattern*) [inline]

Set/Get pattern.

Examples:

ConvertMultiFrameToSingleFrame.cxx, and CreateFakePET.cxx.

10.129.4.9 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

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.

```
#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 typedef std::vector<FileType> gdcm::FileSet::FilesType

10.130.2.2 typedef std::string gdcm::FileSet::FileType

10.130.3 Constructor & Destructor Documentation

10.130.3.1 gdcm::FileSet::FileSet ( ) [inline]

10.130.4 Member Function Documentation

10.130.4.1 void gdcm::FileSet::AddFile (File const &) [inline]

Deprecated . Does nothing

10.130.4.2 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 FilesType const& gdcm::FileSet::GetFiles ( ) const [inline]

10.130.4.4 void gdcm::FileSet::SetFiles (FilesType const & *files*)

10.130.5 Friends And Related Function Documentation

10.130.5.1 std::ostream& operator<< (std::ostream & *_os,* const FileSet & *d*) [friend]

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

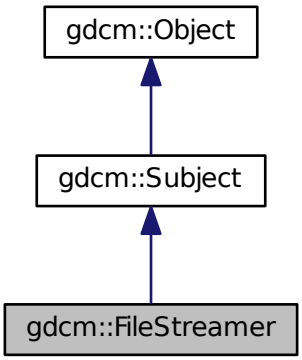
- gdcmFileSet.h

10.131 gdcm::FileStreamer Class Reference

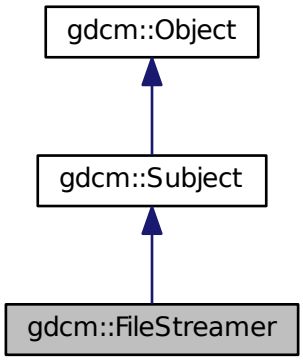
FileStreamer This class let a user create a massive DICOM DataSet from a template DICOM file, by appending chunks of data.

```
#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 gdcm::FileStreamer::FileStreamer ( )

10.131.2.2 gdcm::FileStreamer::~FileStreamer ( )

10.131.3 Member Function Documentation

10.131.3.1 bool gdcm::FileStreamer::AppendToDataElement (const Tag & *t*, const char * *array*, size_t *len*)

Append to previously started Tag t.

10.131.3.2 bool gdcm::FileStreamer::AppendToGroupDataElement (const PrivateTag & *pt*, const char * *array*, size_t *len*)

Append to previously started private creator.

10.131.3.3 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 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 static SmartPointer<FileStreamer> gdcm::FileStreamer::New ( ) [inline], [static]

for wrapped language: instantiate a reference counted object

10.131.3.6 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 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 void gdcm::FileStreamer::SetOutputFileName (const char * *filename_native*)

Set output filename (target file)

10.131.3.9 void gdcm::FileStreamer::SetTemplateFileName (const char * *filename_native*)

Set input DICOM template filename.

Examples:

FileStreaming.cs.

10.131.3.10 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 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 bool gdcm::FileStreamer::StopDataElement (const Tag & *t*)

Stop appending to tag t. This will compute the proper attribute length.

10.131.3.13 bool gdcm::FileStreamer::StopGroupDataElement (const PrivateTag & *pt*)

Stop appending to private creator.

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

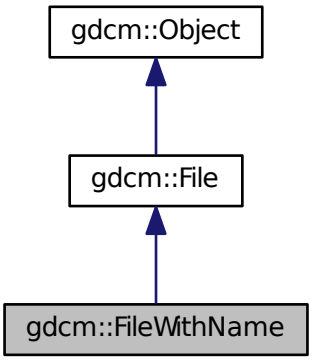
- gdcmFileStreamer.h

10.132 gdcm::FileWithName Class Reference

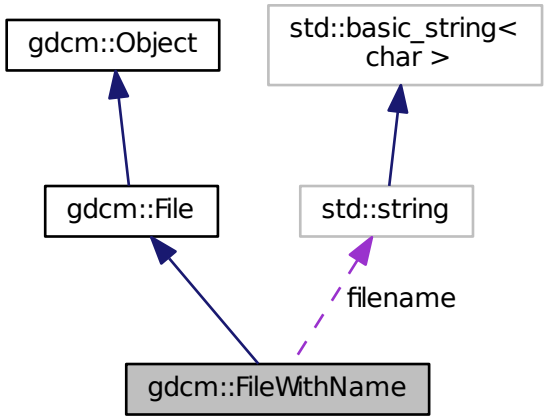
FileWithName.

```
#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 gdcm::FileWithName::FileWithName (File & *f*) [inline]

10.132.3 Member Data Documentation

10.132.3.1 std::string gdcm::FileWithName::filename

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

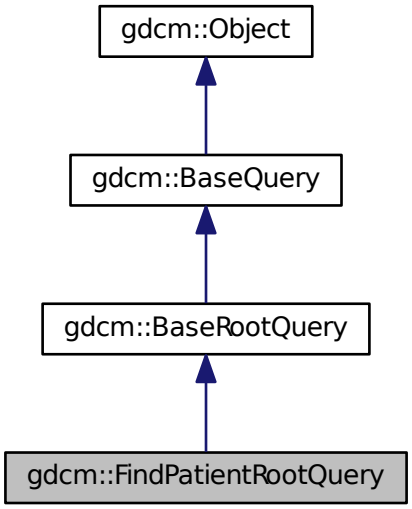
- gdcmSerieHelper.h

10.133 gdcm::FindPatientRootQuery Class Reference

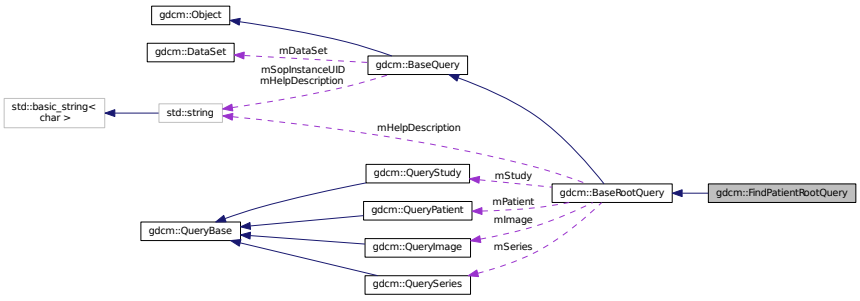
PatientRootQuery contains: the class which will produce a dataset for c-find with patient root.

`#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 gdcm::FindPatientRootQuery::FindPatientRootQuery ( )

10.133.3 Member Function Documentation

10.133.3.1 UIDs::TSName gdcm::FindPatientRootQuery::GetAbstractSyntaxUID ( ) const [virtual]

Implements gdcm::BaseQuery.

10.133.3.2 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 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 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 friend class QueryFactory [friend]

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

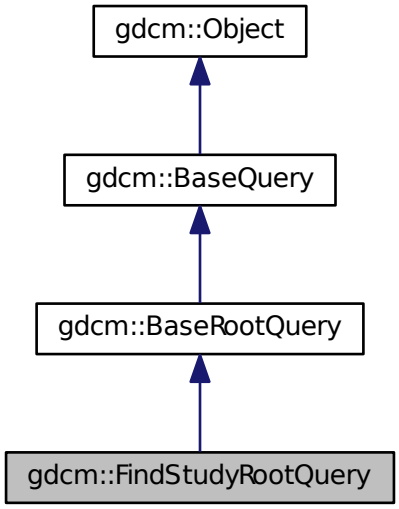
- gdcmFindPatientRootQuery.h

10.134 gdcm::FindStudyRootQuery Class Reference

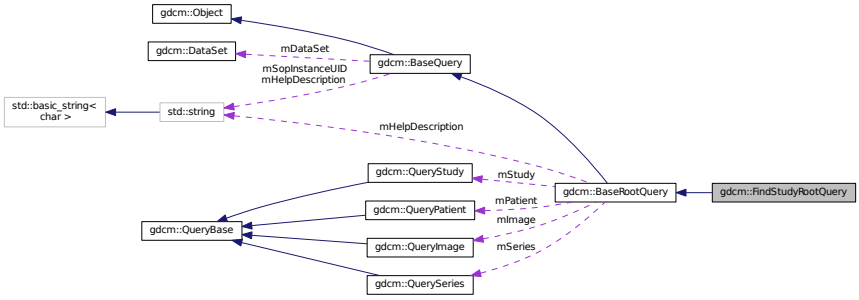
FindStudyRootQuery contains: the class which will produce a dataset for C-FIND with study root.

```
#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 gdcm::FindStudyRootQuery::FindStudyRootQuery ( )

10.134.3 Member Function Documentation

10.134.3.1 UIDs::TSName gdcm::FindStudyRootQuery::GetAbstractSyntaxUID ( ) const [virtual]

Implements gdcm::BaseQuery.

10.134.3.2 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 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 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 friend class QueryFactory [friend]

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

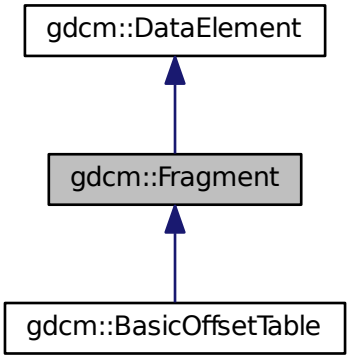
- gdcmFindStudyRootQuery.h

10.135 gdcm::Fragment Class Reference

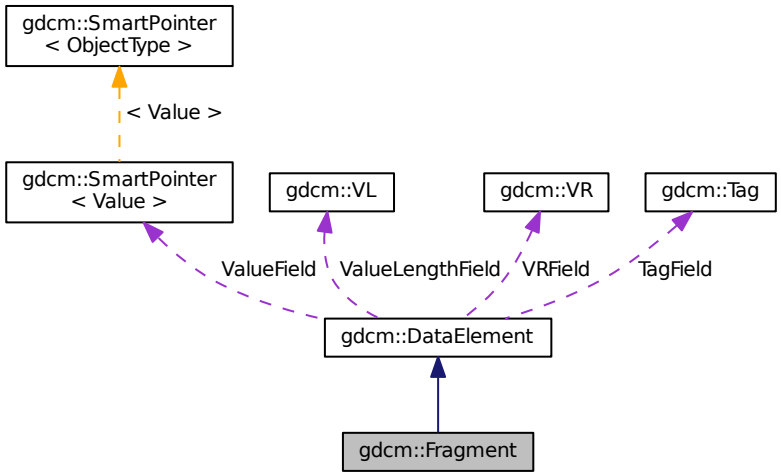
Class to represent a Fragment.

```
#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 gdcm::Fragment::Fragment ( ) [inline]

References gdcm::operator<<().

10.135.3 Member Function Documentation

10.135.3.1 VL gdcm::Fragment::ComputeLength ( ) const

10.135.3.2 VL gdcm::Fragment::GetLength ( ) const

10.135.3.3 template<typename TSwap > std::istream& gdcm::Fragment::Read (std::istream & *is*) [inline]

Referenced by gdcm::SequenceOfFragments::ReadValue().

10.135.3.4 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 template<typename TSwap > std::istream& gdcm::Fragment::ReadPreValue (std::istream & *is*) [inline]

10.135.3.6 template<typename TSwap > std::istream& gdcm::Fragment::ReadValue (std::istream & *is*) [inline]

References gdcmWarningMacro, and gdcm::ParseException::SetLastElement().

10.135.3.7 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::ByteValue::Write().

10.135.4 Friends And Related Function Documentation

10.135.4.1 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 gdcm::Global::Global ( )

10.136.2.2 gdcm::Global::~Global ( )

10.136.3 Member Function Documentation

10.136.3.1 bool gdcm::Global::Append (const char * *path*)

Append path at the end of the path list

Warning

not thread safe !

10.136.3.2 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 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 Dicts& gdcm::Global::GetDicts ( )

10.136.3.5 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 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 const char* gdcm::Global::Locate (const char * *resfile*) const [protected]

Locate a resource file.

10.136.3.8 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 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 typedef std::vector<std::string> gdcm::GroupDict::GroupStringVector

10.137.3 Constructor & Destructor Documentation

10.137.3.1 gdcm::GroupDict::GroupDict ( ) [inline]

10.137.3.2 gdcm::GroupDict::~GroupDict ( ) [inline]

References gdcm::operator<<().

10.137.4 Member Function Documentation

10.137.4.1 void gdcm::GroupDict::Add (std::string const & *abbreviation*, std::string const & *name*) [protected]

10.137.4.2 std::string const& gdcm::GroupDict::GetAbbreviation (uint16_t *num*) const

Referenced by gdcm::operator<<().

10.137.4.3 std::string const& gdcm::GroupDict::GetName (uint16_t *num*) const

Referenced by gdcm::operator<<().

10.137.4.4 void gdcm::GroupDict::Insert (uint16_t *num*, std::string const & *abbreviation*, std::string const & *name*) [protected]

10.137.4.5 size_t gdcm::GroupDict::Size ( ) const [inline]

Referenced by gdcm::operator<<().

10.137.5 Friends And Related Function Documentation

10.137.5.1 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 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.

`#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 gdcm::IconImageFilter::IconImageFilter ( )

10.138.2.2 gdcm::IconImageFilter::~IconImageFilter ( )

10.138.3 Member Function Documentation

10.138.3.1 bool gdcm::IconImageFilter::Extract ( )

Extract all Icon found in File.

Examples:

ExtractIconFromFile.cxx.

10.138.3.2 void gdcm::IconImageFilter::ExtractIconImages ( ) [protected]

10.138.3.3 void gdcm::IconImageFilter::ExtractVeproIconImages ( ) [protected]

10.138.3.4 File& gdcm::IconImageFilter::GetFile ( ) [inline]

10.138.3.5 const File& gdcm::IconImageFilter::GetFile ( ) const [inline]

10.138.3.6 IconImage& gdcm::IconImageFilter::GetIconImage (unsigned int *i*) const

Examples:

ExtractIconFromFile.cxx.

10.138.3.7 unsigned int gdcm::IconImageFilter::GetNumberOfIconImages ( ) const

Retrieve extract IconImage (need to call Extract first)

Examples:

ExtractIconFromFile.cxx.

10.138.3.8 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 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:

```
#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 gdcm::IconImageGenerator::IconImageGenerator ( )

10.139.2.2 gdcm::IconImageGenerator::~IconImageGenerator ( )

10.139.3 Member Function Documentation

10.139.3.1 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 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 bool gdcm::IconImageGenerator::Generate ( )

Generate Icon.

Examples:

ExtractIconFromFile.cxx.

10.139.3.4 const IconImage& gdcm::IconImageGenerator::GetIconImage ( ) const [inline]

Retrieve generated Icon.

Examples:

ExtractIconFromFile.cxx.

10.139.3.5 Pixmap& gdcm::IconImageGenerator::GetPixmap ( ) [inline]

10.139.3.6 const Pixmap& gdcm::IconImageGenerator::GetPixmap ( ) const [inline]

10.139.3.7 void gdcm::IconImageGenerator::SetOutputDimensions (const unsigned int *dims[2]*)

Set Target dimension of output Icon.

Examples:

ExtractIconFromFile.cxx.

10.139.3.8 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 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 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 gdcm::ignore_char::ignore_char (char *c*) [inline]

10.140.2 Member Data Documentation

10.140.2.1 char gdcm::ignore_char::m_char

Referenced by gdcm::operator>>().

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

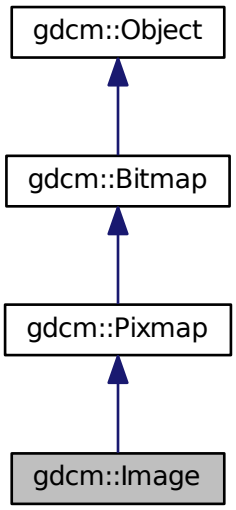
- gdcmElement.h

10.141 gdcm::Image Class Reference

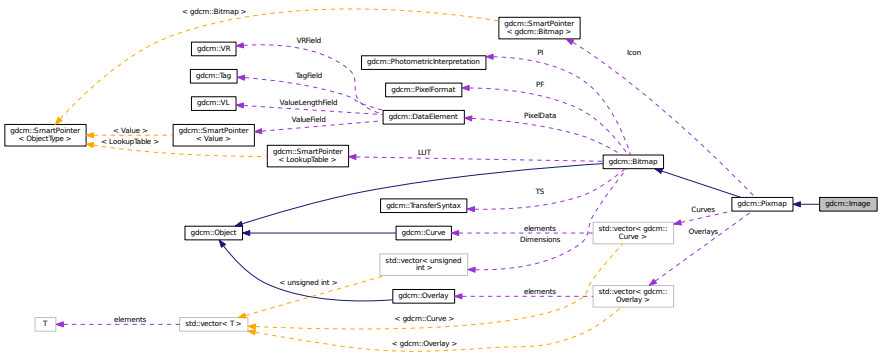
Image This is the container for an Image in the general sense. From this container you should be able to request information like:

```
#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 gdcm::Image::Image ( ) [inline]

10.141.2.2 gdcm::Image::~Image ( ) [inline]

10.141.3 Member Function Documentation

10.141.3.1 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 double gdcm::Image::GetDirectionCosines (unsigned int *idx*) const

10.141.3.3 double gdcm::Image::GetIntercept ( ) const [inline]

10.141.3.4 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 double gdcm::Image::GetOrigin (unsigned int *idx*) const

10.141.3.6 double gdcm::Image::GetSlope ( ) const [inline]

10.141.3.7 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 double gdcm::Image::GetSpacing (unsigned int *idx*) const

10.141.3.9 void gdcm::Image::Print (std::ostream & *os*) const [virtual]

print

Reimplemented from gdcm::Bitmap.

Examples:

CompressImage.cxx, and PatchFile.cxx.

10.141.3.10 void gdcm::Image::SetDirectionCosines (const float * *dircos*)

10.141.3.11 void gdcm::Image::SetDirectionCosines (const double * *dircos*)

10.141.3.12 void gdcm::Image::SetDirectionCosines (unsigned int *idx,* double *dircos*)

10.141.3.13 void gdcm::Image::SetIntercept (double *intercept*) [inline]

intercept

10.141.3.14 void gdcm::Image::SetOrigin (const float * *ori*)

10.141.3.15 void gdcm::Image::SetOrigin (const double * *ori*)

10.141.3.16 void gdcm::Image::SetOrigin (unsigned int *idx,* double *ori*)

10.141.3.17 void gdcm::Image::SetSlope (double *slope*) [inline]

slope

10.141.3.18 void gdcm::Image::SetSpacing (const double * *spacing*)

Examples:

csa2img.cxx, and iU22tomultisc.cxx.

10.141.3.19 void gdcm::Image::SetSpacing (unsigned int *idx,* double *spacing*)

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

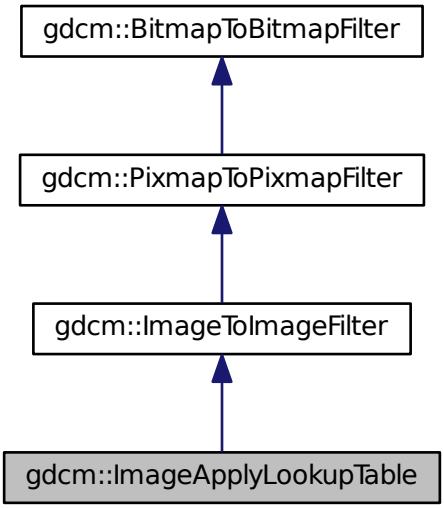
- gdcmImage.h

10.142 gdcm::ImageApplyLookupTable Class Reference

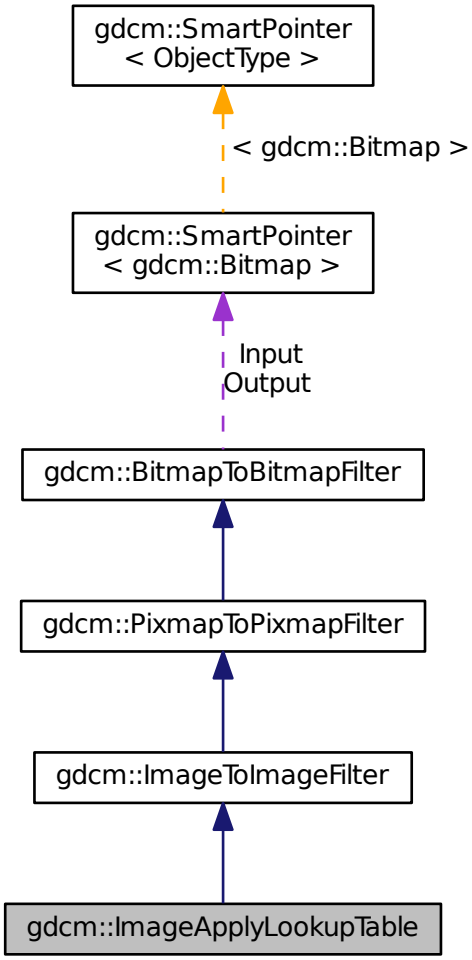
ImageApplyLookupTable class It applies the LUT the PixelData (only PALETTE_COLOR images) Output will be a PhotometricInterpretation=RGB image.

```
#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 gdcm::ImageApplyLookupTable::ImageApplyLookupTable ( ) [inline]

10.142.2.2 gdcm::ImageApplyLookupTable::~ImageApplyLookupTable ( ) [inline]

10.142.3 Member Function Documentation

10.142.3.1 bool gdcm::ImageApplyLookupTable::Apply ( )

Apply.

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

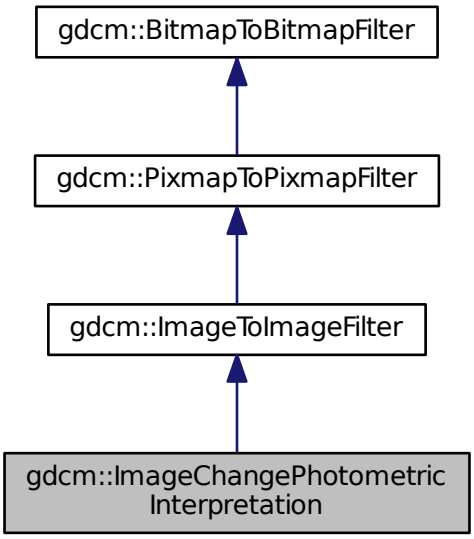
- gdcmImageApplyLookupTable.h

10.143 gdcm::ImageChangePhotometricInterpretation Class Reference

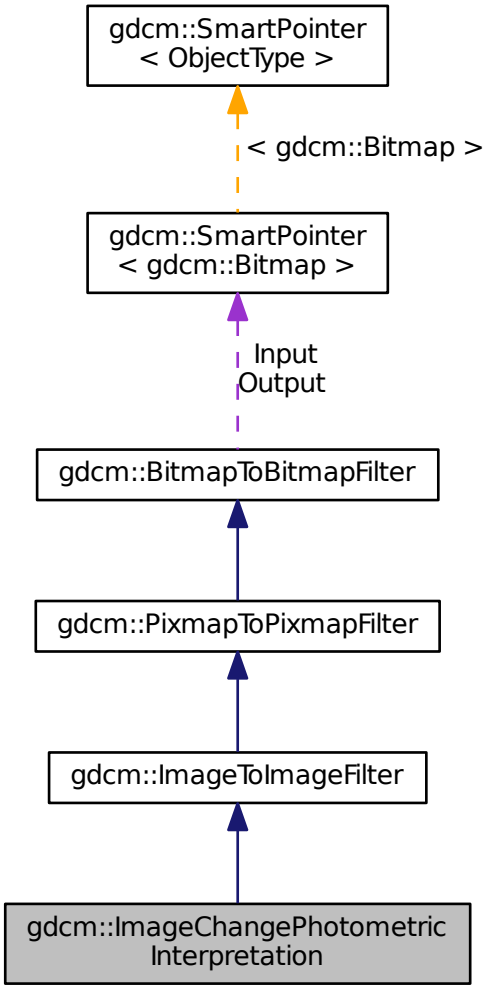
ImageChangePhotometricInterpretation class Class to change the Photometric Interpetation of an input DICOM.

`#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 gdcm::ImageChangePhotometricInterpretation::ImageChangePhotometricInterpretation ( ) [inline]

10.143.2.2 gdcm::ImageChangePhotometricInterpretation::~ImageChangePhotometricInterpretation ( ) [inline]

10.143.3 Member Function Documentation

10.143.3.1 bool gdcm::ImageChangePhotometricInterpretation::Change ( )

Change.

10.143.3.2 bool gdcm::ImageChangePhotometricInterpretation::ChangeMonochrome ( ) [protected]

10.143.3.3 const PhotometricInterpretation& gdcm::ImageChangePhotometricInterpretation::GetPhotometricInterpretation ( ) const [inline]

10.143.3.4 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 void gdcm::ImageChangePhotometricInterpretation::SetPhotometricInterpretation (PhotometricInterpretation const & *pi*) [inline]

Set/Get requested PhotometricInterpretation.

10.143.3.6 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:

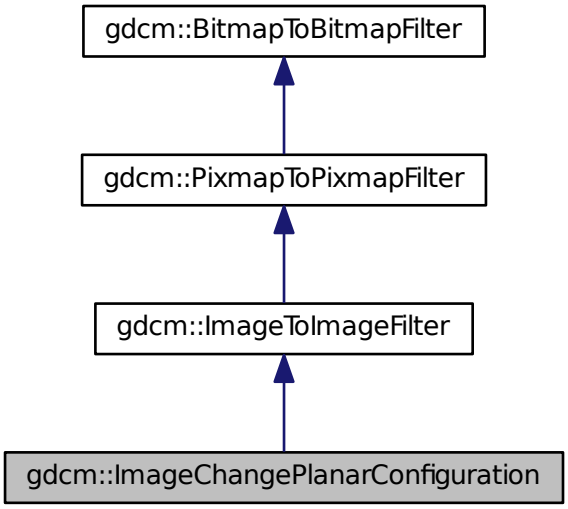
- gdcmImageChangePhotometricInterpretation.h

10.144 gdcm::ImageChangePlanarConfiguration Class Reference

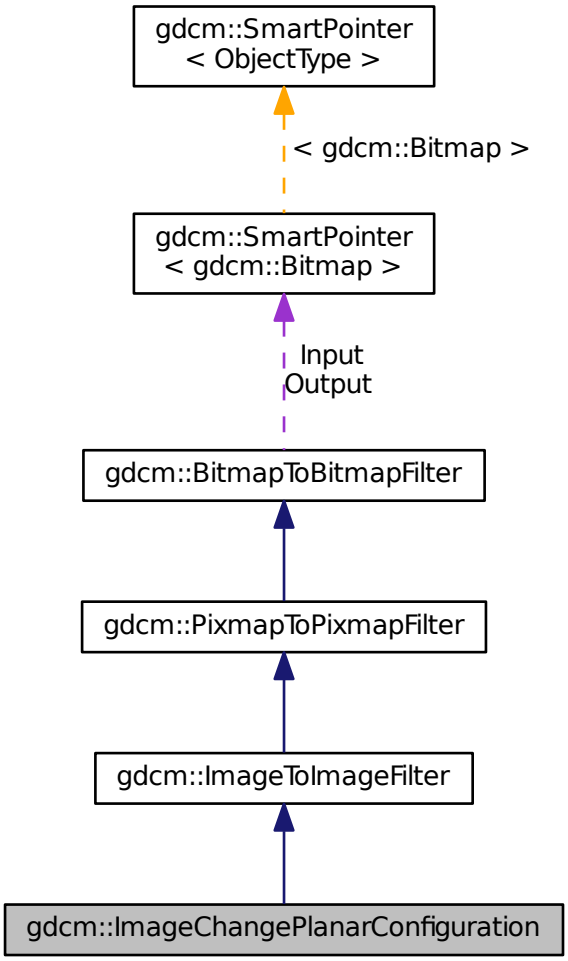
ImageChangePlanarConfiguration class Class to change the Planar configuration of an input DICOM By default it will change into the more usual reprensentation: PlanarConfiguration = 0.

`#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 gdcm::ImageChangePlanarConfiguration::ImageChangePlanarConfiguration ( ) [inline]

10.144.2.2 gdcm::ImageChangePlanarConfiguration::~ImageChangePlanarConfiguration ( ) [inline]

10.144.3 Member Function Documentation

10.144.3.1 bool gdcm::ImageChangePlanarConfiguration::Change ( )

Change.

10.144.3.2 unsigned int gdcm::ImageChangePlanarConfiguration::GetPlanarConfiguration ( ) const [inline]

10.144.3.3 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 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 void gdcm::ImageChangePlanarConfiguration::SetPlanarConfiguration (unsigned int *pc*) [inline]

Set/Get requested PlanarConfigation.

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

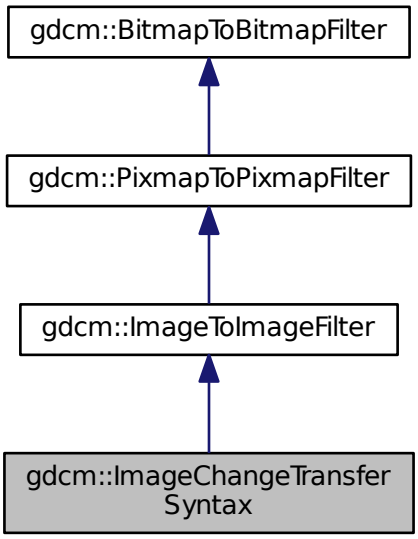
- gdcmImageChangePlanarConfiguration.h

10.145 gdcm::ImageChangeTransferSyntax Class Reference

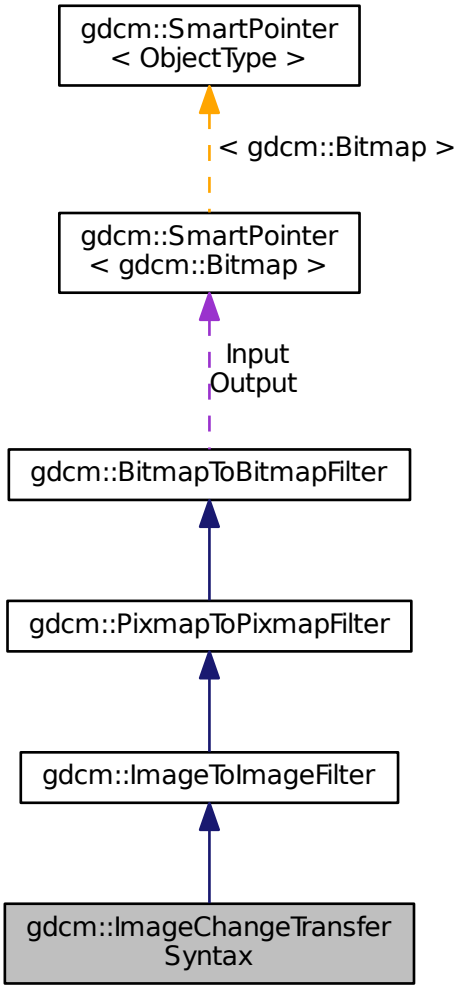
ImageChangeTransferSyntax class Class to change the transfer syntax of an input DICOM.

```
#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 gdcm::ImageChangeTransferSyntax::ImageChangeTransferSyntax ( ) [inline]

10.145.2.2 gdcm::ImageChangeTransferSyntax::~ImageChangeTransferSyntax ( ) [inline]

10.145.3 Member Function Documentation

10.145.3.1 bool gdcm::ImageChangeTransferSyntax::Change ( )

Change.

Examples:

CompressImage.cxx.

10.145.3.2 const TransferSyntax& gdcm::ImageChangeTransferSyntax::GetTransferSyntax ( ) const [inline]

Get Transfer Syntax.

10.145.3.3 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 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 void gdcm::ImageChangeTransferSyntax::SetTransferSyntax (const TransferSyntax & *ts*) [inline]

Set target Transfer Syntax.

Examples:

CompressImage.cxx.

10.145.3.6 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 bool gdcm::ImageChangeTransferSyntax::TryJPEG2000Codec (const DataElement & *pixelde,* Bitmap const & *input,* Bitmap & *output*) [protected]

10.145.3.8 bool gdcm::ImageChangeTransferSyntax::TryJPEGCodec (const DataElement & *pixelde,* Bitmap const & *input,* Bitmap & *output*) [protected]

10.145.3.9 bool gdcm::ImageChangeTransferSyntax::TryJPEGLSCodec (const DataElement & *pixelde,* Bitmap const & *input,* Bitmap & *output*) [protected]

10.145.3.10 bool gdcm::ImageChangeTransferSyntax::TryRAWCodec (const DataElement & *pixelde,* Bitmap const & *input,* Bitmap & *output*) [protected]

10.145.3.11 bool gdcm::ImageChangeTransferSyntax::TryRLECodec (const DataElement & *pixelde,* Bitmap const & *input,* Bitmap & *output*) [protected]

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

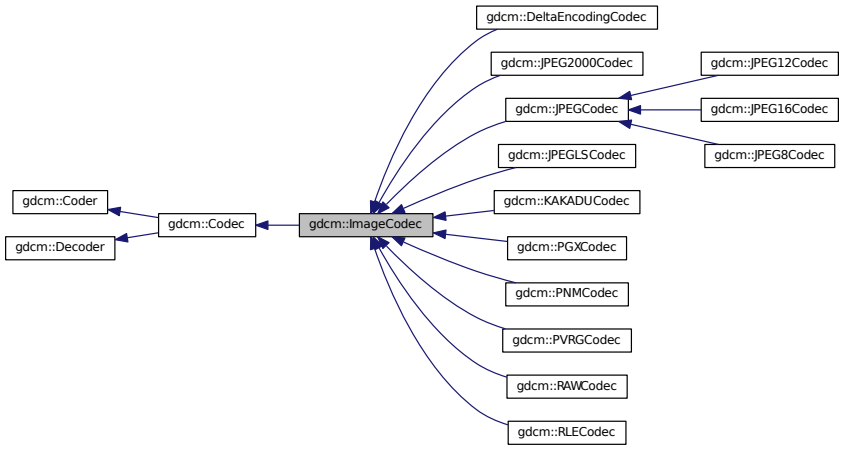
- gdcmImageChangeTransferSyntax.h

10.146 gdcm::ImageCodec Class Reference

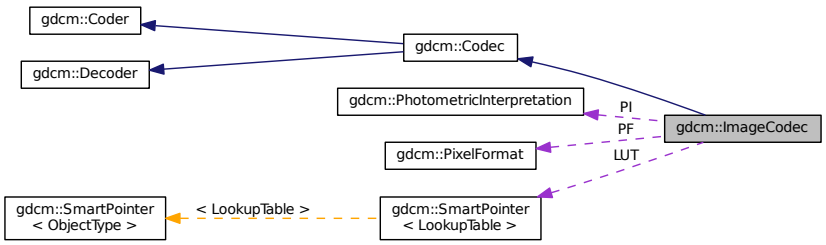
ImageCodec.

`#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 typedef SmartPointer<LookupTable> gdcm::ImageCodec::LUTPtr [protected]

10.146.3 Constructor & Destructor Documentation

10.146.3.1 gdcm::ImageCodec::ImageCodec ( )

10.146.3.2 gdcm::ImageCodec::~ImageCodec ( )

10.146.4 Member Function Documentation

10.146.4.1 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 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 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::JPEGLSCodec, gdcm::PNMCodec, gdcm::PGXCodec, gdcm::KAKADUCodec, and gdcm::RAWCodec.

10.146.4.4 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::JPEGLSCodec, gdcm::PNMCodec, gdcm::RAWCodec, gdcm::PGXCodec, and gdcm::KAKADUCodec.

10.146.4.5 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 bool gdcm::ImageCodec::Decode (DataElement const & , DataElement &) [virtual]

Decode.

Reimplemented from gdcm::Decoder.

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

10.146.4.7 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 bool gdcm::ImageCodec::DoByteSwap (std::istream & *is_*, std::ostream & *os*) [protected]

10.146.4.9 bool gdcm::ImageCodec::DoInvertMonochrome (std::istream & *is_*, std::ostream & *os*) [protected]

10.146.4.10 bool gdcm::ImageCodec::DoOverlayCleanup (std::istream & *is_*, std::ostream & *os*) [protected]

10.146.4.11 bool gdcm::ImageCodec::DoPaddedCompositePixelCode (std::istream & *is_*, std::ostream & *os*) [protected]

10.146.4.12 bool gdcm::ImageCodec::DoPlanarConfiguration (std::istream & *is_*, std::ostream & *os*) [protected]

10.146.4.13 bool gdcm::ImageCodec::DoSimpleCopy (std::istream & *is_*, std::ostream & *os*) [protected]

10.146.4.14 bool gdcm::ImageCodec::DoYBR (std::istream & *is_*, std::ostream & *os*) [protected]

10.146.4.15 const unsigned int* gdcm::ImageCodec::GetDimensions ( ) const [inline]

References gdcm::terminal::dim.

10.146.4.16 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 bool gdcm::ImageCodec::GetLossyFlag ( ) const

10.146.4.18 const **LookupTable**& gdcm::ImageCodec::GetLUT ( ) const [inline]

10.146.4.19 bool gdcm::ImageCodec::GetNeedByteSwap ( ) const [inline]

10.146.4.20 unsigned int gdcm::ImageCodec::GetNumberOfDimensions ( ) const

10.146.4.21 const **PhotometricInterpretation**& gdcm::ImageCodec::GetPhotometricInterpretation ( ) const

10.146.4.22 **PixelFormat**& gdcm::ImageCodec::GetPixelFormat ( ) [inline]

Examples:

GetJPEGSamplePrecision.cxx.

10.146.4.23 const PixelFormat& gdcm::ImageCodec::GetPixelFormat ( ) const [inline]

10.146.4.24 unsigned int gdcm::ImageCodec::GetPlanarConfiguration ( ) const [inline]

10.146.4.25 virtual bool gdcm::ImageCodec::IsFrameEncoder ( ) [protected], [virtual]

Reimplemented in gdcm::JPEGCodec, gdcm::JPEGLSCodec, gdcm::JPEG2000Codec, and gdcm::RLECodec.

10.146.4.26 bool gdcm::ImageCodec::IsLossy ( ) const

10.146.4.27 virtual bool gdcm::ImageCodec::IsRowEncoder ( ) [protected], [virtual]

Reimplemented in gdcm::JPEGCodec, gdcm::JPEGLSCodec, gdcm::JPEG2000Codec, and gdcm::RLECodec.

10.146.4.28 virtual bool gdcm::ImageCodec::IsValid (PhotometricInterpretation const & *pi*) [protected], [virtual]

Reimplemented in gdcm::JPEGCodec.

10.146.4.29 void gdcm::ImageCodec::SetDimensions (const unsigned int *d[3]*)

Examples:

ExtractIconFromFile.cxx.

10.146.4.30 void gdcm::ImageCodec::SetDimensions (const std::vector< unsigned int > & *d*)

10.146.4.31 void gdcm::ImageCodec::SetLossyFlag (bool *l*)

10.146.4.32 void gdcm::ImageCodec::SetLUT (LookupTable const & *lut*) [inline]

Examples:

ExtractIconFromFile.cxx.

10.146.4.33 void gdcm::ImageCodec::SetNeedByteSwap (bool *b*) [inline]

10.146.4.34 void gdcm::ImageCodec::SetNeedOverlayCleanup (bool *b*) [inline]

10.146.4.35 void gdcm::ImageCodec::SetNumberOfDimensions (unsigned int *dim*)

10.146.4.36 void gdcm::ImageCodec::SetPhotometricInterpretation (PhotometricInterpretation const & *pi*)

Examples:

ExtractIconFromFile.cxx.

10.146.4.37 virtual void gdcm::ImageCodec::SetPixelFormat (PixelFormat const & *pf*) [inline], [virtual]

Reimplemented in gdcm::JPEGCodec.

Examples:

ExtractIconFromFile.cxx.

10.146.4.38 void gdcm::ImageCodec::SetPlanarConfiguration (unsigned int *pc*) [inline]

10.146.4.39 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 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 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 friend class ImageChangePhotometricInterpretation [friend]

10.146.6 Member Data Documentation

10.146.6.1 unsigned int gdcm::ImageCodec::Dimensions[3] [protected]

10.146.6.2 bool gdcm::ImageCodec::LossyFlag [protected]

10.146.6.3 LUTPtr gdcm::ImageCodec::LUT [protected]

10.146.6.4 bool gdcm::ImageCodec::NeedByteSwap [protected]

10.146.6.5 bool gdcm::ImageCodec::NeedOverlayCleanup [protected]

10.146.6.6 unsigned int gdcm::ImageCodec::NumberOfDimensions [protected]

10.146.6.7 PixelFormat gdcm::ImageCodec::PF [protected]

10.146.6.8 PhotometricInterpretation gdcm::ImageCodec::PI [protected]

10.146.6.9 unsigned int gdcm::ImageCodec::PlanarConfiguration [protected]

10.146.6.10 bool gdcm::ImageCodec::RequestPaddedCompositePixelCode [protected]

10.146.6.11 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 gdcm::ImageConverter::ImageConverter ( )

10.147.2.2 gdcm::ImageConverter::~ImageConverter ( )

10.147.3 Member Function Documentation

10.147.3.1 void gdcm::ImageConverter::Convert ( )

10.147.3.2 const Image& gdcm::ImageConverter::GetOuput ( ) const

10.147.3.3 void gdcm::ImageConverter::SetInput (Image const & *input*)

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

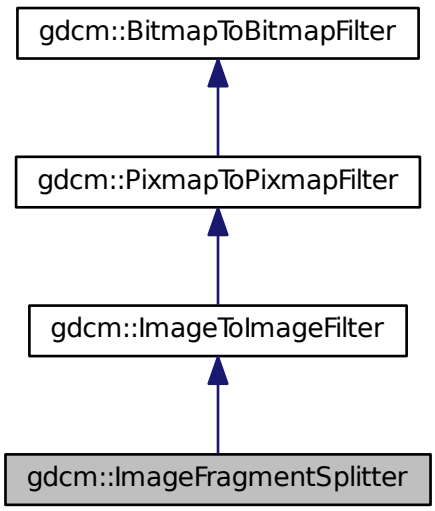
- gdcmImageConverter.h

10.148 gdcm::ImageFragmentSplitter Class Reference

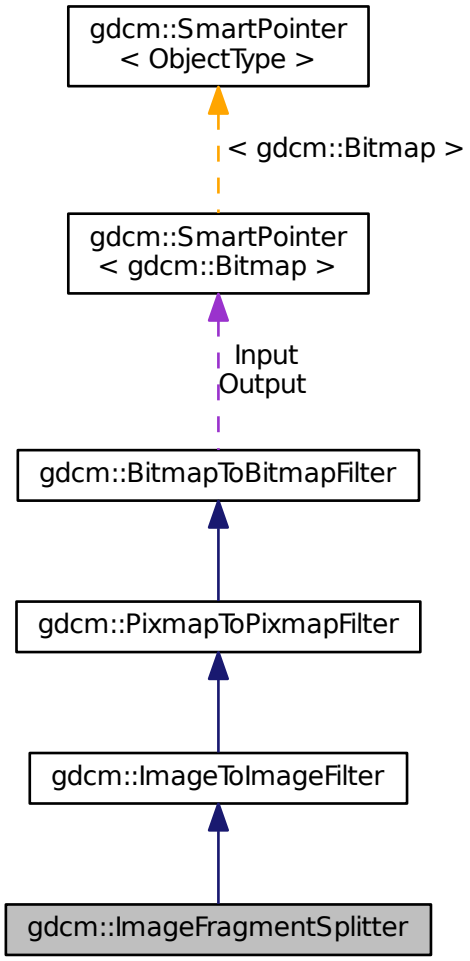
ImageFragmentSplitter class For single frame image, DICOM standard allow splitting the frame into multiple fragments.

`#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 gdcm::ImageFragmentSplitter::ImageFragmentSplitter ( ) [inline]

10.148.2.2 gdcm::ImageFragmentSplitter::~ImageFragmentSplitter ( ) [inline]

10.148.3 Member Function Documentation

10.148.3.1 unsigned int gdcm::ImageFragmentSplitter::GetFragmentSizeMax ( ) const [inline]

10.148.3.2 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 void gdcm::ImageFragmentSplitter::SetFragmentSizeMax (unsigned int *fragsize*)

FragmentSizeMax needs to be an even number.

10.148.3.4 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 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 static bool gdcm::ImageHelper::ComputeSpacingFromImagePositionPatient (const std::vector< double > & *imageposition,* std::vector< double > & *spacing*) [static]

DO NOT USE.

10.149.2.3 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 static bool gdcm::ImageHelper::GetDirectionCosinesFromDataSet (DataSet const & *ds,* std::vector< double > & *dircos*) [static]

10.149.2.5 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 static bool gdcm::ImageHelper::GetForcePixelSpacing ( ) [static]

10.149.2.7 static bool gdcm::ImageHelper::GetForceRescaleInterceptSlope ( ) [static]

10.149.2.8 static SmartPointer<LookupTable> gdcm::ImageHelper::GetLUT (File const & *f*) [static]

returns the lookup table of an image file

10.149.2.9 static std::vector<double> gdcm::ImageHelper::GetOriginValue (File const & *f*) [static]

Set/Get Origin (IPP) from/to a file.

10.149.2.10 static **PhotometricInterpretation** gdcm::ImageHelper::GetPhotometricInterpretationValue (**File** const & *f*) [static]

10.149.2.11 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 static unsigned int gdcm::ImageHelper::GetPlanarConfigurationValue (const **File** & *f*) [static]

10.149.2.13 static bool gdcm::ImageHelper::GetPMSRescaleInterceptSlope ( ) [static]

10.149.2.14 static const **ByteValue** * gdcm::ImageHelper::GetPointerFromElement (**Tag** const & *tag,* **File** const & *f*) [static]

10.149.2.15 static bool gdcm::ImageHelper::GetRealWorldValueMappingContent (**File** const & *f,* **RealWorldValueMappingContent** & *rwvmc*) [static]

10.149.2.16 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 static **Tag** gdcm::ImageHelper::GetSpacingTagFromMediaStorage (**MediaStorage** const & *ms*) [static], [protected]

10.149.2.18 static std::vector<double> gdcm::ImageHelper::GetSpacingValue (**File** const & *f*) [static]

Set/Get Spacing from/to a File.

10.149.2.19 static **Tag** gdcm::ImageHelper::GetZSpacingTagFromMediaStorage (**MediaStorage** const & *ms*) [static], [protected]

10.149.2.20 static void gdcm::ImageHelper::SetDimensionsValue (**File** & *f,* const **Pixmap** & *img*) [static]

10.149.2.21 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 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 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 static void gdcm::ImageHelper::SetOriginValue (DataSet & *ds,* const Image & *img*) [static]

10.149.2.25 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 static void gdcm::ImageHelper::SetRescaleInterceptSlopeValue (File & *f,* const Image & *img*) [static]

10.149.2.27 static void gdcm::ImageHelper::SetSpacingValue (DataSet & *ds,* const std::vector< double > & *spacing*) [static]

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

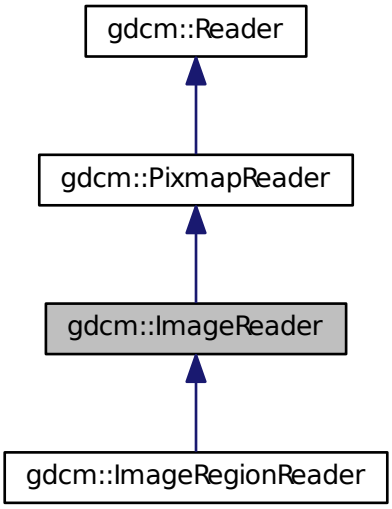
- gdcmImageHelper.h

10.150 gdcm::ImageReader Class Reference

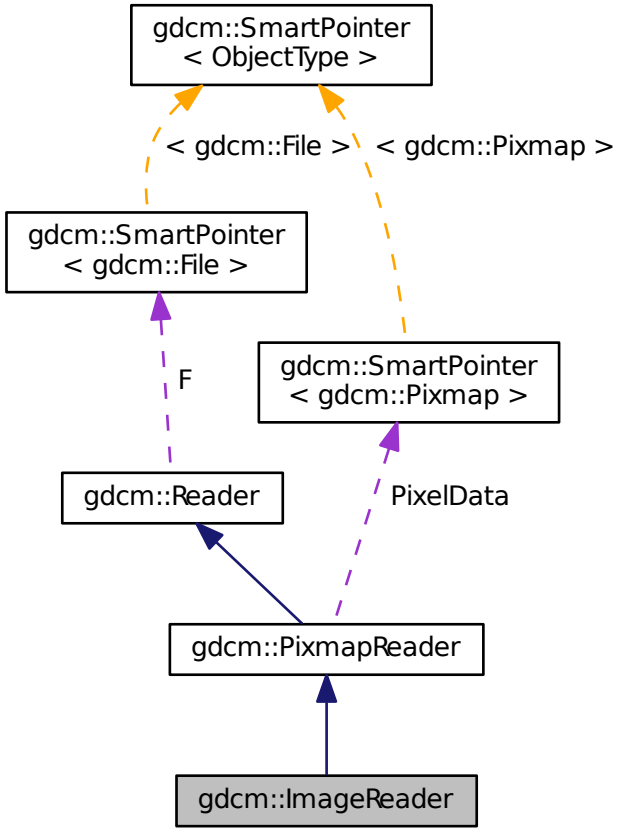
ImageReader.

```
#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 gdcm::ImageReader::ImageReader ( )

10.150.2.2 virtual gdcm::ImageReader::~ImageReader ( ) [virtual]

10.150.3 Member Function Documentation

10.150.3.1 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 Image& gdcm::ImageReader::GetImage ( )

10.150.3.3 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 bool gdcm::ImageReader::ReadACRNEMAImage ( ) [protected], [virtual]

Reimplemented from gdcm::PixmapReader.

10.150.3.5 bool gdcm::ImageReader::ReadImage (MediaStorage const & *ms*) [protected], [virtual]

Reimplemented from gdcm::PixmapReader.

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

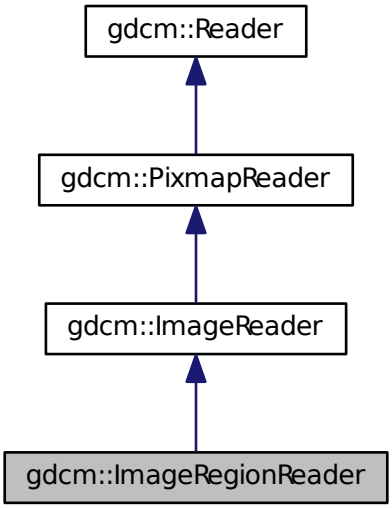
- gdcmImageReader.h

10.151 gdcm::ImageRegionReader Class Reference

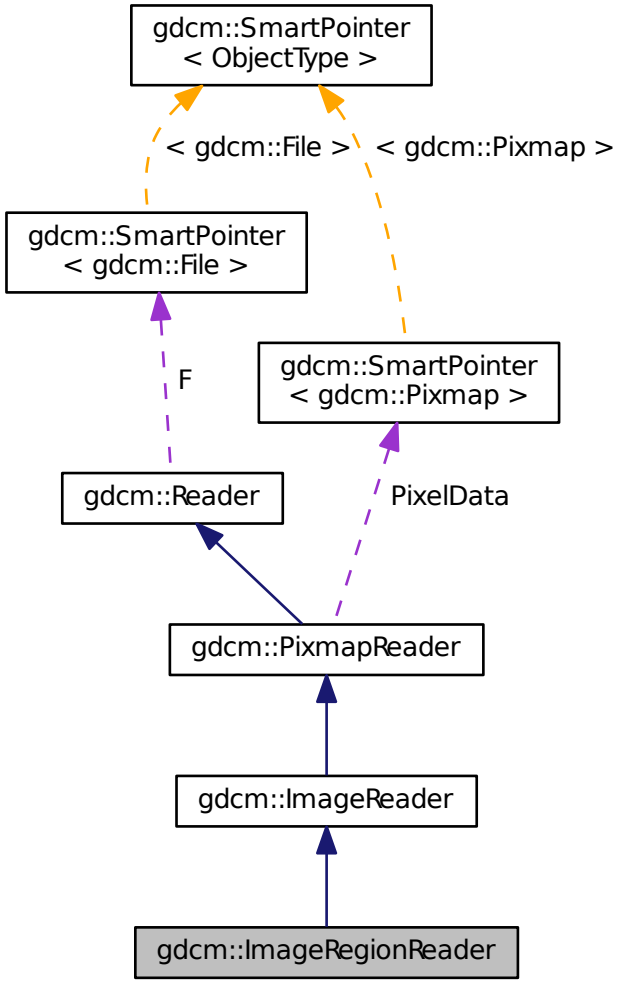
ImageRegionReader.

```
#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.

See also

ImageReader

10.151.2 Constructor & Destructor Documentation

10.151.2.1 gdcm::ImageRegionReader::ImageRegionReader ( )

10.151.2.2 gdcm::ImageRegionReader::~ImageRegionReader ( )

10.151.3 Member Function Documentation

10.151.3.1 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 Region const& gdcm::ImageRegionReader::GetRegion ( ) const

10.151.3.3 bool gdcm::ImageRegionReader::Read ( ) [protected], [virtual]

To prevent user from calling super class Read() function.

Reimplemented from gdcm::ImageReader.

10.151.3.4 bool gdcm::ImageRegionReader::ReadInformation ( )

Read meta information (not Pixel Data) from the DICOM file.

Returns

false upon error

10.151.3.5 bool gdcm::ImageRegionReader::ReadIntoBuffer (char * *inreadbuffer*, size_t *buflen*)

Read into buffer:

Returns

false upon error

10.151.3.6 void gdcm::ImageRegionReader::SetRegion (Region const & *region*)

Set/Get Region to be read.

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

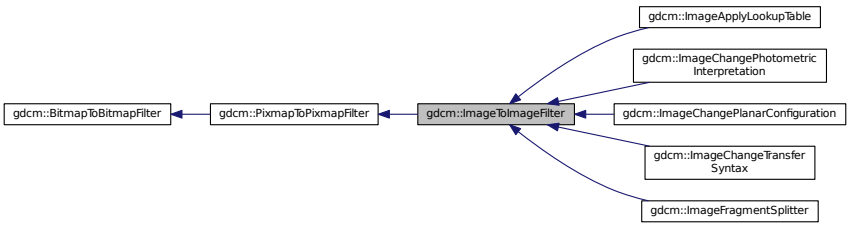
- gdcmImageRegionReader.h

10.152 gdcm::ImageToImageFilter Class Reference

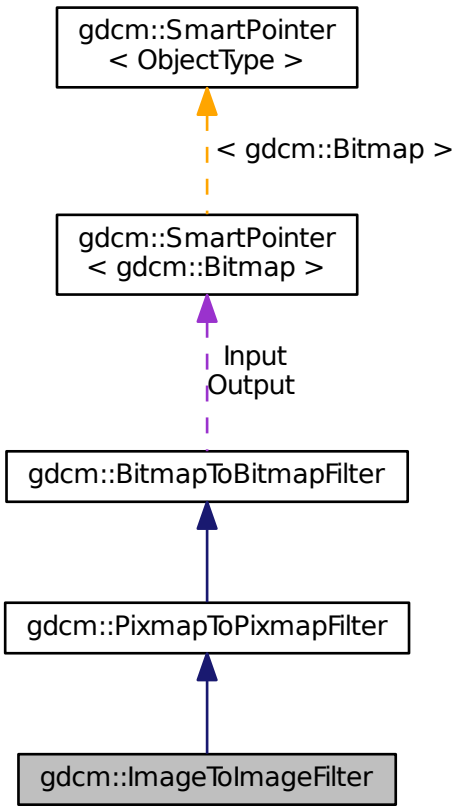
ImageToImageFilter class Super class for all filter taking an image and producing an output image.

`#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 gdcm::ImageToImageFilter::ImageToImageFilter ( )

10.152.2.2 gdcm::ImageToImageFilter::~ImageToImageFilter ( ) [inline]

10.152.3 Member Function Documentation

10.152.3.1 Image& gdcm::ImageToImageFilter::GetInput ( )

10.152.3.2 const Image& gdcm::ImageToImageFilter::GetOutput ( ) const

Get Output image.

Examples:

CompressImage.cxx.

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

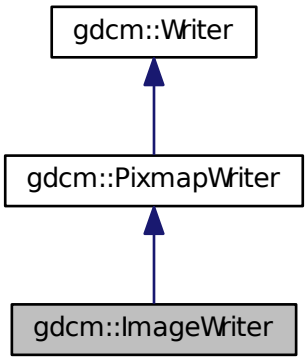
- gdcmImageToImageFilter.h

10.153 gdcm::ImageWriter Class Reference

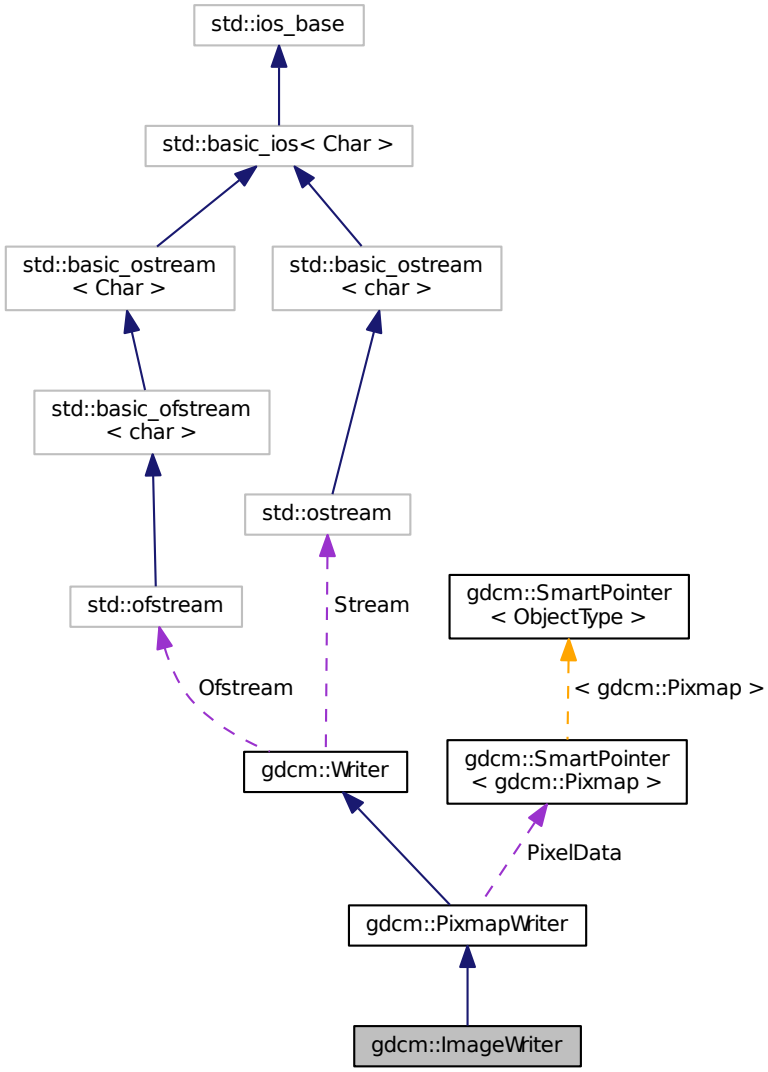
ImageWriter.

`#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 gdcm::ImageWriter::ImageWriter ( )

10.153.2.2 gdcm::ImageWriter::~ImageWriter ( )

10.153.3 Member Function Documentation

10.153.3.1 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 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 Image& gdcm::ImageWriter::GetImage ( ) [inline], [virtual]

Reimplemented from gdcm::PixmapWriter.

10.153.3.4 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 PS 3.7 Table D.3-1 IMPLEMENTATION CLASS UID SUB-ITEM FIELDS (A-ASSOCIATE-RQ)

`#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 gdcm::network::ImplementationClassUIDSub::ImplementationClassUIDSub ( )

10.154.3 Member Function Documentation

10.154.3.1 void gdcm::network::ImplementationClassUIDSub::Print (std::ostream & *os*) const

10.154.3.2 std::istream& gdcm::network::ImplementationClassUIDSub::Read (std::istream & *is*)

10.154.3.3 size_t gdcm::network::ImplementationClassUIDSub::Size ( ) const

10.154.3.4 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 Table D.3-2 IMPLEMENTATION UID SUB-ITEM FIELDS (A-ASSOCIATE-AC)

```
#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 gdcm::network::ImplementationUIDSub::ImplementationUIDSub ( )

10.155.3 Member Function Documentation

10.155.3.1 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 Table D.3-3 IMPLEMENTATION VERSION NAME SUB-ITEM FIELDS (A-ASSOCI↩ ATE-RQ)

```
#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-ASSOCI↩ ATE-RQ)

10.156.2 Constructor & Destructor Documentation

10.156.2.1 gdcm::network::ImplementationVersionNameSub::ImplementationVersionNameSub ( )

10.156.3 Member Function Documentation

10.156.3.1 void gdcm::network::ImplementationVersionNameSub::Print (std::ostream & *os*) const

10.156.3.2 std::istream& gdcm::network::ImplementationVersionNameSub::Read (std::istream & *is*)

10.156.3.3 size_t gdcm::network::ImplementationVersionNameSub::Size ( ) const

10.156.3.4 const std::ostream& gdcm::network::ImplementationVersionNameSub::Write (std::ostream & *os*) const

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

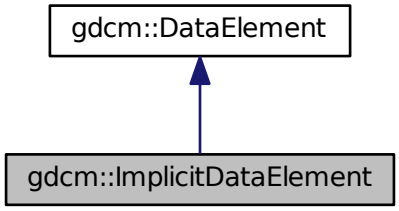
- gdcmImplementationVersionNameSub.h

10.157 gdcm::ImplicitDataElement Class Reference

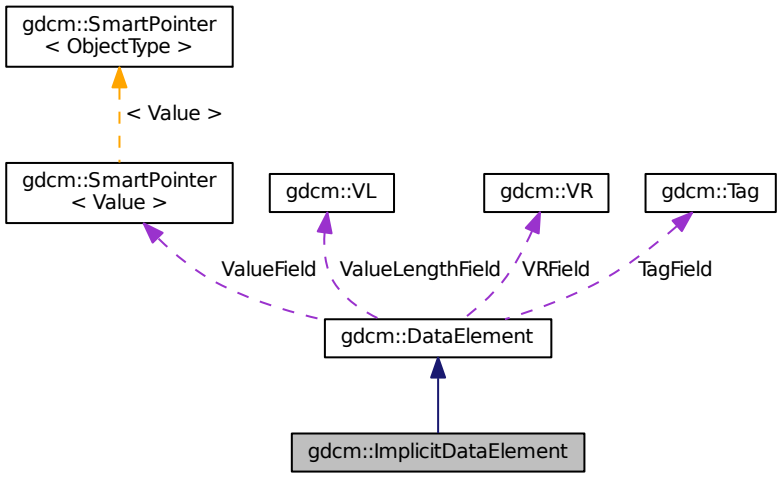
Class to represent an *Implicit VR* Data Element.

`#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 VL gdcm::ImplicitDataElement::GetLength ( ) const

10.157.2.2 template<typename TSwap > std::istream& gdcm::ImplicitDataElement::Read (std::istream & *is*)

10.157.2.3 template<typename TSwap > std::istream& gdcm::ImplicitDataElement::ReadPreValue (std::istream & *is*)

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

10.157.2.5 template<typename TSwap > std::istream& gdcm::ImplicitDataElement::ReadValueWithLength (std::istream & *is*, VL & *length*, bool *readvalues* = `true`)

10.157.2.6 template<typename TSwap > std::istream& gdcm::ImplicitDataElement::ReadWithLength (std::istream & *is*, VL & *length*, bool *readvalues* = `true`)

10.157.2.7 template<typename TSwap > const std::ostream& gdcm::ImplicitDataElement::Write (std::ostream & *os*) const

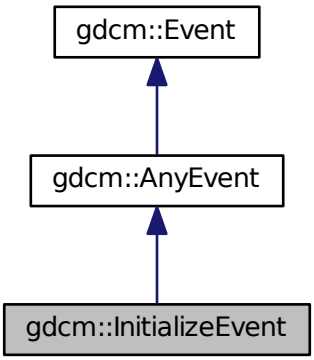
The documentation for this class was generated from the following file:

- gdcmImplicitDataElement.h

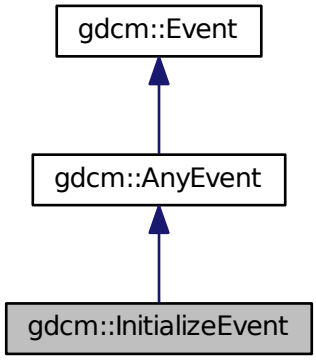
10.158 gdcm::InitializeEvent Class Reference

`#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 typedef std::vector<IODEntry> gdcm::IOD::MapIODEntry

10.159.2.2 typedef MapIODEntry::size_type gdcm::IOD::SizeType

10.159.3 Constructor & Destructor Documentation

10.159.3.1 gdcm::IOD::IOD ( ) [inline]

References gdcm::operator<<().

10.159.4 Member Function Documentation

10.159.4.1 void gdcm::IOD::AddIODEntry (const IODEntry & *iode*) [inline]

10.159.4.2 void gdcm::IOD::Clear ( ) [inline]

10.159.4.3 const IODEntry& gdcm::IOD::GetIODEntry (SizeType *idx*) const [inline]

Examples:

TraverseModules.cxx.

10.159.4.4 SizeType gdcm::IOD::GetNumberOfIODs ( ) const [inline]

Examples:

TraverseModules.cxx.

10.159.4.5 Type gdcm::IOD::GetTypeFromTag (const Defs & *defs,* const Tag & *tag*) const

10.159.5 Friends And Related Function Documentation

10.159.5.1 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 MAN↩ DATORY 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 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 const char* gdcm::IODEntry::GetIE ( ) const [inline]

10.160.3.2 const char* gdcm::IODEntry::GetName ( ) const [inline]

10.160.3.3 const char* gdcm::IODEntry::GetRef ( ) const [inline]

Examples:

TraverseModules.cxx.

10.160.3.4 const char* gdcm::IODEntry::GetUsage ( ) const [inline]

10.160.3.5 Usage::UsageType gdcm::IODEntry::GetUsageType ( ) const

10.160.3.6 void gdcm::IODEntry::SetIE (const char * *ie*) [inline]

10.160.3.7 void gdcm::IODEntry::SetName (const char * *name*) [inline]

10.160.3.8 void gdcm::IODEntry::SetRef (const char * *ref*) [inline]

10.160.3.9 void gdcm::IODEntry::SetUsage (const char * *usag*) [inline]

10.160.4 Friends And Related Function Documentation

10.160.4.1 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 typedef std::map<IODName, IOD> gdcm::IODs::IODMapType

10.161.2.2 typedef IODMapType::const_iterator gdcm::IODs::IODMapTypeConstIterator

10.161.2.3 typedef std::string gdcm::IODs::IODName

10.161.3 Constructor & Destructor Documentation

10.161.3.1 gdcm::IODs::IODs ( ) [inline]

References gdcm::operator<<().

10.161.4 Member Function Documentation

10.161.4.1 void gdcm::IODs::AddIOD (const char * *name*, const IOD & *module*) [inline]

10.161.4.2 IODMapTypeConstIterator gdcm::IODs::Begin ( ) const [inline]

Examples:

TraverseModules.cxx.

10.161.4.3 void gdcm::IODs::Clear ( ) [inline]

10.161.4.4 IODMapTypeConstIterator gdcm::IODs::End ( ) const [inline]

Examples:

TraverseModules.cxx.

10.161.4.5 const IOD& gdcm::IODs::GetIOD (const char * *name*) const [inline]

10.161.5 Friends And Related Function Documentation

10.161.5.1 std::ostream& operator<< (std::ostream & *_os,* const IODs & *_val*) [friend]

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

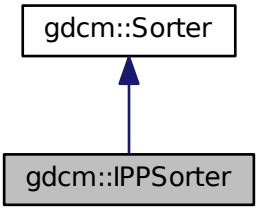
- gdcmIODs.h

10.162 gdcm::IPPSorter Class Reference

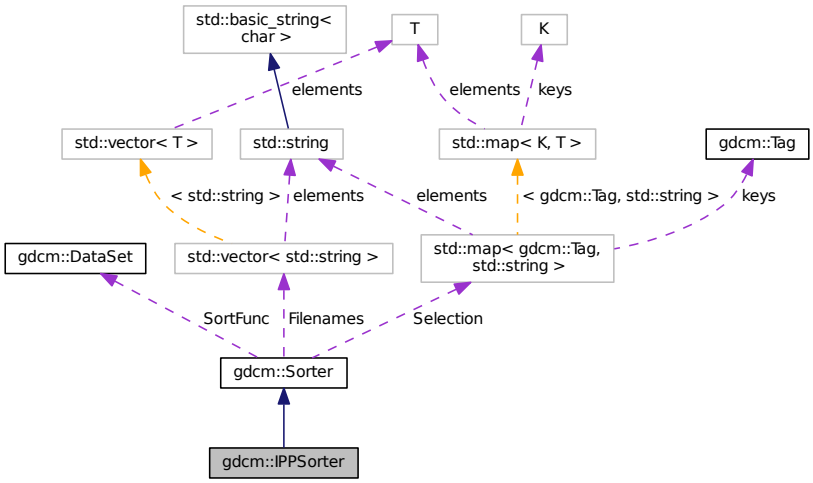
IPPSorter Implement a simple Image Position (Patient) sorter, along the Image Orientation (Patient) direction. This algorithm does NOT support duplicate and will FAIL in case of duplicate IPP.

```
#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 gdcm::IPPSorter::IPPSorter ( )

10.162.3 Member Function Documentation

10.162.3.1 double gdcm::IPPSorter::GetDirectionCosinesTolerance ( ) const [inline]

10.162.3.2 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 double gdcm::IPPSorter::GetZSpacingTolerance ( ) const [inline]

10.162.3.4 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 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 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 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 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 bool gdcm::IPPSorter::ComputeZSpacing [protected]

10.162.4.2 double gdcm::IPPSorter::DirCosTolerance [protected]

10.162.4.3 bool gdcm::IPPSorter::DropDuplicatePositions [protected]

10.162.4.4 double gdcm::IPPSorter::ZSpacing [protected]

10.162.4.5 double gdcm::IPPSorter::ZTolerance [protected]

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

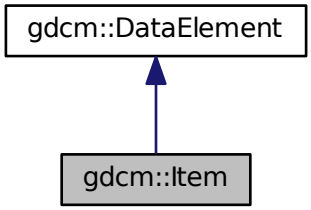
- gdcmIPPSorter.h

10.163 gdcm::Item Class Reference

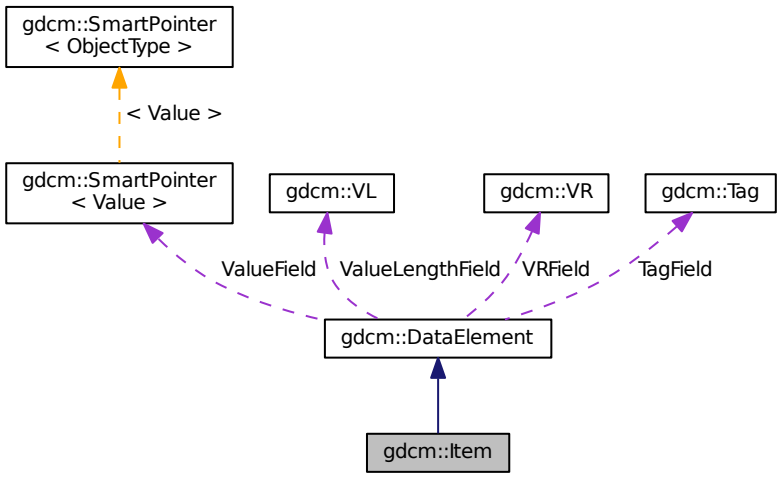
Class to represent an Item A component of the value of a Data Element that is of Value Representation Sequence of Items. An Item contains a Data Set . See PS 3.5 7.5.1 Item Encoding Rules Each Item of a Data Element of VR SQ shall be encoded as a DICOM Standart Data Element with a specific Data Element 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.

```
#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 gdcm::Item::Item ( ) [inline]

References gdcm::operator<<().

10.163.2.2 gdcm::Item::Item (Item const & *val*) [inline]

10.163.3 Member Function Documentation

10.163.3.1 void gdcm::Item::Clear ( ) [inline]

Referenced by gdcm::SequenceOfItems::Read().

10.163.3.2 bool gdcm::Item::FindDataElement (const Tag & *t*) const [inline]

Examples:

ReadAndDumpDICOMDIR.cxx.

10.163.3.3 const DataElement& gdcm::Item::GetDataElement (const Tag & *t*) const [inline]

Examples:

ReadAndDumpDICOMDIR.cxx.

10.163.3.4 template<typename TDE > VL gdcm::Item::GetLength ( ) const

10.163.3.5 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 DataSet& gdcm::Item::GetNestedDataSet ( ) [inline]

10.163.3.7 void gdcm::Item::InsertDataElement (const DataElement & *de*) [inline]

10.163.3.8 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 void gdcm::Item::SetNestedDataSet (const DataSet & *nested*) [inline]

10.163.3.10 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 std::ostream& operator<< (std::ostream & *os,* const Item & *val*) [friend]

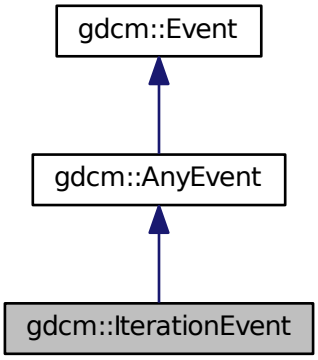
The documentation for this class was generated from the following file:

- gdcmItem.h

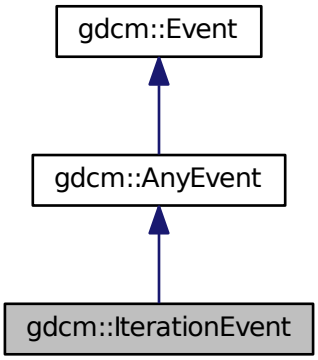
10.164 gdcm::IterationEvent Class Reference

`#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:

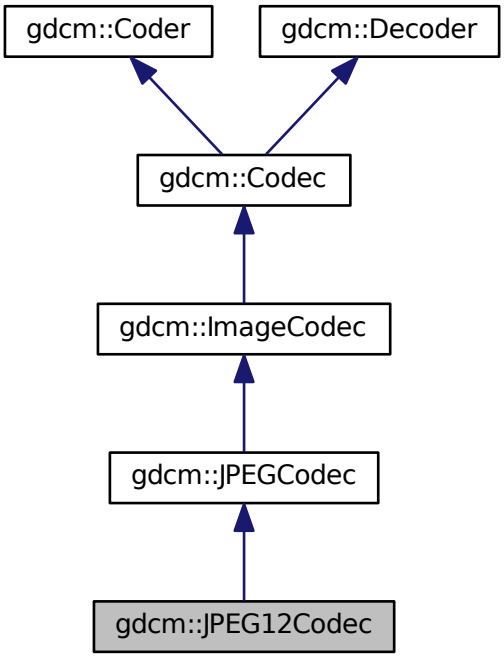
- gdcmEvent.h

10.165 gdcm::JPEG12Codec Class Reference

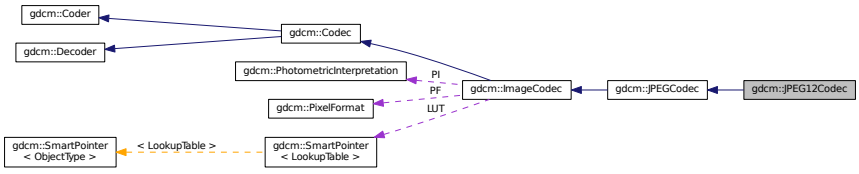
Class to do JPEG 12bits (lossy & lossless)

`#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 gdcm::JPEG12Codec::JPEG12Codec ( )

10.165.2.2 gdcm::JPEG12Codec::∼JPEG12Codec ( )

10.165.3 Member Function Documentation

10.165.3.1 bool gdcm::JPEG12Codec::DecodeByStreams (std::istream & *is,* std::ostream & *os*) [virtual]

Reimplemented from gdcm::ImageCodec.

10.165.3.2 virtual bool gdcm::JPEG12Codec::EncodeBuffer (std::ostream & *os,* const char ∗ *data,* size_t *datalen*) [protected], [virtual]

Reimplemented from gdcm::JPEGCodec.

10.165.3.3 bool gdcm::JPEG12Codec::GetHeaderInfo (std::istream & *is,* TransferSyntax & *ts*) [virtual]

Reimplemented from gdcm::JPEGCodec.

10.165.3.4 bool gdcm::JPEG12Codec::InternalCode (const char * *input*, unsigned long *len*, std::ostream & *os*) [virtual]

Reimplemented from gdcm::Coder.

10.165.3.5 bool gdcm::JPEG12Codec::IsStateSuspension ( ) const [protected], [virtual]

Reimplemented from gdcm::JPEGCodec.

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

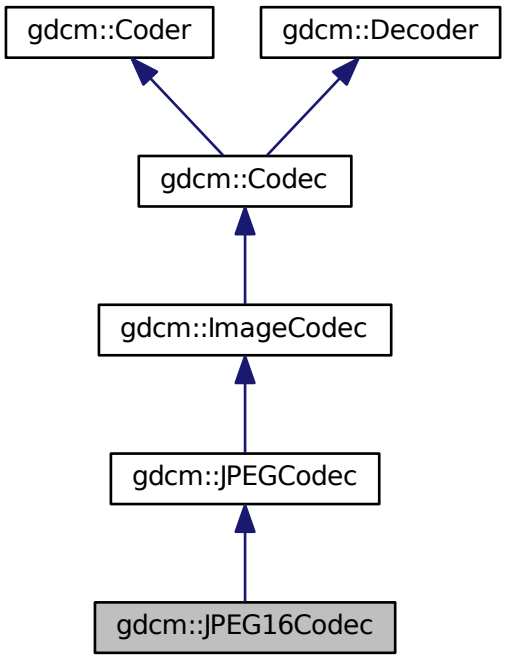
- gdcmJPEG12Codec.h

10.166 gdcm::JPEG16Codec Class Reference

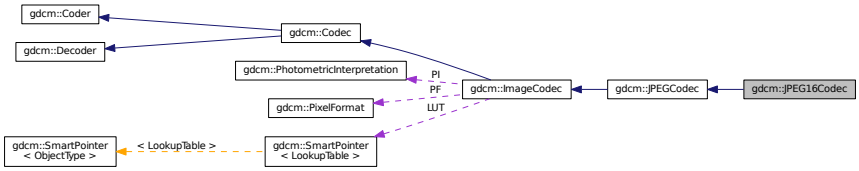
Class to do JPEG 16bits (lossless)

`#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 gdcm::JPEG16Codec::JPEG16Codec ( )

10.166.2.2 gdcm::JPEG16Codec::∼JPEG16Codec ( )

10.166.3 Member Function Documentation

10.166.3.1 bool gdcm::JPEG16Codec::DecodeByStreams (std::istream & *is*, std::ostream & *os*) [virtual]

Reimplemented from gdcm::ImageCodec.

10.166.3.2 virtual bool gdcm::JPEG16Codec::EncodeBuffer (std::ostream & *os,* const char * *data,* size_t *datalen*) [protected], [virtual]

Reimplemented from gdcm::JPEGCodec.

10.166.3.3 bool gdcm::JPEG16Codec::GetHeaderInfo (std::istream & *is,* TransferSyntax & *ts*) [virtual]

Reimplemented from gdcm::JPEGCodec.

10.166.3.4 bool gdcm::JPEG16Codec::InternalCode (const char * *input,* unsigned long *len,* std::ostream & *os*) [virtual]

Reimplemented from gdcm::Coder.

10.166.3.5 bool gdcm::JPEG16Codec::IsStateSuspension ( ) const [protected], [virtual]

Reimplemented from gdcm::JPEGCodec.

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

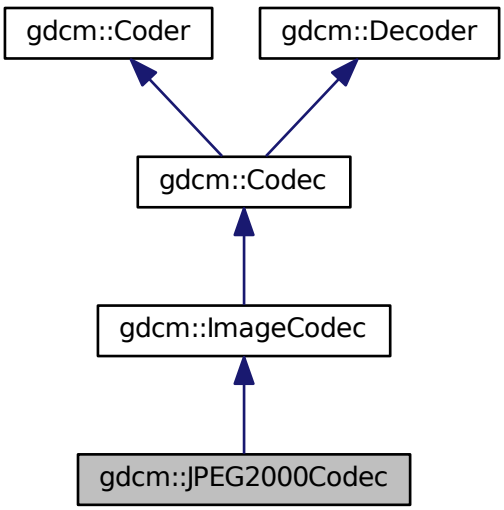
- gdcmJPEG16Codec.h

10.167 gdcm::JPEG2000Codec Class Reference

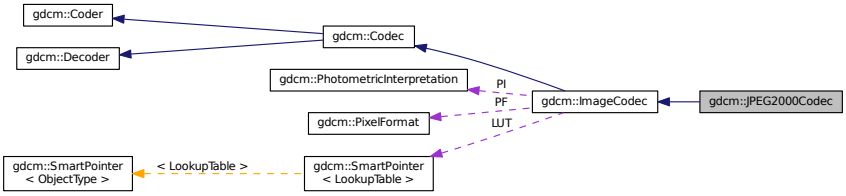
Class to do JPEG 2000.

`#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 gdcm::JPEG2000Codec::JPEG2000Codec ( )

10.167.2.2 gdcm::JPEG2000Codec::~JPEG2000Codec ( )

10.167.3 Member Function Documentation

10.167.3.1 bool gdcm::JPEG2000Codec::AppendFrameEncode (std::ostream & *out,* const char * *data,* size_t *datalen*) [protected], [virtual]

Reimplemented from gdcm::ImageCodec.

10.167.3.2 bool gdcm::JPEG2000Codec::AppendRowEncode (std::ostream & *out,* const char * *data,* size_t *datalen*) [protected], [virtual]

Reimplemented from gdcm::ImageCodec.

10.167.3.3 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 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 virtual ImageCodec* gdcm::JPEG2000Codec::Clone ( ) const [virtual]

Implements gdcm::ImageCodec.

10.167.3.6 bool gdcm::JPEG2000Codec::Code (DataElement const & *in_*, DataElement & *out_*) [virtual]

Code.

Reimplemented from gdcm::Coder.

10.167.3.7 bool gdcm::JPEG2000Codec::Decode (DataElement const &, DataElement &) [virtual]

Decode.

Reimplemented from gdcm::ImageCodec.

10.167.3.8 bool gdcm::JPEG2000Codec::DecodeByStreams (std::istream & *is*, std::ostream & *os*) [protected], [virtual]

Reimplemented from gdcm::ImageCodec.

10.167.3.9 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 virtual bool gdcm::JPEG2000Codec::GetHeaderInfo (std::istream & *is*, TransferSyntax & *ts*) [virtual]

Reimplemented from gdcm::ImageCodec.

10.167.3.11 double gdcm::JPEG2000Codec::GetQuality (unsigned int *idx* = 0) const

10.167.3.12 double gdcm::JPEG2000Codec::GetRate (unsigned int *idx* = 0) const

10.167.3.13 bool gdcm::JPEG2000Codec::IsFrameEncoder ( ) [protected], [virtual]

Reimplemented from gdcm::ImageCodec.

10.167.3.14 bool gdcm::JPEG2000Codec::IsRowEncoder ( ) [protected], [virtual]

Reimplemented from gdcm::ImageCodec.

10.167.3.15 void gdcm::JPEG2000Codec::SetNumberOfResolutions (unsigned int *nres*)

10.167.3.16 void gdcm::JPEG2000Codec::SetQuality (unsigned int *idx*, double *q*)

10.167.3.17 void gdcm::JPEG2000Codec::SetRate (unsigned int *idx*, double *rate*)

10.167.3.18 void gdcm::JPEG2000Codec::SetReversible (bool *res*)

10.167.3.19 void gdcm::JPEG2000Codec::SetTileSize (unsigned int *tx*, unsigned int *ty*)

10.167.3.20 bool gdcm::JPEG2000Codec::StartEncode (std::ostream &) [protected], [virtual]

Reimplemented from gdcm::ImageCodec.

10.167.3.21 bool gdcm::JPEG2000Codec::StopEncode (std::ostream &) [protected], [virtual]

Reimplemented from gdcm::ImageCodec.

10.167.4 Friends And Related Function Documentation

10.167.4.1 friend class Bitmap [friend]

10.167.4.2 friend class ImageRegionReader [friend]

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

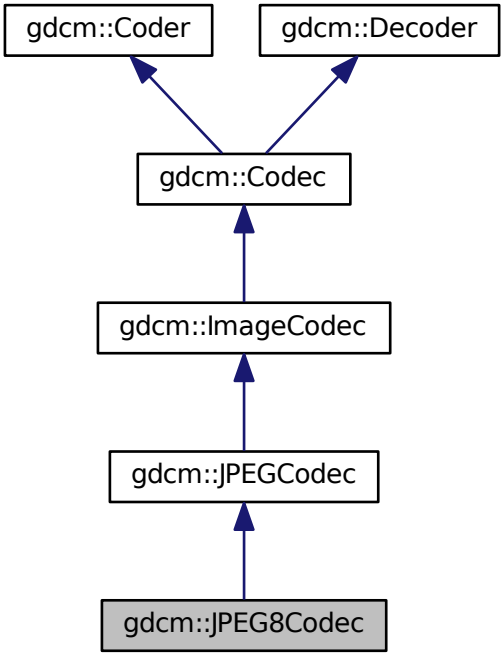
- gdcmJPEG2000Codec.h

10.168 gdcm::JPEG8Codec Class Reference

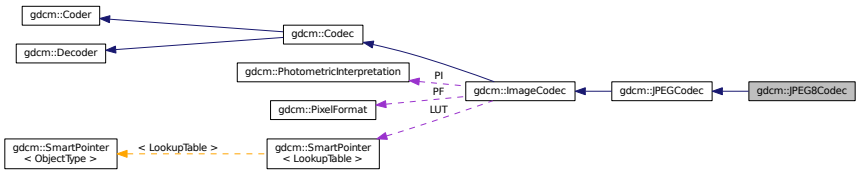
Class to do JPEG 8bits (lossy & lossless)

`#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 gdcm::JPEG8Codec::JPEG8Codec ( )

10.168.2.2 gdcm::JPEG8Codec::∼JPEG8Codec ( )

10.168.3 Member Function Documentation

10.168.3.1 bool gdcm::JPEG8Codec::DecodeByStreams (std::istream & *is*, std::ostream & *os*) [virtual]

Reimplemented from gdcm::ImageCodec.

10.168.3.2 virtual bool gdcm::JPEG8Codec::EncodeBuffer (std::ostream & *os*, const char ∗ *data*, size_t *datalen*) [protected], [virtual]

Reimplemented from gdcm::JPEGCodec.

10.168.3.3 bool gdcm::JPEG8Codec::GetHeaderInfo (std::istream & *is*, TransferSyntax & *ts*) [virtual]

Reimplemented from gdcm::JPEGCodec.

10.168.3.4 bool gdcm::JPEG8Codec::InternalCode (const char * *input,* unsigned long *len,* std::ostream & *os*) [virtual]

Reimplemented from gdcm::Coder.

10.168.3.5 bool gdcm::JPEG8Codec::IsStateSuspension ( ) const [protected], [virtual]

Reimplemented from gdcm::JPEGCodec.

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

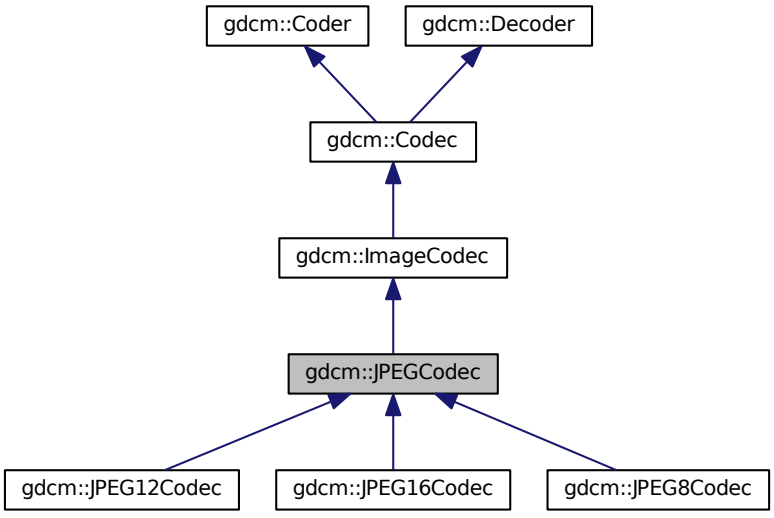
- gdcmJPEG8Codec.h

10.169 gdcm::JPEGCodec Class Reference

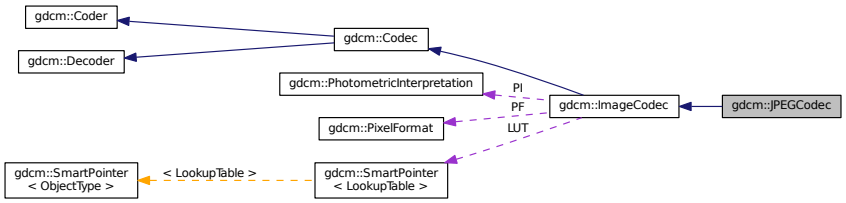
JPEG codec Class to do JPEG (8bits, 12bits, 16bits lossy & lossless). It redispatch in between the different codec implementation: JPEG8Codec, JPEG12Codec & JPEG16Codec It also support inconsistency in between DICOM header and JPEG compressed stream ImageCodec implementation for the JPEG case.

```
#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 gdcm::JPEGCodec::JPEGCodec ( )

10.169.2.2 gdcm::JPEGCodec::~JPEGCodec ( )

10.169.3 Member Function Documentation

10.169.3.1 bool gdcm::JPEGCodec::AppendFrameEncode (std::ostream & *out*, const char * *data*, size_t *datalen*) [protected], [virtual]

Reimplemented from gdcm::ImageCodec.

10.169.3.2 bool gdcm::JPEGCodec::AppendRowEncode (std::ostream & *out*, const char * *data*, size_t *datalen*) [protected], [virtual]

Reimplemented from gdcm::ImageCodec.

10.169.3.3 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 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 virtual ImageCodec* gdcm::JPEGCodec::Clone ( ) const [virtual]

Implements gdcm::ImageCodec.

10.169.3.6 bool gdcm::JPEGCodec::Code (DataElement const & *in*, DataElement & *out*) [virtual]

Compress into JPEG.

Reimplemented from gdcm::Coder.

10.169.3.7 void gdcm::JPEGCodec::ComputeOffsetTable (bool *b*)

Compute the offset table:

10.169.3.8 bool gdcm::JPEGCodec::Decode (DataElement const & , DataElement &) [virtual]

Decode.

Reimplemented from gdcm::ImageCodec.

10.169.3.9 bool gdcm::JPEGCodec::DecodeByStreams (std::istream & *is*, std::ostream & *os*) [protected], [virtual]

Reimplemented from gdcm::ImageCodec.

10.169.3.10 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 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 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 bool gdcm::JPEGCodec::GetLossless ( ) const

10.169.3.14 double gdcm::JPEGCodec::GetQuality ( ) const

10.169.3.15 bool gdcm::JPEGCodec::IsFrameEncoder ( ) [protected], [virtual]

Reimplemented from gdcm::ImageCodec.

10.169.3.16 bool gdcm::JPEGCodec::IsRowEncoder ( ) [protected], [virtual]

Reimplemented from gdcm::ImageCodec.

10.169.3.17 virtual bool gdcm::JPEGCodec::IsStateSuspension ( ) const [protected], [virtual]

Reimplemented in gdcm::JPEG12Codec, gdcm::JPEG16Codec, and gdcm::JPEG8Codec.

10.169.3.18 bool gdcm::JPEGCodec::IsValid (**PhotometricInterpretation** const & *pi*) [protected], [virtual]

Reimplemented from gdcm::ImageCodec.

10.169.3.19 void gdcm::JPEGCodec::SetBitSample (int *bit*) [protected]

10.169.3.20 void gdcm::JPEGCodec::SetLossless (bool *l*)

10.169.3.21 void gdcm::JPEGCodec::SetPixelFormat (PixelFormat const & *pf*) [virtual]

Reimplemented from gdcm::ImageCodec.

Examples:

GetJPEGSamplePrecision.cxx.

10.169.3.22 void gdcm::JPEGCodec::SetQuality (double *q*)

10.169.3.23 bool gdcm::JPEGCodec::StartEncode (std::ostream &) [protected], [virtual]

Reimplemented from gdcm::ImageCodec.

10.169.3.24 bool gdcm::JPEGCodec::StopEncode (std::ostream &) [protected], [virtual]

Reimplemented from gdcm::ImageCodec.

10.169.4 Friends And Related Function Documentation

10.169.4.1 friend class ImageRegionReader [friend]

10.169.5 Member Data Documentation

10.169.5.1 int gdcm::JPEGCodec::BitSample [protected]

10.169.5.2 int gdcm::JPEGCodec::Quality [protected]

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

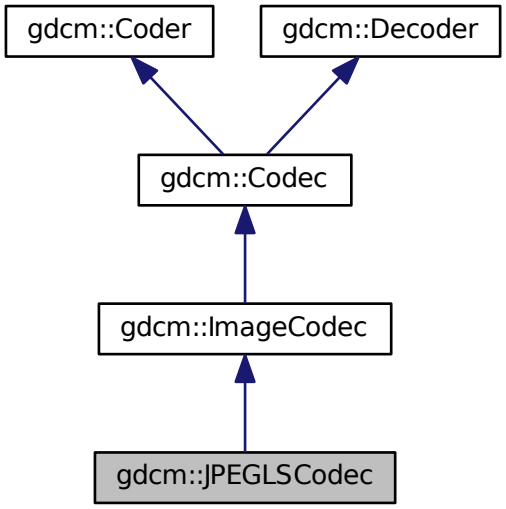
- gdcmJPEGCodec.h

10.170 gdcm::JPEGLSCodec Class Reference

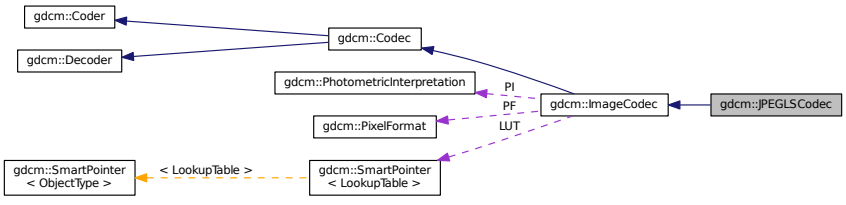
JPEG-LS.

`#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 `http://charls.codeplex.com`

10.170.2 Constructor & Destructor Documentation

10.170.2.1 gdcm::JPEGLSCodec::JPEGLSCodec ( )

10.170.2.2 gdcm::JPEGLSCodec::~JPEGLSCodec ( )

10.170.3 Member Function Documentation

10.170.3.1 bool gdcm::JPEGLSCodec::AppendFrameEncode (std::ostream & *out*, const char * *data*, size_t *datalen*) [protected], [virtual]

Reimplemented from gdcm::ImageCodec.

10.170.3.2 bool gdcm::JPEGLSCodec::AppendRowEncode (std::ostream & *out*, const char * *data*, size_t *datalen*) [protected], [virtual]

Reimplemented from gdcm::ImageCodec.

10.170.3.3 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 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 virtual ImageCodec* gdcm::JPEGLSCodec::Clone ( ) const [virtual]

Implements gdcm::ImageCodec.

10.170.3.6 bool gdcm::JPEGLSCodec::Code (DataElement const & *in_*, DataElement & *out_*) [virtual]

Code.

Reimplemented from gdcm::Coder.

10.170.3.7 bool gdcm::JPEGLSCodec::Decode (DataElement const & , DataElement &) [virtual]

Decode.

Reimplemented from gdcm::ImageCodec.

10.170.3.8 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 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 unsigned long gdcm::JPEGLSCodec::GetBufferLength ( ) const [inline]

10.170.3.11 bool gdcm::JPEGLSCodec::GetHeaderInfo (std::istream & *is*, TransferSyntax & *ts*) [virtual]

Reimplemented from gdcm::ImageCodec.

10.170.3.12 bool gdcm::JPEGLSCodec::GetLossless ( ) const

10.170.3.13 bool gdcm::JPEGLSCodec::IsFrameEncoder ( ) [protected], [virtual]

Reimplemented from gdcm::ImageCodec.

10.170.3.14 bool gdcm::JPEGLSCodec::IsRowEncoder ( ) [protected], [virtual]

Reimplemented from gdcm::ImageCodec.

10.170.3.15 void gdcm::JPEGLSCodec::SetBufferLength (unsigned long *l*) [inline]

10.170.3.16 void gdcm::JPEGLSCodec::SetLossless (bool *l*)

10.170.3.17 void gdcm::JPEGLSCodec::SetLossyError (int *error*)

[0-3] generally

10.170.3.18 bool gdcm::JPEGLSCodec::StartEncode (std::ostream &) [protected], [virtual]

Reimplemented from gdcm::ImageCodec.

10.170.3.19 bool gdcm::JPEGLSCodec::StopEncode (std::ostream &) [protected], [virtual]

Reimplemented from gdcm::ImageCodec.

10.170.4 Friends And Related Function Documentation

10.170.4.1 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 gdcm::JSON::JSON ( )

10.171.2.2 gdcm::JSON::~JSON ( )

10.171.3 Member Function Documentation

10.171.3.1 bool gdcm::JSON::Code (DataSet const & *in*, std::ostream & *os*)

Examples:

QIDO-RS.cxx.

10.171.3.2 bool gdcm::JSON::Decode (std::istream & *is*, DataSet & *out*)

Examples:

QIDO-RS.cxx.

10.171.3.3 bool gdcm::JSON::GetPrettyPrint ( ) const

10.171.3.4 void gdcm::JSON::PrettyPrintOff ( )

10.171.3.5 void gdcm::JSON::PrettyPrintOn ( )

Examples:

QIDO-RS.cxx.

10.171.3.6 void gdcm::JSON::SetPrettyPrint (bool *onoff*)

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

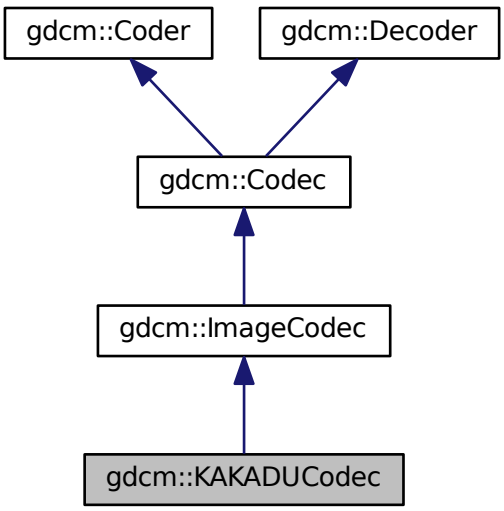
- gdcmJSON.h

10.172 gdcm::KAKADUCodec Class Reference

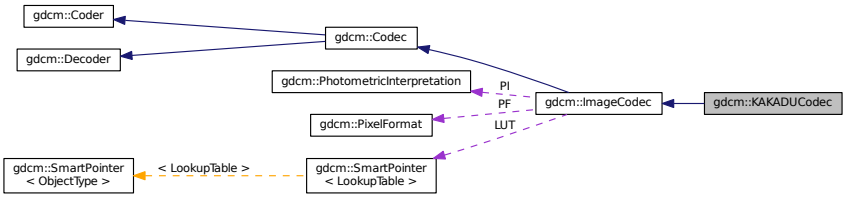
KAKADUCodec.

`#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 gdcm::KAKADUCodec::KAKADUCodec ( )

10.172.2.2 gdcm::KAKADUCodec::~KAKADUCodec ( )

10.172.3 Member Function Documentation

10.172.3.1 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 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 virtual ImageCodec* gdcm::KAKADUCodec::Clone ( ) const [virtual]

Implements gdcm::ImageCodec.

10.172.3.4 bool gdcm::KAKADUCodec::Code (DataElement const & *in_*, DataElement & *out_*) [virtual]

Code.

Reimplemented from gdcm::Coder.

10.172.3.5 bool gdcm::KAKADUCodec::Decode (DataElement const &, DataElement &) [virtual]

Decode.

Reimplemented from gdcm::ImageCodec.

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

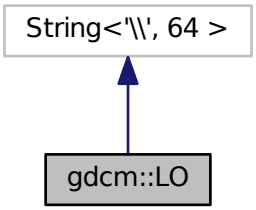
- gdcmKAKADUCodec.h

10.173 gdcm::LO Class Reference

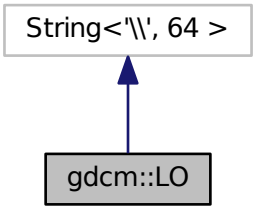
LO.

`#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 typedef Superclass::const_iterator gdcm::LO::const_iterator

10.173.2.2 typedef Superclass::const_reference gdcm::LO::const_reference

10.173.2.3 typedef Superclass::const_reverse_iterator gdcm::LO::const_reverse_iterator

10.173.2.4 typedef Superclass::difference_type gdcm::LO::difference_type

10.173.2.5 typedef Superclass::iterator gdcm::LO::iterator

10.173.2.6 typedef Superclass::pointer gdcm::LO::pointer

10.173.2.7 typedef Superclass::reference gdcm::LO::reference

10.173.2.8 typedef Superclass::reverse_iterator gdcm::LO::reverse_iterator

10.173.2.9 typedef Superclass::size_type gdcm::LO::size_type

10.173.2.10 typedef String<'\\',64> gdcm::LO::Superclass

10.173.2.11 typedef Superclass::value_type gdcm::LO::value_type

10.173.3 Constructor & Destructor Documentation

10.173.3.1 gdcm::LO::LO ( ) [inline]

10.173.3.2 gdcm::LO::LO (const value_type * *s*) [inline]

10.173.3.3 gdcm::LO::LO (const value_type * *s*, size_type *n*) [inline]

10.173.3.4 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 bool gdcm::LO::IsValid ( ) const [inline]

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

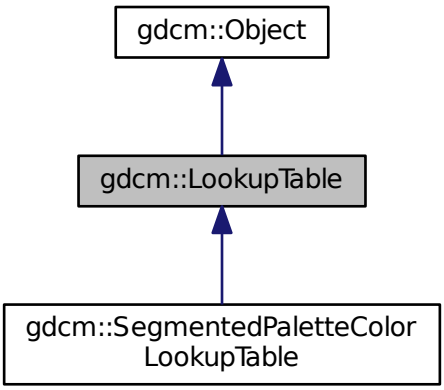
- gdcmLO.h

10.174 gdcm::LookupTable Class Reference

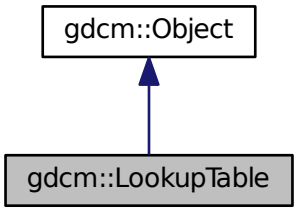
LookupTable class.

`#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 enum gdcm::LookupTable::LookupTableType

Enumerator

RED

GREEN

BLUE

GRAY

UNKNOWN

10.174.3 Constructor & Destructor Documentation

10.174.3.1 gdcm::LookupTable::LookupTable ( )

10.174.3.2 gdcm::LookupTable::~LookupTable ( )

10.174.3.3 gdcm::LookupTable::LookupTable (LookupTable const & *lut*) [inline]

10.174.4 Member Function Documentation

10.174.4.1 void gdcm::LookupTable::Allocate (unsigned short *bitsample* = 8)

Allocate the LUT.

10.174.4.2 void gdcm::LookupTable::Clear ( )

Clear the LUT.

10.174.4.3 void gdcm::LookupTable::Decode (std::istream & *is*, std::ostream & *os*) const

Decode the LUT.

10.174.4.4 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 unsigned short gdcm::LookupTable::GetBitSample ( ) const [inline]

return the bit sample

10.174.4.6 bool gdcm::LookupTable::GetBufferAsRGBA (unsigned char * *rgba*) const

return the LUT as RGBA buffer

10.174.4.7 void gdcm::LookupTable::GetLUT (LookupTableType *type,* unsigned char * *array,* unsigned int & *length*) const

Examples:

ExtractImageRegionWithLUT.cs.

10.174.4.8 void gdcm::LookupTable::GetLUTDescriptor (LookupTableType *type,* unsigned short & *length,* unsigned short & *subscript,* unsigned short & *bitsize*) const

10.174.4.9 unsigned int gdcm::LookupTable::GetLUTLength (LookupTableType *type*) const

10.174.4.10 const unsigned char* gdcm::LookupTable::GetPointer ( ) const

return a raw pointer to the LUT

10.174.4.11 void gdcm::LookupTable::InitializeBlueLUT (unsigned short *length,* unsigned short *subscript,* unsigned short *bitsize*)

10.174.4.12 bool gdcm::LookupTable::Initialized ( ) const

return whether the LUT has been initialized

10.174.4.13 void gdcm::LookupTable::InitializeGreenLUT (unsigned short *length,* unsigned short *subscript,* unsigned short *bitsize*)

10.174.4.14 void gdcm::LookupTable::InitializeLUT (LookupTableType *type,* unsigned short *length,* unsigned short *subscript,* unsigned short *bitsize*)

Generic interface:

10.174.4.15 void gdcm::LookupTable::InitializeRedLUT (unsigned short *length,* unsigned short *subscript,* unsigned short *bitsize*)

RED / GREEN / BLUE specific:

10.174.4.16 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 void gdcm::LookupTable::SetBlueLUT (const unsigned char * *blue*, unsigned int *length*)

10.174.4.18 void gdcm::LookupTable::SetGreenLUT (const unsigned char * *green*, unsigned int *length*)

10.174.4.19 virtual void gdcm::LookupTable::SetLUT (LookupTableType *type*, const unsigned char * *array*, unsigned int *length*) [virtual]

Reimplemented in gdcm::SegmentedPaletteColorLookupTable.

10.174.4.20 void gdcm::LookupTable::SetRedLUT (const unsigned char * *red*, unsigned int *length*)

10.174.4.21 bool gdcm::LookupTable::WriteBufferAsRGBA (const unsigned char * *rgba*)

Write the LUT as RGBA.

10.174.5 Member Data Documentation

10.174.5.1 unsigned short gdcm::LookupTable::BitSample [protected]

10.174.5.2 bool gdcm::LookupTable::IncompleteLUT [protected]

10.174.5.3 LookupTableInternal* gdcm::LookupTable::Internal [protected]

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

- gdcmLookupTable.h

10.175 gdcm::Scanner::Itstr 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 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 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 typedef std::vector<std::string> gdcm::Macro::ArrayIncludeMacrosType

10.177.2.2 typedef std::map<Tag, MacroEntry> gdcm::Macro::MapModuleEntry

10.177.3 Constructor & Destructor Documentation

10.177.3.1 gdcm::Macro::Macro ( ) [inline]

References gdcm::operator<<().

10.177.4 Member Function Documentation

10.177.4.1 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 void gdcm::Macro::Clear ( ) [inline]

10.177.4.3 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 const MacroEntry& gdcm::Macro::GetMacroEntry (const Tag & *tag*) const

10.177.4.5 const char* gdcm::Macro::GetName ( ) const [inline]

10.177.4.6 void gdcm::Macro::SetName (const char * *name*) [inline]

10.177.4.7 bool gdcm::Macro::Verify (const DataSet & *ds,* Usage const & *usage*) const

10.177.5 Friends And Related Function Documentation

10.177.5.1 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 typedef std::map<std::string, Macro> gdcm::Macros::ModuleMapType

10.178.3 Constructor & Destructor Documentation

10.178.3.1 gdcm::Macros::Macros ( ) [inline]

References gdcm::operator<<().

10.178.4 Member Function Documentation

10.178.4.1 void gdcm::Macros::AddMacro (const char * *ref,* const Macro & *module*) [inline]

10.178.4.2 void gdcm::Macros::Clear ( ) [inline]

10.178.4.3 const Macro& gdcm::Macros::GetMacro (const char * *name*) const [inline]

10.178.4.4 bool gdcm::Macros::IsEmpty ( ) const [inline]

10.178.5 Friends And Related Function Documentation

10.178.5.1 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 Annex D Table D.1-1 MAXIMUM LENGTH SUB-ITEM FIELDS (A-ASSOCIATE-RQ)

`#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 gdcm::network::MaximumLengthSub::MaximumLengthSub ( )

10.179.3 Member Function Documentation

10.179.3.1 uint32_t gdcm::network::MaximumLengthSub::GetMaximumLength ( ) const [inline]

References Print(), and SetMaximumLength().

10.179.3.2 void gdcm::network::MaximumLengthSub::Print (std::ostream & *os*) const

Referenced by GetMaximumLength().

10.179.3.3 std::istream& gdcm::network::MaximumLengthSub::Read (std::istream & *is*)

10.179.3.4 void gdcm::network::MaximumLengthSub::SetMaximumLength (uint32_t *maximumlength*)

Referenced by GetMaximumLength().

10.179.3.5 size_t gdcm::network::MaximumLengthSub::Size ( ) const

10.179.3.6 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 gdcm::MD5::MD5 ( )

10.180.2.2 gdcm::MD5::~MD5 ( )

10.180.3 Member Function Documentation

10.180.3.1 static bool gdcm::MD5::Compute (const char * *buffer,* unsigned long *buf_len,* char *digest_str[33]*) [static]

10.180.3.2 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,

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 enum gdcm::MediaStorage::MSType

Enumerator

MediaStorageDirectoryStorage

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
MS_END

Examples:

GenerateStandardSOPClasses.cxx.

10.181.2.2 enum gdcm::MediaStorage::ObjectType

Enumerator

NoObject
Video
Waveform
Audio
PDF
URI
Segmentation
ObjectEnd

10.181.3 Constructor & Destructor Documentation

10.181.3.1 gdcm::MediaStorage::MediaStorage (MSType *type* = MS_END) [inline]

10.181.4 Member Function Documentation

10.181.4.1 const char* gdcm::MediaStorage::GetModality ( ) const

10.181.4.2 unsigned int gdcm::MediaStorage::GetModalityDimension ( ) const

10.181.4.3 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 static MSType gdcm::MediaStorage::GetMSType (const char * *str*) [static]

Examples:

MetaImageMD5Activiz.cs, and TestReader.cxx.

10.181.4.5 static unsigned int gdcm::MediaStorage::GetNumberOfModality ( ) [static]

10.181.4.6 static unsigned int gdcm::MediaStorage::GetNumberOfMSString ( ) [static]

10.181.4.7 static unsigned int gdcm::MediaStorage::GetNumberOfMSType ( ) [static]

10.181.4.8 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 void gdcm::MediaStorage::GuessFromModality (const char * *modality,* unsigned int *dimension* = 2)

10.181.4.10 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 bool gdcm::MediaStorage::IsUndefined ( ) const [inline]

Examples:

TestReader.cxx.

10.181.4.12 gdcm::MediaStorage::operator MSType ( ) const [inline]

References gdcm::operator<<().

10.181.4.13 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 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 bool gdcm::MediaStorage::SetFromHeader (FileMetaInformation const & *fmi*)

10.181.4.16 bool gdcm::MediaStorage::SetFromModality (DataSet const & *ds*)

10.181.4.17 void gdcm::MediaStorage::SetFromSourceImageSequence (DataSet const & *ds*) [protected]

10.181.5 Friends And Related Function Documentation

10.181.5.1 std::ostream& operator<< (std::ostream & *os*, const MediaStorage & *ms*) [friend]

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

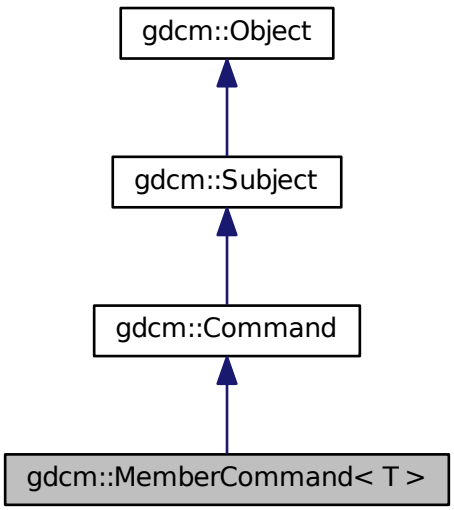
- gdcmMediaStorage.h

10.182 gdcm::MemberCommand< T > Class Template Reference

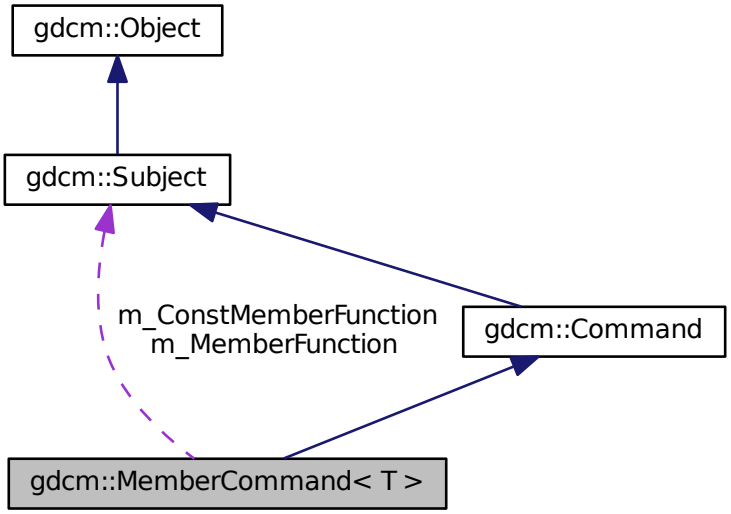
Command subclass that calls a pointer to a member function.

`#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 template<class T > typedef MemberCommand gdcm::MemberCommand< T >::Self

Standard class typedefs.

10.182.2.2 template<class T > typedef void(T::* gdcm::MemberCommand< T >::TConstMemberFunctionPointer) (const Subject *, const Event &)

10.182.2.3 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 template<class T > gdcm::MemberCommand< T >::MemberCommand ( ) [inline], [protected]

10.182.3.2 template<class T > virtual gdcm::MemberCommand< T >::~MemberCommand ( ) [inline], [protected], [virtual]

10.182.4 Member Function Documentation

10.182.4.1 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 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 template<class T > static SmartPointer<MemberCommand> gdcm::MemberCommand< T >::New ( ) [inline], [static]

Method for creation through the object factory.

10.182.4.4 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 template<class T > void gdcm::MemberCommand< T >::SetCallbackFunction (T * *object,* TConstMemberFunctionPointer *memberFunction*) [inline]

10.182.5 Member Data Documentation

10.182.5.1 template<class T > TConstMemberFunctionPointer gdcm::MemberCommand< T >::m_ConstMemberFunction [protected]

10.182.5.2 template<class T > TMemberFunctionPointer gdcm::MemberCommand< T >::m_MemberFunction [protected]

10.182.5.3 template<class T > T* gdcm::MemberCommand< T >::m_This [protected]

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

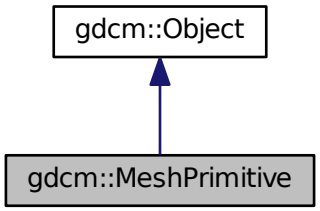
- gdcmCommand.h

10.183 gdcm::MeshPrimitive Class Reference

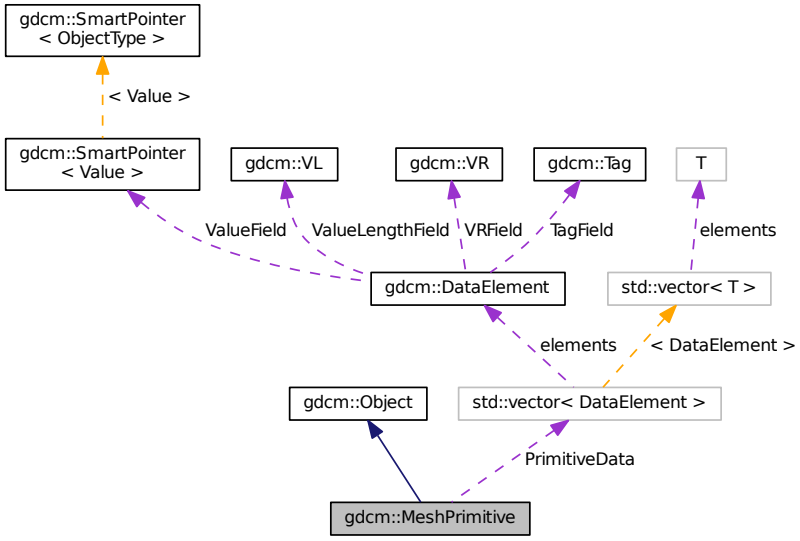
This class defines surface mesh primitives. It is designed from surface mesh primitives macro.

`#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 typedef std::vector< DataElement > gdcm::MeshPrimitive::PrimitivesData

10.183.3 Member Enumeration Documentation

10.183.3.1 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 gdcm::MeshPrimitive::MeshPrimitive ( )

10.183.4.2 virtual gdcm::MeshPrimitive::~MeshPrimitive ( ) [virtual]

10.183.5 Member Function Documentation

10.183.5.1 void gdcm::MeshPrimitive::AddPrimitiveData (DataElement const & *de*)

10.183.5.2 static MPType gdcm::MeshPrimitive::GetMPType (const char * *type*) [static]

10.183.5.3 static const char* gdcm::MeshPrimitive::GetMPTypeString (const MPType *type*) [static]

10.183.5.4 unsigned int gdcm::MeshPrimitive::GetNumberOfPrimitivesData ( ) const

10.183.5.5 const DataElement& gdcm::MeshPrimitive::GetPrimitiveData ( ) const

10.183.5.6 DataElement& gdcm::MeshPrimitive::GetPrimitiveData ( )

10.183.5.7 const DataElement& gdcm::MeshPrimitive::GetPrimitiveData (const unsigned int *idx*) const

10.183.5.8 DataElement& gdcm::MeshPrimitive::GetPrimitiveData (const unsigned int *idx*)

10.183.5.9 const PrimitivesData& gdcm::MeshPrimitive::GetPrimitivesData ( ) const

10.183.5.10 PrimitivesData& gdcm::MeshPrimitive::GetPrimitivesData ( )

10.183.5.11 MPType gdcm::MeshPrimitive::GetPrimitiveType ( ) const

10.183.5.12 void gdcm::MeshPrimitive::SetPrimitiveData (DataElement const & *de*)

10.183.5.13 void gdcm::MeshPrimitive::SetPrimitiveData (const unsigned int *idx,* DataElement const & *de*)

10.183.5.14 void gdcm::MeshPrimitive::SetPrimitivesData (PrimitivesData const & *DEs*)

10.183.5.15 void gdcm::MeshPrimitive::SetPrimitiveType (const MPType *type*)

10.183.6 Member Data Documentation

10.183.6.1 PrimitivesData gdcm::MeshPrimitive::PrimitiveData [protected]

10.183.6.2 MPType gdcm::MeshPrimitive::PrimitiveType [protected]

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

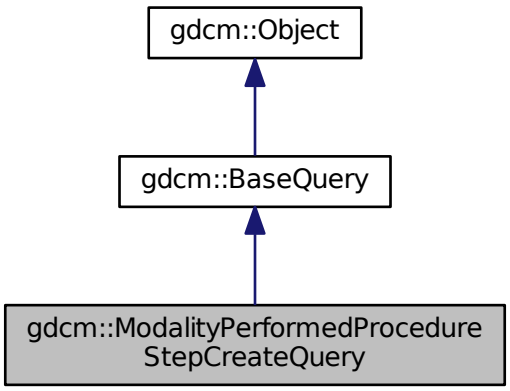
- gdcmMeshPrimitive.h

10.184 gdcm::ModalityPerformedProcedureStepCreateQuery Class Reference

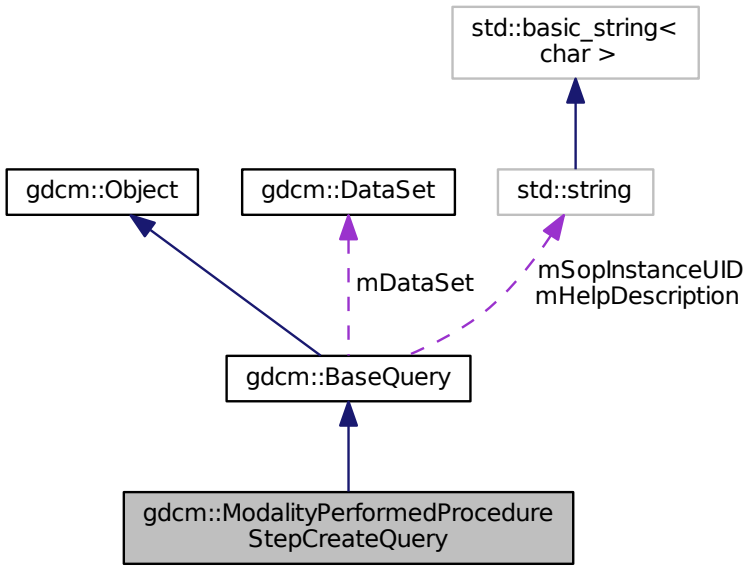
ModalityPerformedProcedureStepCreateQuery contains: the class which will produce a dataset for n-create for Modality Performed Procedure Step sop class.

```
#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 gdcm::ModalityPerformedProcedureStepCreateQuery::ModalityPerformedProcedureStepCreateQuery (const std::string & *iSopInstanceUID*)

10.184.3 Member Function Documentation

10.184.3.1 UIDs::TSName gdcm::ModalityPerformedProcedureStepCreateQuery::GetAbstractSyntaxUID ( ) const [virtual]

Implements gdcm::BaseQuery.

10.184.3.2 gdcm::DataSet gdcm::ModalityPerformedProcedureStepCreateQuery::GetRequiredDataSet ( ) const

10.184.3.3 bool gdcm::ModalityPerformedProcedureStepCreateQuery::ValidateQuery (bool *inStrict* = `true`) const [virtual]

Implements gdcm::BaseQuery.

10.184.4 Friends And Related Function Documentation

10.184.4.1 friend class QueryFactory [friend]

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

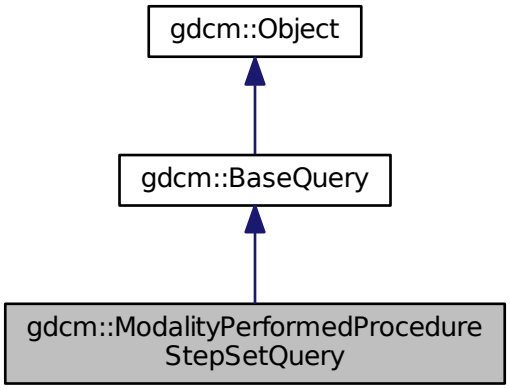
- gdcmModalityPerformedProcedureStepCreateQuery.h

10.185 gdcm::ModalityPerformedProcedureStepSetQuery Class Reference

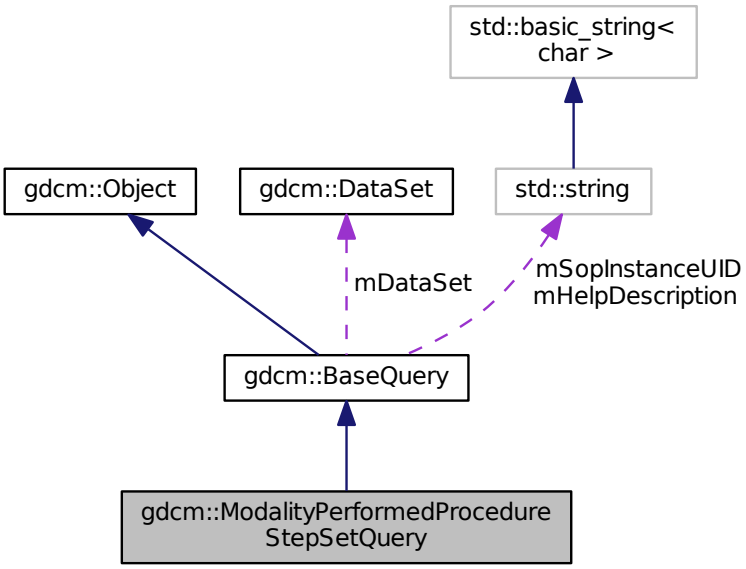
ModalityPerformedProcedureStepSetQuery contains: the class which will produce a dataset for n-set for Modality Performed Procedure Step sop class.

```
#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 gdcm::ModalityPerformedProcedureStepSetQuery::ModalityPerformedProcedureStepSetQuery (const std::string & *iSopInstanceUID*)

10.185.3 Member Function Documentation

10.185.3.1 UIDs::TSName gdcm::ModalityPerformedProcedureStepSetQuery::GetAbstractSyntaxUID ( ) const [virtual]

Implements gdcm::BaseQuery.

10.185.3.2 gdcm::DataSet gdcm::ModalityPerformedProcedureStepSetQuery::GetRequiredDataSet ( ) const

10.185.3.3 bool gdcm::ModalityPerformedProcedureStepSetQuery::ValidateQuery (bool *inStrict* = `true`) const [virtual]

Implements gdcm::BaseQuery.

10.185.4 Friends And Related Function Documentation

10.185.4.1 friend class QueryFactory [friend]

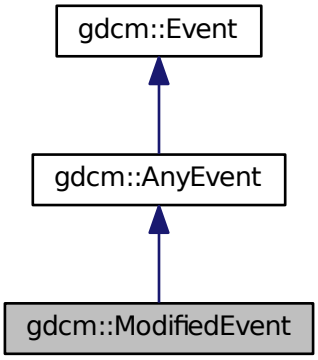
The documentation for this class was generated from the following file:

- gdcmModalityPerformedProcedureStepSetQuery.h

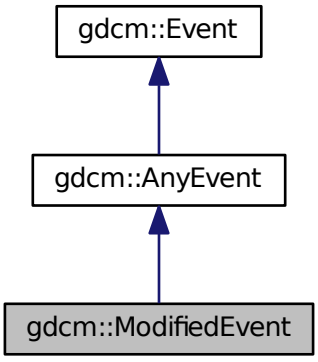
10.186 gdcm::ModifiedEvent Class Reference

`#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 typedef std::vector<std::string> gdcm::Module::ArrayIncludeMacrosType

10.187.2.2 typedef std::map<Tag, ModuleEntry> gdcm::Module::MapModuleEntry

10.187.3 Constructor & Destructor Documentation

10.187.3.1 gdcm::Module::Module ( ) [inline]

References gdcm::operator<<().

10.187.4 Member Function Documentation

10.187.4.1 void gdcm::Module::AddMacro (const char * *include*) [inline]

10.187.4.2 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 void gdcm::Module::Clear ( ) [inline]

10.187.4.4 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 const ModuleEntry& gdcm::Module::GetModuleEntryInMacros (Macros const & *macros,* const Tag & *tag*) const

Examples:

TraverseModules.cxx.

10.187.4.6 const char* gdcm::Module::GetName ( ) const [inline]

10.187.4.7 void gdcm::Module::SetName (const char * *name*) [inline]

10.187.4.8 bool gdcm::Module::Verify (const DataSet & *ds,* Usage const & *usage*) const

10.187.5 Friends And Related Function Documentation

10.187.5.1 std::ostream& operator<< (std::ostream & *_os,* const Module & *_val*) [friend]

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

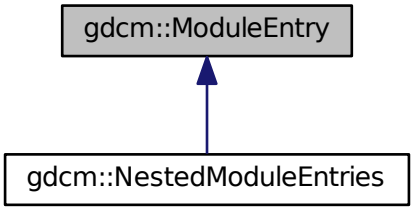
- gdcmModule.h

10.188 gdcm::ModuleEntry Class Reference

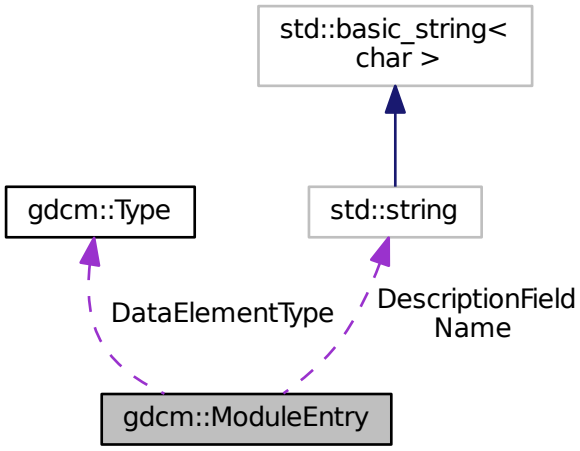
Class for representing a ModuleEntry.

`#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 typedef std::string gdcm::ModuleEntry::Description

10.188.3 Constructor & Destructor Documentation

10.188.3.1 gdcm::ModuleEntry::ModuleEntry (const char * *name* = " ", const char * *type* = "3", const char * *description* = " ") [inline]

References gdcm::Type::GetTypeType().

10.188.3.2 virtual gdcm::ModuleEntry::~ModuleEntry ( ) [inline], [virtual]

References gdcm::operator<<().

10.188.4 Member Function Documentation

10.188.4.1 const Description& gdcm::ModuleEntry::GetDescription ( ) const [inline]

10.188.4.2 const char* gdcm::ModuleEntry::GetName ( ) const [inline]

10.188.4.3 const Type& gdcm::ModuleEntry::GetType ( ) const [inline]

Examples:

TraverseModules.cxx.

10.188.4.4 void gdcm::ModuleEntry::SetDescription (const char * *d*) [inline]

10.188.4.5 void gdcm::ModuleEntry::SetName (const char * *name*) [inline]

10.188.4.6 void gdcm::ModuleEntry::SetType (const Type & *type*) [inline]

10.188.5 Friends And Related Function Documentation

10.188.5.1 std::ostream& operator<< (std::ostream & *_os,* const ModuleEntry & *_val*) [friend]

10.188.6 Member Data Documentation

10.188.6.1 Type gdcm::ModuleEntry::DataElementType [protected]

Referenced by gdcm::operator<<().

10.188.6.2 Description gdcm::ModuleEntry::DescriptionField [protected]

Referenced by gdcm::operator<<().

10.188.6.3 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 typedef std::map<std::string, Module> gdcm::Modules::ModuleMapType

10.189.3 Constructor & Destructor Documentation

10.189.3.1 gdcm::Modules::Modules ( ) [inline]

References gdcm::operator<<().

10.189.4 Member Function Documentation

10.189.4.1 void gdcm::Modules::AddModule (const char * *ref*, const Module & *module*) [inline]

10.189.4.2 void gdcm::Modules::Clear ( ) [inline]

10.189.4.3 const Module& gdcm::Modules::GetModule (const char * *name*) const [inline]

Examples:

TraverseModules.cxx.

10.189.4.4 bool gdcm::Modules::IsEmpty ( ) const [inline]

10.189.5 Friends And Related Function Documentation

10.189.5.1 std::ostream& operator<< (std::ostream & *_os,* const Modules & *_val*) [friend]

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

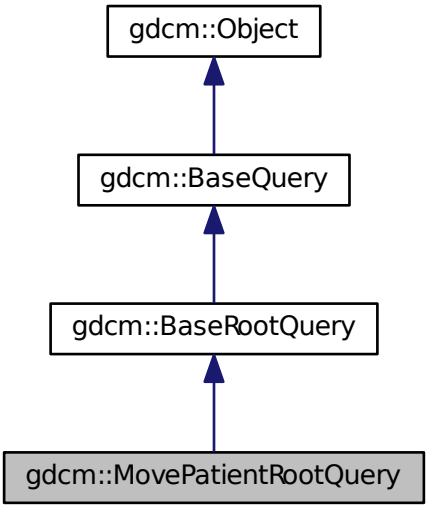
- gdcmModules.h

10.190 gdcm::MovePatientRootQuery Class Reference

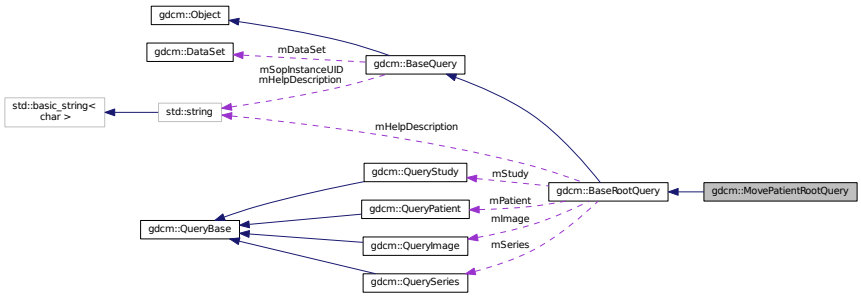
MovePatientRootQuery contains: the class which will produce a dataset for c-move with patient root.

```
#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 gdcm::MovePatientRootQuery::MovePatientRootQuery ( )

10.190.3 Member Function Documentation

10.190.3.1 UIDs::TSName gdcm::MovePatientRootQuery::GetAbstractSyntaxUID ( ) const [virtual]

Implements gdcm::BaseQuery.

10.190.3.2 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 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 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 friend class QueryFactory [friend]

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

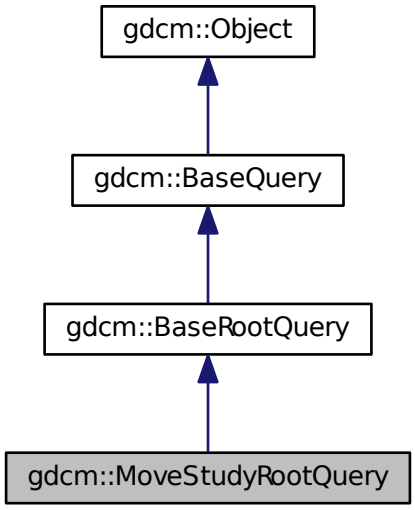
- gdcmMovePatientRootQuery.h

10.191 gdcm::MoveStudyRootQuery Class Reference

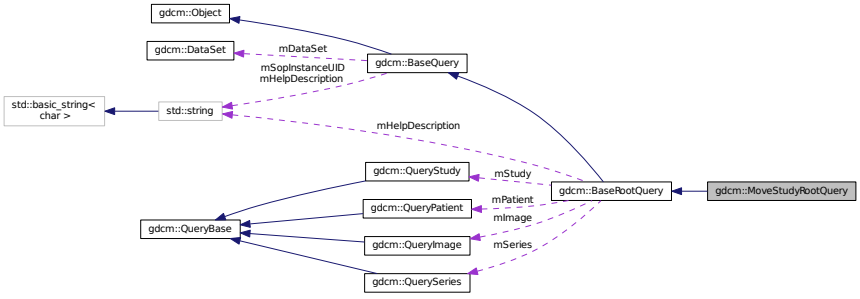
MoveStudyRootQuery contains: the class which will produce a dataset for C-MOVE with study root.

`#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 gdcm::MoveStudyRootQuery::MoveStudyRootQuery ( )

10.191.3 Member Function Documentation

10.191.3.1 UIDs::TSName gdcm::MoveStudyRootQuery::GetAbstractSyntaxUID ( ) const [virtual]

Implements gdcm::BaseQuery.

10.191.3.2 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 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 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 friend class QueryFactory [friend]

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

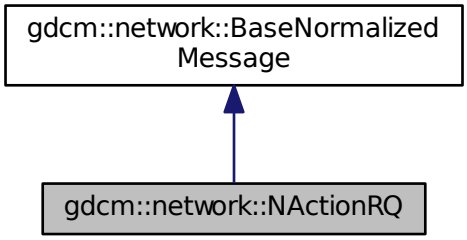
- gdcmMoveStudyRootQuery.h

10.192 gdcm::network::NActionRQ Class Reference

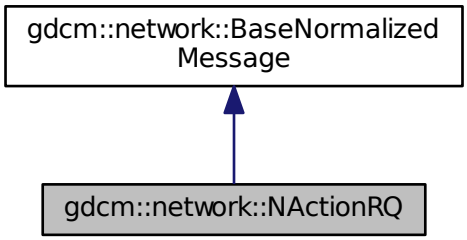
NActionRQ this file defines the messages for the NAction action.

`#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 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:

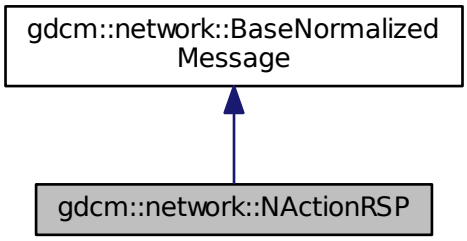
- gdcmNActionMessages.h

10.193 gdcm::network::NActionRSP Class Reference

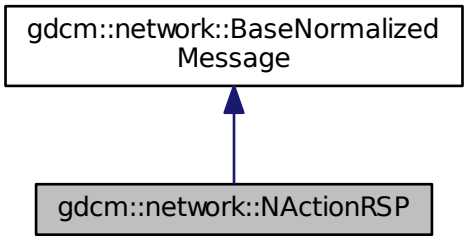
NActionRSP this file defines the messages for the NAction action.

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

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

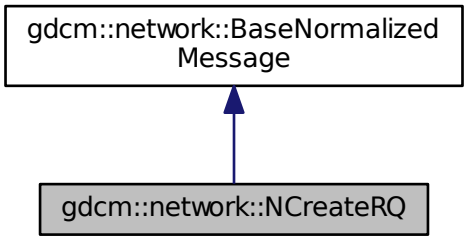
- gdcmNActionMessages.h

10.194 gdcm::network::NCreateRQ Class Reference

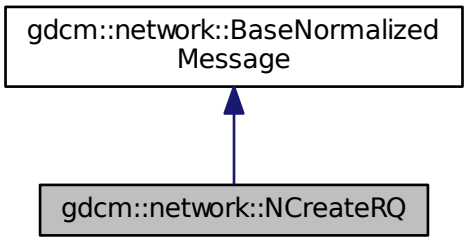
NCreateRQ this file defines the messages for the ncreate action.

```
#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 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:

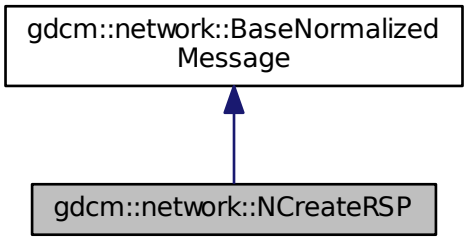
- gdcmNCreateMessages.h

10.195 gdcm::network::NCreateRSP Class Reference

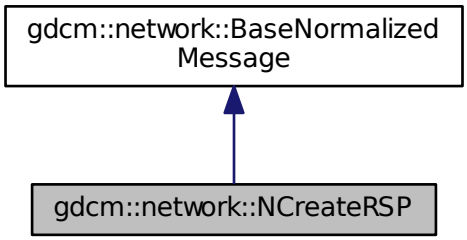
NCreateRSP this file defines the messages for the ncreate action.

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

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

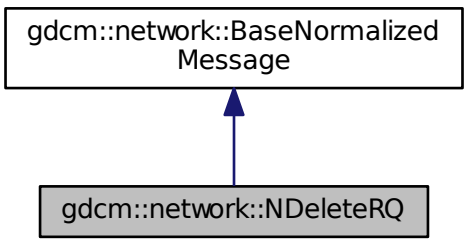
- gdcmNCreateMessages.h

10.196 gdcm::network::NDeleteRQ Class Reference

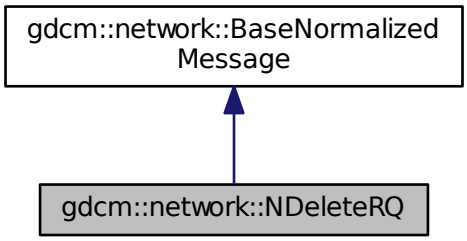
NDeleteRQ this file defines the messages for the ndelete action.

```
#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 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:

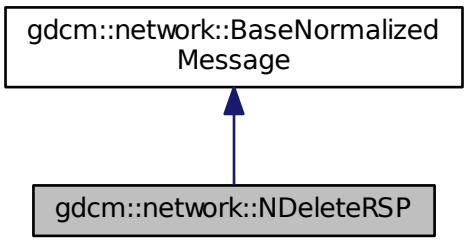
- gdcmNDeleteMessages.h

10.197 gdcm::network::NDeleteRSP Class Reference

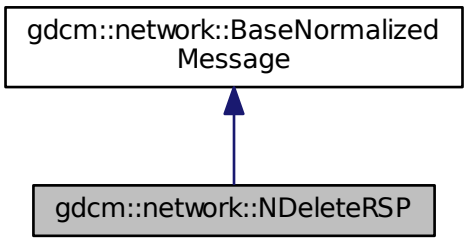
NDeleteRSP this file defines the messages for the ndelete action.

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

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

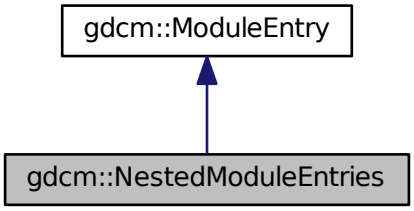
- gdcmNDeleteMessages.h

10.198 gdcm::NestedModuleEntries Class Reference

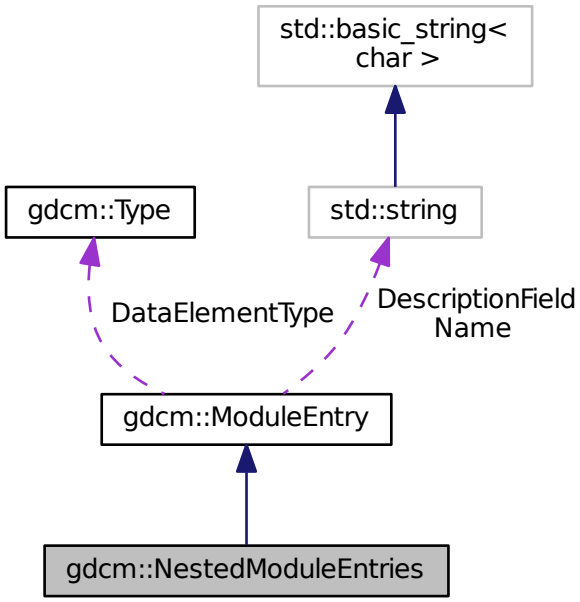
Class for representing a NestedModuleEntries.

```
#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 typedef std::vector<ModuleEntry>::size_type gdcm::NestedModuleEntries::SizeType

10.198.3 Constructor & Destructor Documentation

10.198.3.1 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 void gdcm::NestedModuleEntries::AddModuleEntry (const ModuleEntry & *me*) [inline]

10.198.4.2 const ModuleEntry& gdcm::NestedModuleEntries::GetModuleEntry (SizeType *idx*) const [inline]

10.198.4.3 ModuleEntry& gdcm::NestedModuleEntries::GetModuleEntry (SizeType *idx*) [inline]

10.198.4.4 SizeType gdcm::NestedModuleEntries::GetNumberOfModuleEntries ( ) [inline]

10.198.5 Friends And Related Function Documentation

10.198.5.1 std::ostream& operator<< (std::ostream & *_os*, const NestedModuleEntries & *_val*) [friend]

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

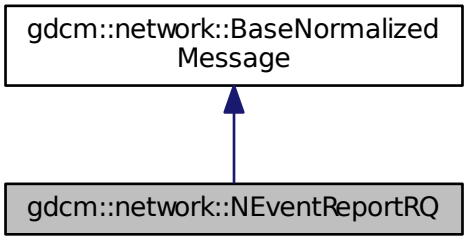
- gdcmNestedModuleEntries.h

10.199 gdcm::network::NEventReportRQ Class Reference

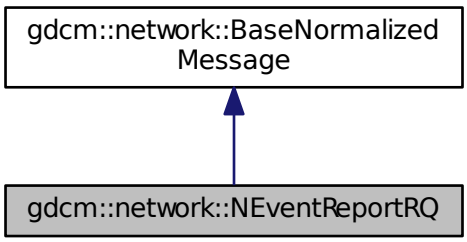
NEventReportRQ this file defines the messages for the neventreport action.

`#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 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:

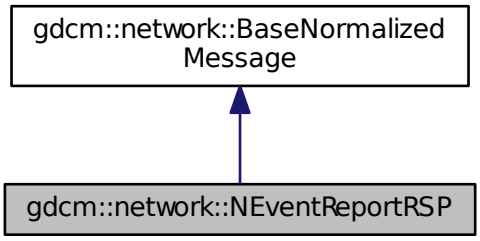
- gdcmNEventReportMessages.h

10.200 gdcm::network::NEventReportRSP Class Reference

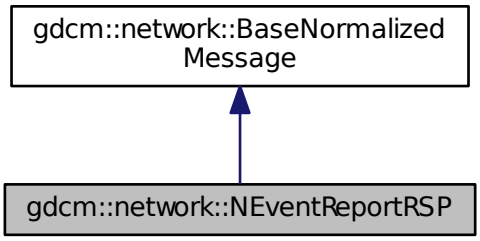
NEventReportRSP this file defines the messages for the neventreport action.

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

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

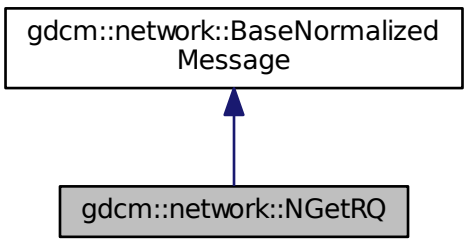
- gdcmNEventReportMessages.h

10.201 gdcm::network::NGetRQ Class Reference

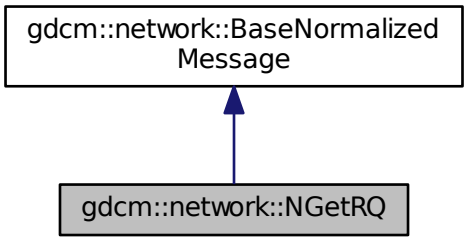
NGetRQ this file defines the messages for the nget action.

```
#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 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:

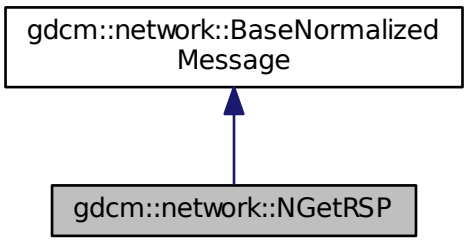
- gdcmNGetMessages.h

10.202 gdcm::network::NGetRSP Class Reference

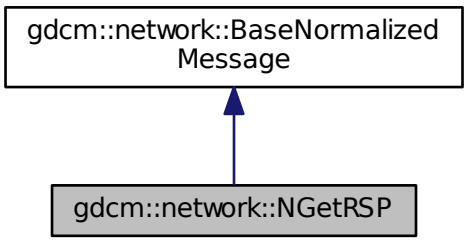
NGetRSP this file defines the messages for the nget action.

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

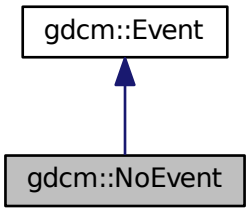
The documentation for this class was generated from the following file:

- gdcmNGetMessages.h

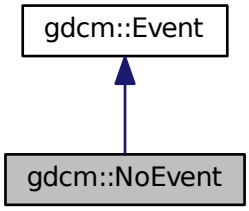
10.203 gdcm::NoEvent Class Reference

`#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 **static std::vector<PresentationDataValue> gdcm::network::NormalizedMessageFactory::ConstructNAction (const ULConnection & *inConnection,* const BaseQuery * *inQuery*) [static]**

10.204.1.2 **static std::vector<PresentationDataValue> gdcm::network::NormalizedMessageFactory::ConstructNCreate (const ULConnection & *inConnection,* const BaseQuery * *inQuery*) [static]**

10.204.1.3 **static std::vector<PresentationDataValue> gdcm::network::NormalizedMessageFactory::ConstructNDelete (const ULConnection & *inConnection,* const BaseQuery * *inQuery*) [static]**

10.204.1.4 **static std::vector<PresentationDataValue> gdcm::network::NormalizedMessageFactory::ConstructNEventReport (const ULConnection & *inConnection,* const BaseQuery * *inQuery*) [static]**

10.204.1.5 **static std::vector<PresentationDataValue> gdcm::network::NormalizedMessageFactory::ConstructNGet (const ULConnection & *inConnection,* const BaseQuery * *inQuery*) [static]**

10.204.1.6 **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 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:

`#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 static **BaseQuery** * gdcm::NormalizedNetworkFunctions::ConstructQuery (const std::string & *sopInstanceUID,* const **DataSet** & *queryds,* **ENQueryType** *queryType* = **eCreateMMPS**) [static]

10.205.2.2 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 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 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 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 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 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:

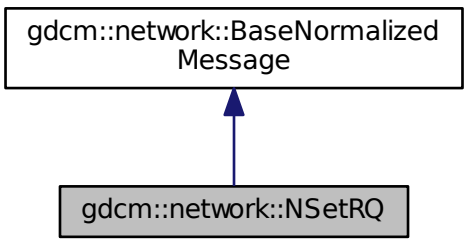
- gdcmNormalizedNetworkFunctions.h

10.206 gdcm::network::NSetRQ Class Reference

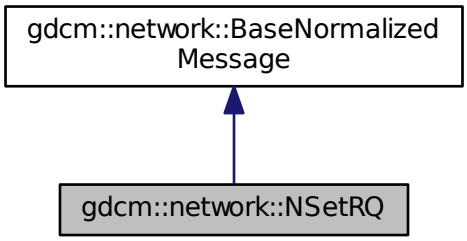
NSetRQ this file defines the messages for the nset action.

```
#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 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:

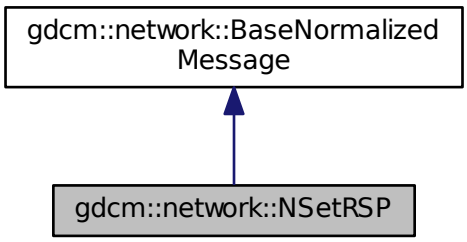
- gdcmNSetMessages.h

10.207 gdcm::network::NSetRSP Class Reference

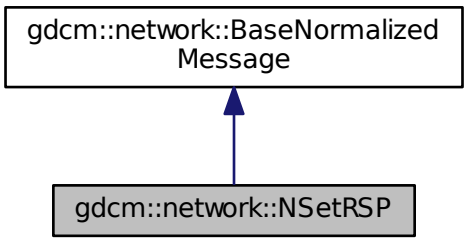
NSetRSP this file defines the messages for the nset action.

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

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

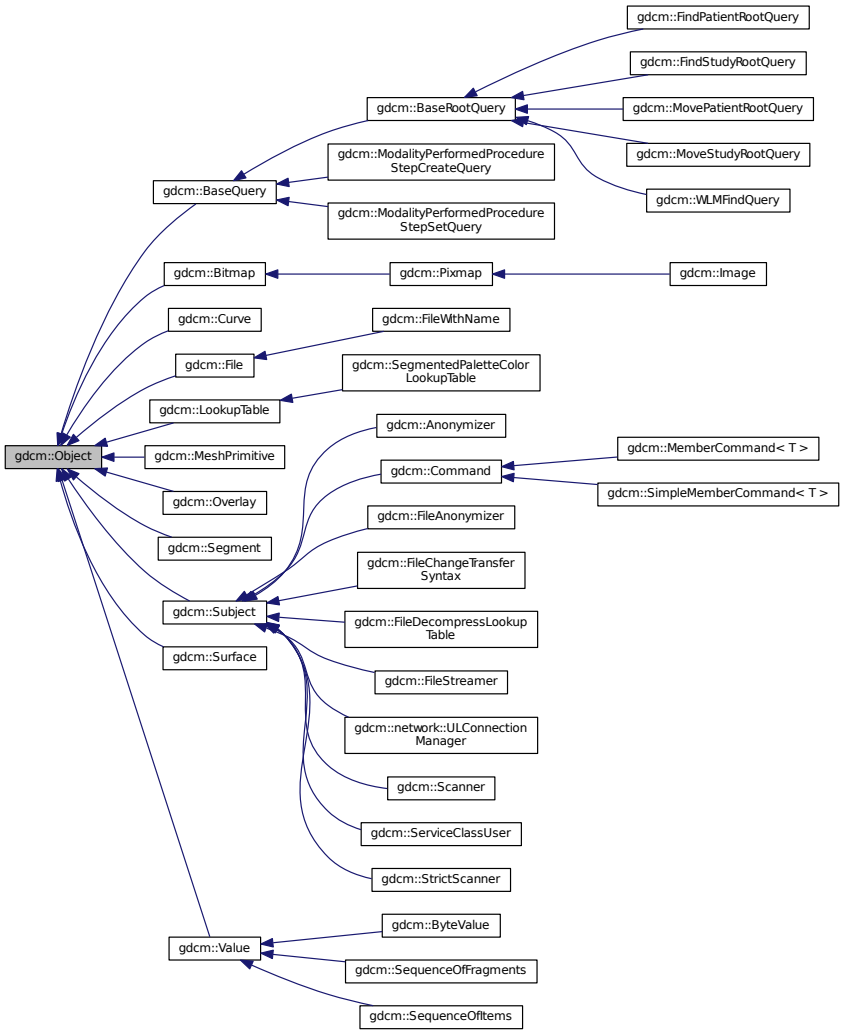
- gdcmNSetMessages.h

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 gdcm::Object::Object ( ) [inline]

10.208.2.2 virtual gdcm::Object::~Object ( ) [inline], [virtual]

10.208.2.3 gdcm::Object::Object (const Object &) [inline]

Special requirement for copy/cstor, assignment operator.

10.208.3 Member Function Documentation

10.208.3.1 void gdcm::Object::operator= (const Object &) [inline]

10.208.3.2 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 void gdcm::Object::Register ( ) [inline], [protected]

10.208.3.4 void gdcm::Object::UnRegister ( ) [inline], [protected]

10.208.4 Friends And Related Function Documentation

10.208.4.1 std::ostream& operator<< (std::ostream & *os,* const Object & *obj*) [friend]

10.208.4.2 template<class ObjectType > friend class SmartPointer [friend]

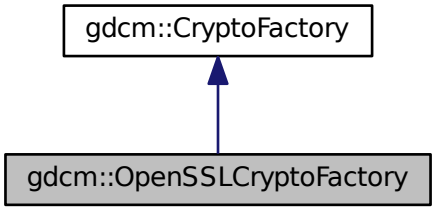
The documentation for this class was generated from the following file:

- gdcmObject.h

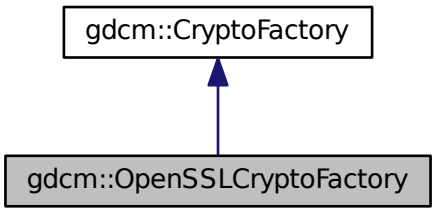
10.209 gdcm::OpenSSLCryptoFactory Class Reference

`#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 gdcm::OpenSSLCryptoFactory::OpenSSLCryptoFactory (CryptoLib *id*) [inline]

References gdcmDebugMacro.

10.209.2 Member Function Documentation

10.209.2.1 CryptographicMessageSyntax* gdcm::OpenSSLCryptoFactory::CreateCMSProvider ( ) [inline], [virtual]

Implements gdcm::CryptoFactory.

10.209.2.2 void gdcm::OpenSSLCryptoFactory::InitOpenSSL ( ) [protected]

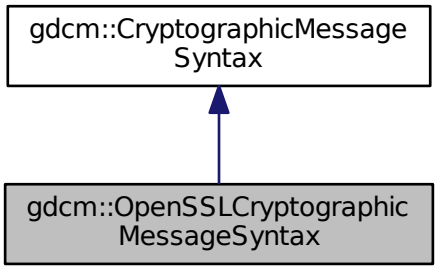
The documentation for this class was generated from the following file:

- gdcmOpenSSLCryptoFactory.h

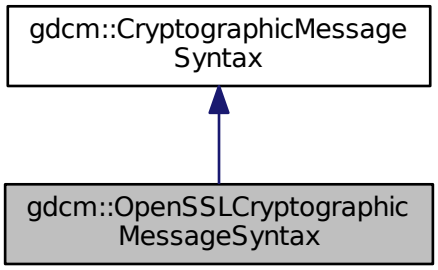
10.210 gdcm::OpenSSLCryptographicMessageSyntax Class Reference

`#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 gdcm::OpenSSLCryptographicMessageSyntax::OpenSSLCryptographicMessageSyntax ( )

10.210.1.2 gdcm::OpenSSLCryptographicMessageSyntax::~OpenSSLCryptographicMessageSyntax ( )

10.210.2 Member Function Documentation

10.210.2.1 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 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 CipherTypes gdcm::OpenSSLCryptographicMessageSyntax::GetCipherType ( ) const [virtual]

Implements gdcm::CryptographicMessageSyntax.

10.210.2.4 bool gdcm::OpenSSLCryptographicMessageSyntax::ParseCertificateFile (const char * *filename*) [virtual]

Implements gdcm::CryptographicMessageSyntax.

10.210.2.5 bool gdcm::OpenSSLCryptographicMessageSyntax::ParseKeyFile (const char * *filename*) [virtual]

Implements gdcm::CryptographicMessageSyntax.

10.210.2.6 void gdcm::OpenSSLCryptographicMessageSyntax::SetCipherType (CipherTypes *type*) [virtual]

Set Cipher Type. Default is: AES256_CIPHER

Implements gdcm::CryptographicMessageSyntax.

10.210.2.7 bool gdcm::OpenSSLCryptographicMessageSyntax::SetPassword (const char * *pass*, size_t *passLen*) [virtual]

Implements gdcm::CryptographicMessageSyntax.

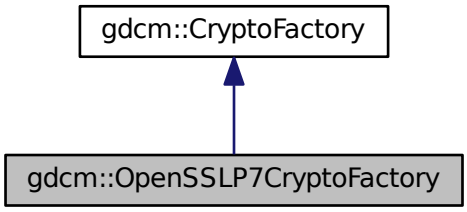
The documentation for this class was generated from the following file:

- gdcmOpenSSLCryptographicMessageSyntax.h

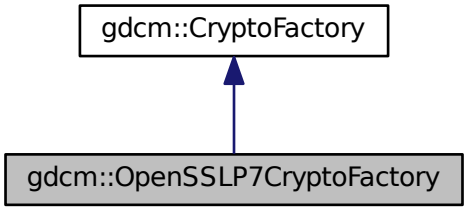
10.211 gdcm::OpenSSLP7CryptoFactory Class Reference

`#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 gdcm::OpenSSLP7CryptoFactory::OpenSSLP7CryptoFactory (CryptoLib *id*) [inline]

References gdcmDebugMacro.

10.211.2 Member Function Documentation

10.211.2.1 CryptographicMessageSyntax* gdcm::OpenSSLP7CryptoFactory::CreateCMSProvider ( ) [inline], [virtual]

Implements gdcm::CryptoFactory.

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

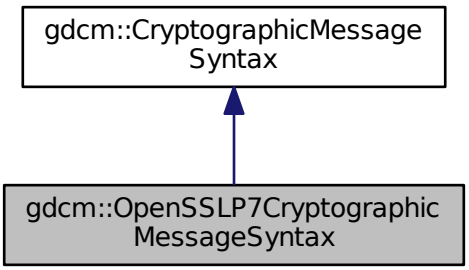
- gdcmOpenSSLP7CryptoFactory.h

10.212 gdcm::OpenSSLP7CryptographicMessageSyntax Class Reference

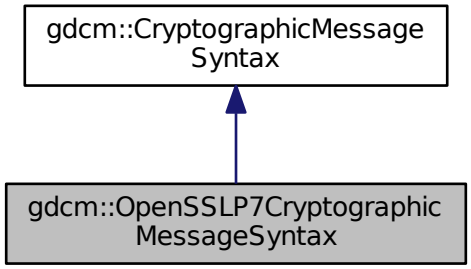
Class for CryptographicMessageSyntax encryption. This is just a simple wrapper around openssl PKCS7_encrypt functionalities.

`#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 gdcm::OpenSSLP7CryptographicMessageSyntax::OpenSSLP7CryptographicMessageSyntax ( )

10.212.2.2 gdcm::OpenSSLP7CryptographicMessageSyntax::~OpenSSLP7CryptographicMessageSyntax ( )

10.212.3 Member Function Documentation

10.212.3.1 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 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 CipherTypes gdcm::OpenSSLP7CryptographicMessageSyntax::GetCipherType ( ) const [virtual]

Implements gdcm::CryptographicMessageSyntax.

10.212.3.4 bool gdcm::OpenSSLP7CryptographicMessageSyntax::ParseCertificateFile (const char * *filename*) [virtual]

Implements gdcm::CryptographicMessageSyntax.

10.212.3.5 bool gdcm::OpenSSLP7CryptographicMessageSyntax::ParseKeyFile (const char * *filename*) [virtual]

Implements gdcm::CryptographicMessageSyntax.

10.212.3.6 void gdcm::OpenSSLP7CryptographicMessageSyntax::SetCipherType (CipherTypes *type*) [virtual]

Set Cipher Type. Default is: AES256_CIPHER

Implements gdcm::CryptographicMessageSyntax.

10.212.3.7 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 enum gdcm::Orientation::OrientationType

Enumerator

UNKNOWN
AXIAL
CORONAL
SAGITTAL
OBLIQUE

10.213.3 Constructor & Destructor Documentation

10.213.3.1 gdcm::Orientation::Orientation ( )

10.213.3.2 gdcm::Orientation::~Orientation ( )

10.213.4 Member Function Documentation

10.213.4.1 static const char* gdcm::Orientation::GetLabel (OrientationType *type*) [static]

Return the label of an Orientation.

Examples:

FixOrientation.cxx.

10.213.4.2 static char gdcm::Orientation::GetMajorAxisFromPatientRelativeDirectionCosine (double *x*, double *y*, double *z*) [static], [protected]

10.213.4.3 static double gdcm::Orientation::GetObliquityThresholdCosineValue ( ) [static]

10.213.4.4 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 void gdcm::Orientation::Print (std::ostream &) const

Print.

Referenced by gdcm::operator<<().

10.213.4.6 static void gdcm::Orientation::SetObliquityThresholdCosineValue (double *val*) [static]

ObliquityThresholdCosineValue stuff.

10.213.5 Friends And Related Function Documentation

10.213.5.1 std::ostream& operator<< (std::ostream & *_os,* const Orientation & *o*) [friend]

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

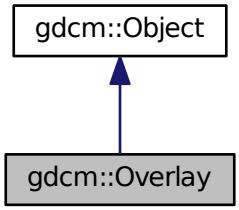
- gdcmOrientation.h

10.214 gdcm::Overlay Class Reference

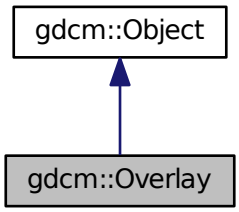
Overlay class.

`#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 enum gdcm::Overlay::OverlayType

Enumerator

Invalid

Graphics

ROI

10.214.3 Constructor & Destructor Documentation

10.214.3.1 gdcm::Overlay::Overlay ( )

10.214.3.2 gdcm::Overlay::~Overlay ( )

10.214.3.3 gdcm::Overlay::Overlay (Overlay const & *ov*)

10.214.4 Member Function Documentation

10.214.4.1 void gdcm::Overlay::Decompress (std::ostream & *os*) const

Decode the internal OverlayData (packed bits) into unpacked representation.

10.214.4.2 unsigned short gdcm::Overlay::GetBitPosition ( ) const

return bit position

10.214.4.3 unsigned short gdcm::Overlay::GetBitsAllocated ( ) const

return bits allocated

10.214.4.4 unsigned short gdcm::Overlay::GetColumns ( ) const

get columns

10.214.4.5 const char* gdcm::Overlay::GetDescription ( ) const

get description

10.214.4.6 unsigned short gdcm::Overlay::GetGroup ( ) const

Get Group number.

10.214.4.7 const signed short* gdcm::Overlay::GetOrigin ( ) const

get origin

10.214.4.8 const ByteValue& gdcm::Overlay::GetOverlayData ( ) const

Return the Overlay Data as ByteValue: Not thread safe

10.214.4.9 static const char* gdcm::Overlay::GetOverlayTypeAsString (OverlayType *ot*) [static]

10.214.4.10 static OverlayType gdcm::Overlay::GetOverlayTypeFromString (const char *) [static]

10.214.4.11 unsigned short gdcm::Overlay::GetRows ( ) const

get rows

10.214.4.12 const char* gdcm::Overlay::GetType ( ) const

get type

10.214.4.13 OverlayType gdcm::Overlay::GetTypeAsEnum ( ) const

10.214.4.14 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 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 bool gdcm::Overlay::GrabOverlayFromPixelData (DataSet const & *ds*)

10.214.4.17 bool gdcm::Overlay::IsEmpty ( ) const

Return whether or not the Overlay is empty:

10.214.4.18 bool gdcm::Overlay::IsInPixelData ( ) const

return if the Overlay is stored in the pixel data or not

10.214.4.19 void gdcm::Overlay::IsInPixelData (bool *b*)

Set whether or no the OverlayData is in the Pixel Data:

10.214.4.20 bool gdcm::Overlay::IsZero ( ) const

return true if all bits are set to 0

10.214.4.21 Overlay& gdcm::Overlay::operator= (Overlay const & *ov*)

10.214.4.22 void gdcm::Overlay::Print (std::ostream &) const [virtual]

Print.

Reimplemented from gdcm::Object.

10.214.4.23 void gdcm::Overlay::SetBitPosition (unsigned short *bitposition*)

set bit position

10.214.4.24 void gdcm::Overlay::SetBitsAllocated (unsigned short *bitsallocated*)

set bits allocated

10.214.4.25 void gdcm::Overlay::SetColumns (unsigned short *columns*)

set columns

10.214.4.26 void gdcm::Overlay::SetDescription (const char * *description*)

set description

10.214.4.27 void gdcm::Overlay::SetFrameOrigin (unsigned short *frameorigin*)

set frame origin

10.214.4.28 void gdcm::Overlay::SetGroup (unsigned short *group*)

Set Group number.

10.214.4.29 void gdcm::Overlay::SetNumberOfFrames (unsigned int *numberofframes*)

set number of frames

10.214.4.30 void gdcm::Overlay::SetOrigin (const signed short *origin[2]*)

set origin

10.214.4.31 void gdcm::Overlay::SetOverlay (const char * *array,* size_t *length*)

set overlay from byte array + length

10.214.4.32 void gdcm::Overlay::SetRows (unsigned short *rows*)

set rows

10.214.4.33 void gdcm::Overlay::SetType (const char * *type*)

set type

10.214.4.34 void gdcm::Overlay::Update (const DataElement & *de*)

Update overlay from data element de:

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

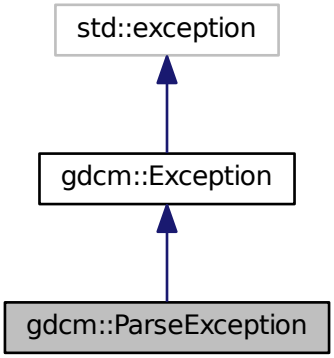
- gdcmOverlay.h

10.215 gdcm::ParseException Class Reference

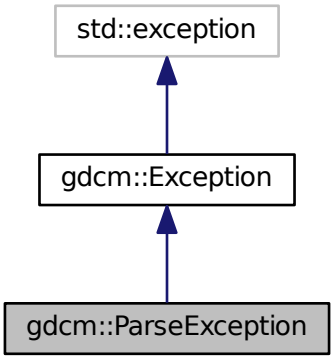
ParseException Standard exception handling object.

```
#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 gdcm::ParseException::ParseException ( ) [inline]

10.215.2.2 virtual gdcm::ParseException::~ParseException ( ) throw) [inline], [virtual]

10.215.3 Member Function Documentation

10.215.3.1 const DataElement& gdcm::ParseException::GetLastElement ( ) const [inline]

10.215.3.2 ParseException& gdcm::ParseException::operator= (const ParseException & *orig*) [inline]

Assignment operator.

10.215.3.3 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 typedef void(* gdcm::Parser::EndElementHandler) (void *userData, const Tag &name)

10.216.2.2 typedef void(* gdcm::Parser::StartElementHandler) (void *userData, const Tag &tag, const char *atts[ ])

10.216.3 Member Enumeration Documentation

10.216.3.1 enum gdcm::Parser::ErrorType

Enumerator

NoError

NoMemoryError

SyntaxError

NoElementsError

TagMismatchError

DuplicateAttributeError

JunkAfterDocElementError

UndefinedEntityError

UnexpectedStateError

10.216.4 Constructor & Destructor Documentation

10.216.4.1 gdcm::Parser::Parser ( ) [inline]

10.216.4.2 gdcm::Parser::~Parser ( ) [inline]

10.216.5 Member Function Documentation

10.216.5.1 char* gdcm::Parser::GetBuffer (int *len*) [protected]

10.216.5.2 unsigned long gdcm::Parser::GetCurrentByteIndex ( ) const

10.216.5.3 ErrorType gdcm::Parser::GetErrorCode ( ) const

10.216.5.4 static const char* gdcm::Parser::GetErrorString (ErrorType const & *err*) [static]

10.216.5.5 void* gdcm::Parser::GetUserData ( ) const

10.216.5.6 bool gdcm::Parser::Parse (const char * *s,* int *len,* bool *isFinal*)

10.216.5.7 bool gdcm::Parser::ParseBuffer (int *len,* bool *isFinal*) [protected]

10.216.5.8 ErrorType gdcm::Parser::Process ( ) [protected]

10.216.5.9 void gdcm::Parser::SetElementHandler (StartElementHandler *start,* EndElementHandler *end*)

10.216.5.10 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 gdcm::Patient::Patient ( ) [inline]

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

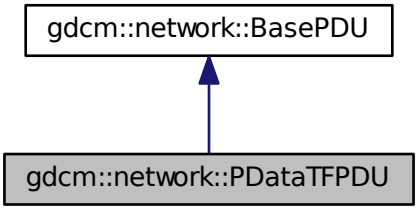
- gdcmPatient.h

10.218 gdcm::network::PDataTFPDU Class Reference

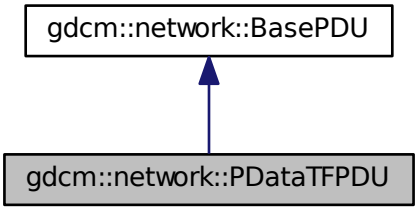
PDataTFPDU Table 9-22 P-DATA-TF PDU FIELDS.

`#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 typedef std::vector<PresentationDataValue>::size_type gdcm::network::PDataTFPDU::SizeType

10.218.3 Constructor & Destructor Documentation

10.218.3.1 gdcm::network::PDataTFPDU::PDataTFPDU ( )

10.218.4 Member Function Documentation

10.218.4.1 void gdcm::network::PDataTFPDU::AddPresentationDataValue (PresentationDataValue const & *pdv*) [inline]

10.218.4.2 SizeType gdcm::network::PDataTFPDU::GetNumberOfPresentationDataValues ( ) const [inline]

10.218.4.3 PresentationDataValue const& gdcm::network::PDataTFPDU::GetPresentationDataValue (SizeType *i*) const [inline]

10.218.4.4 bool gdcm::network::PDataTFPDU::IsLastFragment ( ) const [virtual]

Implements gdcm::network::BasePDU.

10.218.4.5 void gdcm::network::PDataTFPDU::Print (std::ostream & *os*) const [virtual]

Implements gdcm::network::BasePDU.

10.218.4.6 std::istream& gdcm::network::PDataTFPDU::Read (std::istream & *is*) [virtual]

Implements gdcm::network::BasePDU.

10.218.4.7 std::istream& gdcm::network::PDataTFPDU::ReadInto (std::istream & *is*, std::ostream & *os*) [protected]

10.218.4.8 size_t gdcm::network::PDataTFPDU::Size ( ) const [virtual]

Implements gdcm::network::BasePDU.

10.218.4.9 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:

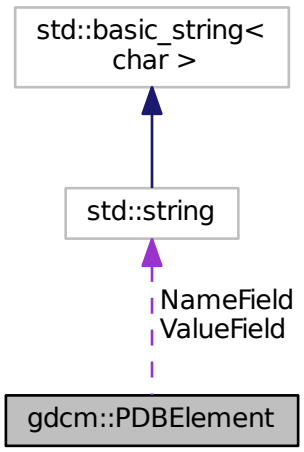
- gdcmPDataTFPDU.h

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 gdcm::PDBElement::PDBElement ( ) [inline]

References gdcm::operator<<().

10.219.3 Member Function Documentation

10.219.3.1 const char* gdcm::PDBElement::GetName ( ) const [inline]

Set/Get Name.

10.219.3.2 const char* gdcm::PDBElement::GetValue ( ) const [inline]

Set/Get Value.

10.219.3.3 bool gdcm::PDBElement::operator== (const PDBElement & *de*) const [inline]

References NameField, and ValueField.

10.219.3.4 void gdcm::PDBElement::SetName (const char * *name*) [inline]

10.219.3.5 void gdcm::PDBElement::SetValue (const char * *value*) [inline]

10.219.4 Friends And Related Function Documentation

10.219.4.1 std::ostream& operator<< (std::ostream & *os,* const PDBElement & *val*) [friend]

10.219.5 Member Data Documentation

10.219.5.1 std::string gdcm::PDBElement::NameField [protected]

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

10.219.5.2 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 gdcm::PDBHeader::PDBHeader ( ) [inline]

10.220.2.2 gdcm::PDBHeader::~PDBHeader ( ) [inline]

10.220.3 Member Function Documentation

10.220.3.1 bool gdcm::PDBHeader::FindPDBElementByName (const char * *name*)

Return true if the PDB element matching name is found or not.

10.220.3.2 const PDBElement& gdcm::PDBHeader::GetPDBEEnd ( ) const [protected]

10.220.3.3 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 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 bool gdcm::PDBHeader::LoadFromDataElement (DataElement const & *de*)

Load the PDB Header from a DataElement of a DataSet.

10.220.3.6 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 std::ostream& operator<< (std::ostream & *_os,* const PDBHeader & *d*) [friend]

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

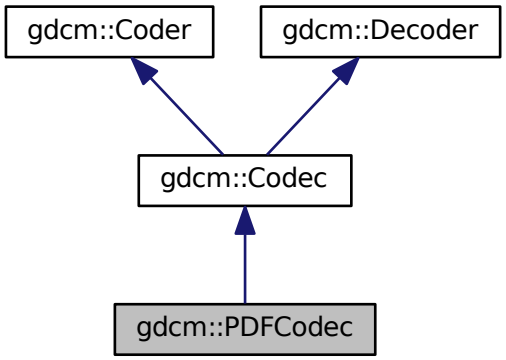
- gdcmPDBHeader.h

10.221 gdcm::PDFCodec Class Reference

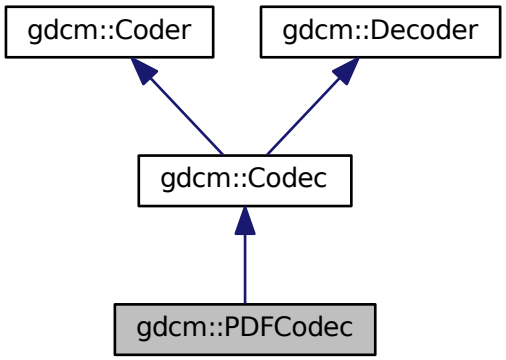
PDFCodec class.

```
#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 gdcm::PDFCodec::PDFCodec ( )

10.221.2.2 gdcm::PDFCodec::~PDFCodec ( )

10.221.3 Member Function Documentation

10.221.3.1 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 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 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 appropriate PDU. This way, the event loop doesn't have to know about all the different PDU types.

`#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 static **BasePDU*** gdcm::network::PDUFactory::ConstructAbortPDU ( ) [static]

10.222.2.2 static **BasePDU*** gdcm::network::PDUFactory::ConstructPDU (uint8_t *itemtype*) [static]

10.222.2.3 static **BasePDU*** gdcm::network::PDUFactory::ConstructReleasePDU ( ) [static]

10.222.2.4 static std::vector<**BasePDU***> gdcm::network::PDUFactory::CreateCEchoPDU (const **ULConnection** & *inConnection*) [static]

10.222.2.5 static std::vector<**BasePDU***> gdcm::network::PDUFactory::CreateCFindPDU (const **ULConnection** & *inConnection,* const **BaseRootQuery** * *inRootQuery*) [static]

10.222.2.6 static std::vector<**BasePDU***> gdcm::network::PDUFactory::CreateCMovePDU (const **ULConnection** & *inConnection,* const **BaseRootQuery** * *inRootQuery*) [static]

10.222.2.7 static std::vector<**BasePDU***> gdcm::network::PDUFactory::CreateCStoreRQPDU (const **ULConnection** & *inConnection,* const **File** & *file,* bool *writeDataSet* = true) [static]

10.222.2.8 static std::vector<**BasePDU***> gdcm::network::PDUFactory::CreateCStoreRSPPDU (const **DataSet** * *inDataSet,* const **BasePDU** * *inPC*) [static]

10.222.2.9 static std::vector<**BasePDU***> gdcm::network::PDUFactory::CreateNActionPDU (const **ULConnection** & *inConnection,* const **BaseQuery** * *inQuery*) [static]

10.222.2.10 static std::vector<**BasePDU***> gdcm::network::PDUFactory::CreateNCreatePDU (const **ULConnection** & *inConnection,* const **BaseQuery** * *inQuery*) [static]

10.222.2.11 static std::vector<**BasePDU***> gdcm::network::PDUFactory::CreateNDeletePDU (const **ULConnection** & *inConnection,* const **BaseQuery** * *inQuery*) [static]

10.222.2.12 static std::vector<**BasePDU***> gdcm::network::PDUFactory::CreateNEventReportPDU (const **ULConnection** & *inConnection,* const **BaseQuery** * *inQuery*) [static]

10.222.2.13 static std::vector<**BasePDU***> gdcm::network::PDUFactory::CreateNGetPDU (const **ULConnection** & *inConnection,* const **BaseQuery** * *inQuery*) [static]

10.222.2.14 static std::vector<**BasePDU***> gdcm::network::PDUFactory::CreateNSetPDU (const **ULConnection** & *inConnection,* const **BaseQuery** * *inQuery*) [static]

10.222.2.15 static **EEventID** gdcm::network::PDUFactory::DetermineEventByPDU (const **BasePDU** * *inPDU*) [static]

10.222.2.16 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 unsigned int gdcm::PersonName::GetMaxLength ( ) const [inline]

10.223.2.2 unsigned int gdcm::PersonName::GetNumberOfComponents ( ) const [inline]

10.223.2.3 void gdcm::PersonName::Print (std::ostream & *os*) const [inline]

10.223.2.4 void gdcm::PersonName::SetBlob (const std::vector< char > & *v*) [inline]

10.223.2.5 void gdcm::PersonName::SetComponents (const char * *comp1* = " ", const char * *comp2* = " ", const char * *comp3* = " ", const char * *comp4* = " ", const char * *comp5* = " ") [inline]

10.223.2.6 void gdcm::PersonName::SetComponents (const char * *components[ ]*) [inline]

10.223.3 Member Data Documentation

10.223.3.1 char gdcm::PersonName::Component[MaxNumberOfComponents][MaxLength+1]

10.223.3.2 const unsigned int gdcm::PersonName::MaxLength = 64 [static]

10.223.3.3 const unsigned int gdcm::PersonName::MaxNumberOfComponents = 5 [static]

10.223.3.4 const char gdcm::PersonName::Padding = '' [static]

10.223.3.5 const char gdcm::PersonName::Separator = '^' [static]

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

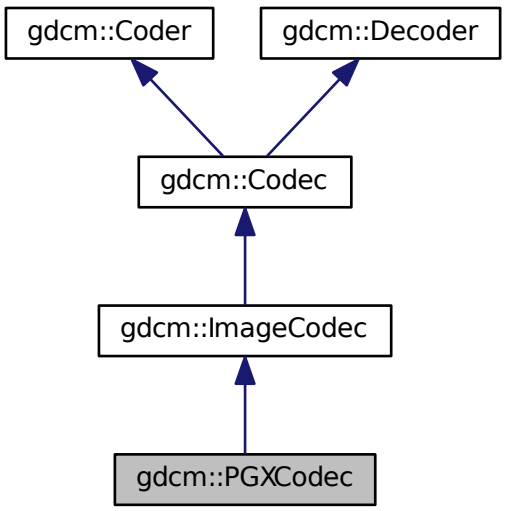
- gdcmPersonName.h

10.224 gdcm::PGXCodec Class Reference

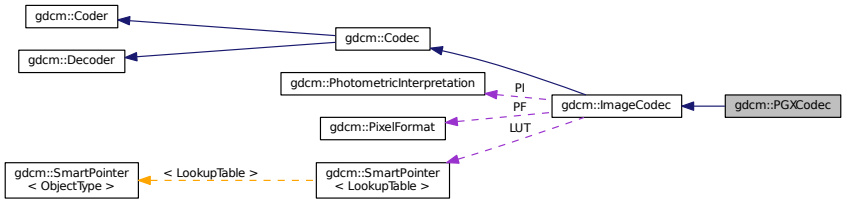
Class to do PGX See PGX as used in JPEG 2000 implementation and reference images.

`#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 gdcm::PGXCodec::PGXCodec ( )

10.224.2.2 gdcm::PGXCodec::~PGXCodec ( )

10.224.3 Member Function Documentation

10.224.3.1 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 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 virtual ImageCodec* gdcm::PGXCodec::Clone ( ) const [virtual]

Implements gdcm::ImageCodec.

10.224.3.4 bool gdcm::PGXCodec::GetHeaderInfo (std::istream & *is*, TransferSyntax & *ts*) [virtual]

Reimplemented from gdcm::ImageCodec.

10.224.3.5 bool gdcm::PGXCodec::Read (const char * *filename*, DataElement & *out*) const

10.224.3.6 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 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 gdcm::PhotometricInterpretation::PhotometricInterpretation (PIType *pi* = UNKNOWN) [inline]

References gdcm::operator<<().

10.225.4 Member Function Documentation

10.225.4.1 static const char* gdcm::PhotometricInterpretation::GetPIString (PIType *pi*) [static]

Referenced by gdcm::operator<<().

10.225.4.2 static PIType gdcm::PhotometricInterpretation::GetPIType (const char * *pi*) [static]

10.225.4.3 unsigned short gdcm::PhotometricInterpretation::GetSamplesPerPixel ( ) const

return the value for Sample Per Pixel associated with a particular Photometric Interpretation

10.225.4.4 const char* gdcm::PhotometricInterpretation::GetString ( ) const

10.225.4.5 PIType gdcm::PhotometricInterpretation::GetType ( ) const [inline]

10.225.4.6 bool gdcm::PhotometricInterpretation::IsLossless ( ) const

10.225.4.7 bool gdcm::PhotometricInterpretation::IsLossy ( ) const

10.225.4.8 static bool gdcm::PhotometricInterpretation::IsRetired (PIType *pi*) [static]

10.225.4.9 bool gdcm::PhotometricInterpretation::IsSameColorSpace (PhotometricInterpretation const & *pi*) const

10.225.4.10 gdcm::PhotometricInterpretation::operator PIType ( ) const [inline]

10.225.5 Friends And Related Function Documentation

10.225.5.1 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.

Note

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 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 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 gdcm::PixelFormat::PixelFormat (ScalarType *st*)

10.226.4 Member Function Documentation

10.226.4.1 unsigned short gdcm::PixelFormat::GetBitsAllocated ( ) const [inline]

BitsAllocated see Tag (0028,0100) US Bits Allocated.

Examples:

GetJPEGSamplePrecision.cxx.

10.226.4.2 unsigned short gdcm::PixelFormat::GetBitsStored ( ) const [inline]

BitsStored see Tag (0028,0101) US Bits Stored.

Examples:

GetJPEGSamplePrecision.cxx.

10.226.4.3 unsigned short gdcm::PixelFormat::GetHighBit ( ) const [inline]

HighBit see Tag (0028,0102) US High Bit.

10.226.4.4 int64_t gdcm::PixelFormat::GetMax ( ) const

return the max possible of the pixel

10.226.4.5 int64_t gdcm::PixelFormat::GetMin ( ) const

return the min possible of the pixel

10.226.4.6 unsigned short gdcm::PixelFormat::GetPixelRepresentation ( ) const [inline]

PixelRepresentation: 0 or 1, see Tag (0028,0103) US Pixel Representation.

10.226.4.7 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 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 ScalarType gdcm::PixelFormat::GetScalarType ( ) const

ScalarType does not take into account the sample per pixel.

10.226.4.10 const char* gdcm::PixelFormat::GetScalarTypeAsString ( ) const

10.226.4.11 bool gdcm::PixelFormat::IsCompatible (const TransferSyntax & *ts*) const

10.226.4.12 bool gdcm::PixelFormat::IsValid ( ) const

return IsValid

10.226.4.13 gdcm::PixelFormat::operator ScalarType ( ) const [inline]

10.226.4.14 bool gdcm::PixelFormat::operator!= (ScalarType *st*) const [inline]

10.226.4.15 bool gdcm::PixelFormat::operator!= (const PixelFormat & *pf*) const [inline]

10.226.4.16 bool gdcm::PixelFormat::operator== (ScalarType *st*) const [inline]

10.226.4.17 bool gdcm::PixelFormat::operator== (const PixelFormat & *pf*) const [inline]

10.226.4.18 void gdcm::PixelFormat::Print (std::ostream & *os*) const

Print.

Referenced by gdcm::operator<<().

10.226.4.19 void gdcm::PixelFormat::SetBitsAllocated (unsigned short *ba*) [inline]

10.226.4.20 void gdcm::PixelFormat::SetBitsStored (unsigned short *bs*) [inline]

10.226.4.21 void gdcm::PixelFormat::SetHighBit (unsigned short *hb*) [inline]

10.226.4.22 void gdcm::PixelFormat::SetPixelRepresentation (unsigned short *pr*) [inline]

10.226.4.23 void gdcm::PixelFormat::SetSamplesPerPixel (unsigned short *spp*) [inline]

Examples:

CreateARGBImage.cxx, CreateCMYKImage.cxx, and GenFakeImage.cxx.

References gdcmAssertMacro.

10.226.4.24 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 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 friend class Bitmap [friend]

10.226.5.2 std::ostream& operator<< (std::ostream & *_os,* const PixelFormat & *pf*) [friend]

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

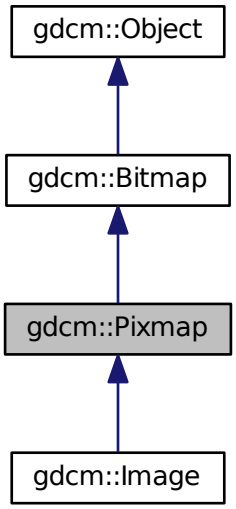
- gdcmPixelFormat.h

10.227 gdcm::Pixmap Class Reference

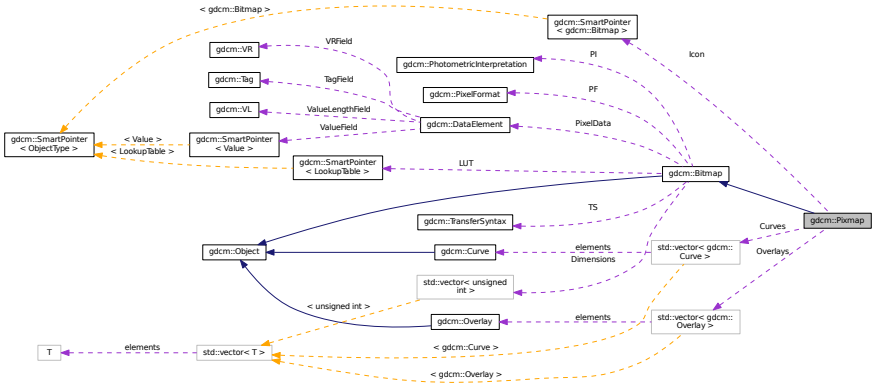
Pixmap class A bitmap based image. Used as parent for both IconImage and the main Pixel Data Image It does not contains any World Space information (IPP, IOP)

`#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 gdcm::Pixmap::Pixmap ( )

10.227.2.2 gdcm::Pixmap::~Pixmap ( )

10.227.3 Member Function Documentation

10.227.3.1 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 Curve& gdcm::Pixmap::GetCurve (size_t *i* = 0) [inline]

Curve: group 50xx.

10.227.3.3 const Curve& gdcm::Pixmap::GetCurve (size_t *i* = 0) const [inline]

10.227.3.4 const IconImage& gdcm::Pixmap::GetIconImage ( ) const [inline]

Set/Get Icon Image.

10.227.3.5 IconImage& gdcm::Pixmap::GetIconImage ( ) [inline]

10.227.3.6 size_t gdcm::Pixmap::GetNumberOfCurves ( ) const [inline]

10.227.3.7 size_t gdcm::Pixmap::GetNumberOfOverlays ( ) const [inline]

10.227.3.8 Overlay& gdcm::Pixmap::GetOverlay (size_t *i* = 0) [inline]

Overlay: group 60xx.

10.227.3.9 const Overlay& gdcm::Pixmap::GetOverlay (size_t *i* = 0) const [inline]

10.227.3.10 void gdcm::Pixmap::Print (std::ostream &) const [virtual]

Reimplemented from gdcm::Bitmap.

10.227.3.11 void gdcm::Pixmap::RemoveOverlay (size_t *i*) [inline]

10.227.3.12 void gdcm::Pixmap::SetIconImage (IconImage const & *ii*) [inline]

10.227.3.13 void gdcm::Pixmap::SetNumberOfCurves (size_t *n*) [inline]

10.227.3.14 void gdcm::Pixmap::SetNumberOfOverlays (size_t *n*) [inline]

10.227.4 Member Data Documentation

10.227.4.1 std::vector<Curve> gdcm::Pixmap::Curves [protected]

10.227.4.2 SmartPointer<IconImage> gdcm::Pixmap::Icon [protected]

10.227.4.3 std::vector<Overlay> gdcm::Pixmap::Overlays [protected]

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

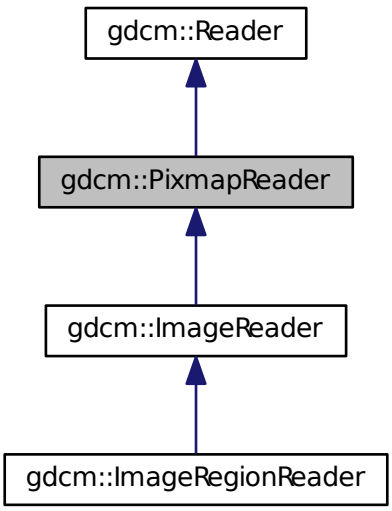
- gdcmPixmap.h

10.228 gdcm::PixmapReader Class Reference

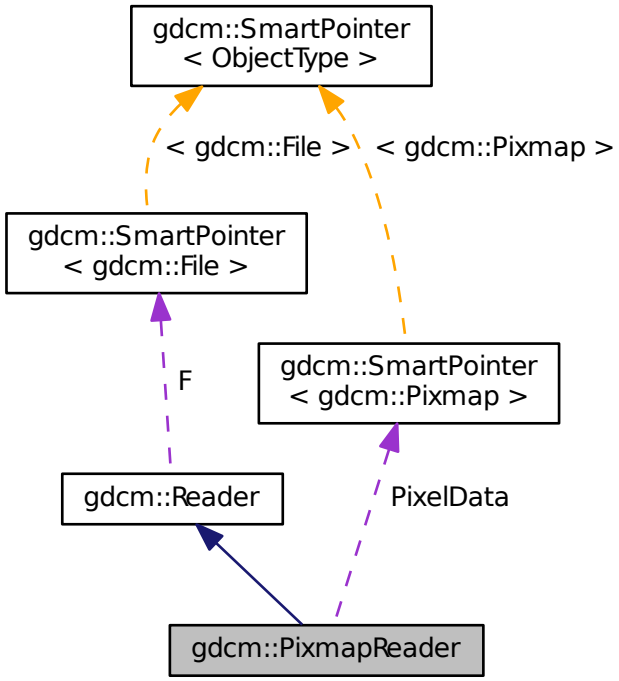
PixmapReader.

`#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 gdcm::PixmapReader::PixmapReader ( )

10.228.2.2 virtual gdcm::PixmapReader::~PixmapReader ( ) [virtual]

10.228.3 Member Function Documentation

10.228.3.1 const Pixmap& gdcm::PixmapReader::GetPixmap ( ) const

Return the read image (need to call Read() first)

10.228.3.2 Pixmap& gdcm::PixmapReader::GetPixmap ( )

10.228.3.3 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 virtual bool gdcm::PixmapReader::ReadACRNEMAImage ( ) [protected], [virtual]

Reimplemented in gdcm::ImageReader.

10.228.3.5 virtual bool gdcm::PixmapReader::ReadImage (MediaStorage const & *ms*) [protected], [virtual]

Reimplemented in gdcm::ImageReader.

10.228.3.6 bool gdcm::PixmapReader::ReadImageInternal (MediaStorage const & *ms*, bool *handlepixeldata* = true) [protected]

10.228.4 Member Data Documentation

10.228.4.1 SmartPointer<Pixmap> gdcm::PixmapReader::PixelData [protected]

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

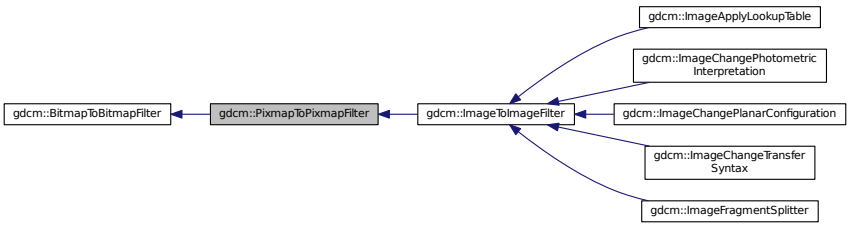
- gdcmPixmapReader.h

10.229 gdcm::PixmapToPixmapFilter Class Reference

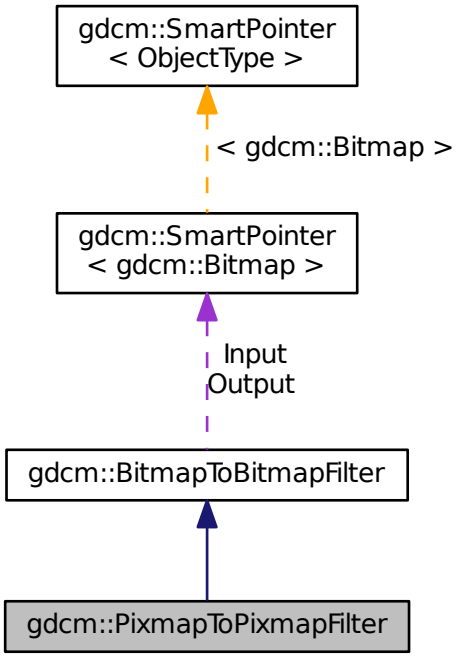
PixmapToPixmapFilter class Super class for all filter taking an image and producing an output image.

```
#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 gdcm::PixmapToPixmapFilter::PixmapToPixmapFilter ( )

10.229.2.2 gdcm::PixmapToPixmapFilter::~PixmapToPixmapFilter ( ) [inline]

10.229.3 Member Function Documentation

10.229.3.1 Pixmap& gdcm::PixmapToPixmapFilter::GetInput ( )

10.229.3.2 const Pixmap& gdcm::PixmapToPixmapFilter::GetOutput ( ) const

Get Output image.

10.229.3.3 const Pixmap& gdcm::PixmapToPixmapFilter::GetOutputAsPixmap ( ) const

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

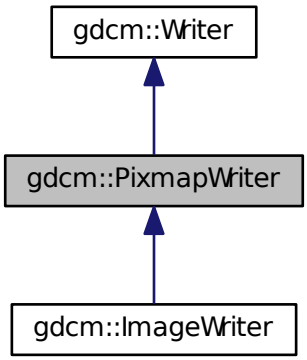
- gdcmPixmapToPixmapFilter.h

10.230 gdcm::PixmapWriter Class Reference

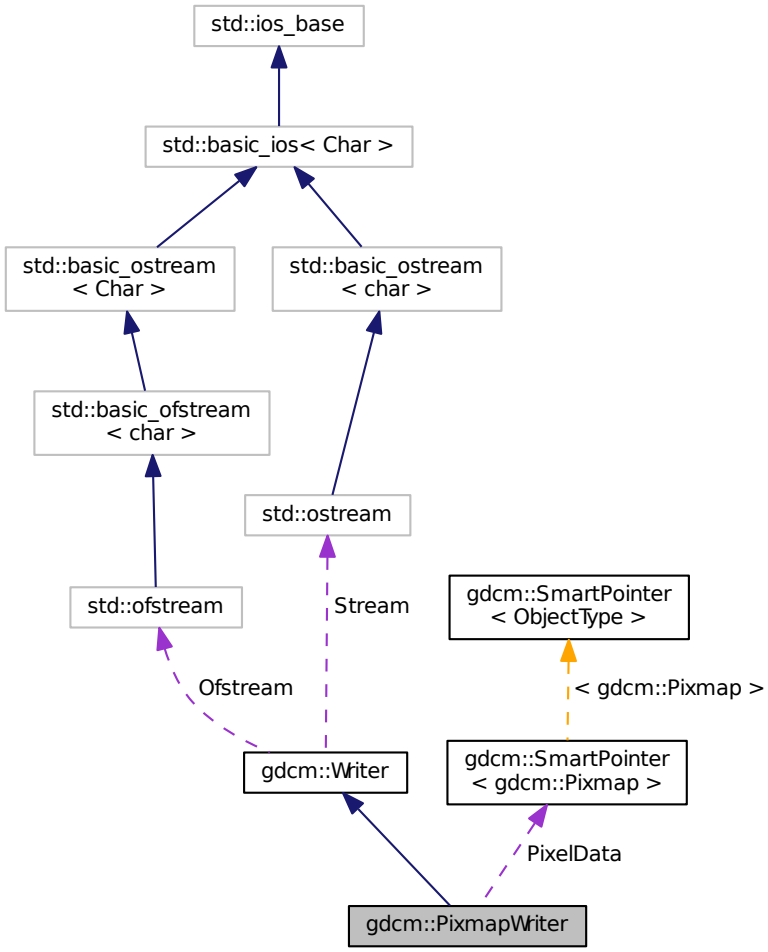
PixmapWriter This class will takes two inputs:

```
#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 gdcm::PixmapWriter::PixmapWriter ( )

10.230.2.2 gdcm::PixmapWriter::~PixmapWriter ( )

10.230.3 Member Function Documentation

10.230.3.1 void gdcm::PixmapWriter::DoIconImage (DataSet & *ds*, Pixmap const & *image*) [protected]

10.230.3.2 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 virtual Pixmap& gdcm::PixmapWriter::GetImage ( ) [inline], [virtual]

Reimplemented in gdcm::ImageWriter.

10.230.3.4 const Pixmap& gdcm::PixmapWriter::GetPixmap ( ) const [inline]

10.230.3.5 Pixmap& gdcm::PixmapWriter::GetPixmap ( ) [inline]

10.230.3.6 bool gdcm::PixmapWriter::PrepareWrite ( ) [protected]

10.230.3.7 bool gdcm::PixmapWriter::PrepareWrite (MediaStorage const & *refms*) [protected]

10.230.3.8 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 void gdcm::PixmapWriter::SetPixmap (Pixmap const & *img*)

10.230.3.10 bool gdcm::PixmapWriter::Write ( ) [virtual]

Write.

Reimplemented from gdcm::Writer.

10.230.4 Member Data Documentation

10.230.4.1 SmartPointer<Pixmap> gdcm::PixmapWriter::PixelData [protected]

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

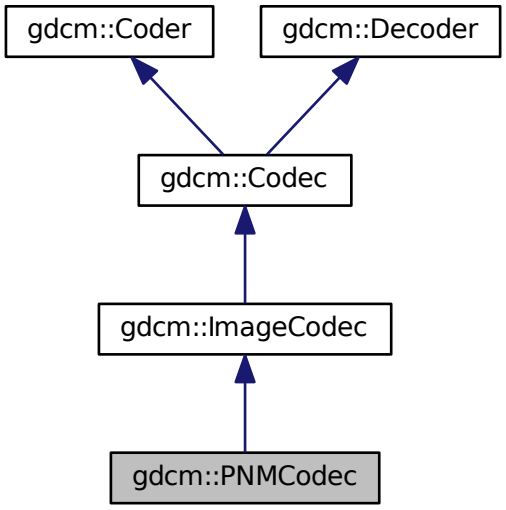
- gdcmPixmapWriter.h

10.231 gdcm::PNMCodec Class Reference

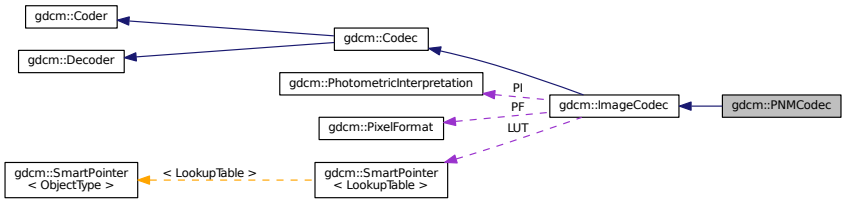
Class to do PNM PNM is the Portable anymap file format. The main web page can be found at: `http://netpbm.sourceforge.net/`.

`#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 gdcm::PNMCodec::PNMCodec ( )

10.231.2.2 gdcm::PNMCodec::~PNMCodec ( )

10.231.3 Member Function Documentation

10.231.3.1 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 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 virtual ImageCodec* gdcm::PNMCodec::Clone ( ) const [virtual]

Implements gdcm::ImageCodec.

10.231.3.4 unsigned long gdcm::PNMCodec::GetBufferLength ( ) const [inline]

10.231.3.5 bool gdcm::PNMCodec::GetHeaderInfo (std::istream & *is*, TransferSyntax & *ts*) [virtual]

Reimplemented from gdcm::ImageCodec.

10.231.3.6 bool gdcm::PNMCodec::Read (const char * *filename*, DataElement & *out*) const

10.231.3.7 void gdcm::PNMCodec::SetBufferLength (unsigned long *l*) [inline]

10.231.3.8 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 ()
- void Create ()
- const char * GetInternal () const
- VL GetLength () const
- bool IsEmpty () const
- Preamble & operator= (Preamble const &)
- void Print (std::ostream &os) const
- std::istream & Read (std::istream &is)
- void Remove ()
- void Valid ()
- std::ostream const & Write (std::ostream &os) const

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 gdcm::Preamble::Preamble ( )

10.232.2.2 gdcm::Preamble::~Preamble ( )

10.232.2.3 gdcm::Preamble::Preamble (Preamble const &) [inline]

10.232.3 Member Function Documentation

10.232.3.1 void gdcm::Preamble::Clear ( )

10.232.3.2 void gdcm::Preamble::Create ( )

10.232.3.3 const char* gdcm::Preamble::GetInternal ( ) const [inline]

10.232.3.4 VL gdcm::Preamble::GetLength ( ) const [inline]

10.232.3.5 bool gdcm::Preamble::IsEmpty ( ) const [inline]

10.232.3.6 bool gdcm::Preamble::IsValid ( ) const [inline], [protected]

10.232.3.7 Preamble& gdcm::Preamble::operator= (Preamble const &) [inline]

10.232.3.8 void gdcm::Preamble::Print (std::ostream & *os*) const

10.232.3.9 std::istream& gdcm::Preamble::Read (std::istream & *is*)

10.232.3.10 void gdcm::Preamble::Remove ( )

10.232.3.11 void gdcm::Preamble::Valid ( )

10.232.3.12 std::ostream const& gdcm::Preamble::Write (std::ostream & *os*) const

10.232.4 Friends And Related Function Documentation

10.232.4.1 std::ostream& operator<< (std::ostream & *_os,* const Preamble & *_val*) [friend]

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

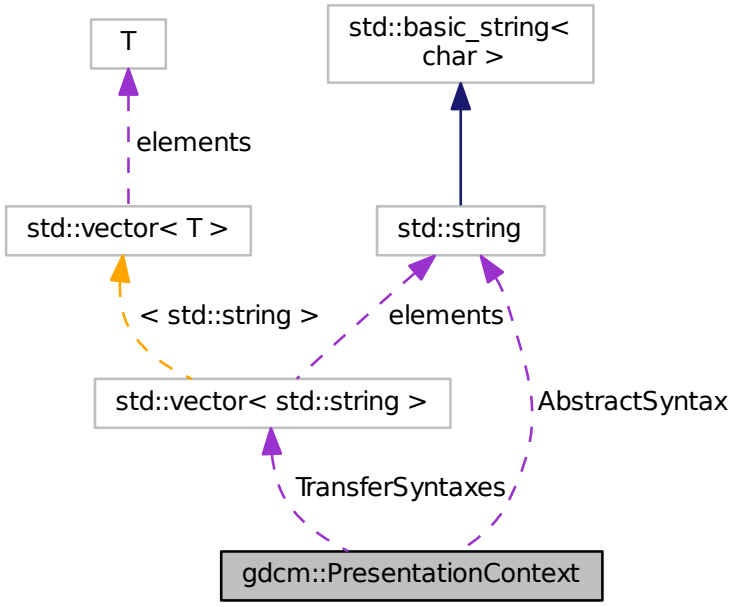
- gdcmPreamble.h

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 typedef TransferSyntaxArrayType::size_type gdcm::PresentationContext::SizeType

10.233.2.2 typedef std::vector<std::string> gdcm::PresentationContext::TransferSyntaxArrayType

10.233.3 Constructor & Destructor Documentation

10.233.3.1 gdcm::PresentationContext::PresentationContext ( )

10.233.3.2 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 void gdcm::PresentationContext::AddTransferSyntax (const char * *tsstr*)

10.233.4.2 const char* gdcm::PresentationContext::GetAbstractSyntax ( ) const [inline]

10.233.4.3 SizeType gdcm::PresentationContext::GetNumberOfTransferSyntaxes ( ) const [inline]

10.233.4.4 uint8_t gdcm::PresentationContext::GetPresentationContextID ( ) const

10.233.4.5 const char* gdcm::PresentationContext::GetTransferSyntax (SizeType *i*) const [inline]

10.233.4.6 bool gdcm::PresentationContext::operator== (const PresentationContext & *pc*) const [inline]

References AbstractSyntax, and TransferSyntaxes.

10.233.4.7 void gdcm::PresentationContext::Print (std::ostream & *os*) const

10.233.4.8 void gdcm::PresentationContext::SetAbstractSyntax (const char * *absyn*) [inline]

10.233.4.9 void gdcm::PresentationContext::SetPresentationContextID (uint8_t *id*)

10.233.5 Member Data Documentation

10.233.5.1 std::string gdcm::PresentationContext::AbstractSyntax [protected]

Referenced by operator==().

10.233.5.2 uint8_t gdcm::PresentationContext::ID [protected]

10.233.5.3 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 Table 9-18 PRESENTATION CONTEXT ITEM FIELDS.

`#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 gdcm::network::PresentationContextAC::PresentationContextAC ( )

10.234.3 Member Function Documentation

10.234.3.1 uint8_t gdcm::network::PresentationContextAC::GetPresentationContextID ( ) const [inline]

10.234.3.2 uint8_t gdcm::network::PresentationContextAC::GetReason ( ) const [inline]

10.234.3.3 TransferSyntaxSub const& gdcm::network::PresentationContextAC::GetTransferSyntax ( ) const [inline]

10.234.3.4 void gdcm::network::PresentationContextAC::Print (std::ostream & *os*) const

10.234.3.5 std::istream& gdcm::network::PresentationContextAC::Read (std::istream & *is*)

10.234.3.6 void gdcm::network::PresentationContextAC::SetPresentationContextID (uint8_t *id*)

10.234.3.7 void gdcm::network::PresentationContextAC::SetReason (uint8_t *r*) [inline]

10.234.3.8 void gdcm::network::PresentationContextAC::SetTransferSyntax (TransferSyntaxSub const & *ts*)

10.234.3.9 size_t gdcm::network::PresentationContextAC::Size ( ) const

10.234.3.10 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 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.

```
#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 typedef std::vector<PresentationContext> gdcm::PresentationContextGenerator::PresentationContextArrayType

10.235.2.2 typedef PresentationContextArrayType::size_type gdcm::PresentationContextGenerator::SizeType

10.235.3 Constructor & Destructor Documentation

10.235.3.1 gdcm::PresentationContextGenerator::PresentationContextGenerator ( )

10.235.4 Member Function Documentation

10.235.4.1 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 bool gdcm::PresentationContextGenerator::AddPresentationContext (const char * *absyn*, const char * *ts*) [protected]

10.235.4.3 bool gdcm::PresentationContextGenerator::GenerateFromFilenames (const Directory::FilenamesType & *files*)

Generate the PresentationContext array from a File-Set. File specified needs to be valid DICOM files. Used for C-STORE operations

Examples:

CStoreQtProgress.cxx.

10.235.4.4 bool gdcm::PresentationContextGenerator::GenerateFromUID (UIDs::TSName *asname*)

Generate the PresentationContext array from a UID (eg. VerificationSOPClass)

10.235.4.5 const char* gdcm::PresentationContextGenerator::GetDefaultTransferSyntax ( ) const [protected]

10.235.4.6 PresentationContextArrayType const& gdcm::PresentationContextGenerator::GetPresentationContexts ( ) [inline]

Examples:

CStoreQtProgress.cxx.

10.235.4.7 void gdcm::PresentationContextGenerator::SetDefaultTransferSyntax (const TransferSyntax & *ts*)

Not implemented for now. GDCM internally uses Implicit Little Endian.

10.235.4.8 void gdcm::PresentationContextGenerator::SetMergeModeToAbstractSyntax ( )

10.235.4.9 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 Table 9-13 PRESENTATION CONTEXT ITEM FIELDS.

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

10.236.3 Constructor & Destructor Documentation

10.236.3.1 gdcm::network::PresentationContextRQ::PresentationContextRQ ( )

10.236.3.2 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 gdcm::network::PresentationContextRQ::PresentationContextRQ (const PresentationContext & *pc*)

10.236.4 Member Function Documentation

10.236.4.1 void gdcm::network::PresentationContextRQ::AddTransferSyntax (TransferSyntaxSub const & *ts*)

10.236.4.2 AbstractSyntax const& gdcm::network::PresentationContextRQ::GetAbstractSyntax ( ) const [inline]

10.236.4.3 AbstractSyntax& gdcm::network::PresentationContextRQ::GetAbstractSyntax ( ) [inline]

10.236.4.4 SizeType gdcm::network::PresentationContextRQ::GetNumberOfTransferSyntaxes ( ) const [inline]

10.236.4.5 uint8_t gdcm::network::PresentationContextRQ::GetPresentationContextID ( ) const

10.236.4.6 TransferSyntaxSub const& gdcm::network::PresentationContextRQ::GetTransferSyntax (SizeType *i*) const [inline]

10.236.4.7 TransferSyntaxSub& gdcm::network::PresentationContextRQ::GetTransferSyntax (SizeType *i*) [inline]

10.236.4.8 std::vector<TransferSyntaxSub> const& gdcm::network::PresentationContextRQ::GetTransferSyntaxes ( ) const [inline]

10.236.4.9 bool gdcm::network::PresentationContextRQ::operator== (const PresentationContextRQ & *pc*) const [inline]

10.236.4.10 void gdcm::network::PresentationContextRQ::Print (std::ostream & *os*) const

10.236.4.11 std::istream& gdcm::network::PresentationContextRQ::Read (std::istream & *is*)

10.236.4.12 void gdcm::network::PresentationContextRQ::SetAbstractSyntax (AbstractSyntax const & *absyn*)

10.236.4.13 void gdcm::network::PresentationContextRQ::SetPresentationContextID (uint8_t *id*)

10.236.4.14 size_t gdcm::network::PresentationContextRQ::Size ( ) const

10.236.4.15 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 Table 9-23 PRESENTATION-DATA-VALUE ITEM FIELDS.

`#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 gdcm::network::PresentationDataValue::PresentationDataValue ( )

10.237.3 Member Function Documentation

10.237.3.1 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 static DataSet gdcm::network::PresentationDataValue::ConcatenatePDVBlobsAsExplicit (const std::vector< PresentationDataValue > & *inPDVs*) [static]

10.237.3.3 const std::string& gdcm::network::PresentationDataValue::GetBlob ( ) const

10.237.3.4 bool gdcm::network::PresentationDataValue::GetIsCommand ( ) const

10.237.3.5 bool gdcm::network::PresentationDataValue::GetIsLastFragment ( ) const

10.237.3.6 uint8_t gdcm::network::PresentationDataValue::GetMessageHeader ( ) const [inline]

10.237.3.7 uint8_t gdcm::network::PresentationDataValue::GetPresentationContextID ( ) const [inline]

10.237.3.8 void gdcm::network::PresentationDataValue::Print (std::ostream & *os*) const

10.237.3.9 std::istream& gdcm::network::PresentationDataValue::Read (std::istream & *is*)

10.237.3.10 std::istream& gdcm::network::PresentationDataValue::ReadInto (std::istream & *is*, std::ostream & *os*)

10.237.3.11 void gdcm::network::PresentationDataValue::SetBlob (const std::string & *partialblob*)

10.237.3.12 void gdcm::network::PresentationDataValue::SetCommand (bool *inCommand*)

10.237.3.13 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 void gdcm::network::PresentationDataValue::SetLastFragment (bool *inLast*)

10.237.3.15 void gdcm::network::PresentationDataValue::SetMessageHeader (uint8_t *messageheader*) [inline]

10.237.3.16 void gdcm::network::PresentationDataValue::SetPresentationContextID (uint8_t *id*) [inline]

10.237.3.17 size_t gdcm::network::PresentationDataValue::Size ( ) const

10.237.3.18 const std::ostream& gdcm::network::PresentationDataValue::Write (std::ostream & *os*) const

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

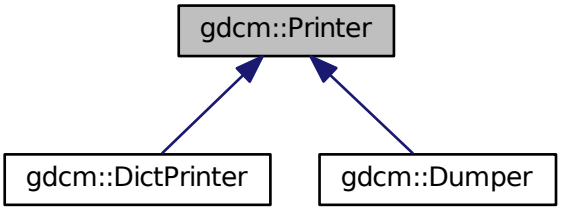
- gdcmPresentationDataValue.h

10.238 gdcm::Printer Class Reference

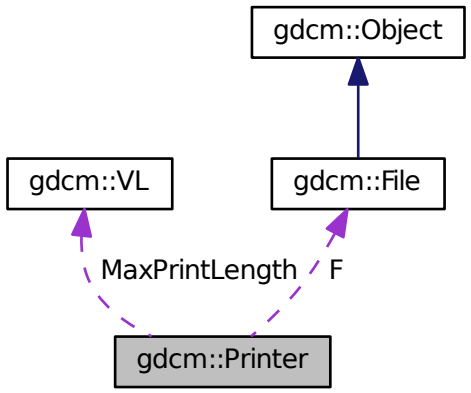
Printer class.

```
#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 enum gdcm::Printer::PrintStyles

Enumerator

VERBOSE_STYLE

CONDENSED_STYLE

XML

10.238.3 Constructor & Destructor Documentation

10.238.3.1 gdcm::Printer::Printer ( )

10.238.3.2 gdcm::Printer::~Printer ( )

10.238.4 Member Function Documentation

10.238.4.1 PrintStyles gdcm::Printer::GetPrintStyle ( ) const [inline]

Get PrintStyle value.

10.238.4.2 void gdcm::Printer::Print (std::ostream & *os*)

Print.

Examples:

DumpToshibaDTI.cxx.

10.238.4.3 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 void gdcm::Printer::PrintDataSet (const DataSet & *ds,* std::ostream & *os,* const std::string & *s* = " ")

Print an individual dataset.

10.238.4.5 void gdcm::Printer::PrintSQ (const SequenceOfItems * *sqi,* std::ostream & *os,* std::string const & *indent*) [protected]

10.238.4.6 void gdcm::Printer::SetColor (bool *c*)

Set color mode or not.

Examples:

DumpToshibaDTI.cxx.

10.238.4.7 void gdcm::Printer::SetFile (File const & *f*) [inline]

Set file.

Examples:

DumpToshibaDTI.cxx.

10.238.4.8 void gdcm::Printer::SetStyle (PrintStyles *ps*) [inline]

Set PrintStyle value.

10.238.5 Member Data Documentation

10.238.5.1 const File* gdcm::Printer::F [protected]

10.238.5.2 VL gdcm::Printer::MaxPrintLength [protected]

10.238.5.3 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 gdcm::PrivateDict::PrivateDict ( ) [inline]

10.239.2.2 gdcm::PrivateDict::~PrivateDict ( ) [inline]

10.239.3 Member Function Documentation

10.239.3.1 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 bool gdcm::PrivateDict::FindDictEntry (const PrivateTag & *tag*) const [inline]

10.239.3.3 const DictEntry& gdcm::PrivateDict::GetDictEntry (const PrivateTag & *tag*) const [inline]

10.239.3.4 bool gdcm::PrivateDict::IsEmpty ( ) const [inline]

10.239.3.5 void gdcm::PrivateDict::LoadDefault ( ) [protected]

10.239.3.6 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 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 friend class Dicts [friend]

10.239.4.2 std::ostream& operator<< (std::ostream & *os,* const PrivateDict & *val*) [friend]

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

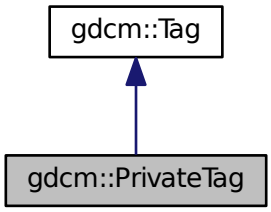
- gdcmDict.h

10.240 gdcm::PrivateTag Class Reference

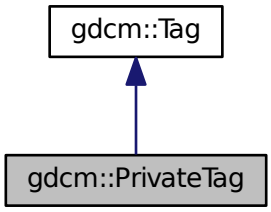
Class to represent a Private DICOM Data Element (Attribute) Tag (Group, Element, Owner)

`#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 gdcm::PrivateTag::PrivateTag (uint16_t *group* = 0, uint16_t *element* = 0, const char * *owner* = " ") [inline]

10.240.2.2 gdcm::PrivateTag::PrivateTag (Tag const & *t*, const char * *owner* = " ") [inline]

References gdcm::Tag::GetElement().

10.240.3 Member Function Documentation

10.240.3.1 DataElement gdcm::PrivateTag::GetAsDataElement ( ) const

10.240.3.2 const char* gdcm::PrivateTag::GetOwner ( ) const [inline]

Examples:

PublicDict.cxx.

Referenced by gdcm::PrivateDict::PrintXML().

10.240.3.3 bool gdcm::PrivateTag::operator< (const PrivateTag & *_val*) const

10.240.3.4 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 void gdcm::PrivateTag::SetOwner (const char * *owner*) [inline]

10.240.4 Friends And Related Function Documentation

10.240.4.1 std::ostream& operator<< (std::ostream & *_os,* const PrivateTag & *_val*) [friend]

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

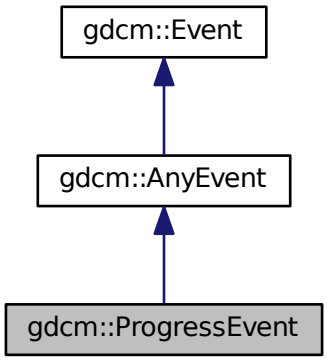
- gdcmPrivateTag.h

10.241 gdcm::ProgressEvent Class Reference

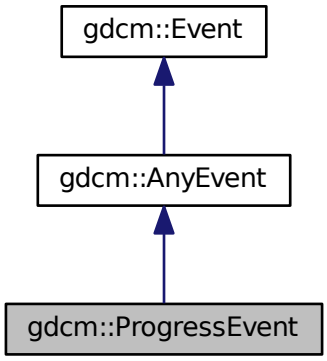
ProgressEvent Special type of event triggered during.

```
#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 typedef ProgressEvent gdcm::ProgressEvent::Self

10.241.2.2 typedef AnyEvent gdcm::ProgressEvent::Superclass

10.241.3 Constructor & Destructor Documentation

10.241.3.1 gdcm::ProgressEvent::ProgressEvent (double *p* = 0) [inline]

10.241.3.2 virtual gdcm::ProgressEvent::~ProgressEvent ( ) [inline], [virtual]

10.241.3.3 gdcm::ProgressEvent::ProgressEvent (const Self & *s*) [inline]

10.241.4 Member Function Documentation

10.241.4.1 virtual bool gdcm::ProgressEvent::CheckEvent (const ::gdcm::Event * *e*) const [inline], [virtual]

10.241.4.2 virtual const char* gdcm::ProgressEvent::GetEventName ( ) const [inline], [virtual]

Return the StringName associated with the event.

Implements gdcm::Event.

10.241.4.3 double gdcm::ProgressEvent::GetProgress ( ) const [inline]

10.241.4.4 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 void gdcm::ProgressEvent::SetProgress (double *p*) [inline]

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

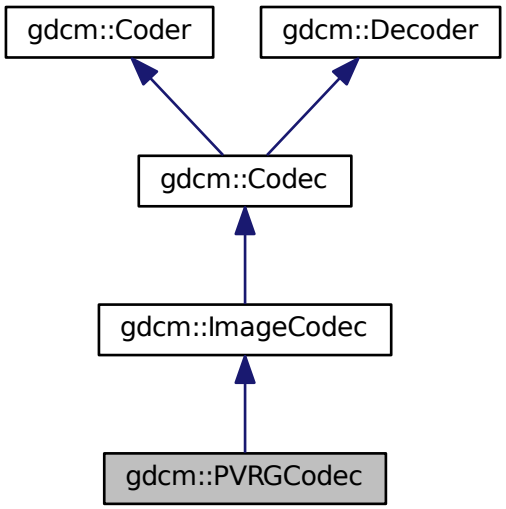
- gdcmProgressEvent.h

10.242 gdcm::PVRGCodec Class Reference

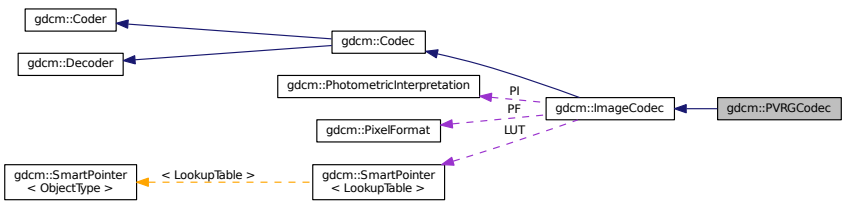
PVRGCodec.

`#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 gdcm::PVRGCodec::PVRGCodec ( )

10.242.2.2 gdcm::PVRGCodec::~PVRGCodec ( )

10.242.3 Member Function Documentation

10.242.3.1 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 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 virtual ImageCodec* gdcm::PVRGCodec::Clone ( ) const [virtual]

Implements gdcm::ImageCodec.

10.242.3.4 bool gdcm::PVRGCodec::Code (DataElement const & *in_*, DataElement & *out_*) [virtual]

Code.

Reimplemented from gdcm::Coder.

10.242.3.5 bool gdcm::PVRGCodec::Decode (DataElement const &, DataElement &) [virtual]

Decode.

Reimplemented from gdcm::ImageCodec.

10.242.3.6 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 gdcm::PythonFilter::PythonFilter ( )

10.243.2.2 gdcm::PythonFilter::∼PythonFilter ( )

10.243.3 Member Function Documentation

10.243.3.1 File& gdcm::PythonFilter::GetFile ( ) [inline]

10.243.3.2 const File& gdcm::PythonFilter::GetFile ( ) const [inline]

10.243.3.3 void gdcm::PythonFilter::SetDicts (const Dicts & *dicts*)

10.243.3.4 void gdcm::PythonFilter::SetFile (const File & *f*) [inline]

10.243.3.5 PyObject∗ gdcm::PythonFilter::ToPyObject (const Tag & *t*) const

10.243.3.6 void gdcm::PythonFilter::UseDictAlways (bool) [inline]

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

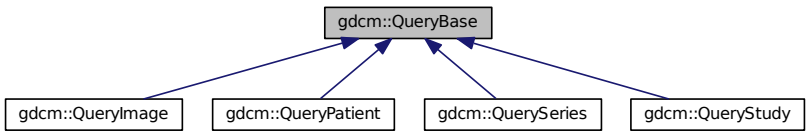
- gdcmPythonFilter.h

10.244 gdcm::QueryBase Class Reference

QueryBase contains: the base class for constructing a query dataset for a C-FIND and a C-MOVE.

`#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 virtual gdcm::QueryBase::~QueryBase ( ) [inline], [virtual]

10.244.3 Member Function Documentation

10.244.3.1 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 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 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 virtual const char* gdcm::QueryBase::GetName ( ) const [pure virtual]

Implemented in gdcm::QueryImage, gdcm::QueryPatient, gdcm::QuerySeries, and gdcm::QueryStudy.

10.244.3.5 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 virtual DataElement gdcm::QueryBase::GetQueryLevel ( ) const [pure virtual]

Implemented in gdcm::QueryImage, gdcm::QueryPatient, gdcm::QuerySeries, and gdcm::QueryStudy.

10.244.3.7 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 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 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 static void gdcm::QueryFactory::ListCharSets (std::ostream & *os*) [static]

List all possible CharSet.

10.245.2.3 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 static BaseQuery* gdcm::QueryFactory::ProduceQuery (const std::string & *sopInstanceUID,* ENQueryType *inQueryType*) [static]

10.245.2.5 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:

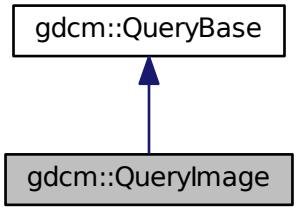
- gdcmQueryFactory.h

10.246 gdcm::QueryImage Class Reference

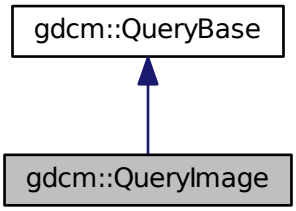
QueryImage contains: class to construct an image-based query for C-FIND and C-MOVE.

`#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 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 const char* gdcm::QueryImage::GetName ( ) const [virtual]

Implements gdcm::QueryBase.

10.246.2.3 std::vector<Tag> gdcm::QueryImage::GetOptionalTags (const ERootType & *inRootType*) const [virtual]

Implements gdcm::QueryBase.

10.246.2.4 DataElement gdcm::QueryImage::GetQueryLevel ( ) const [virtual]

Implements gdcm::QueryBase.

10.246.2.5 std::vector<Tag> gdcm::QueryImage::GetRequiredTags (const ERootType & *inRootType*) const [virtual]

Implements gdcm::QueryBase.

10.246.2.6 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:

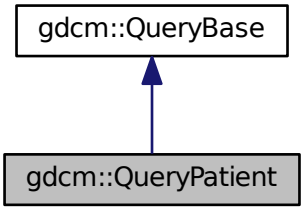
- gdcmQueryImage.h

10.247 gdcm::QueryPatient Class Reference

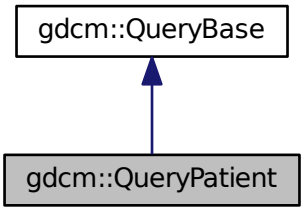
QueryPatient contains: class to construct a patient-based query for c-find and c-move.

`#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 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 const char* gdcm::QueryPatient::GetName ( ) const [virtual]

Implements gdcm::QueryBase.

10.247.2.3 std::vector<Tag> gdcm::QueryPatient::GetOptionalTags (const ERootType & *inRootType*) const [virtual]

Implements gdcm::QueryBase.

10.247.2.4 DataElement gdcm::QueryPatient::GetQueryLevel ( ) const [virtual]

Implements gdcm::QueryBase.

10.247.2.5 std::vector<Tag> gdcm::QueryPatient::GetRequiredTags (const ERootType & *inRootType*) const [virtual]

Implements gdcm::QueryBase.

10.247.2.6 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:

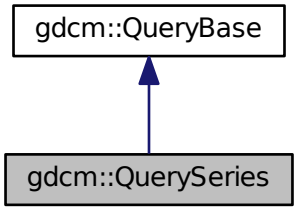
- gdcmQueryPatient.h

10.248 gdcm::QuerySeries Class Reference

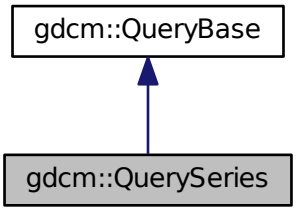
QuerySeries contains: class to construct a series-based query for c-find and c-move.

`#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 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 const char* gdcm::QuerySeries::GetName ( ) const [virtual]

Implements gdcm::QueryBase.

10.248.2.3 std::vector<Tag> gdcm::QuerySeries::GetOptionalTags (const ERootType & *inRootType*) const [virtual]

Implements gdcm::QueryBase.

10.248.2.4 DataElement gdcm::QuerySeries::GetQueryLevel ( ) const [virtual]

Implements gdcm::QueryBase.

10.248.2.5 std::vector<Tag> gdcm::QuerySeries::GetRequiredTags (const ERootType & *inRootType*) const [virtual]

Implements gdcm::QueryBase.

10.248.2.6 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:

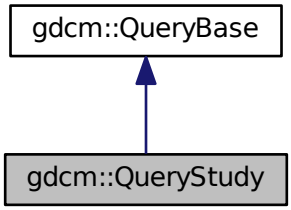
- gdcmQuerySeries.h

10.249 gdcm::QueryStudy Class Reference

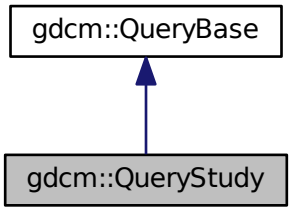
QueryStudy.h contains: class to construct a study-based query for C-FIND and C-MOVE.

`#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 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 const char* gdcm::QueryStudy::GetName ( ) const [virtual]

Implements gdcm::QueryBase.

10.249.2.3 std::vector<Tag> gdcm::QueryStudy::GetOptionalTags (const ERootType & *inRootType*) const [virtual]

Implements gdcm::QueryBase.

10.249.2.4 DataElement gdcm::QueryStudy::GetQueryLevel ( ) const [virtual]

Implements gdcm::QueryBase.

10.249.2.5 std::vector<Tag> gdcm::QueryStudy::GetRequiredTags (const ERootType & *inRootType*) const [virtual]

Implements gdcm::QueryBase.

10.249.2.6 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:

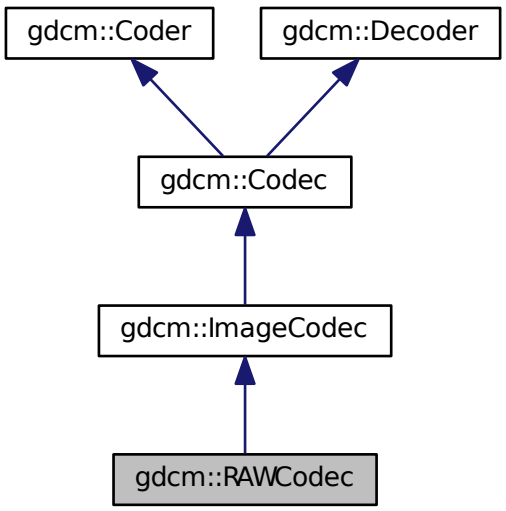
- gdcmQueryStudy.h

10.250 gdcm::RAWCodec Class Reference

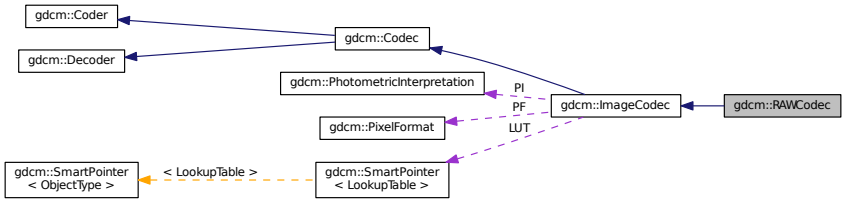
RAWCodec class.

`#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 gdcm::RAWCodec::RAWCodec ( )

10.250.2.2 gdcm::RAWCodec::~RAWCodec ( )

10.250.3 Member Function Documentation

10.250.3.1 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 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 virtual ImageCodec* gdcm::RAWCodec::Clone ( ) const [virtual]

Implements gdcm::ImageCodec.

10.250.3.4 bool gdcm::RAWCodec::Code (DataElement const & *in_*, DataElement & *out_*) [virtual]

Code.

Reimplemented from gdcm::Coder.

10.250.3.5 bool gdcm::RAWCodec::Decode (DataElement const &, DataElement &) [virtual]

Decode.

Reimplemented from gdcm::ImageCodec.

10.250.3.6 bool gdcm::RAWCodec::DecodeByStreams (std::istream & *is*, std::ostream & *os*) [protected], [virtual]

Reimplemented from gdcm::ImageCodec.

10.250.3.7 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 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:

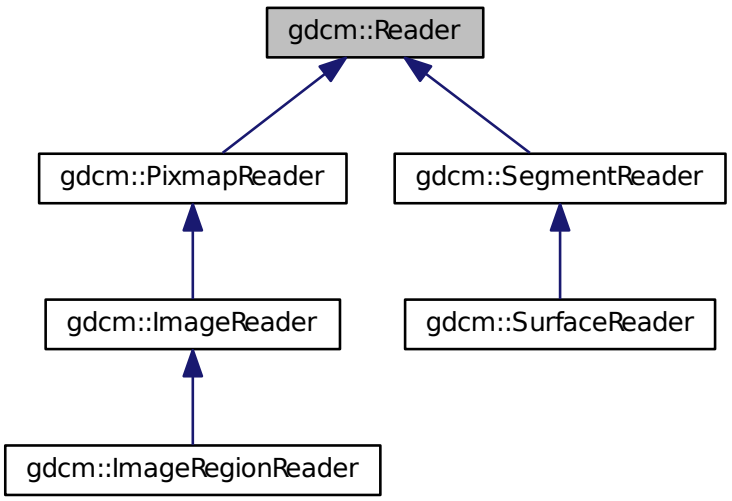
- gdcmRAWCodec.h

10.251 gdcm::Reader Class Reference

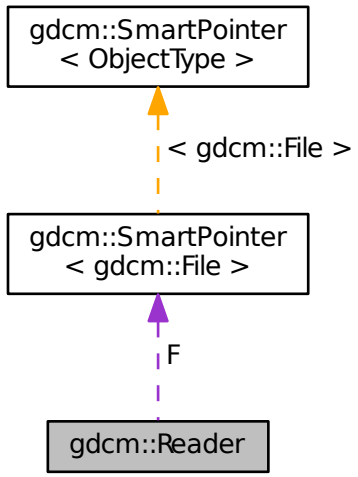
Reader ala DOM (Document Object Model)

```
#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 gdcm::Reader::Reader ( )

10.251.2.2 virtual gdcm::Reader::~Reader ( ) [virtual]

10.251.3 Member Function Documentation

10.251.3.1 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 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 File& gdcm::Reader::GetFile ( ) [inline]

Set/Get File.

10.251.3.4 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 std::istream* gdcm::Reader::GetStreamPtr ( ) const [inline], [protected]

10.251.3.6 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 bool gdcm::Reader::ReadDataSet ( ) [protected]

10.251.3.8 bool gdcm::Reader::ReadMetaInformation ( ) [protected]

10.251.3.9 bool gdcm::Reader::ReadPreamble ( ) [protected]

10.251.3.10 bool gdcm::Reader::ReadSelectedPrivateTags (std::set< PrivateTag > const & *ptags*, bool *readvalues* = true)

Will only read the specified selected private tags.

10.251.3.11 bool gdcm::Reader::ReadSelectedTags (std::set< Tag > const & *tags*, bool *readvalues* = true)

Will only read the specified selected tags.

10.251.3.12 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 void gdcm::Reader::SetFile (File & *file*) [inline]

Set/Get File.

10.251.3.14 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, 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 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 friend class StreamImageReader [friend]

10.251.5 Member Data Documentation

10.251.5.1 SmartPointer<File> gdcm::Reader::F [protected]

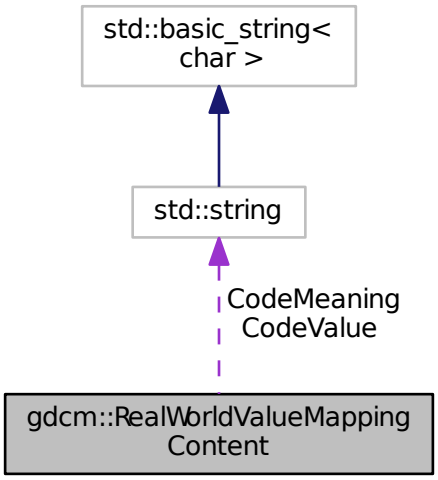
The documentation for this class was generated from the following file:

- 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 std::string gdcm::RealWorldValueMappingContent::CodeMeaning

10.252.1.2 std::string gdcm::RealWorldValueMappingContent::CodeValue

10.252.1.3 double gdcm::RealWorldValueMappingContent::RealWorldValueIntercept

10.252.1.4 double gdcm::RealWorldValueMappingContent::RealWorldValueSlope

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

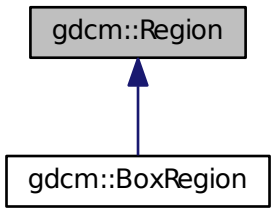
- gdcmImageHelper.h

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 gdcm::Region::Region ( )

10.253.2.2 virtual gdcm::Region::~Region ( ) [virtual]

10.253.3 Member Function Documentation

10.253.3.1 virtual size_t gdcm::Region::Area ( ) const [pure virtual]

compute the area

Implemented in gdcm::BoxRegion.

10.253.3.2 virtual Region* gdcm::Region::Clone ( ) const [pure virtual]

Implemented in gdcm::BoxRegion.

10.253.3.3 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 virtual bool gdcm::Region::Empty ( ) const [pure virtual]

return whether this domain is empty:

Implemented in gdcm::BoxRegion.

10.253.3.5 virtual bool gdcm::Region::IsValid ( ) const [pure virtual]

return whether this is valid domain

Implemented in gdcm::BoxRegion.

10.253.3.6 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 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.

```
#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 gdcm::Rescaler::Rescaler ( ) [inline]

10.254.2.2 gdcm::Rescaler::~Rescaler ( ) [inline]

10.254.3 Member Function Documentation

10.254.3.1 PixelFormat::ScalarType gdcm::Rescaler::ComputeInterceptSlopePixelType ( )

Compute the Pixel Format of the output data Used for direct transformation

10.254.3.2 PixelFormat gdcm::Rescaler::ComputePixelTypeFromMinMax ( )

Compute the Pixel Format of the output data Used for inverse transformation

10.254.3.3 double gdcm::Rescaler::GetIntercept ( ) const [inline]

10.254.3.4 double gdcm::Rescaler::GetSlope ( ) const [inline]

10.254.3.5 bool gdcm::Rescaler::InverseRescale (char * *out*, const char * *in*, size_t *n*)

Inverse transform.

10.254.3.6 template<typename TIn > void gdcm::Rescaler::InverseRescaleFunctionIntoBestFit (char * *out*, const TIn * *in*, size_t *n*) [protected]

10.254.3.7 bool gdcm::Rescaler::Rescale (char * *out*, const char * *in*, size_t *n*)

Direct transform.

10.254.3.8 template<typename TIn > void gdcm::Rescaler::RescaleFunctionIntoBestFit (char * *out*, const TIn * *in*, size_t *n*) [protected]

10.254.3.9 void gdcm::Rescaler::SetIntercept (double *i*) [inline]

Set Intercept: used for both direct&inverse transformation.

10.254.3.10 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 void gdcm::Rescaler::SetPixelFormat (PixelFormat const & *pf*) [inline]

Set Pixel Format of input data.

10.254.3.12 void gdcm::Rescaler::SetSlope (double *s*) [inline]

Set Slope: user for both direct&inverse transformation.

10.254.3.13 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 void gdcm::Rescaler::SetUseTargetPixelType (bool *b*)

Override default behavior of Rescale.

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

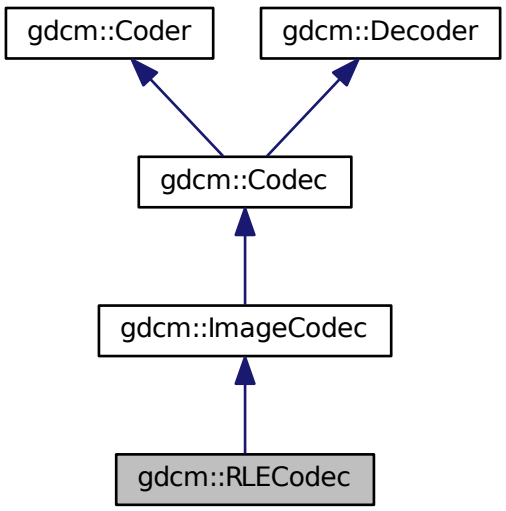
- gdcmRescaler.h

10.255 gdcm::RLECodec Class Reference

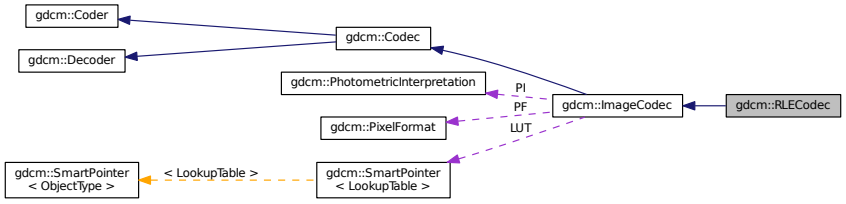
Class to do RLE.

`#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 gdcm::RLECodec::RLECodec ( )

10.255.2.2 gdcm::RLECodec::~RLECodec ( )

10.255.3 Member Function Documentation

10.255.3.1 bool gdcm::RLECodec::AppendFrameEncode (std::ostream & *out,* const char * *data,* size_t *datalen*) [protected], [virtual]

Reimplemented from gdcm::ImageCodec.

10.255.3.2 bool gdcm::RLECodec::AppendRowEncode (std::ostream & *out,* const char * *data,* size_t *datalen*) [protected], [virtual]

Reimplemented from gdcm::ImageCodec.

10.255.3.3 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 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 virtual ImageCodec* gdcm::RLECodec::Clone ( ) const [virtual]

Implements gdcm::ImageCodec.

10.255.3.6 bool gdcm::RLECodec::Code (DataElement const & *in_*, DataElement & *out_*) [virtual]

Code.

Reimplemented from gdcm::Coder.

10.255.3.7 bool gdcm::RLECodec::Decode (DataElement const &, DataElement &) [virtual]

Decode.

Reimplemented from gdcm::ImageCodec.

10.255.3.8 bool gdcm::RLECodec::DecodeByStreams (std::istream & *is*, std::ostream & *os*) [protected], [virtual]

Reimplemented from gdcm::ImageCodec.

10.255.3.9 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 unsigned long gdcm::RLECodec::GetBufferLength ( ) const [inline]

10.255.3.11 bool gdcm::RLECodec::GetHeaderInfo (std::istream & *is*, TransferSyntax & *ts*) [virtual]

Reimplemented from gdcm::ImageCodec.

10.255.3.12 bool gdcm::RLECodec::IsFrameEncoder ( ) [protected], [virtual]

Reimplemented from gdcm::ImageCodec.

10.255.3.13 bool gdcm::RLECodec::IsRowEncoder ( ) [protected], [virtual]

Reimplemented from gdcm::ImageCodec.

10.255.3.14 void gdcm::RLECodec::SetBufferLength (unsigned long l) [inline]

10.255.3.15 void gdcm::RLECodec::SetLength (unsigned long l) [inline]

10.255.3.16 bool gdcm::RLECodec::StartEncode (std::ostream &) [protected], [virtual]

Reimplemented from gdcm::ImageCodec.

10.255.3.17 bool gdcm::RLECodec::StopEncode (std::ostream &) [protected], [virtual]

Reimplemented from gdcm::ImageCodec.

10.255.4 Friends And Related Function Documentation

10.255.4.1 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 PS 3.7 Table D.3-9 SCP/SCU ROLE SELECTION SUB-ITEM FIELDS (A-ASSOCIATE-RQ)

`#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 gdcm::network::RoleSelectionSub::RoleSelectionSub ( )

10.256.3 Member Function Documentation

10.256.3.1 void gdcm::network::RoleSelectionSub::Print (std::ostream & *os*) const

10.256.3.2 std::istream& gdcm::network::RoleSelectionSub::Read (std::istream & *is*)

10.256.3.3 void gdcm::network::RoleSelectionSub::SetTuple (const char * *uid*, uint8_t *scurole*, uint8_t *scprole*)

10.256.3.4 size_t gdcm::network::RoleSelectionSub::Size ( ) const

10.256.3.5 const std::ostream& gdcm::network::RoleSelectionSub::Write (std::ostream & *os*) const

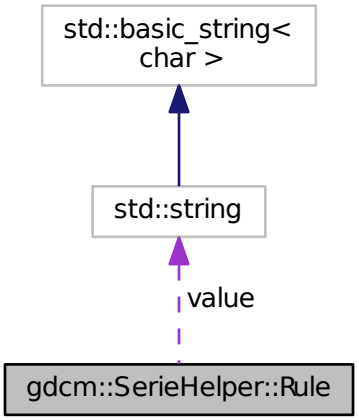
The documentation for this class was generated from the following file:

- 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 uint16_t gdcm::SerieHelper::Rule::elem

10.257.1.2 uint16_t gdcm::SerieHelper::Rule::group

10.257.1.3 int gdcm::SerieHelper::Rule::op

10.257.1.4 std::string gdcm::SerieHelper::Rule::value

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

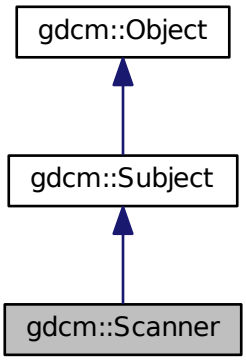
- gdcmSerieHelper.h

10.258 gdcm::Scanner Class Reference

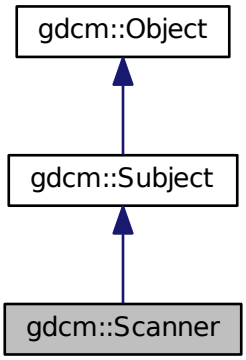
Scanner This filter is meant for quickly browsing a FileSet (a set of files on disk). Special consideration are taken so as to read the mimimum amount of information in each file in order to retrieve the user specified set of DICOM Attribute.

`#include <gdcmScanner.h>`

Inheritance diagram for gdcm::Scanner:



Collaboration diagram for gdcm::Scanner:



Classes

- struct ltstr

Public Types

- typedef MappingType::const_iterator ConstIterator
- typedef std::map< const char *, TagToValue, ltstr > MappingType
- typedef std::map< Tag, const char * > TagToValue
- typedef TagToValue::value_type TagToValueValueType
- typedef std::set< std::string > ValuesType

Public Member Functions

- Scanner ()
- ~Scanner ()
- 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 typedef MappingType::const_iterator gdcm::Scanner::ConstIterator

10.258.2.2 typedef std::map<const char *,TagToValue, ltstr> gdcm::Scanner::MappingType

10.258.2.3 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 typedef TagToValue::value_type gdcm::Scanner::TagToValueValueType

10.258.2.5 typedef std::set< std::string > gdcm::Scanner::ValuesType

10.258.3 Constructor & Destructor Documentation

10.258.3.1 gdcm::Scanner::Scanner ( ) [inline]

10.258.3.2 gdcm::Scanner::~Scanner ( )

10.258.4 Member Function Documentation

10.258.4.1 void gdcm::Scanner::AddPrivateTag (PrivateTag const & *t*)

10.258.4.2 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 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 ConstIterator gdcm::Scanner::Begin ( ) const [inline]

10.258.4.5 void gdcm::Scanner::ClearSkipTags ( )

10.258.4.6 void gdcm::Scanner::ClearTags ( )

10.258.4.7 ConstIterator gdcm::Scanner::End ( ) const [inline]

10.258.4.8 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 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 Directory::FilenamesType const& gdcm::Scanner::GetFilenames ( ) const [inline]

10.258.4.11 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 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 TagToValue const& gdcm::Scanner::GetMappingFromTagToValue (Tag const & *t,* const char * *value*) const

See GetFilenameFromTagToValue(). This is simply GetFilenameFromTagToValue followed.

10.258.4.14 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 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 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 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 ValuesType gdcm::Scanner::GetValues (Tag const & *t*) const

Get all the values found (in lexicographic order) associated with Tag 't'.

10.258.4.19 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 static SmartPointer<Scanner> gdcm::Scanner::New ( ) [inline], [static]

for wrapped language: instanciate a reference counted object

10.258.4.21 void gdcm::Scanner::Print (std::ostream & *os*) const [virtual]

Print result.

Reimplemented from gdcm::Object.

Referenced by gdcm::operator<<().

10.258.4.22 void gdcm::Scanner::ProcessPublicTag (StringFilter & *sf,* const char * *filename*) [protected]

10.258.4.23 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 std::ostream& operator<< (std::ostream & *_os,* const Scanner & *s*) [friend]

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

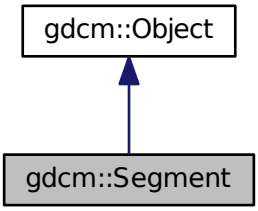
- gdcmScanner.h

10.259 gdcm::Segment Class Reference

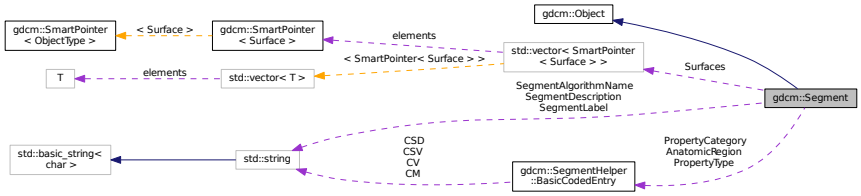
This class defines a segment. It mainly contains attributes of group 0x0062. In addition, it can be associated with surface.

`#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 typedef std::vector< SmartPointer< Surface > > gdcm::Segment::SurfaceVector

10.259.3 Member Enumeration Documentation

10.259.3.1 enum gdcm::Segment::ALGOType

Enumerator

MANUAL

AUTOMATIC

ALGOType_END

10.259.4 Constructor & Destructor Documentation

10.259.4.1 gdcm::Segment::Segment ( )

10.259.4.2 virtual gdcm::Segment::~Segment ( ) [virtual]

10.259.5 Member Function Documentation

10.259.5.1 void gdcm::Segment::AddSurface (SmartPointer< Surface > *surface*)

10.259.5.2 static ALGOType gdcm::Segment::GetALGOType (const char * *type*) [static]

10.259.5.3 static const char* gdcm::Segment::GetALGOTypeString (ALGOType *type*) [static]

10.259.5.4 SegmentHelper::BasicCodedEntry const& gdcm::Segment::GetAnatomicRegion ( ) const

10.259.5.5 SegmentHelper::BasicCodedEntry& gdcm::Segment::GetAnatomicRegion ( )

10.259.5.6 SegmentHelper::BasicCodedEntry const& gdcm::Segment::GetPropertyCategory ( ) const

10.259.5.7 SegmentHelper::BasicCodedEntry& gdcm::Segment::GetPropertyCategory ( )

10.259.5.8 SegmentHelper::BasicCodedEntry const& gdcm::Segment::GetPropertyType ( ) const

10.259.5.9 SegmentHelper::BasicCodedEntry& gdcm::Segment::GetPropertyType ( )

10.259.5.10 const char* gdcm::Segment::GetSegmentAlgorithmName ( ) const

10.259.5.11 ALGOType gdcm::Segment::GetSegmentAlgorithmType ( ) const

10.259.5.12 const char* gdcm::Segment::GetSegmentDescription ( ) const

10.259.5.13 const char* gdcm::Segment::GetSegmentLabel ( ) const

10.259.5.14 unsigned short gdcm::Segment::GetSegmentNumber ( ) const

10.259.5.15 SmartPointer< Surface > gdcm::Segment::GetSurface (const unsigned int *idx* = 0) const

10.259.5.16 unsigned long gdcm::Segment::GetSurfaceCount ( )

10.259.5.17 SurfaceVector const& gdcm::Segment::GetSurfaces ( ) const

10.259.5.18 SurfaceVector& gdcm::Segment::GetSurfaces ( )

10.259.5.19 void gdcm::Segment::SetAnatomicRegion (SegmentHelper::BasicCodedEntry const & *BSE*)

10.259.5.20 void gdcm::Segment::SetPropertyCategory (SegmentHelper::BasicCodedEntry const & *BSE*)

10.259.5.21 void gdcm::Segment::SetPropertyType (SegmentHelper::BasicCodedEntry const & *BSE*)

10.259.5.22 void gdcm::Segment::SetSegmentAlgorithmName (const char * *name*)

10.259.5.23 void gdcm::Segment::SetSegmentAlgorithmType (ALGOType *type*)

10.259.5.24 void gdcm::Segment::SetSegmentAlgorithmType (const char * *typeStr*)

10.259.5.25 void gdcm::Segment::SetSegmentDescription (const char * *description*)

10.259.5.26 void gdcm::Segment::SetSegmentLabel (const char * *label*)

10.259.5.27 void gdcm::Segment::SetSegmentNumber (const unsigned short *num*)

10.259.5.28 void gdcm::Segment::SetSurfaceCount (const unsigned long *nb*)

10.259.6 Member Data Documentation

10.259.6.1 **SegmentHelper::BasicCodedEntry gdcm::Segment::AnatomicRegion** [protected]

10.259.6.2 **SegmentHelper::BasicCodedEntry gdcm::Segment::PropertyCategory** [protected]

10.259.6.3 **SegmentHelper::BasicCodedEntry gdcm::Segment::PropertyType** [protected]

10.259.6.4 **std::string gdcm::Segment::SegmentAlgorithmName** [protected]

10.259.6.5 **ALGOType gdcm::Segment::SegmentAlgorithmType** [protected]

10.259.6.6 **std::string gdcm::Segment::SegmentDescription** [protected]

10.259.6.7 **std::string gdcm::Segment::SegmentLabel** [protected]

10.259.6.8 **unsigned short gdcm::Segment::SegmentNumber** [protected]

10.259.6.9 **unsigned long gdcm::Segment::SurfaceCount** [protected]

10.259.6.10 **SurfaceVector gdcm::Segment::Surfaces** [protected]

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

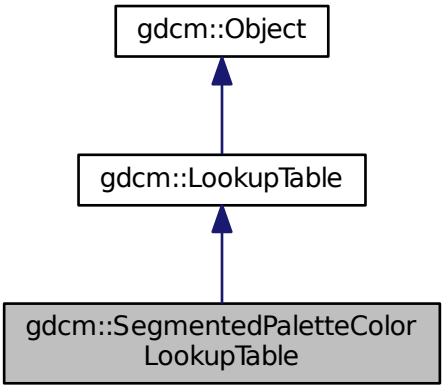
- gdcmSegment.h

10.260 gdcm::SegmentedPaletteColorLookupTable Class Reference

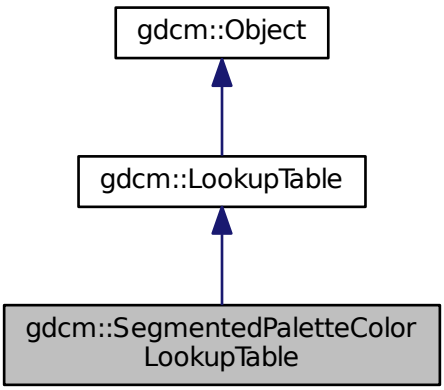
SegmentedPaletteColorLookupTable class.

`#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 gdcm::SegmentedPaletteColorLookupTable::SegmentedPaletteColorLookupTable ( )

10.260.2.2 gdcm::SegmentedPaletteColorLookupTable::~SegmentedPaletteColorLookupTable ( )

10.260.3 Member Function Documentation

10.260.3.1 void gdcm::SegmentedPaletteColorLookupTable::Print (std::ostream &) const [inline], [virtual]

Reimplemented from gdcm::LookupTable.

10.260.3.2 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:

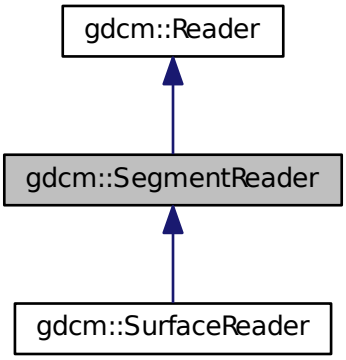
- gdcmSegmentedPaletteColorLookupTable.h

10.261 gdcm::SegmentReader Class Reference

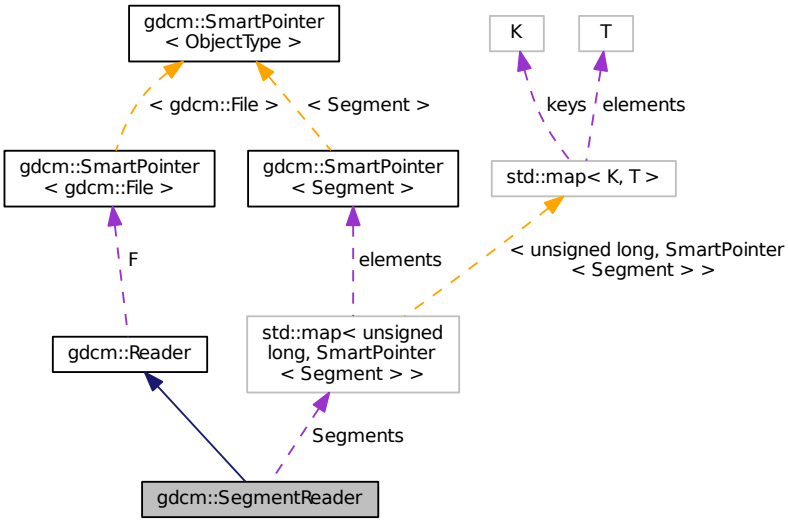
This class defines a segment reader. It reads attributes of group 0x0062.

`#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 typedef std::map< unsigned long, SmartPointer< Segment > > gdcm::SegmentReader::SegmentMap [protected]

10.261.2.2 typedef std::vector< SmartPointer< Segment > > gdcm::SegmentReader::SegmentVector

10.261.3 Constructor & Destructor Documentation

10.261.3.1 gdcm::SegmentReader::SegmentReader ( )

10.261.3.2 virtual gdcm::SegmentReader::~SegmentReader ( ) [virtual]

10.261.4 Member Function Documentation

10.261.4.1 const SegmentVector gdcm::SegmentReader::GetSegments ( ) const

10.261.4.2 SegmentVector gdcm::SegmentReader::GetSegments ( )

10.261.4.3 virtual bool gdcm::SegmentReader::Read ( ) [virtual]

Read.

Reimplemented from gdcm::Reader.

Reimplemented in gdcm::SurfaceReader.

10.261.4.4 bool gdcm::SegmentReader::ReadSegment (const Item & *segmentItem,* const unsigned int *idx*) [protected]

10.261.4.5 bool gdcm::SegmentReader::ReadSegments ( ) [protected]

10.261.5 Member Data Documentation

10.261.5.1 SegmentMap gdcm::SegmentReader::Segments [protected]

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

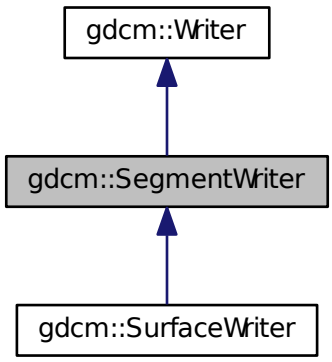
- gdcmSegmentReader.h

10.262 gdcm::SegmentWriter Class Reference

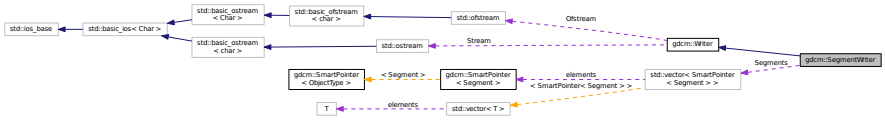
This class defines a segment writer. It writes attributes of group 0x0062.

`#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 typedef std::vector< SmartPointer< Segment > > gdcm::SegmentWriter::SegmentVector

10.262.3 Constructor & Destructor Documentation

10.262.3.1 gdcm::SegmentWriter::SegmentWriter ( )

10.262.3.2 virtual gdcm::SegmentWriter::~SegmentWriter ( ) [virtual]

10.262.4 Member Function Documentation

10.262.4.1 void gdcm::SegmentWriter::AddSegment (SmartPointer< Segment > *segment*)

10.262.4.2 unsigned int gdcm::SegmentWriter::GetNumberOfSegments ( ) const

10.262.4.3 SmartPointer< Segment > gdcm::SegmentWriter::GetSegment (const unsigned int *idx* = 0) const

10.262.4.4 const SegmentVector& gdcm::SegmentWriter::GetSegments ( ) const

10.262.4.5 SegmentVector& gdcm::SegmentWriter::GetSegments ( )

10.262.4.6 bool gdcm::SegmentWriter::PrepareWrite ( ) [protected]

10.262.4.7 void gdcm::SegmentWriter::SetNumberOfSegments (const unsigned int *size*)

10.262.4.8 void gdcm::SegmentWriter::SetSegments (SegmentVector & *segments*)

10.262.4.9 bool gdcm::SegmentWriter::Write ( ) [virtual]

Write.

Reimplemented from gdcm::Writer.

Reimplemented in gdcm::SurfaceWriter.

10.262.5 Member Data Documentation

10.262.5.1 SegmentVector gdcm::SegmentWriter::Segments [protected]

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

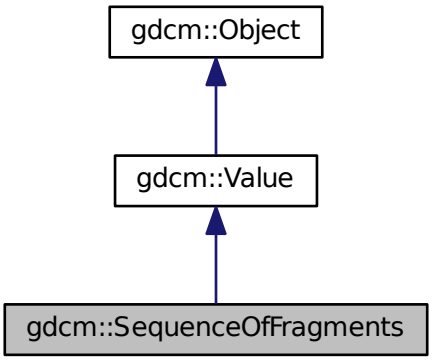
- gdcmSegmentWriter.h

10.263 gdcm::SequenceOfFragments Class Reference

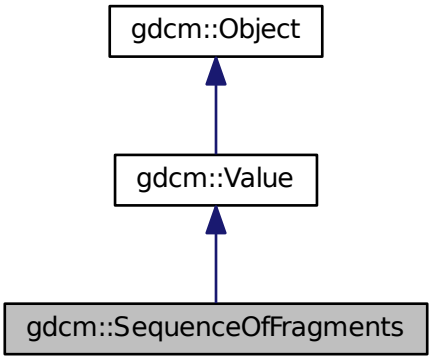
Class to represent a Sequence Of Fragments.

```
#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 typedef FragmentVector::const_iterator gdcm::SequenceOfFragments::ConstIterator

10.263.2.2 typedef std::vector<Fragment> gdcm::SequenceOfFragments::FragmentVector

10.263.2.3 typedef FragmentVector::iterator gdcm::SequenceOfFragments::Iterator

10.263.2.4 typedef FragmentVector::size_type gdcm::SequenceOfFragments::SizeType

10.263.3 Constructor & Destructor Documentation

10.263.3.1 gdcm::SequenceOfFragments::SequenceOfFragments ( ) [inline]

constructor (UndefinedLength by default)

10.263.4 Member Function Documentation

10.263.4.1 void gdcm::SequenceOfFragments::AddFragment (Fragment const & *item*)

Appends a Fragment to the already added ones.

Examples:

FixBrokenJ2K.cxx.

10.263.4.2 Iterator gdcm::SequenceOfFragments::Begin ( ) [inline]

10.263.4.3 ConstIterator gdcm::SequenceOfFragments::Begin ( ) const [inline]

10.263.4.4 void gdcm::SequenceOfFragments::Clear ( ) [virtual]

Clear.

Implements gdcm::Value.

10.263.4.5 unsigned long gdcm::SequenceOfFragments::ComputeByteLength ( ) const

10.263.4.6 VL gdcm::SequenceOfFragments::ComputeLength ( ) const

10.263.4.7 Iterator gdcm::SequenceOfFragments::End ( ) [inline]

10.263.4.8 ConstIterator gdcm::SequenceOfFragments::End ( ) const [inline]

10.263.4.9 bool gdcm::SequenceOfFragments::GetBuffer (char * *buffer,* unsigned long *length*) const

10.263.4.10 bool gdcm::SequenceOfFragments::GetFragBuffer (unsigned int *fragNb,* char * *buffer,* unsigned long & *length*) const

10.263.4.11 const Fragment& gdcm::SequenceOfFragments::GetFragment (SizeType *num*) const

Examples:

FixBrokenJ2K.cxx, and FixJAIBugJPEGLS.cxx.

10.263.4.12 VL gdcm::SequenceOfFragments::GetLength ( ) const [inline], [virtual]

Returns the SQ length, as read from disk.

Implements gdcm::Value.

10.263.4.13 SizeType gdcm::SequenceOfFragments::GetNumberOfFragments ( ) const

Examples:

FixJAIBugJPEGLS.cxx.

10.263.4.14 const BasicOffsetTable& gdcm::SequenceOfFragments::GetTable ( ) const [inline]

10.263.4.15 BasicOffsetTable& gdcm::SequenceOfFragments::GetTable ( ) [inline]

10.263.4.16 static SmartPointer<SequenceOfFragments> gdcm::SequenceOfFragments::New ( ) [inline], [static]

10.263.4.17 bool gdcm::SequenceOfFragments::operator== (const Value & *val*) const [inline], [virtual]

Implements gdcm::Value.

10.263.4.18 void gdcm::SequenceOfFragments::Print (std::ostream & *os*) const [inline], [virtual]

Reimplemented from gdcm::Object.

10.263.4.19 template<typename TSwap > std::istream& gdcm::SequenceOfFragments::Read (std::istream & *is*, bool *readvalues* = true) [inline]

10.263.4.20 template<typename TSwap > std::istream& gdcm::SequenceOfFragments::ReadPreValue (std::istream & *is*) [inline]

References gdcmDebugMacro.

10.263.4.21 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 void gdcm::SequenceOfFragments::SetLength (VL *length*) [inline], [virtual]

Sets the actual SQ length.

Implements gdcm::Value.

10.263.4.23 template<typename TSwap > std::ostream const& gdcm::SequenceOfFragments::Write (std::ostream & *os*) const [inline]

References gdcm::VL::Write().

10.263.4.24 bool gdcm::SequenceOfFragments::WriteBuffer (std::ostream & *os*) const

Examples:

GetJPEGSamplePrecision.cxx.

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

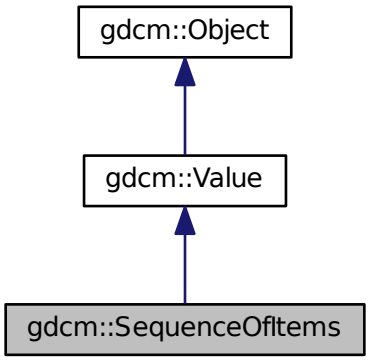
- gdcmSequenceOfFragments.h

10.264 gdcm::SequenceOfItems Class Reference

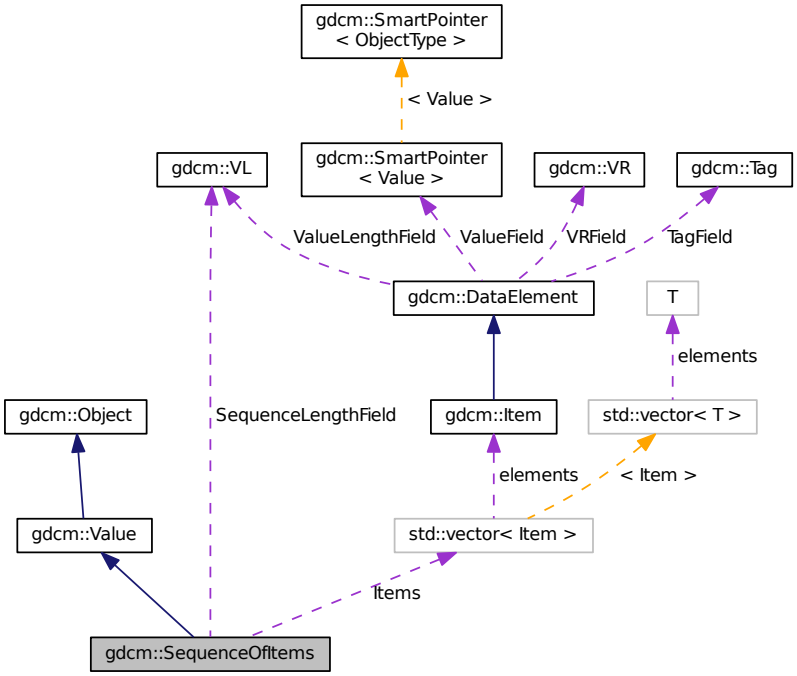
Class to represent a Sequence Of Items (value representation : SQ)

`#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 typedef ItemVector::const_iterator gdcm::SequenceOfItems::ConstIterator

10.264.2.2 typedef std::vector< Item > gdcm::SequenceOfItems::ItemVector

10.264.2.3 typedef ItemVector::iterator gdcm::SequenceOfItems::Iterator

10.264.2.4 typedef ItemVector::size_type gdcm::SequenceOfItems::SizeType

10.264.3 Constructor & Destructor Documentation

10.264.3.1 gdcm::SequenceOfItems::SequenceOfItems ( ) [inline]

constructor (UndefinedLength by default)

10.264.4 Member Function Documentation

10.264.4.1 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 Item& gdcm::SequenceOfItems::AddNewUndefinedLengthItem ( )

Appends an Item to the already added ones.

10.264.4.3 Iterator gdcm::SequenceOfItems::Begin ( ) [inline]

10.264.4.4 ConstIterator gdcm::SequenceOfItems::Begin ( ) const [inline]

10.264.4.5 void gdcm::SequenceOfItems::Clear ( ) [virtual]

remove all items within the sequence

Implements gdcm::Value.

10.264.4.6 template<typename TDE > VL gdcm::SequenceOfItems::ComputeLength ( ) const

10.264.4.7 Iterator gdcm::SequenceOfItems::End ( ) [inline]

10.264.4.8 ConstIterator gdcm::SequenceOfItems::End ( ) const [inline]

10.264.4.9 bool gdcm::SequenceOfItems::FindDataElement (const Tag & *t*) const

10.264.4.10 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 Item& gdcm::SequenceOfItems::GetItem (SizeType *position*)

10.264.4.12 VL gdcm::SequenceOfItems::GetLength ( ) const [inline], [virtual]

Returns the SQ length, as read from disk.

Implements gdcm::Value.

10.264.4.13 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 bool gdcm::SequenceOfItems::IsUndefinedLength ( ) const [inline]

return if Value Length if of undefined length

10.264.4.15 static SmartPointer<SequenceOfItems> gdcm::SequenceOfItems::New ( ) [inline], [static]

Examples:

NewSequence.cs.

10.264.4.16 SequenceOfItems& gdcm::SequenceOfItems::operator= (const SequenceOfItems & *val*) [inline]

References Items, and SequenceLengthField.

10.264.4.17 bool gdcm::SequenceOfItems::operator== (const Value & *val*) const [inline], [virtual]

Implements gdcm::Value.

References Items, and SequenceLengthField.

10.264.4.18 void gdcm::SequenceOfItems::Print (std::ostream & *os*) const [inline], [virtual]

Reimplemented from gdcm::Object.

10.264.4.19 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 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 void gdcm::SequenceOfItems::SetLength (VL *length*) [inline], [virtual]

Sets the actual SQ length.

Implements gdcm::Value.

Examples:

ReadExplicitLengthSQIVR.cxx.

10.264.4.22 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 void gdcm::SequenceOfItems::SetNumberOfItems (SizeType *n*) [inline]

10.264.4.24 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 ItemVector gdcm::SequenceOfItems::Items

Vector of Sequence Items.

Referenced by operator=(), and operator==().

10.264.5.2 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:

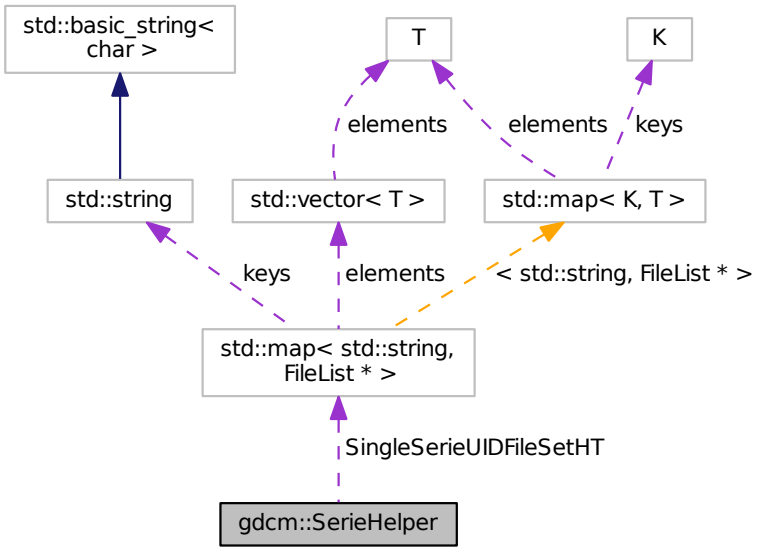
- gdcmSequenceOfItems.h

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 typedef std::vector<Rule> gdcm::SerieHelper::SerieRestrictions [protected]

10.265.2.2 typedef std::map<std::string, FileList *> gdcm::SerieHelper::SingleSerieUIDFileSetmap [protected]

10.265.3 Constructor & Destructor Documentation

10.265.3.1 gdcm::SerieHelper::SerieHelper ( )

10.265.3.2 gdcm::SerieHelper::~SerieHelper ( )

10.265.4 Member Function Documentation

10.265.4.1 bool gdcm::SerieHelper::AddFile (FileWithName & *header*) [protected]

10.265.4.2 void gdcm::SerieHelper::AddFileName (std::string const & *filename*) [protected]

10.265.4.3 void gdcm::SerieHelper::AddRestriction (const std::string & *tag*)

10.265.4.4 void gdcm::SerieHelper::AddRestriction (uint16_t *group*, uint16_t *elem*, std::string const & *value*, int *op*)

10.265.4.5 void gdcm::SerieHelper::AddRestriction (const Tag & *tag*) [protected]

10.265.4.6 void gdcm::SerieHelper::Clear ( )

10.265.4.7 void gdcm::SerieHelper::CreateDefaultUniqueSeriesIdentifier ( )

10.265.4.8 std::string gdcm::SerieHelper::CreateUniqueSeriesIdentifier (File * *inFile*)

10.265.4.9 bool gdcm::SerieHelper::FileNameOrdering (FileList * *fileList*) [protected]

10.265.4.10 FileList* gdcm::SerieHelper::GetFirstSingleSerieUIDFileSet ( )

10.265.4.11 FileList* gdcm::SerieHelper::GetNextSingleSerieUIDFileSet ( )

10.265.4.12 bool gdcm::SerieHelper::ImagePositionPatientOrdering (FileList * *fileSet*) [protected]

10.265.4.13 void gdcm::SerieHelper::OrderFileList (FileList * *fileSet*)

10.265.4.14 void gdcm::SerieHelper::SetDirectory (std::string const & *dir*, bool *recursive* = false)

10.265.4.15 void gdcm::SerieHelper::SetLoadMode (int) [inline]

10.265.4.16 void gdcm::SerieHelper::SetUseSeriesDetails (bool *useSeriesDetails*)

10.265.4.17 bool gdcm::SerieHelper::UserOrdering (FileList * *fileSet*) [protected]

10.265.5 Member Data Documentation

10.265.5.1 SingleSerieUIDFileSetmap::iterator gdcm::SerieHelper::ItFileSetHt [protected]

10.265.5.2 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 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 gdcm::network::ServiceClassApplicationInformation::ServiceClassApplicationInformation ( )

10.267.3 Member Function Documentation

10.267.3.1 void gdcm::network::ServiceClassApplicationInformation::Print (std::ostream & *os*) const

10.267.3.2 std::istream& gdcm::network::ServiceClassApplicationInformation::Read (std::istream & *is*)

10.267.3.3 void gdcm::network::ServiceClassApplicationInformation::SetTuple (uint8_t *levelofsupport,* uint8_t *levelofdigitalsig,* uint8_t *elementcoercion*)

10.267.3.4 size_t gdcm::network::ServiceClassApplicationInformation::Size ( ) const

10.267.3.5 const std::ostream& gdcm::network::ServiceClassApplicationInformation::Write (std::ostream & *os*) const

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

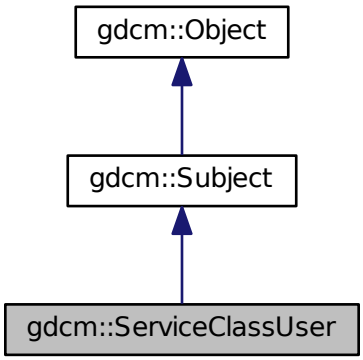
- gdcmServiceClassApplicationInformation.h

10.268 gdcm::ServiceClassUser Class Reference

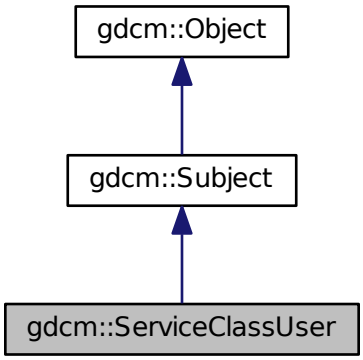
ServiceClassUser.

`#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 gdcm::ServiceClassUser::ServiceClassUser ( )

Construct a SCU with default:

- hostname = localhost
- port = 104

10.268.2.2 gdcm::ServiceClassUser::~ServiceClassUser ( )

10.268.3 Member Function Documentation

10.268.3.1 const char* gdcm::ServiceClassUser::GetAETitle ( ) const

10.268.3.2 const char* gdcm::ServiceClassUser::GetCalledAETitle ( ) const

10.268.3.3 double gdcm::ServiceClassUser::GetTimeout ( ) const

10.268.3.4 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 bool gdcm::ServiceClassUser::IsPresentationContextAccepted (const PresentationContext & *pc*) const

Return if the passed in presentation was accepted during association negotiation.

10.268.3.6 static SmartPointer<ServiceClassUser> gdcm::ServiceClassUser::New ( ) [inline], [static]

for wrapped language: instanciate a reference counted object

10.268.3.7 bool gdcm::ServiceClassUser::SendEcho ( )

C-ECHO.

10.268.3.8 bool gdcm::ServiceClassUser::SendFind (const BaseRootQuery * *query*, std::vector< DataSet > & *retDatasets*)

C-FIND a query, return result are in retDatasets.

10.268.3.9 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 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 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 bool gdcm::ServiceClassUser::SendStore (const char * *filename*)

Execute a C-STORE on file on disk, named filename.

Examples:

CStoreQtProgress.cxx.

10.268.3.13 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 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 void gdcm::ServiceClassUser::SetAETitle (const char * *aetitle*)

set calling ae title

10.268.3.16 void gdcm::ServiceClassUser::SetCalledAETitle (const char * *aetitle*)

set called ae title

Examples:

CStoreQtProgress.cxx.

10.268.3.17 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 void gdcm::ServiceClassUser::SetPort (uint16_t *port*)

Set port of remote host (called application)

Examples:

CStoreQtProgress.cxx.

10.268.3.19 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 void gdcm::ServiceClassUser::SetPresentationContexts (std::vector< **PresentationContext** > const & *pcs*)

Set the Presentation Context used for the Association.

Examples:

CStoreQtProgress.cxx.

10.268.3.21 void gdcm::ServiceClassUser::SetTimeout (double *t*)

set/get Timeout

Examples:

CStoreQtProgress.cxx.

10.268.3.22 bool gdcm::ServiceClassUser::StartAssociation ( )

Start the association. Need to call SetPresentationContexts before.

Examples:

CStoreQtProgress.cxx.

10.268.3.23 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 gdcm::SHA1::SHA1 ( )

10.269.2.2 gdcm::SHA1::~SHA1 ( )

10.269.3 Member Function Documentation

10.269.3.1 static bool gdcm::SHA1::Compute (const char * *buffer,* unsigned long *buf_len,* char *digest_str[20 *2+1]*) [static]

10.269.3.2 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:

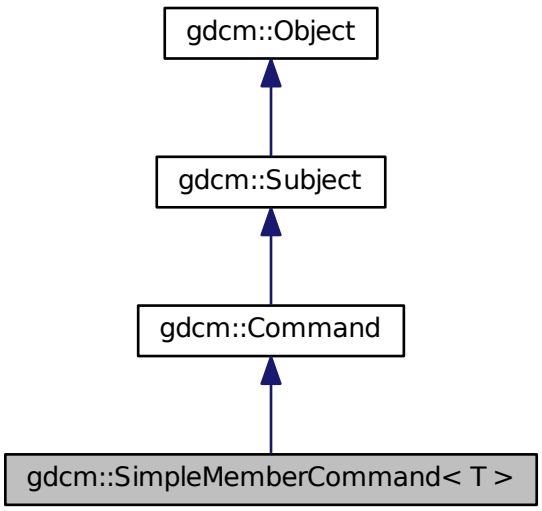
- gdcmSHA1.h

10.270 gdcm::SimpleMemberCommand< T > Class Template Reference

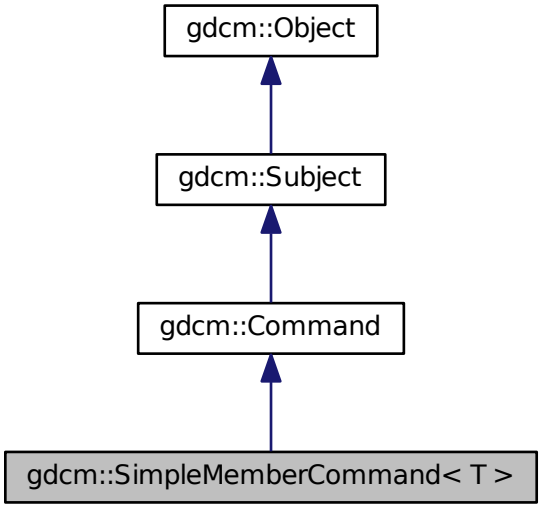
Command subclass that calls a pointer to a member function.

`#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 template<typename T > typedef SimpleMemberCommand gdcm::SimpleMemberCommand< T >::Self

Standard class typedefs.

10.270.2.2 template<typename T > typedef void(T::* gdcm::SimpleMemberCommand< T >::TMemberFunctionPointer) ()

A method callback.

10.270.3 Constructor & Destructor Documentation

10.270.3.1 template<typename T > gdcm::SimpleMemberCommand< T >::SimpleMemberCommand ( ) [inline], [protected]

10.270.3.2 template<typename T > virtual gdcm::SimpleMemberCommand< T >::~SimpleMemberCommand ( ) [inline], [protected], [virtual]

10.270.4 Member Function Documentation

10.270.4.1 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 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 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 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 template<typename T > TMemberFunctionPointer gdcm::SimpleMemberCommand< T >::m_MemberFunction [protected]

10.270.5.2 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 This is a typical Subject Watcher class. It will observe all events.

```
#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 gdcm::SimpleSubjectWatcher::SimpleSubjectWatcher (Subject * *s*, const char * *comment* = " ")

10.271.2.2 virtual gdcm::SimpleSubjectWatcher::~SimpleSubjectWatcher ( ) [virtual]

10.271.3 Member Function Documentation

10.271.3.1 virtual void gdcm::SimpleSubjectWatcher::EndFilter ( ) [protected], [virtual]

10.271.3.2 virtual void gdcm::SimpleSubjectWatcher::ShowAbort ( ) [protected], [virtual]

10.271.3.3 virtual void gdcm::SimpleSubjectWatcher::ShowAnonymization (Subject * *caller*, const Event & *evt*) [protected], [virtual]

10.271.3.4 virtual void gdcm::SimpleSubjectWatcher::ShowData (Subject * *caller*, const Event & *evt*) [protected], [virtual]

10.271.3.5 virtual void gdcm::SimpleSubjectWatcher::ShowDataSet (Subject * *caller*, const Event & *evt*) [protected], [virtual]

10.271.3.6 virtual void gdcm::SimpleSubjectWatcher::ShowFileName (Subject * *caller*, const Event & *evt*) [protected], [virtual]

Examples:

SimpleScanner.cxx.

10.271.3.7 virtual void gdcm::SimpleSubjectWatcher::ShowIteration ( ) [protected], [virtual]

10.271.3.8 virtual void gdcm::SimpleSubjectWatcher::ShowProgress (Subject * *caller,* const Event & *evt*) [protected], [virtual]

10.271.3.9 virtual void gdcm::SimpleSubjectWatcher::StartFilter ( ) [protected], [virtual]

10.271.3.10 void gdcm::SimpleSubjectWatcher::TestAbortOff ( ) [protected]

10.271.3.11 void gdcm::SimpleSubjectWatcher::TestAbortOn ( ) [protected]

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

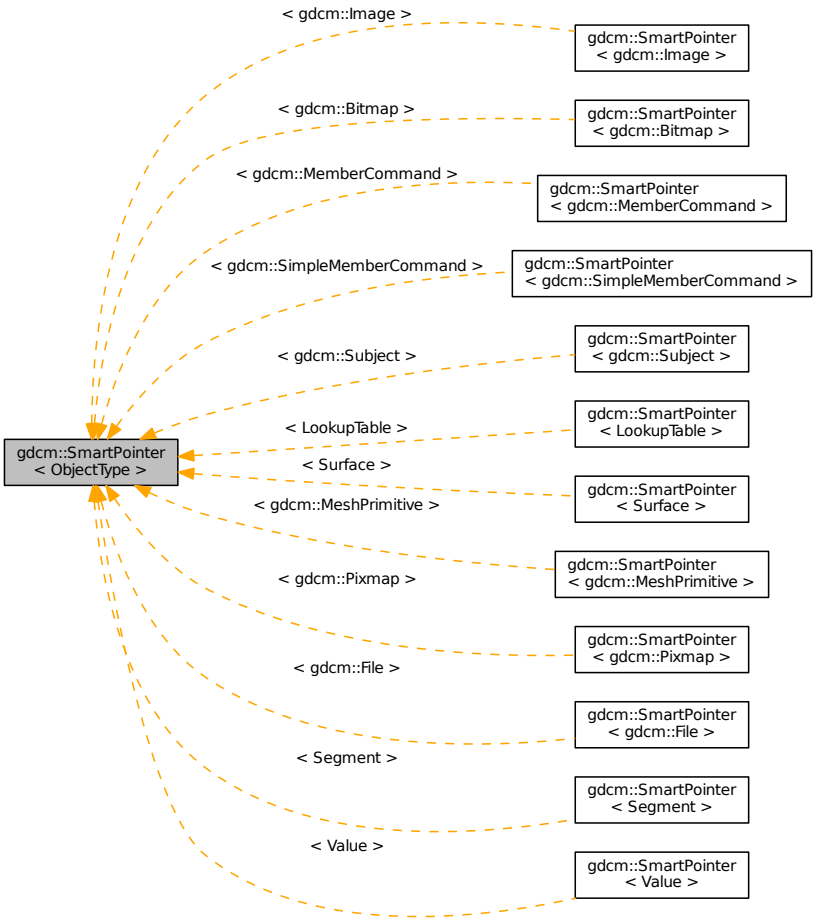
- gdcmSimpleSubjectWatcher.h

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 template<class ObjectType> gdcm::SmartPointer< ObjectType >::SmartPointer ( ) [inline]

10.272.2.2 template<class ObjectType> gdcm::SmartPointer< ObjectType >::SmartPointer (const SmartPointer< ObjectType > & *p*) [inline]

10.272.2.3 template<class ObjectType> gdcm::SmartPointer< ObjectType >::SmartPointer (ObjectType * *p*) [inline]

10.272.2.4 template<class ObjectType> gdcm::SmartPointer< ObjectType >::SmartPointer (ObjectType const & *p*) [inline]

10.272.2.5 template<class ObjectType> gdcm::SmartPointer< ObjectType >::~SmartPointer ( ) [inline]

10.272.3 Member Function Documentation

10.272.3.1 template<class ObjectType> ObjectType* gdcm::SmartPointer< ObjectType >::GetPointer ( ) const [inline]

Explicit function to retrieve the pointer.

10.272.3.2 template<class ObjectType> gdcm::SmartPointer< ObjectType >::operator ObjectType * ( ) const [inline]

Return pointer to object.

10.272.3.3 template<class ObjectType> ObjectType& gdcm::SmartPointer< ObjectType >::operator* ( ) const [inline]

10.272.3.4 template<class ObjectType> ObjectType* gdcm::SmartPointer< ObjectType >::operator-> ( ) const [inline]

Overload operator ->

10.272.3.5 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 template<class ObjectType> SmartPointer& gdcm::SmartPointer< ObjectType >::operator= (ObjectType * *r*) [inline]

Overload operator assignment.

10.272.3.7 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 PS 3.7 Table D.3-11 SOP CLASS EXTENDED NEGOTIATION SUB-ITEM FIELDS (A-ASSOCIATE-RQ and A-ASSOCIATE-AC)

```
#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 gdcm::network::SOPClassExtendedNegociationSub::SOPClassExtendedNegociationSub ( )

10.273.3 Member Function Documentation

10.273.3.1 void gdcm::network::SOPClassExtendedNegociationSub::Print (std::ostream & *os*) const

10.273.3.2 std::istream& gdcm::network::SOPClassExtendedNegociationSub::Read (std::istream & *is*)

10.273.3.3 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_t gdcm::network::SOPClassExtendedNegociationSub::Size ( ) const

10.273.3.5 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 typedef const char* gdcm::SOPClassUIDToIOD::const(SOPClassUIDToIODType)[2]

10.274.3 Member Function Documentation

10.274.3.1 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 static const char* gdcm::SOPClassUIDToIOD::GetIODFromSOPClassUID (const char * *sopclassuid*) [static]

10.274.3.3 static unsigned int gdcm::SOPClassUIDToIOD::GetNumberOfSOPClassToIOD ( ) [static]

Return the number of SOP Class UID listed internally.

10.274.3.4 static const char* gdcm::SOPClassUIDToIOD::GetSOPClassUIDFromIOD (const char * *iod*) [static]

10.274.3.5 static SOPClassUIDToIODType& gdcm::SOPClassUIDToIOD::GetSOPClassUIDToIOD (unsigned int *i*) [static]

10.274.3.6 static SOPClassUIDToIODType* gdcm::SOPClassUIDToIOD::GetSOPClassUIDToIODs ( ) [static]

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

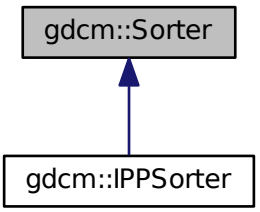
- gdcmSOPClassUIDToIOD.h

10.275 gdcm::Sorter Class Reference

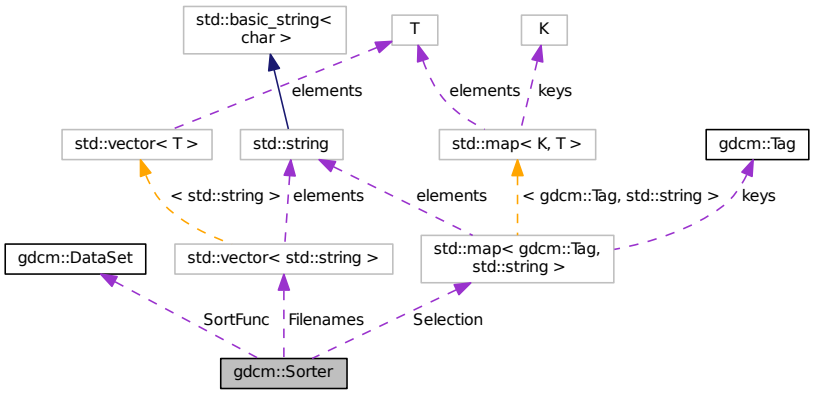
Sorter General class to do sorting using a custom function You simply need to provide a function of type: Sorter::SortFunction.

`#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::Sort↩ Function.

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 typedef std::map<Tag,std::string> gdcm::Sorter::SelectionMap [protected]

10.275.2.2 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 gdcm::Sorter::Sorter ( )

10.275.3.2 virtual gdcm::Sorter::~Sorter ( ) [virtual]

10.275.4 Member Function Documentation

10.275.4.1 bool gdcm::Sorter::AddSelect (Tag const & *tag,* const char * *value*)

UNSUPPORTED FOR NOW.

10.275.4.2 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 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 void gdcm::Sorter::SetSortFunction (SortFunction *f*)

Examples:

SortImage.cxx, and VolumeSorter.cxx.

10.275.4.5 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 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 std::ostream& operator<< (std::ostream & *_os,* const Sorter & *s*) [friend]

10.275.6 Member Data Documentation

10.275.6.1 std::vector<std::string> gdcm::Sorter::Filenames [protected]

10.275.6.2 std::map<Tag,std::string> gdcm::Sorter::Selection [protected]

10.275.6.3 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 enum gdcm::Spacing::SpacingType

Enumerator

DETECTOR

MAGNIFIED

CALIBRATED

UNKNOWN

10.276.3 Constructor & Destructor Documentation

10.276.3.1 gdcm::Spacing::Spacing ( )

10.276.3.2 gdcm::Spacing::~Spacing ( )

10.276.4 Member Function Documentation

10.276.4.1 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 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 Class to reshuffle bytes for a SIEMENS Mosaic image Siemens CSA Image Header CSA:= Common Siemens Architecture, sometimes also known as Common syngo Architecture.

`#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 gdcm::SplitMosaicFilter::SplitMosaicFilter ( )

10.278.2.2 gdcm::SplitMosaicFilter::~SplitMosaicFilter ( )

10.278.3 Member Function Documentation

10.278.3.1 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 File& gdcm::SplitMosaicFilter::GetFile ( ) [inline]

10.278.3.3 const File& gdcm::SplitMosaicFilter::GetFile ( ) const [inline]

10.278.3.4 const Image& gdcm::SplitMosaicFilter::GetImage ( ) const [inline]

10.278.3.5 Image& gdcm::SplitMosaicFilter::GetImage ( ) [inline]

10.278.3.6 void gdcm::SplitMosaicFilter::SetFile (const File & *f*) [inline]

10.278.3.7 void gdcm::SplitMosaicFilter::SetImage (const Image & *image*)

10.278.3.8 bool gdcm::SplitMosaicFilter::Split ( )

Split the SIEMENS MOSAIC image.

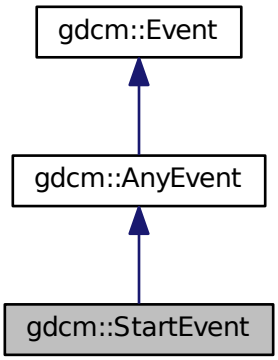
The documentation for this class was generated from the following file:

- gdcmSplitMosaicFilter.h

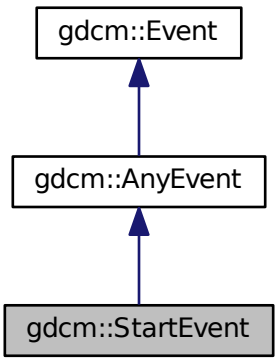
10.279 gdcm::StartEvent Class Reference

`#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

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 gdcm::StreamImageReader::StreamImageReader ( )

10.283.2.2 virtual gdcm::StreamImageReader::~StreamImageReader ( ) [virtual]

10.283.3 Member Function Documentation

10.283.3.1 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 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 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 std::vector<unsigned int> gdcm::StreamImageReader::GetDimensionsValueForResolution (unsigned *int*)

10.283.3.5 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 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 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 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 void gdcm::StreamImageReader::SetStream (std::istream & *inStream*)

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

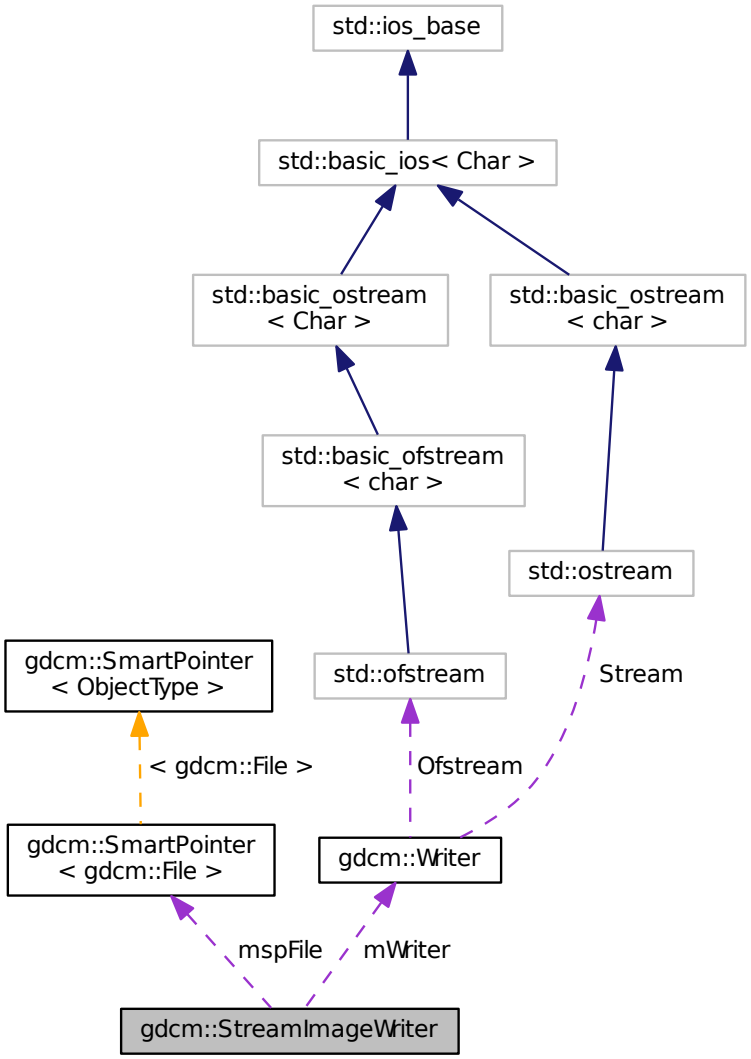
- gdcmStreamImageReader.h

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 gdcm::StreamImageWriter::StreamImageWriter ( )

10.284.2.2 virtual gdcm::StreamImageWriter::~StreamImageWriter ( ) [virtual]

10.284.3 Member Function Documentation

10.284.3.1 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 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 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 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 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 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 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 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 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 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 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 int gdcm::StreamImageWriter::mElementOffsets1 [protected]

10.284.4.3 SmartPointer<File> gdcm::StreamImageWriter::mspFile [protected]

10.284.4.4 Writer gdcm::StreamImageWriter::mWriter [protected]

10.284.4.5 uint16_t gdcm::StreamImageWriter::mXMax [protected]

10.284.4.6 uint16_t gdcm::StreamImageWriter::mXMin [protected]

10.284.4.7 uint16_t gdcm::StreamImageWriter::mYMax [protected]

10.284.4.8 uint16_t gdcm::StreamImageWriter::mYMin [protected]

10.284.4.9 uint16_t gdcm::StreamImageWriter::mZMax [protected]

10.284.4.10 uint16_t gdcm::StreamImageWriter::mZMin [protected]

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

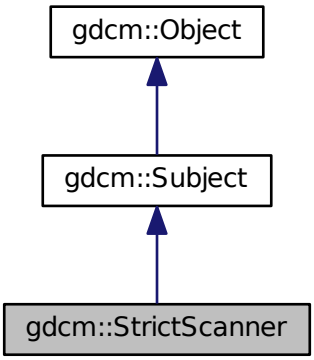
- gdcmStreamImageWriter.h

10.285 gdcm::StrictScanner Class Reference

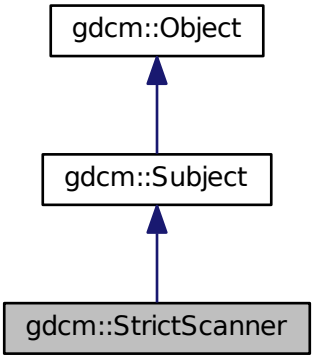
StrictScanner This filter is meant for quickly browsing a FileSet (a set of files on disk). Special consideration are taken so as to read the mimimum amount of information in each file in order to retrieve the user specified set of DICOM Attribute.

`#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 typedef MappingType::const_iterator gdcm::StrictScanner::ConstIterator

10.285.2.2 typedef std::map<const char *,TagToValue, ltstr> gdcm::StrictScanner::MappingType

10.285.2.3 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 typedef TagToValue::value_type gdcm::StrictScanner::TagToValueValueType

10.285.2.5 typedef std::set< std::string > gdcm::StrictScanner::ValuesType

10.285.3 Constructor & Destructor Documentation

10.285.3.1 gdcm::StrictScanner::StrictScanner ( ) [inline]

10.285.3.2 gdcm::StrictScanner::~StrictScanner ( )

10.285.4 Member Function Documentation

10.285.4.1 void gdcm::StrictScanner::AddPrivateTag (PrivateTag const & *t*)

10.285.4.2 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 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 ConstIterator gdcm::StrictScanner::Begin ( ) const [inline]

10.285.4.5 void gdcm::StrictScanner::ClearSkipTags ( )

10.285.4.6 void gdcm::StrictScanner::ClearTags ( )

10.285.4.7 ConstIterator gdcm::StrictScanner::End ( ) const [inline]

10.285.4.8 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 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 Directory::FilenamesType const& gdcm::StrictScanner::GetFilenames ( ) const [inline]

10.285.4.11 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 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 TagToValue const& gdcm::StrictScanner::GetMappingFromTagToValue (Tag const & *t*, const char * *value*) const

See GetFilenameFromTagToValue(). This is simply GetFilenameFromTagToValue followed.

10.285.4.14 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 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 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 ValuesType const& gdcm::StrictScanner::GetValues ( ) const [inline]

Get all the values found (in lexicographic order)

10.285.4.18 ValuesType gdcm::StrictScanner::GetValues (Tag const & *t*) const

Get all the values found (in lexicographic order) associated with Tag 't'.

10.285.4.19 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 static SmartPointer<StrictScanner> gdcm::StrictScanner::New ( ) [inline], [static]

for wrapped language: instanciate a reference counted object

10.285.4.21 void gdcm::StrictScanner::Print (std::ostream & *os*) const [virtual]

Print result.

Reimplemented from gdcm::Object.

Referenced by gdcm::operator<<().

10.285.4.22 void gdcm::StrictScanner::ProcessPublicTag (StringFilter & *sf*, const char * *filename*) [protected]

10.285.4.23 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 std::ostream& operator<< (std::ostream & *_os*, const StrictScanner & *s*) [friend]

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

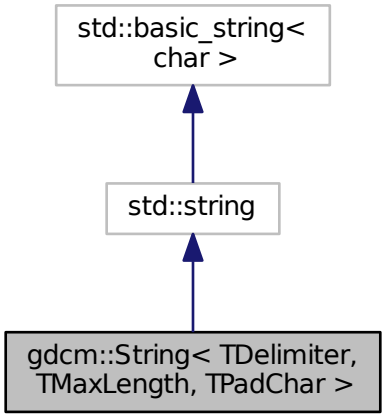
- gdcmStrictScanner.h

10.286 gdcm::String< TDelimiter, TMaxLength, TPadChar > Class Template Reference

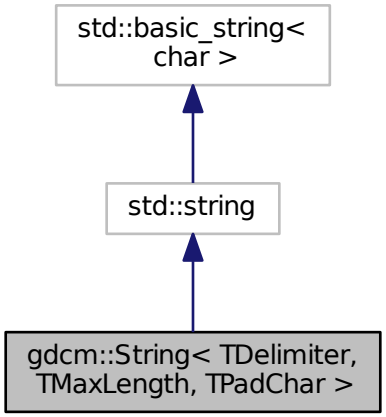
String.

`#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 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 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 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 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 template<char TDelimiter = '\\', unsigned int TMaxLength = 64, char TPadChar = ' '> typedef std::string::iterator gdcm::String< TDelimiter, TMaxLength, TPadChar >::iterator

10.286.2.6 template<char TDelimiter = '\\', unsigned int TMaxLength = 64, char TPadChar = ' '> typedef std::string::pointer gdcm::String< TDelimiter, TMaxLength, TPadChar >::pointer

10.286.2.7 template<char TDelimiter = '\\', unsigned int TMaxLength = 64, char TPadChar = ' '> typedef std::string::reference gdcm::String< TDelimiter, TMaxLength, TPadChar >::reference

10.286.2.8 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 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 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 template<char TDelimiter = '\\', unsigned int TMaxLength = 64, char TPadChar = ' '> gdcm::String< TDelimiter, TMaxLength, TPadChar >::String ( ) [inline]

String constructors.

10.286.3.2 template<char TDelimiter = '\\', unsigned int TMaxLength = 64, char TPadChar = ' '> gdcm::String< TDelimiter, TMaxLength, TPadChar >::String (const value_type * *s*) [inline]

10.286.3.3 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 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 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 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 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 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 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 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.

`#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 gdcm::StringFilter::StringFilter ( )

10.287.2.2 gdcm::StringFilter::~StringFilter ( )

10.287.3 Member Function Documentation

10.287.3.1 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 bool gdcm::StringFilter::ExecuteQuery (std::string const & *query*, DataSet const & *ds*, std::string & *value*) const [protected]

10.287.3.3 std::string gdcm::StringFilter::FromString (const Tag & *t*, const char * *value*, VL const & *vl*)

10.287.3.4 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 File& gdcm::StringFilter::GetFile ( ) [inline]

10.287.3.6 const File& gdcm::StringFilter::GetFile ( ) const [inline]

10.287.3.7 void gdcm::StringFilter::SetDicts (const Dicts & *dicts*)

Allow user to pass in there own dicts.

10.287.3.8 void gdcm::StringFilter::SetFile (const File & *f*) [inline]

Set/Get File.

Examples:

ReadAndPrintAttributes.cxx, and SimplePrintPatientName.cs.

10.287.3.9 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 std::string gdcm::StringFilter::ToString (const Tag & *t*) const

Directly from a Tag:

10.287.3.11 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 std::pair<std::string, std::string> gdcm::StringFilter::ToStringPair (const Tag & *t*) const

Directly from a Tag:

10.287.3.13 std::pair<std::string, std::string> gdcm::StringFilter::ToStringPair (const Tag & *t*, DataSet const & *ds*) const [protected]

10.287.3.14 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 gdcm::Study::Study ( ) [inline]

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

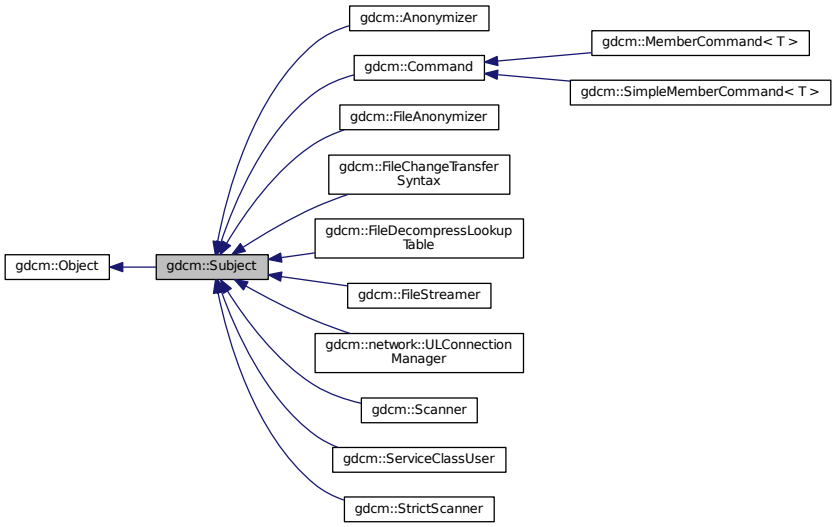
- gdcmStudy.h

10.289 gdcm::Subject Class Reference

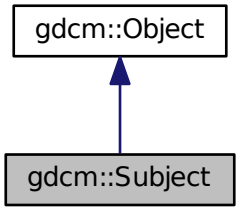
Subject.

```
#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 gdcm::Subject::Subject ( )

10.289.2.2 gdcm::Subject::~Subject ( )

10.289.3 Member Function Documentation

10.289.3.1 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 unsigned long gdcm::Subject::AddObserver (const Event & *event,* Command *) const

10.289.3.3 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 bool gdcm::Subject::HasObserver (const Event & *event*) const

Return true if an observer is registered for this event.

10.289.3.5 void gdcm::Subject::InvokeEvent (const Event &)

Call Execute on all the Commands observing this event id.

10.289.3.6 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 void gdcm::Subject::RemoveAllObservers ( )

Remove all observers .

10.289.3.8 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:

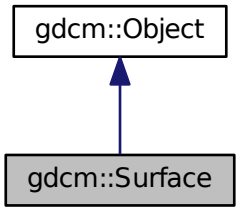
- gdcmSubject.h

10.290 gdcm::Surface Class Reference

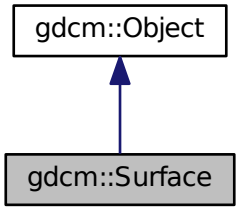
This class defines a SURFACE IE. This members are taken from required surface mesh module attributes.

`#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 enum gdcm::Surface::STATES

Enumerator

NO
YES
UNKNOWN
STATES_END

10.290.2.2 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 gdcm::Surface::Surface ( )

10.290.3.2 virtual gdcm::Surface::~Surface ( ) [virtual]

10.290.4 Member Function Documentation

10.290.4.1 SegmentHelper::BasicCodedEntry const& gdcm::Surface::GetAlgorithmFamily ( ) const

10.290.4.2 SegmentHelper::BasicCodedEntry& gdcm::Surface::GetAlgorithmFamily ( )

10.290.4.3 const char* gdcm::Surface::GetAlgorithmName ( ) const

10.290.4.4 const char* gdcm::Surface::GetAlgorithmVersion ( ) const

10.290.4.5 const float* gdcm::Surface::GetAxisOfRotation ( ) const

Note

Pointer is null if undefined

10.290.4.6 const float* gdcm::Surface::GetCenterOfRotation ( ) const

Note

Pointer is null if undefined

10.290.4.7 STATES gdcm::Surface::GetFiniteVolume ( ) const

10.290.4.8 STATES gdcm::Surface::GetManifold ( ) const

10.290.4.9 float gdcm::Surface::GetMaximumPointDistance ( ) const

10.290.4.10 float gdcm::Surface::GetMeanPointDistance ( ) const

10.290.4.11 MeshPrimitive const& gdcm::Surface::GetMeshPrimitive ( ) const

10.290.4.12 MeshPrimitive& gdcm::Surface::GetMeshPrimitive ( )

10.290.4.13 unsigned long gdcm::Surface::GetNumberOfSurfacePoints ( ) const

10.290.4.14 unsigned long gdcm::Surface::GetNumberOfVectors ( ) const

10.290.4.15 const DataElement& gdcm::Surface::GetPointCoordinatesData ( ) const

10.290.4.16 DataElement& gdcm::Surface::GetPointCoordinatesData ( )

10.290.4.17 const float* gdcm::Surface::GetPointPositionAccuracy ( ) const

Note

Pointer is null if undefined

10.290.4.18 const float* gdcm::Surface::GetPointsBoundingBoxCoordinates ( ) const

Note

Pointer is null if undefined

10.290.4.19 **SegmentHelper::BasicCodedEntry** const& gdcm::Surface::GetProcessingAlgorithm ( ) const

10.290.4.20 **SegmentHelper::BasicCodedEntry**& gdcm::Surface::GetProcessingAlgorithm ( )

10.290.4.21 const unsigned short* gdcm::Surface::GetRecommendedDisplayCIELabValue ( ) const

10.290.4.22 unsigned short gdcm::Surface::GetRecommendedDisplayCIELabValue (const unsigned int *idx*) const

10.290.4.23 unsigned short gdcm::Surface::GetRecommendedDisplayGrayscaleValue ( ) const

10.290.4.24 float gdcm::Surface::GetRecommendedPresentationOpacity ( ) const

10.290.4.25 **VIEWType** gdcm::Surface::GetRecommendedPresentationType ( ) const

10.290.4.26 static **STATES** gdcm::Surface::GetSTATES (const char * *state*) [static]

10.290.4.27 static const char* gdcm::Surface::GetSTATESString (**STATES** *state*) [static]

10.290.4.28 const char* gdcm::Surface::GetSurfaceComments ( ) const

10.290.4.29 unsigned long gdcm::Surface::GetSurfaceNumber ( ) const

10.290.4.30 bool gdcm::Surface::GetSurfaceProcessing ( ) const

10.290.4.31 const char* gdcm::Surface::GetSurfaceProcessingDescription ( ) const

10.290.4.32 float gdcm::Surface::GetSurfaceProcessingRatio ( ) const

10.290.4.33 const float* gdcm::Surface::GetVectorAccuracy ( ) const

10.290.4.34 const **DataElement**& gdcm::Surface::GetVectorCoordinateData ( ) const

10.290.4.35 **DataElement**& gdcm::Surface::GetVectorCoordinateData ( )

10.290.4.36 unsigned short gdcm::Surface::GetVectorDimensionality ( ) const

10.290.4.37 static **VIEWType** gdcm::Surface::GetVIEWType (const char * *type*) [static]

10.290.4.38 static const char* gdcm::Surface::GetVIEWTypeString (**VIEWType** *type*) [static]

10.290.4.39 void gdcm::Surface::SetAlgorithmFamily (**SegmentHelper::BasicCodedEntry** const & *BSE*)

10.290.4.40 void gdcm::Surface::SetAlgorithmName (const char * *str*)

10.290.4.41 void gdcm::Surface::SetAlgorithmVersion (const char * *str*)

10.290.4.42 void gdcm::Surface::SetAxisOfRotation (const float * *axis*)

10.290.4.43 void gdcm::Surface::SetCenterOfRotation (const float * *center*)

10.290.4.44 void gdcm::Surface::SetFiniteVolume (**STATES** *state*)

10.290.4.45 void gdcm::Surface::SetManifold (**STATES** *state*)

10.290.4.46 void gdcm::Surface::SetMaximumPointDistance (float *maximum*)

10.290.4.47 void gdcm::Surface::SetMeanPointDistance (float *average*)

10.290.4.48 void gdcm::Surface::SetMeshPrimitive (**MeshPrimitive** & *mp*)

10.290.4.49 void gdcm::Surface::SetNumberOfSurfacePoints (const unsigned long *nb*)

10.290.4.50 void gdcm::Surface::SetNumberOfVectors (const unsigned long *nb*)

10.290.4.51 void gdcm::Surface::SetPointCoordinatesData (**DataElement** const & *de*)

10.290.4.52 void gdcm::Surface::SetPointPositionAccuracy (const float * *accuracies*)

10.290.4.53 void gdcm::Surface::SetPointsBoundingBoxCoordinates (const float * *coordinates*)

10.290.4.54 void gdcm::Surface::SetProcessingAlgorithm (**SegmentHelper::BasicCodedEntry** const & *BSE*)

10.290.4.55 void gdcm::Surface::SetRecommendedDisplayCIELabValue (const unsigned short *vl[3]*)

10.290.4.56 void gdcm::Surface::SetRecommendedDisplayCIELabValue (const unsigned short *vl,* const unsigned int *idx* = 0)

10.290.4.57 void gdcm::Surface::SetRecommendedDisplayCIELabValue (const std::vector< unsigned short > & *vl*)

10.290.4.58 void gdcm::Surface::SetRecommendedDisplayGrayscaleValue (const unsigned short *vl*)

10.290.4.59 void gdcm::Surface::SetRecommendedPresentationOpacity (const float *opacity*)

10.290.4.60 void gdcm::Surface::SetRecommendedPresentationType (**VIEWType** *type*)

10.290.4.61 void gdcm::Surface::SetSurfaceComments (const char * *comment*)

10.290.4.62 void gdcm::Surface::SetSurfaceNumber (const unsigned long *nb*)

10.290.4.63 void gdcm::Surface::SetSurfaceProcessing (bool *b*)

10.290.4.64 void gdcm::Surface::SetSurfaceProcessingDescription (const char * *description*)

10.290.4.65 void gdcm::Surface::SetSurfaceProcessingRatio (const float *ratio*)

10.290.4.66 void gdcm::Surface::SetVectorAccuracy (const float * *accuracy*)

10.290.4.67 void gdcm::Surface::SetVectorCoordinateData (**DataElement** const & *de*)

10.290.4.68 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 Helper class for Surface object.

`#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 range↩ Max=255)

  Convert a DICOM CIE-Lab (after reading) color into RGB.
- template<typename U >
  static std::vector< float > RecommendedDisplayCIELabToRGB (const ColorArray &CIELab, const U range↩ Max=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 range↩ Max=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 typedef std::vector< unsigned short > gdcm::SurfaceHelper::ColorArray

10.291.3 Member Function Documentation

10.291.3.1 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 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 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 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:

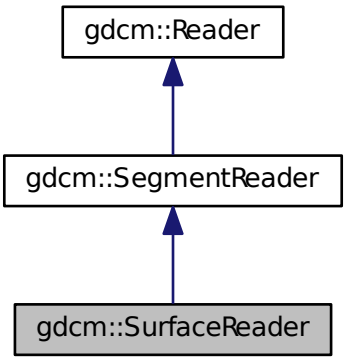
- gdcmSurfaceHelper.h

10.292 gdcm::SurfaceReader Class Reference

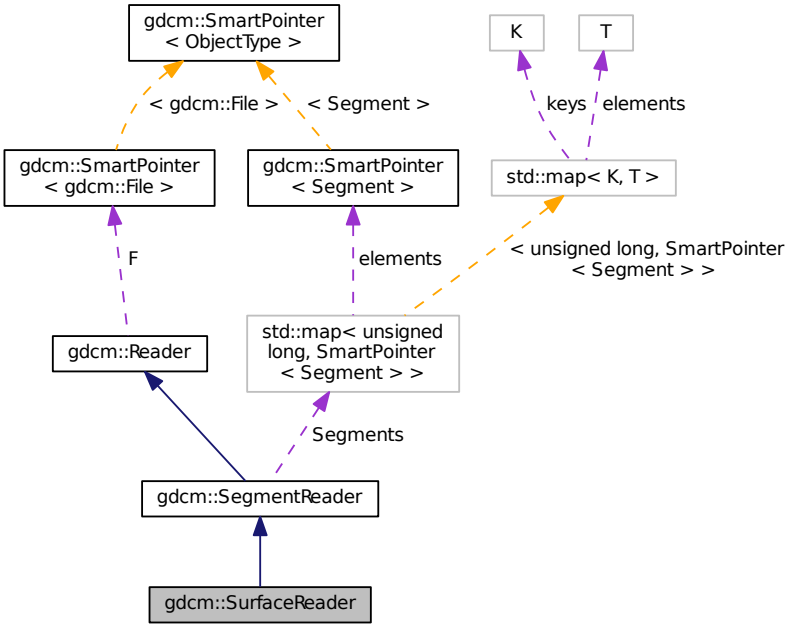
This class defines a SURFACE IE reader. It reads surface mesh module attributes.

```
#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 gdcm::SurfaceReader::SurfaceReader ( )

10.292.2.2 virtual gdcm::SurfaceReader::~SurfaceReader ( ) [virtual]

10.292.3 Member Function Documentation

10.292.3.1 unsigned long gdcm::SurfaceReader::GetNumberOfSurfaces ( ) const

10.292.3.2 virtual bool gdcm::SurfaceReader::Read ( ) [virtual]

Read.

Reimplemented from gdcm::SegmentReader.

10.292.3.3 bool gdcm::SurfaceReader::ReadPointMacro (SmartPointer< Surface > *surface,* const DataSet & *surfaceDS*) [protected]

10.292.3.4 bool gdcm::SurfaceReader::ReadSurface (const Item & *surfaceItem,* const unsigned long *idx*) [protected]

10.292.3.5 bool gdcm::SurfaceReader::ReadSurfaces ( ) [protected]

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

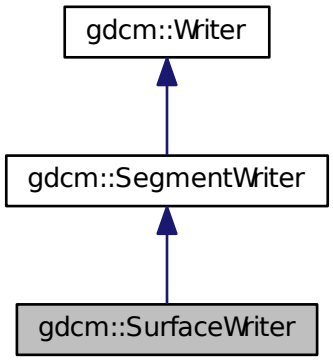
- gdcmSurfaceReader.h

10.293 gdcm::SurfaceWriter Class Reference

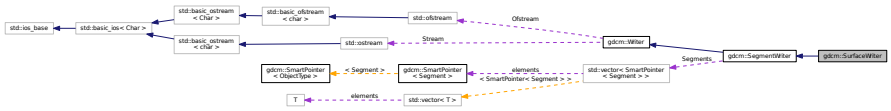
This class defines a SURFACE IE writer. It writes surface mesh module attributes.

`#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 gdcm::SurfaceWriter::SurfaceWriter ( )

10.293.2.2 virtual gdcm::SurfaceWriter::~SurfaceWriter ( ) [virtual]

10.293.3 Member Function Documentation

10.293.3.1 void gdcm::SurfaceWriter::ComputeNumberOfSurfaces ( ) [protected]

10.293.3.2 unsigned long gdcm::SurfaceWriter::GetNumberOfSurfaces ( )

10.293.3.3 bool gdcm::SurfaceWriter::PrepareWrite ( ) [protected]

10.293.3.4 bool gdcm::SurfaceWriter::PrepareWritePointMacro (SmartPointer< Surface > *surface*, DataSet & *surfaceDS*, const TransferSyntax & *ts*) [protected]

10.293.3.5 void gdcm::SurfaceWriter::SetNumberOfSurfaces (const unsigned long *nb*)

10.293.3.6 bool gdcm::SurfaceWriter::Write ( ) [virtual]

Write.

Reimplemented from gdcm::SegmentWriter.

10.293.4 Member Data Documentation

10.293.4.1 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 enum gdcm::SwapCode::SwapCodeType

Enumerator

Unknown

LittleEndian

BigEndian

BadLittleEndian

BadBigEndian

10.294.3 Constructor & Destructor Documentation

10.294.3.1 gdcm::SwapCode::SwapCode (SwapCodeType *sc* = Unknown) [inline]

References gdcm::operator<<().

10.294.4 Member Function Documentation

10.294.4.1 static int gdcm::SwapCode::GetIndex (SwapCode const & *sc*) [static], [protected]

10.294.4.2 static const char* gdcm::SwapCode::GetSwapCodeString (SwapCode const & *sc*) [static]

Referenced by gdcm::operator<<().

10.294.4.3 gdcm::SwapCode::operator SwapCode::SwapCodeType ( ) const [inline]

10.294.5 Friends And Related Function Documentation

10.294.5.1 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 template<typename T > static T gdcm::SwapperDoOp::Swap (T *val*) [static]

10.295.1.2 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 template<typename T > static T gdcm::SwapperNoOp::Swap (T *val*) [inline], [static]

10.296.2.2 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 static bool gdcm::System::DeleteDirectory (const char * *source*) [static]

remove a directory named source

10.297.2.2 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 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 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 static bool gdcm::System::FileIsSymlink (const char * *name*) [static]

Check whether name is a symlink.

10.297.2.6 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 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 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 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 static const char* gdcm::System::GetCurrentModuleFileName ( ) [static]

Return the directory the current module is located: NOT THREAD SAFE

10.297.2.11 static const char* gdcm::System::GetCurrentProcessFileName ( ) [static]

Return the directory the current process (executable) is located: NOT THREAD SAFE

10.297.2.12 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 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 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 static const char* gdcm::System::GetLastSystemError ( ) [static]

Return the last error.

10.297.2.16 static const char* gdcm::System::GetLocaleCharset ( ) [static]

return `locale charmap`

10.297.2.17 static bool gdcm::System::GetPermissions (const char * *file,* unsigned short & *mode*) [static], [protected]

NOT THREAD SAFE.

10.297.2.18 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 static bool gdcm::System::MakeDirectory (const char * *path*) [static]

Create a directory name path.

10.297.2.20 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 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 static bool gdcm::System::RemoveFile (const char * *source*) [static]

remove a file named source

10.297.2.23 static bool gdcm::System::SetPermissions (const char * *file,* unsigned short *mode*) [static], [protected]

10.297.2.24 static int gdcm::System::StrCaseCmp (const char * *s1,* const char * *s2*) [static]

consistent func for C99 spec of strcasecmp/strncasecmp

10.297.2.25 static int gdcm::System::StrNCaseCmp (const char * *s1,* const char * *s2,* size_t *n*) [static]

Precondition

n != 0

10.297.2.26 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 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 typedef std::map<Tag, TableEntry> gdcm::Table::MapTableEntry

10.298.3 Constructor & Destructor Documentation

10.298.3.1 gdcm::Table::Table ( ) [inline]

Referenced by GetTableEntry().

10.298.3.2 gdcm::Table::~Table ( ) [inline]

References operator<<.

10.298.4 Member Function Documentation

10.298.4.1 const TableEntry& gdcm::Table::GetTableEntry (const Tag & *tag*) const [inline]

References Table().

10.298.4.2 void gdcm::Table::InsertEntry (Tag const & *tag,* TableEntry const & *te*) [inline]

10.298.5 Friends And Related Function Documentation

10.298.5.1 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 gdcm::TableEntry::TableEntry (const char * *attribute* = 0, Type const & *type* = Type (), const char * *des* = 0) [inline]

10.299.2.2 gdcm::TableEntry::~TableEntry ( ) [inline]

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

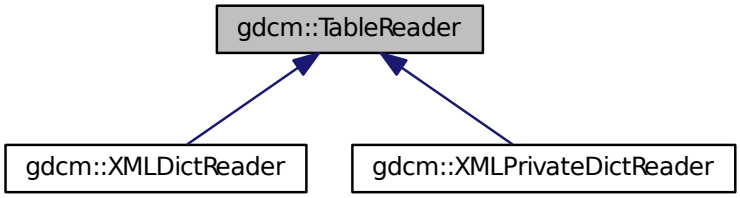
- gdcmTableEntry.h

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 gdcm::TableReader::TableReader (Defs & *defs*) [inline]

10.300.2.2 virtual gdcm::TableReader::~TableReader ( ) [inline], [virtual]

10.300.3 Member Function Documentation

10.300.3.1 virtual void gdcm::TableReader::CharacterDataHandler (const char * *data*, int *length*) [virtual]

Reimplemented in gdcm::XMLDictReader, and gdcm::XMLPrivateDictReader.

10.300.3.2 virtual void gdcm::TableReader::EndElement (const char * *name*) [virtual]

Reimplemented in gdcm::XMLDictReader, and gdcm::XMLPrivateDictReader.

10.300.3.3 const Defs& gdcm::TableReader::GetDefs ( ) const [inline]

10.300.3.4 const char* gdcm::TableReader::GetFilename ( ) [inline]

10.300.3.5 void gdcm::TableReader::HandleIOD (const char ** *atts*)

10.300.3.6 void gdcm::TableReader::HandleIODEntry (const char ** *atts*)

10.300.3.7 void gdcm::TableReader::HandleMacro (const char ** *atts*)

10.300.3.8 void gdcm::TableReader::HandleMacroEntry (const char ** *atts*)

10.300.3.9 void gdcm::TableReader::HandleMacroEntryDescription (const char ** *atts*)

10.300.3.10 void gdcm::TableReader::HandleModule (const char ** *atts*)

10.300.3.11 void gdcm::TableReader::HandleModuleEntry (const char ** *atts*)

10.300.3.12 void gdcm::TableReader::HandleModuleEntryDescription (const char ** *atts*)

10.300.3.13 void gdcm::TableReader::HandleModuleInclude (const char ** *atts*)

10.300.3.14 int gdcm::TableReader::Read ( )

10.300.3.15 void gdcm::TableReader::SetFilename (const char * *filename*) [inline]

10.300.3.16 virtual void gdcm::TableReader::StartElement (const char * *name*, const char ** *atts*) [virtual]

Reimplemented in gdcm::XMLDictReader, and gdcm::XMLPrivateDictReader.

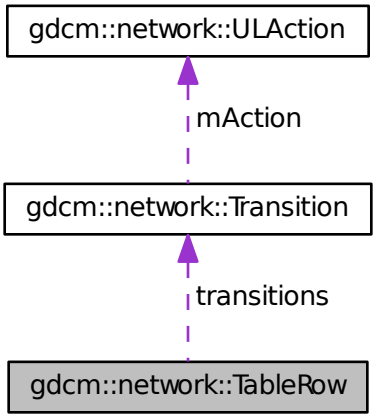
The documentation for this class was generated from the following file:

- 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 gdcm::network::TableRow::TableRow ( ) [inline]

References gdcm::network::cMaxStateID.

10.301.1.2 gdcm::network::TableRow::~TableRow ( ) [inline]

References gdcm::network::cMaxStateID.

10.301.2 Member Data Documentation

10.301.2.1 Transition* gdcm::network::TableRow::transitions[cMaxStateID]

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

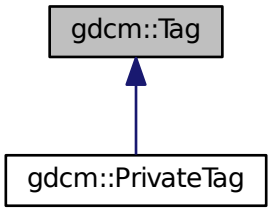
- gdcmULTransitionTable.h

10.302 gdcm::Tag Class Reference

Class to represent a DICOM Data Element (Attribute) Tag (Group, Element). Basically an uint32_t which can also be expressed as two uint16_t (group and 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 gdcm::Tag::Tag (uint16_t *group*, uint16_t *element*) [inline]

Constructor with 2∗uint16_t.

10.302.2.2 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 gdcm::Tag::Tag (const Tag & *_val*) [inline]

References tag.

10.302.3 Member Function Documentation

10.302.3.1 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::PrivateTag::PrivateTag(), gdcm::SequenceOfFragments::ReadValue(), and SetPrivateCreator().

10.302.3.2 uint32_t gdcm::Tag::GetElementTag ( ) const [inline]

Returns the full tag value of the given Tag.

10.302.3.3 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 uint32_t gdcm::Tag::GetLength ( ) const [inline]

return the length of tag (read: size on disk)

10.302.3.5 Tag gdcm::Tag::GetPrivateCreator ( ) const [inline]

Return the Private Creator Data Element tag of a private data element.

References SetElement().

10.302.3.6 bool gdcm::Tag::IsGroupLength ( ) const [inline]

return whether the tag correspond to a group length tag:

10.302.3.7 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 bool gdcm::Tag::IsIllegal ( ) const [inline]

return if the tag is considered to be an illegal tag

10.302.3.9 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 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 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 bool gdcm::Tag::operator!= (const Tag & *_val*) const [inline]

References tag.

10.302.3.13 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 bool gdcm::Tag::operator<= (const Tag & *t2*) const [inline]

10.302.3.15 Tag& gdcm::Tag::operator= (const Tag & *_val*) [inline]

References tag.

10.302.3.16 bool gdcm::Tag::operator== (const Tag & *_val*) const [inline]

References tag.

10.302.3.17 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 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 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 std::string gdcm::Tag::PrintAsContinuousUpperCaseString ( ) const

Same as PrintAsContinuousString, but hexadecimal [a-f] are printed using upper case.

10.302.3.21 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 template<typename TSwap > std::istream& gdcm::Tag::Read (std::istream & *is*) [inline]

Read a tag from binary representation.

10.302.3.23 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 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 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 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 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 void gdcm::Tag::SetElementTag (uint32_t *tag*) [inline]

Sets the full tag value of the given Tag.

10.302.3.29 void gdcm::Tag::SetGroup (uint16_t *group*) [inline]

Sets the 'Group number' of the given Tag.

Referenced by gdcm::operator>>().

10.302.3.30 void gdcm::Tag::SetPrivateCreator (Tag const & *t*) [inline]

Set private creator:

Examples:

DuplicatePCDE.cxx.

References GetElement(), GetGroup(), and IsPrivate().

10.302.3.31 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 std::ostream& operator<< (std::ostream & *_os,* const Tag & *_val*) [friend]

10.302.4.2 std::istream& operator>> (std::istream & *_is,* Tag & *_val*) [friend]

10.302.5 Member Data Documentation

10.302.5.1 char gdcm::Tag::bytes[4]

10.302.5.2 uint32_t gdcm::Tag::tag

Referenced by operator!=(), operator<(), operator=(), operator==(), and Tag().

10.302.5.3 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 gdcm::TagPath::TagPath ( )

10.303.2.2 gdcm::TagPath::~TagPath ( )

10.303.3 Member Function Documentation

10.303.3.1 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 bool gdcm::TagPath::ConstructFromTagList (Tag const * *l*, unsigned int *n*)

Construct from a list of tags.

10.303.3.3 static bool gdcm::TagPath::IsValid (const char * *path*) [static]

Return if path is valid or not.

10.303.3.4 void gdcm::TagPath::Print (std::ostream &) const

10.303.3.5 bool gdcm::TagPath::Push (Tag const & *t*)

10.303.3.6 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 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 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 gdcm::Testing::Testing ( ) [inline]

10.304.3.2 gdcm::Testing::~Testing ( ) [inline]

10.304.4 Member Function Documentation

10.304.4.1 static bool gdcm::Testing::ComputeFileMD5 (const char * *filename,* char *digest_str[33]*) [static]

Examples:

MetaImageMD5Activiz.cs.

10.304.4.2 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 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 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 static const char* gdcm::Testing::GetFileName (unsigned int *file*) [static]

Examples:

MetaImageMD5Activiz.cs.

10.304.4.6 static const char* const* gdcm::Testing::GetFileNames ( ) [static]

return the table of fullpath to gdcmData DICOM files:

Examples:

TestReader.cxx.

10.304.4.7 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 static const char* const* gdcm::Testing::GetMD5DataImage (unsigned int *file*) [static]

10.304.4.9 static **MD5DataImagesType** gdcm::Testing::GetMD5DataImages ( ) [static]

10.304.4.10 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 static const char* gdcm::Testing::GetMD5FromFile (const char * *filepath*) [static]

10.304.4.12 static const char* const* gdcm::Testing::GetMediaStorageDataFile (unsigned int *file*) [static]

10.304.4.13 static **MediaStorageDataFilesType** gdcm::Testing::GetMediaStorageDataFiles ( ) [static]

10.304.4.14 static const char* gdcm::Testing::GetMediaStorageFromFile (const char * *filepath*) [static]

Examples:

MetaImageMD5Activiz.cs, and TestReader.cxx.

10.304.4.15 static unsigned int gdcm::Testing::GetNumberOfFileNames ( ) [static]

Examples:

MetaImageMD5Activiz.cs.

10.304.4.16 static unsigned int gdcm::Testing::GetNumberOfMD5DataImages ( ) [static]

10.304.4.17 static unsigned int gdcm::Testing::GetNumberOfMediaStorageDataFiles ( ) [static]

10.304.4.18 static const char* gdcm::Testing::GetPixelSpacingDataRoot ( ) [static]

Return the GDCM PIXEL SPACING DATA ROOT (See David Clunie website for dataset)

10.304.4.19 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 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 static const char* gdcm::Testing::GetSourceDirectory ( ) [static]

Examples:

BasicAnonymizer.cs, and ClinicalTrialIdentificationWorkflow.cs.

10.304.4.22 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 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 static const wchar_t* gdcm::Testing::GetTempDirectoryW (const wchar_t * *subdir* = 0) [static]

NOT THREAD SAFE.

10.304.4.25 static const char* gdcm::Testing::GetTempFilename (const char * *filename,* const char * *subdir* = `0`) [static]

NOT THREAD SAFE.

Examples:

MetaImageMD5Activiz.cs.

10.304.4.26 static const wchar_t* gdcm::Testing::GetTempFilenameW (const wchar_t * *filename,* const wchar_t * *subdir* = `0`) [static]

NOT THREAD SAFE.

10.304.4.27 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 gdcm::Trace::Trace ( )

10.305.2.2 gdcm::Trace::~Trace ( )

10.305.3 Member Function Documentation

10.305.3.1 static void gdcm::Trace::DebugOff ( ) [static]

Examples:

MetaImageMD5Activiz.cs, and TestReader.cxx.

10.305.3.2 static void gdcm::Trace::DebugOn ( ) [static]

Examples:

CreateFakePET.cxx, Fake_Image_Using_Stream_Image_Writer.cxx, and StreamImageReaderTest.cxx.

10.305.3.3 static void gdcm::Trace::ErrorOff ( ) [static]

Examples:

MetaImageMD5Activiz.cs.

10.305.3.4 static void gdcm::Trace::ErrorOn ( ) [static]

10.305.3.5 static bool gdcm::Trace::GetDebugFlag ( ) [static]

10.305.3.6 static std::ostream& gdcm::Trace::GetDebugStream ( ) [static]

10.305.3.7 static bool gdcm::Trace::GetErrorFlag ( ) [static]

10.305.3.8 static std::ostream& gdcm::Trace::GetErrorStream ( ) [static]

10.305.3.9 static std::ostream& gdcm::Trace::GetStream ( ) [static]

10.305.3.10 static bool gdcm::Trace::GetWarningFlag ( ) [static]

10.305.3.11 static std::ostream& gdcm::Trace::GetWarningStream ( ) [static]

10.305.3.12 static void gdcm::Trace::SetDebug (bool *debug*) [static]

Turn debug messages on (default: false)

Examples:

DumpToSQLITE3.cxx.

10.305.3.13 static void gdcm::Trace::SetDebugStream (std::ostream & *os*) [static]

Explicitly set the stream which receive Debug messages:

10.305.3.14 static void gdcm::Trace::SetError (bool *debug*) [static]

Turn error messages on (default: true)

10.305.3.15 static void gdcm::Trace::SetErrorStream (std::ostream & *os*) [static]

Explicitly set the stream which receive Error messages:

Examples:

CStoreQtProgress.cxx.

10.305.3.16 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 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 static void gdcm::Trace::SetWarning (bool *debug*) [static]

Turn warning messages on (default: true)

Examples:

DumpToSQLITE3.cxx.

10.305.3.19 static void gdcm::Trace::SetWarningStream (std::ostream & *os*) [static]

Explicitly set the stream which receive Warning messages:

10.305.3.20 static void gdcm::Trace::WarningOff ( ) [static]

Examples:

MetaImageMD5Activiz.cs, and TestReader.cxx.

10.305.3.21 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 enum gdcm::TransferSyntax::NegociatedType

Enumerator

Unknown

Explicit

Implicit

10.306.2.2 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 gdcm::TransferSyntax::TransferSyntax (TSType *type* = ImplicitVRLittleEndian) [inline]

10.306.4 Member Function Documentation

10.306.4.1 bool gdcm::TransferSyntax::CanStoreLossy ( ) const

return true if TransFer Syntax Allow storing of Lossy Pixel Data

10.306.4.2 NegociatedType gdcm::TransferSyntax::GetNegociatedType ( ) const

10.306.4.3 const char* gdcm::TransferSyntax::GetString ( ) const [inline]

References GetTSString(), and gdcm::operator<<().

10.306.4.4 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 static const char* gdcm::TransferSyntax::GetTSString (TSType *ts*) [static]

Examples:

LargeVRDSExplicit.cxx.

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

10.306.4.6 static TSType gdcm::TransferSyntax::GetTSType (const char * *str*) [static]

10.306.4.7 bool gdcm::TransferSyntax::IsEncapsulated ( ) const

Examples:

ExtractIconFromFile.cxx.

10.306.4.8 bool gdcm::TransferSyntax::IsEncoded ( ) const

10.306.4.9 bool gdcm::TransferSyntax::IsExplicit ( ) const

10.306.4.10 bool gdcm::TransferSyntax::IsImplicit ( ) const

10.306.4.11 bool gdcm::TransferSyntax::IsLossless ( ) const

Return true if the transfer syntax algorithm is a lossless algorithm

10.306.4.12 bool gdcm::TransferSyntax::IsLossy ( ) const

Return true if the transfer syntax algorithm is a lossy algorithm

10.306.4.13 bool gdcm::TransferSyntax::IsValid ( ) const [inline]

10.306.4.14 gdcm::TransferSyntax::operator TSType ( ) const [inline]

10.306.5 Friends And Related Function Documentation

10.306.5.1 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 Table 9-15 TRANSFER SYNTAX SUB-ITEM FIELDS.

`#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 gdcm::network::TransferSyntaxSub::TransferSyntaxSub ( )

10.307.3 Member Function Documentation

10.307.3.1 const char* gdcm::network::TransferSyntaxSub::GetName ( ) const [inline]

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

10.307.3.2 bool gdcm::network::TransferSyntaxSub::operator== (const TransferSyntaxSub & *ts*) const [inline]

10.307.3.3 void gdcm::network::TransferSyntaxSub::Print (std::ostream & *os*) const

Referenced by GetName().

10.307.3.4 std::istream& gdcm::network::TransferSyntaxSub::Read (std::istream & *is*)

Referenced by GetName().

10.307.3.5 void gdcm::network::TransferSyntaxSub::SetName (const char * *name*)

10.307.3.6 void gdcm::network::TransferSyntaxSub::SetNameFromUID (UIDs::TSName *tsname*)

Referenced by GetName().

10.307.3.7 size_t gdcm::network::TransferSyntaxSub::Size ( ) const

Referenced by GetName().

10.307.3.8 const std::ostream& gdcm::network::TransferSyntaxSub::Write (std::ostream & *os*) const

Referenced by GetName().

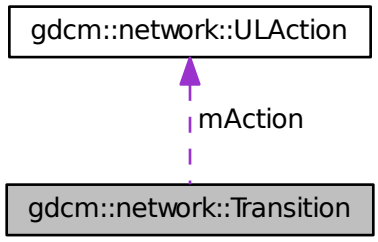
The documentation for this class was generated from the following file:

- 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 gdcm::network::Transition::Transition ( ) [inline]

References gdcm::network::eStaDoesNotExist.

Referenced by MakeNew().

10.308.1.2 gdcm::network::Transition::~Transition ( ) [inline]

References mAction.

10.308.1.3 gdcm::network::Transition::Transition (int *inEndState,* ULAction * *inAction*) [inline]

10.308.2 Member Function Documentation

10.308.2.1 static Transition* gdcm::network::Transition::MakeNew (int *inEndState,* ULAction * *inAction*) [inline], [static]

References Transition().

10.308.3 Member Data Documentation

10.308.3.1 ULAction* gdcm::network::Transition::mAction

Referenced by ~Transition().

10.308.3.2 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 ELEME↩ NTS 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 enum gdcm::Type::TypeType

Enumerator

T1

T1C

T2

T2C

T3

UNKNOWN

10.309.3 Constructor & Destructor Documentation

10.309.3.1 gdcm::Type::Type (TypeType *type* = UNKNOWN) [inline]

10.309.4 Member Function Documentation

10.309.4.1 static const char* gdcm::Type::GetTypeString (TypeType *type*) [static]

Referenced by gdcm::operator<<().

10.309.4.2 static TypeType gdcm::Type::GetTypeType (const char * *type*) [static]

Referenced by gdcm::ModuleEntry::ModuleEntry().

10.309.4.3 gdcm::Type::operator TypeType ( ) const [inline]

References gdcm::operator<<().

10.309.5 Friends And Related Function Documentation

10.309.5.1 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 std::ostream& operator<< (std::ostream & *_os,* const UI & *_val*) [friend]

10.310.2 Member Data Documentation

10.310.2.1 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 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 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 static bool gdcm::UIDGenerator::GenerateUUID (unsigned char * *uuid_data*) [static], [protected]

10.311.3.3 static const char* gdcm::UIDGenerator::GetGDCMUID ( ) [static]

Return the default (GDCM) root UID:

10.311.3.4 static const char* gdcm::UIDGenerator::GetRoot ( ) [static]

Examples:

ClinicalTrialIdentificationWorkflow.cs, ReformatFile.cs, and StandardizeFiles.cs.

10.311.3.5 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 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,

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_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 typedef const char* const(* gdcm::UIDs::TransferSyntaxStringsType)[2]

10.312.3 Member Enumeration Documentation

10.312.3.1 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
JPEGLosslessNonHierarchicalFirstOrderPredictionProcess14SelectionValue1DefaultTransferSyntaxforLosslessJPEGImag

JPEGLSLosslessImageCompression
JPEGLSLossyNearLosslessImageCompression
JPEG2000ImageCompressionLosslessOnly
JPEG2000ImageCompression
JPEG2000Part2MulticomponentImageCompressionLosslessOnly
JPEG2000Part2MulticomponentImageCompression
JPIPReferenced
JPIPReferencedDeflate
MPEG2MainProfileMainLevel
RLELossless
RFC2557MIMEencapsulation
XMLEncoding
MediaStorageDirectoryStorage
TalairachBrainAtlasFrameofReference
SPM2T1FrameofReference
SPM2T2FrameofReference
SPM2PDFrameofReference
SPM2EPIFrameofReference
SPM2FILT1FrameofReference
SPM2PETFrameofReference
SPM2TRANSMFrameofReference
SPM2SPECTFrameofReference
SPM2GRAYFrameofReference
SPM2WHITEFrameofReference
SPM2CSFFrameofReference
SPM2BRAINMASKFrameofReference
SPM2AVG305T1FrameofReference

SPM2AVG152T1FrameofReference

SPM2AVG152T2FrameofReference

SPM2AVG152PDFrameofReference

SPM2SINGLESUBJT1FrameofReference

ICBM452T1FrameofReference

ICBMSingleSubjectMRIFrameofReference

BasicStudyContentNotificationSOPClassRetired

StorageCommitmentPushModelSOPClass

StorageCommitmentPushModelSOPInstance

StorageCommitmentPullModelSOPClassRetired

StorageCommitmentPullModelSOPInstanceRetired

ProceduralEventLoggingSOPClass

ProceduralEventLoggingSOPInstance

SubstanceAdministrationLoggingSOPClass

SubstanceAdministrationLoggingSOPInstance

DICOMUIDRegistry

DICOMControlledTerminology

DICOMApplicationContextName

DetachedPatientManagementSOPClassRetired

DetachedPatientManagementMetaSOPClassRetired

DetachedVisitManagementSOPClassRetired

DetachedStudyManagementSOPClassRetired

StudyComponentManagementSOPClassRetired

ModalityPerformedProcedureStepSOPClass

ModalityPerformedProcedureStepRetrieveSOPClass

ModalityPerformedProcedureStepNotificationSOPClass

DetachedResultsManagementSOPClassRetired

DetachedResultsManagementMetaSOPClassRetired

DetachedStudyManagementMetaSOPClassRetired

DetachedInterpretationManagementSOPClassRetired

StorageServiceClass

BasicFilmSessionSOPClass

BasicFilmBoxSOPClass

BasicGrayscaleImageBoxSOPClass

BasicColorImageBoxSOPClass

ReferencedImageBoxSOPClassRetired

BasicGrayscalePrintManagementMetaSOPClass

ReferencedGrayscalePrintManagementMetaSOPClassRetired

PrintJobSOPClass

BasicAnnotationBoxSOPClass

PrinterSOPClass

PrinterConfigurationRetrievalSOPClass

PrinterSOPInstance
PrinterConfigurationRetrievalSOPInstance
BasicColorPrintManagementMetaSOPClass
ReferencedColorPrintManagementMetaSOPClassRetired
VOILUTBoxSOPClass
PresentationLUTSOPClass
ImageOverlayBoxSOPClassRetired
BasicPrintImageOverlayBoxSOPClassRetired
PrintQueueSOPInstanceRetired
PrintQueueManagementSOPClassRetired
StoredPrintStorageSOPClassRetired
HardcopyGrayscaleImageStorageSOPClassRetired
HardcopyColorImageStorageSOPClassRetired
PullPrintRequestSOPClassRetired
PullStoredPrintManagementMetaSOPClassRetired
MediaCreationManagementSOPClassUID
ComputedRadiographyImageStorage
DigitalXRayImageStorageForPresentation
DigitalXRayImageStorageForProcessing
DigitalMammographyXRayImageStorageForPresentation
DigitalMammographyXRayImageStorageForProcessing
DigitalIntraoralXRayImageStorageForPresentation
DigitalIntraoralXRayImageStorageForProcessing
CTImageStorage
EnhancedCTImageStorage
UltrasoundMultiframeImageStorageRetired
UltrasoundMultiframeImageStorage
MRImageStorage
EnhancedMRImageStorage
MRSpectroscopyStorage
NuclearMedicineImageStorageRetired
UltrasoundImageStorageRetired
UltrasoundImageStorage
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

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

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
dicomIssuerOfPatientID
dicomPreferredCallingAETitle
dicomSupportedCharacterSet
dicomConfigurationRoot

dicomDevicesRoot

dicomUniqueAETitlesRegistryRoot

dicomDevice

dicomNetworkAE

dicomNetworkConnection

dicomUniqueAETitle

dicomTransferCapability

VLWholeSlideMicroscopyImageStorage

EnhancedUSVolumeStorage

SurfaceSegmentationStorage

BreastTomosynthesisImageStorage

10.312.3.2 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

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

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
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
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
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

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 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 static unsigned int gdcm::UIDs::GetNumberOfTransferSyntaxStrings ( ) [static]

10.312.4.3 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 static const char* const* gdcm::UIDs::GetTransferSyntaxString (unsigned int *ts*) [static]

10.312.4.5 static TransferSyntaxStringsType gdcm::UIDs::GetTransferSyntaxStrings ( ) [static]

10.312.4.6 static const char* gdcm::UIDs::GetUIDName (unsigned int *ts*) [static]

10.312.4.7 static const char* gdcm::UIDs::GetUIDString (unsigned int *ts*) [static]

10.312.4.8 gdcm::UIDs::operator TSType ( ) const [inline]

10.312.4.9 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:

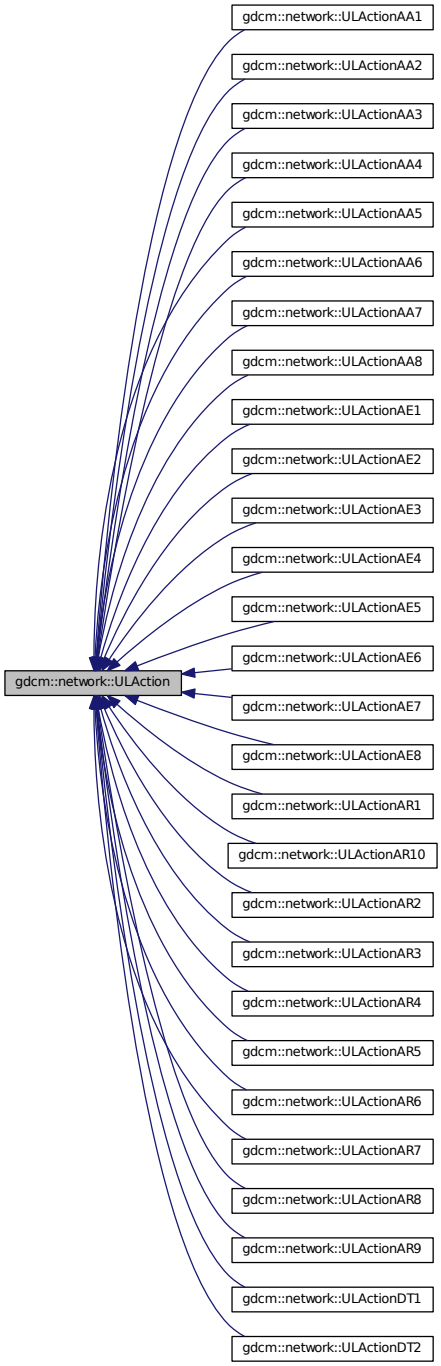
- gdcmUIDs.h

10.313 gdcm::network::ULAction Class Reference

ULAction A ULConnection in a given ULState can perform certain ULActions. This base class provides the interface for running those ULActions on a given ULConnection.

```
#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 gdcm::network::ULAction::ULAction ( ) [inline]

10.313.2.2 virtual gdcm::network::ULAction::~ULAction ( ) [inline], [virtual]

References PerformAction().

10.313.3 Member Function Documentation

10.313.3.1 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().

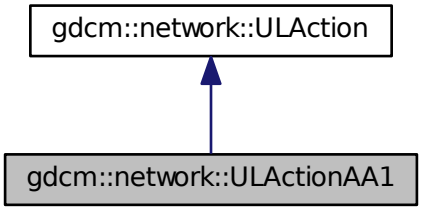
The documentation for this class was generated from the following file:

- gdcmULAction.h

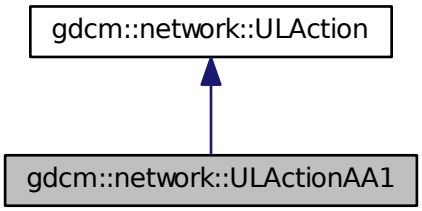
10.314 gdcm::network::ULActionAA1 Class Reference

`#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 EStateID gdcm::network::ULActionAA1::PerformAction (Subject * *s*, ULEvent & *inEvent*, ULConnection & *inConnection*, bool & *outWaitingForEvent*, EEventID & *outRaisedEvent*) [virtual]

Implements gdcm::network::ULAction.

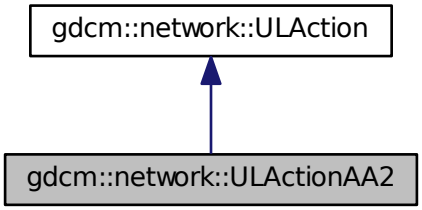
The documentation for this class was generated from the following file:

- gdcmULActionAA.h

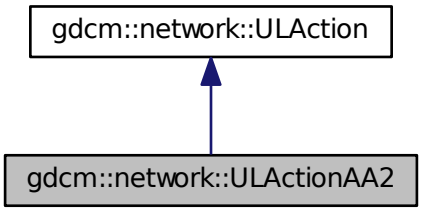
10.315 gdcm::network::ULActionAA2 Class Reference

`#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 EStateID gdcm::network::ULActionAA2::PerformAction (Subject * *s*, ULEvent & *inEvent*, ULConnection & *inConnection*, bool & *outWaitingForEvent*, EEventID & *outRaisedEvent*) [virtual]

Implements gdcm::network::ULAction.

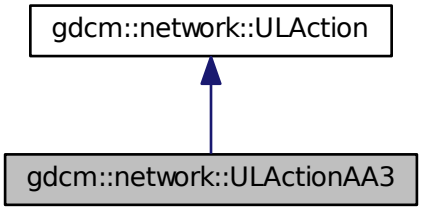
The documentation for this class was generated from the following file:

- gdcmULActionAA.h

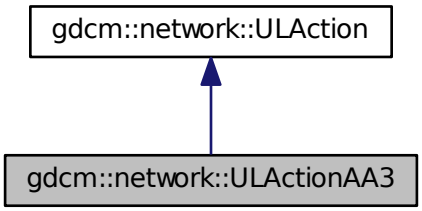
10.316 gdcm::network::ULActionAA3 Class Reference

`#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 &outWaitingFor↩ Event, EEventID &outRaisedEvent)

10.316.1 Member Function Documentation

10.316.1.1 EStateID gdcm::network::ULActionAA3::PerformAction (Subject * *s*, ULEvent & *inEvent*, ULConnection & *inConnection*, bool & *outWaitingForEvent*, EEventID & *outRaisedEvent*) [virtual]

Implements gdcm::network::ULAction.

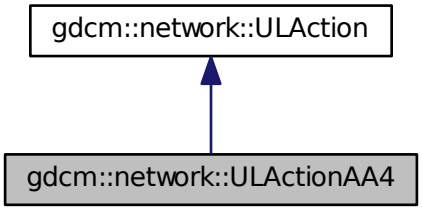
The documentation for this class was generated from the following file:

- gdcmULActionAA.h

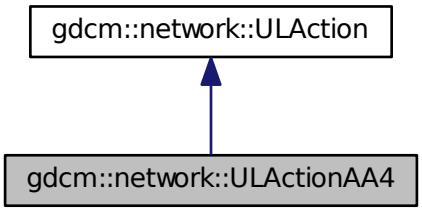
10.317 gdcm::network::ULActionAA4 Class Reference

`#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 EStateID gdcm::network::ULActionAA4::PerformAction (Subject * *s*, ULEvent & *inEvent*, ULConnection & *inConnection*, bool & *outWaitingForEvent*, EEventID & *outRaisedEvent*) [virtual]

Implements gdcm::network::ULAction.

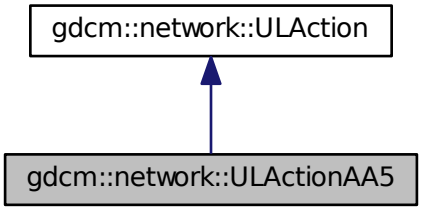
The documentation for this class was generated from the following file:

- gdcmULActionAA.h

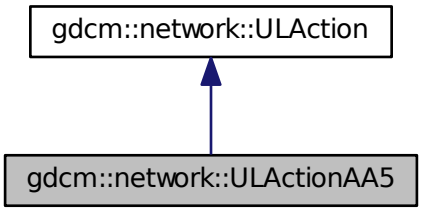
10.318 gdcm::network::ULActionAA5 Class Reference

`#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 EStateID gdcm::network::ULActionAA5::PerformAction (Subject * *s*, ULEvent & *inEvent*, ULConnection & *inConnection*, bool & *outWaitingForEvent*, EEventID & *outRaisedEvent*) [virtual]

Implements gdcm::network::ULAction.

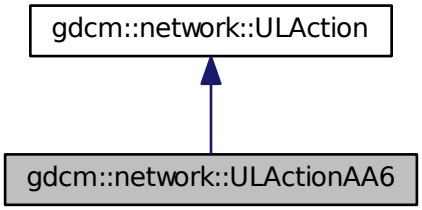
The documentation for this class was generated from the following file:

- gdcmULActionAA.h

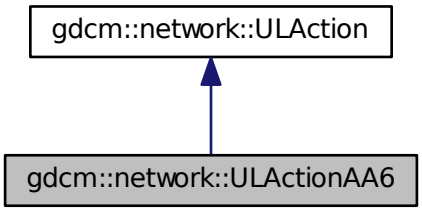
10.319 gdcm::network::ULActionAA6 Class Reference

`#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 EStateID gdcm::network::ULActionAA6::PerformAction (Subject * *s*, ULEvent & *inEvent*, ULConnection & *inConnection*, bool & *outWaitingForEvent*, EEventID & *outRaisedEvent*) [virtual]

Implements gdcm::network::ULAction.

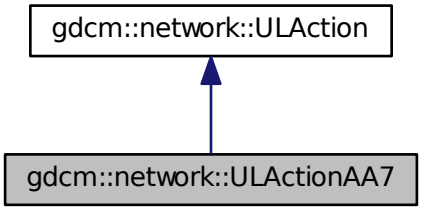
The documentation for this class was generated from the following file:

- gdcmULActionAA.h

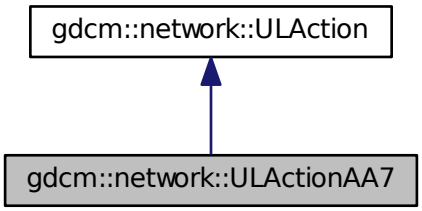
10.320 gdcm::network::ULActionAA7 Class Reference

`#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 EStateID gdcm::network::ULActionAA7::PerformAction (Subject * *s*, ULEvent & *inEvent*, ULConnection & *inConnection*, bool & *outWaitingForEvent*, EEventID & *outRaisedEvent*) [virtual]

Implements gdcm::network::ULAction.

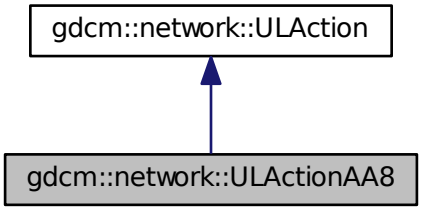
The documentation for this class was generated from the following file:

- gdcmULActionAA.h

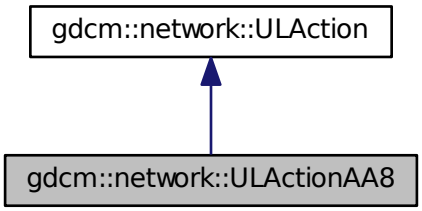
10.321 gdcm::network::ULActionAA8 Class Reference

`#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 &outWaitingFor↩ Event, EEventID &outRaisedEvent)

10.321.1 Member Function Documentation

10.321.1.1 EStateID gdcm::network::ULActionAA8::PerformAction (Subject * *s*, ULEvent & *inEvent*, ULConnection & *inConnection*, bool & *outWaitingForEvent*, EEventID & *outRaisedEvent*) [virtual]

Implements gdcm::network::ULAction.

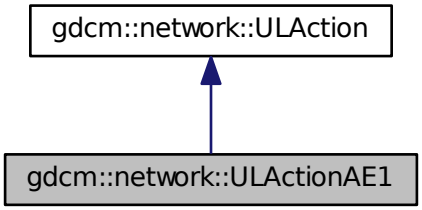
The documentation for this class was generated from the following file:

- gdcmULActionAA.h

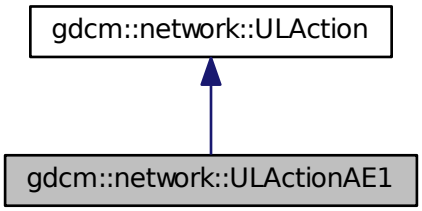
10.322 gdcm::network::ULActionAE1 Class Reference

`#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 EStateID gdcm::network::ULActionAE1::PerformAction (Subject ∗ *s,* ULEvent & *inEvent,* ULConnection & *inConnection,* bool & *outWaitingForEvent,* EEventID & *outRaisedEvent*) [virtual]

Implements gdcm::network::ULAction.

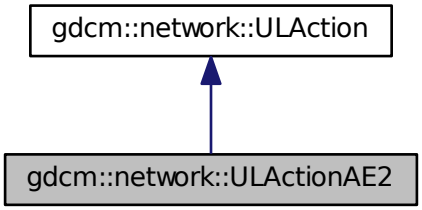
The documentation for this class was generated from the following file:

- gdcmULActionAE.h

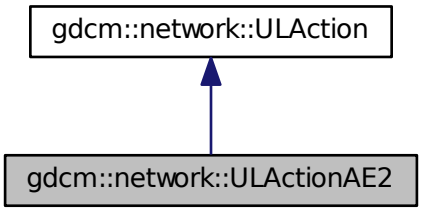
10.323 gdcm::network::ULActionAE2 Class Reference

`#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 EStateID gdcm::network::ULActionAE2::PerformAction (Subject * *s*, ULEvent & *inEvent*, ULConnection & *inConnection*, bool & *outWaitingForEvent*, EEventID & *outRaisedEvent*) [virtual]

Implements gdcm::network::ULAction.

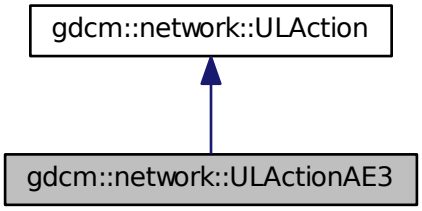
The documentation for this class was generated from the following file:

- gdcmULActionAE.h

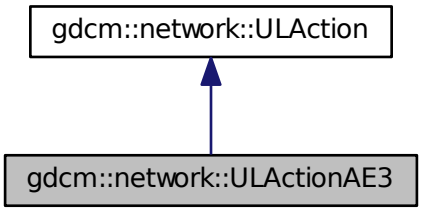
10.324 gdcm::network::ULActionAE3 Class Reference

`#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 &outWaitingFor↩ Event, EEventID &outRaisedEvent)

10.324.1 Member Function Documentation

10.324.1.1 EStateID gdcm::network::ULActionAE3::PerformAction (Subject ∗ *s,* ULEvent & *inEvent,* ULConnection & *inConnection,* bool & *outWaitingForEvent,* EEventID & *outRaisedEvent*) [virtual]

Implements gdcm::network::ULAction.

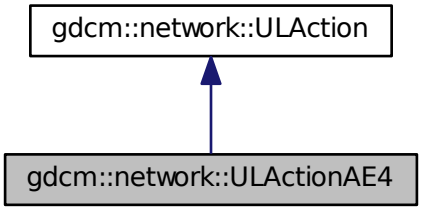
The documentation for this class was generated from the following file:

- gdcmULActionAE.h

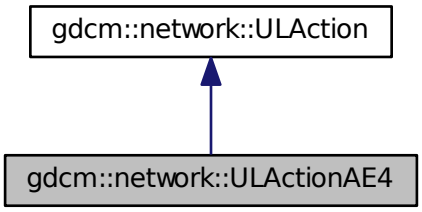
10.325 gdcm::network::ULActionAE4 Class Reference

`#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 EStateID gdcm::network::ULActionAE4::PerformAction (Subject * *s*, ULEvent & *inEvent*, ULConnection & *inConnection*, bool & *outWaitingForEvent*, EEventID & *outRaisedEvent*) [virtual]

Implements gdcm::network::ULAction.

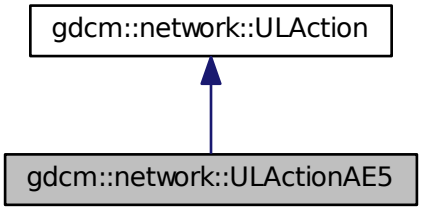
The documentation for this class was generated from the following file:

- gdcmULActionAE.h

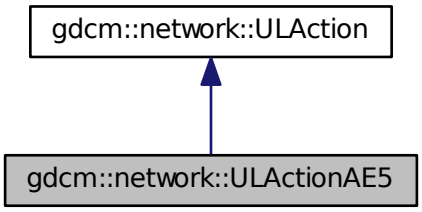
10.326 gdcm::network::ULActionAE5 Class Reference

`#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 EStateID gdcm::network::ULActionAE5::PerformAction (Subject ∗ *s,* ULEvent & *inEvent,* ULConnection & *inConnection,* bool & *outWaitingForEvent,* EEventID & *outRaisedEvent*) [virtual]

Implements gdcm::network::ULAction.

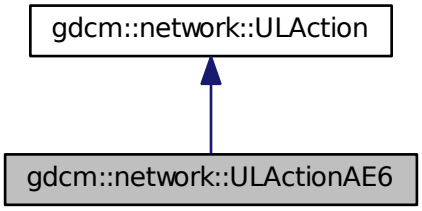
The documentation for this class was generated from the following file:

- gdcmULActionAE.h

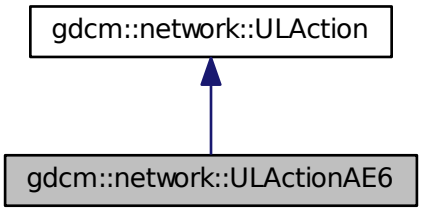
10.327 gdcm::network::ULActionAE6 Class Reference

`#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 EStateID gdcm::network::ULActionAE6::PerformAction (Subject * *s*, ULEvent & *inEvent*, ULConnection & *inConnection*, bool & *outWaitingForEvent*, EEventID & *outRaisedEvent*) [virtual]

Implements gdcm::network::ULAction.

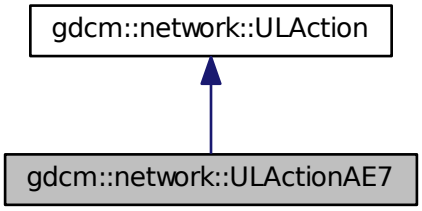
The documentation for this class was generated from the following file:

- gdcmULActionAE.h

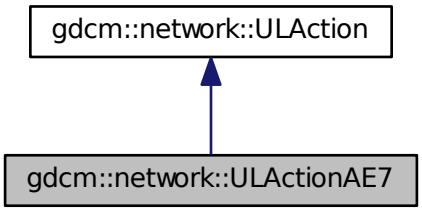
10.328 gdcm::network::ULActionAE7 Class Reference

`#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 EStateID gdcm::network::ULActionAE7::PerformAction (Subject * *s*, ULEvent & *inEvent*, ULConnection & *inConnection*, bool & *outWaitingForEvent*, EEventID & *outRaisedEvent*) [virtual]

Implements gdcm::network::ULAction.

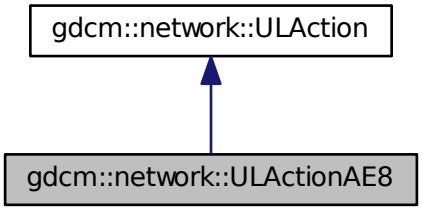
The documentation for this class was generated from the following file:

- gdcmULActionAE.h

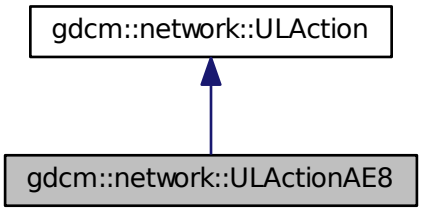
10.329 gdcm::network::ULActionAE8 Class Reference

`#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 &outWaitingFor↩ Event, EEventID &outRaisedEvent)

10.329.1 Member Function Documentation

10.329.1.1 EStateID gdcm::network::ULActionAE8::PerformAction (Subject * *s*, ULEvent & *inEvent*, ULConnection & *inConnection*, bool & *outWaitingForEvent*, EEventID & *outRaisedEvent*) [virtual]

Implements gdcm::network::ULAction.

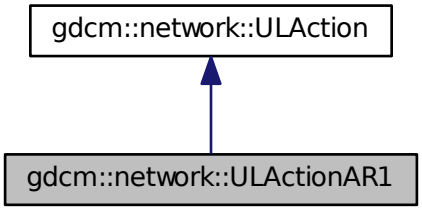
The documentation for this class was generated from the following file:

- gdcmULActionAE.h

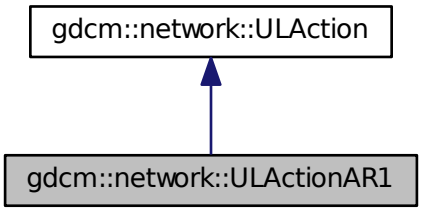
10.330 gdcm::network::ULActionAR1 Class Reference

`#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 EStateID gdcm::network::ULActionAR1::PerformAction (Subject * *s*, ULEvent & *inEvent*, ULConnection & *inConnection*, bool & *outWaitingForEvent*, EEventID & *outRaisedEvent*) [virtual]

Implements gdcm::network::ULAction.

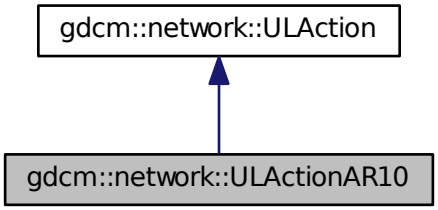
The documentation for this class was generated from the following file:

- gdcmULActionAR.h

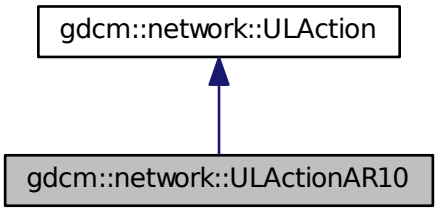
10.331 gdcm::network::ULActionAR10 Class Reference

`#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 EStateID gdcm::network::ULActionAR10::PerformAction (Subject * *s*, ULEvent & *inEvent*, ULConnection & *inConnection*, bool & *outWaitingForEvent*, EEventID & *outRaisedEvent*) [virtual]

Implements gdcm::network::ULAction.

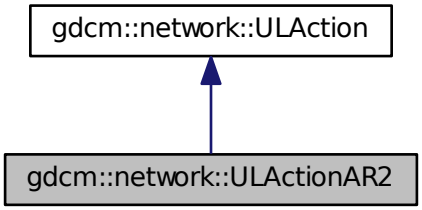
The documentation for this class was generated from the following file:

- gdcmULActionAR.h

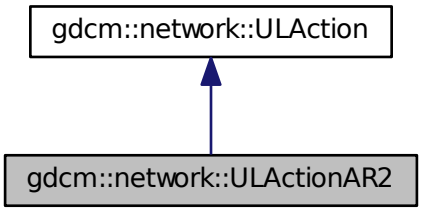
10.332 gdcm::network::ULActionAR2 Class Reference

`#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 &outWaitingFor↩ Event, EEventID &outRaisedEvent)

10.332.1 Member Function Documentation

10.332.1.1 EStateID gdcm::network::ULActionAR2::PerformAction (Subject * *s*, ULEvent & *inEvent*, ULConnection & *inConnection*, bool & *outWaitingForEvent*, EEventID & *outRaisedEvent*) [virtual]

Implements gdcm::network::ULAction.

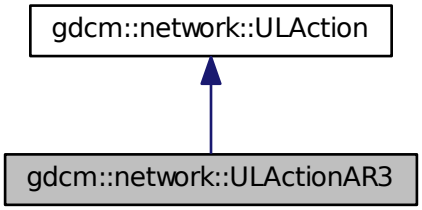
The documentation for this class was generated from the following file:

- gdcmULActionAR.h

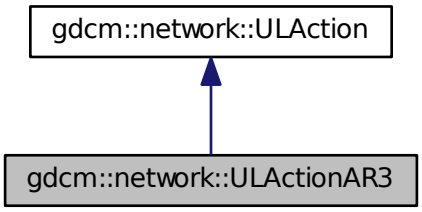
10.333 gdcm::network::ULActionAR3 Class Reference

`#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 EStateID gdcm::network::ULActionAR3::PerformAction (Subject * *s*, ULEvent & *inEvent*, ULConnection & *inConnection*, bool & *outWaitingForEvent*, EEventID & *outRaisedEvent*) [virtual]

Implements gdcm::network::ULAction.

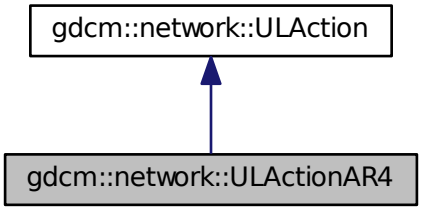
The documentation for this class was generated from the following file:

- gdcmULActionAR.h

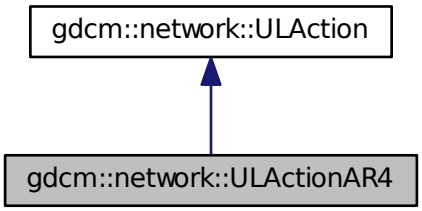
10.334 gdcm::network::ULActionAR4 Class Reference

`#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 EStateID gdcm::network::ULActionAR4::PerformAction (Subject * *s*, ULEvent & *inEvent*, ULConnection & *inConnection*, bool & *outWaitingForEvent*, EEventID & *outRaisedEvent*) [virtual]

Implements gdcm::network::ULAction.

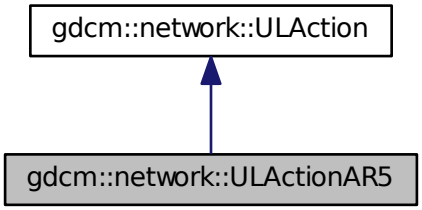
The documentation for this class was generated from the following file:

- gdcmULActionAR.h

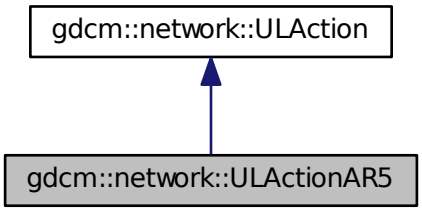
10.335 gdcm::network::ULActionAR5 Class Reference

`#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 EStateID gdcm::network::ULActionAR5::PerformAction (Subject * *s*, ULEvent & *inEvent*, ULConnection & *inConnection*, bool & *outWaitingForEvent*, EEventID & *outRaisedEvent*) [virtual]

Implements gdcm::network::ULAction.

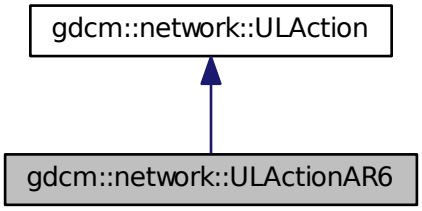
The documentation for this class was generated from the following file:

- gdcmULActionAR.h

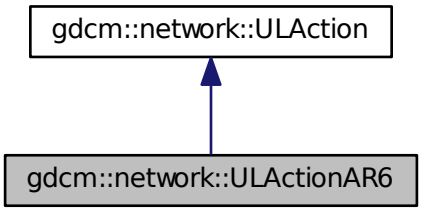
10.336 gdcm::network::ULActionAR6 Class Reference

`#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 EStateID gdcm::network::ULActionAR6::PerformAction (Subject * *s*, ULEvent & *inEvent*, ULConnection & *inConnection*, bool & *outWaitingForEvent*, EEventID & *outRaisedEvent*) [virtual]

Implements gdcm::network::ULAction.

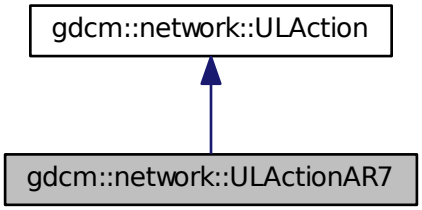
The documentation for this class was generated from the following file:

- gdcmULActionAR.h

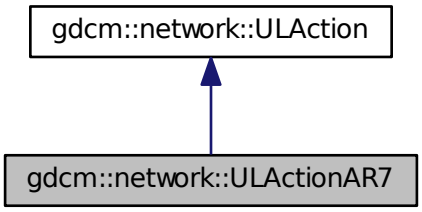
10.337 gdcm::network::ULActionAR7 Class Reference

`#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 &outWaitingFor↩ Event, EEventID &outRaisedEvent)

10.337.1 Member Function Documentation

10.337.1.1 EStateID gdcm::network::ULActionAR7::PerformAction (Subject * *s*, ULEvent & *inEvent*, ULConnection & *inConnection*, bool & *outWaitingForEvent*, EEventID & *outRaisedEvent*) [virtual]

Implements gdcm::network::ULAction.

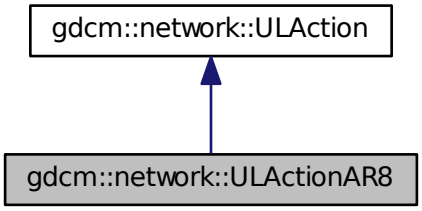
The documentation for this class was generated from the following file:

- gdcmULActionAR.h

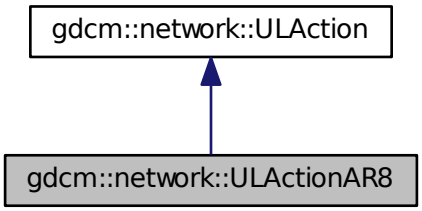
10.338 gdcm::network::ULActionAR8 Class Reference

`#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 EStateID gdcm::network::ULActionAR8::PerformAction (Subject * *s*, ULEvent & *inEvent*, ULConnection & *inConnection*, bool & *outWaitingForEvent*, EEventID & *outRaisedEvent*) [virtual]

Implements gdcm::network::ULAction.

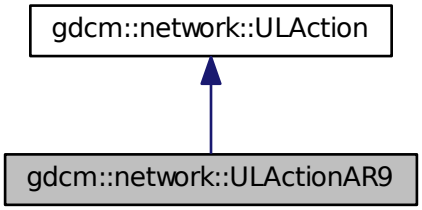
The documentation for this class was generated from the following file:

- gdcmULActionAR.h

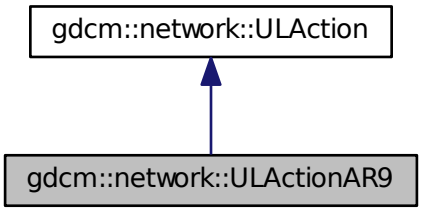
10.339 gdcm::network::ULActionAR9 Class Reference

`#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 EStateID gdcm::network::ULActionAR9::PerformAction (Subject * *s*, ULEvent & *inEvent*, ULConnection & *inConnection*, bool & *outWaitingForEvent*, EEventID & *outRaisedEvent*) [virtual]

Implements gdcm::network::ULAction.

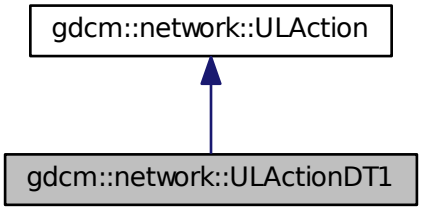
The documentation for this class was generated from the following file:

- gdcmULActionAR.h

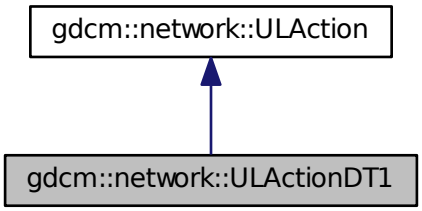
10.340 gdcm::network::ULActionDT1 Class Reference

`#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 &outWaitingFor↩ Event, EEventID &outRaisedEvent)

10.340.1 Member Function Documentation

10.340.1.1 EStateID gdcm::network::ULActionDT1::PerformAction (Subject * *s*, ULEvent & *inEvent*, ULConnection & *inConnection*, bool & *outWaitingForEvent*, EEventID & *outRaisedEvent*) [virtual]

Implements gdcm::network::ULAction.

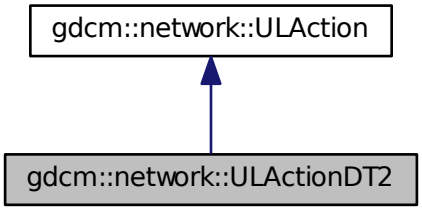
The documentation for this class was generated from the following file:

- gdcmULActionDT.h

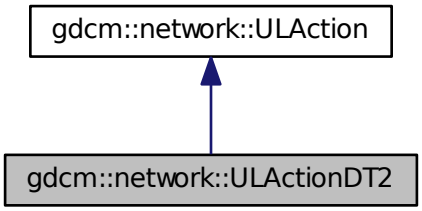
10.341 gdcm::network::ULActionDT2 Class Reference

`#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 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:

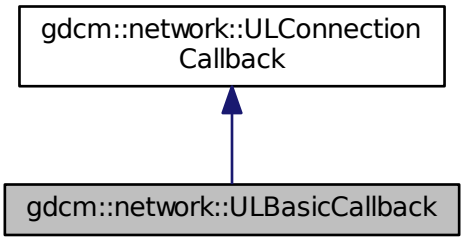
- gdcmULActionDT.h

10.342 gdcm::network::ULBasicCallback Class Reference

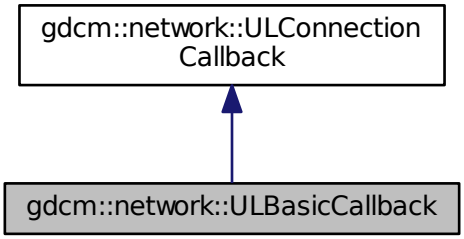
ULBasicCallback This is the most basic of callbacks for how the ULConnectionManager handles incoming datasets. DataSets are just concatenated to the mDataSets vector, and the result can be pulled out of the vector by later code. Alternatives to this method include progress updates, saving to disk, etc. This class is NOT THREAD SAFE. Access the dataset vector after the entire set of datasets has been returned by the ULConnectionManager.

```
#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 gdcm::network::ULBasicCallback::ULBasicCallback ( ) [inline]

10.342.2.2 virtual gdcm::network::ULBasicCallback::~ULBasicCallback ( ) [inline], [virtual]

10.342.3 Member Function Documentation

10.342.3.1 std::vector<DataSet> const& gdcm::network::ULBasicCallback::GetDataSets ( ) const

10.342.3.2 std::vector<DataSet> const& gdcm::network::ULBasicCallback::GetResponses ( ) const

10.342.3.3 virtual void gdcm::network::ULBasicCallback::HandleDataSet (const DataSet & *inDataSet*) [virtual]

Implements gdcm::network::ULConnectionCallback.

10.342.3.4 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 This is the class that contains the socket to another machine, and passes data through itself, as well as maintaining a sense of state.

```
#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 gdcm::network::ULConnection::ULConnection (const **ULConnectionInfo** & *inUserInformation*)

10.343.2.2 virtual gdcm::network::ULConnection::~ULConnection ( ) [virtual]

10.343.3 Member Function Documentation

10.343.3.1 void gdcm::network::ULConnection::AddAcceptedPresentationContext (const **PresentationContextAC** & *inPC*)

10.343.3.2 **PresentationContextRQ** gdcm::network::ULConnection::FindContext (const **DataElement** & *de*) const

10.343.3.3 std::vector<**PresentationContextAC**> const& gdcm::network::ULConnection::GetAcceptedPresentationContexts ( ) const

10.343.3.4 std::vector<**PresentationContextAC**>& gdcm::network::ULConnection::GetAcceptedPresentationContexts ( )

10.343.3.5 const **ULConnectionInfo**& gdcm::network::ULConnection::GetConnectionInfo ( ) const

10.343.3.6 uint32_t gdcm::network::ULConnection::GetMaxPDUSize ( ) const

10.343.3.7 const **PresentationContextAC*** gdcm::network::ULConnection::GetPresentationContextACByID (uint8_t *id*) const

10.343.3.8 uint8_t gdcm::network::ULConnection::GetPresentationContextIDFromPresentationContext (**PresentationContextRQ** const & *pc*) const

return 0 upon error

10.343.3.9 const **PresentationContextRQ*** gdcm::network::ULConnection::GetPresentationContextRQByID (uint8_t *id*) const

10.343.3.10 std::vector<**PresentationContextRQ**> const& gdcm::network::ULConnection::GetPresentationContexts ( ) const

10.343.3.11 std::iostream* gdcm::network::ULConnection::GetProtocol ( )

10.343.3.12 **EStateID** gdcm::network::ULConnection::GetState ( ) const

10.343.3.13 **ARTIMTimer**& gdcm::network::ULConnection::GetTimer ( )

10.343.3.14 bool gdcm::network::ULConnection::InitializeConnection ( )

used to establish scu connections

10.343.3.15 bool gdcm::network::ULConnection::InitializeIncomingConnection ( )

used to establish scp connections

10.343.3.16 void gdcm::network::ULConnection::SetMaxPDUSize (uint32_t *inSize*)

10.343.3.17 void gdcm::network::ULConnection::SetPresentationContexts (const std::vector< **PresentationContextRQ** > & *inContexts*)

10.343.3.18 void gdcm::network::ULConnection::SetPresentationContexts (const std::vector< **PresentationContext** > & *inContexts*)

10.343.3.19 void gdcm::network::ULConnection::SetState (const **EStateID** & *inState*)

10.343.3.20 void gdcm::network::ULConnection::StopProtocol ( )

10.343.4 Friends And Related Function Documentation

10.343.4.1 friend class ULActionAE6 [friend]

10.343.4.2 friend class ULConnectionManager [friend]

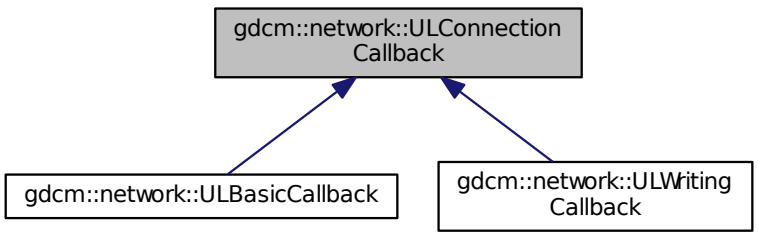
The documentation for this class was generated from the following file:

- 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 gdcm::network::ULConnectionCallback::ULConnectionCallback ( ) [inline]

10.344.2.2 virtual gdcm::network::ULConnectionCallback::~ULConnectionCallback ( ) [inline], [virtual]

10.344.3 Member Function Documentation

10.344.3.1 void gdcm::network::ULConnectionCallback::DataSetHandled ( ) [inline], [protected]

10.344.3.2 bool gdcm::network::ULConnectionCallback::DataSetHandles ( ) const [inline]

10.344.3.3 virtual void gdcm::network::ULConnectionCallback::HandleDataSet (const DataSet & *inDataSet*) [pure virtual]

Implemented in gdcm::network::ULBasicCallback, and gdcm::network::ULWritingCallback.

10.344.3.4 virtual void gdcm::network::ULConnectionCallback::HandleResponse (const DataSet & *inDataSet*) [pure virtual]

Implemented in gdcm::network::ULBasicCallback, and gdcm::network::ULWritingCallback.

10.344.3.5 void gdcm::network::ULConnectionCallback::ResetHandledDataSet ( ) [inline]

10.344.3.6 void gdcm::network::ULConnectionCallback::SetImplicitFlag (const bool *imp*) [inline]

10.344.4 Member Data Documentation

10.344.4.1 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 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.

`#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 gdcm::network::ULConnectionInfo::ULConnectionInfo ( )

10.345.3 Member Function Documentation

10.345.3.1 const char* gdcm::network::ULConnectionInfo::GetCalledAETitle ( ) const

10.345.3.2 std::string gdcm::network::ULConnectionInfo::GetCalledComputerName ( ) const

10.345.3.3 unsigned long gdcm::network::ULConnectionInfo::GetCalledIPAddress ( ) const

10.345.3.4 int gdcm::network::ULConnectionInfo::GetCalledIPPort ( ) const

10.345.3.5 const char* gdcm::network::ULConnectionInfo::GetCallingAETitle ( ) const

10.345.3.6 unsigned long gdcm::network::ULConnectionInfo::GetMaxPDULength ( ) const

10.345.3.7 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 void gdcm::network::ULConnectionInfo::SetMaxPDULength (unsigned long *inMaxPDULength*)

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

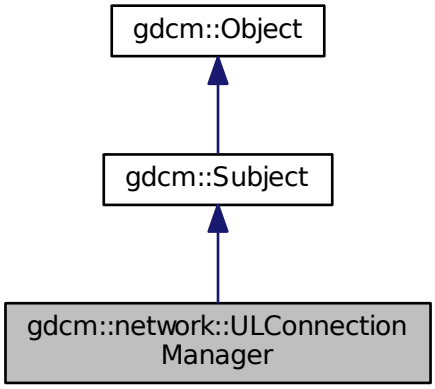
- gdcmULConnectionInfo.h

10.346 gdcm::network::ULConnectionManager Class Reference

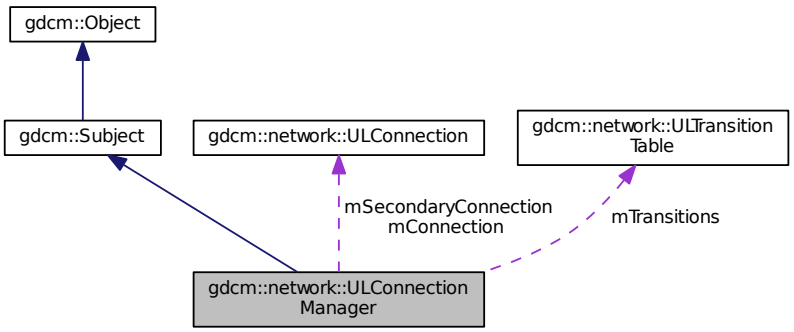
ULConnectionManager The ULConnectionManager performs actions on the ULConnection given inputs from the user and from the state of what's going on around the connection (ie, timeouts of the ARTIM timer, responses from the peer across the connection, etc).

```
#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 dataSet↩ Offset=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 *in↩ Callback, 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 gdcm::network::ULConnectionManager::ULConnectionManager (const ULConnectionManager & *inCM*) [protected]

10.346.2.2 gdcm::network::ULConnectionManager::ULConnectionManager ( )

10.346.2.3 virtual gdcm::network::ULConnectionManager::~ULConnectionManager ( ) [virtual]

10.346.3 Member Function Documentation

10.346.3.1 bool gdcm::network::ULConnectionManager::BreakConnection (const double & *inTimeout*)

10.346.3.2 void gdcm::network::ULConnectionManager::BreakConnectionNow ( )

10.346.3.3 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 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 EStateID gdcm::network::ULConnectionManager::RunEventLoop (ULEvent & *inEvent,* ULConnection * *inWhichConnection,* ULConnectionCallback * *inCallback,* const bool & *startWaiting*) [protected]

10.346.3.6 EStateID gdcm::network::ULConnectionManager::RunMoveEventLoop (ULEvent & *inEvent,* ULConnectionCallback * *inCallback*) [protected]

10.346.3.7 std::vector<PresentationDataValue> gdcm::network::ULConnectionManager::SendEcho ( )

10.346.3.8 std::vector<DataSet> gdcm::network::ULConnectionManager::SendFind (const BaseRootQuery * *inRootQuery*)

10.346.3.9 void gdcm::network::ULConnectionManager::SendFind (const BaseRootQuery * *inRootQuery,* ULConnectionCallback * *inCallback*)

10.346.3.10 std::vector<DataSet> gdcm::network::ULConnectionManager::SendMove (const BaseRootQuery * *inRootQuery*)

10.346.3.11 bool gdcm::network::ULConnectionManager::SendMove (const BaseRootQuery * *inRootQuery,* ULConnectionCallback * *inCallback*)

return false upon error

10.346.3.12 std::vector<DataSet> gdcm::network::ULConnectionManager::SendNAction (const BaseQuery * *inQuery*)

10.346.3.13 void gdcm::network::ULConnectionManager::SendNAction (const BaseQuery * *inQuery*, ULConnectionCallback * *inCallback*)

10.346.3.14 std::vector<DataSet> gdcm::network::ULConnectionManager::SendNCreate (const BaseQuery * *inQuery*)

10.346.3.15 void gdcm::network::ULConnectionManager::SendNCreate (const BaseQuery * *inQuery*, ULConnectionCallback * *inCallback*)

10.346.3.16 std::vector<DataSet> gdcm::network::ULConnectionManager::SendNDelete (const BaseQuery * *inQuery*)

10.346.3.17 void gdcm::network::ULConnectionManager::SendNDelete (const BaseQuery * *inQuery*, ULConnectionCallback * *inCallback*)

10.346.3.18 std::vector<DataSet> gdcm::network::ULConnectionManager::SendNEventReport (const BaseQuery * *inQuery*)

10.346.3.19 void gdcm::network::ULConnectionManager::SendNEventReport (const BaseQuery * *inQuery*, ULConnectionCallback * *inCallback*)

10.346.3.20 std::vector<DataSet> gdcm::network::ULConnectionManager::SendNGet (const BaseQuery * *inQuery*)

10.346.3.21 void gdcm::network::ULConnectionManager::SendNGet (const BaseQuery * *inQuery*, ULConnectionCallback * *inCallback*)

10.346.3.22 std::vector<DataSet> gdcm::network::ULConnectionManager::SendNSet (const BaseQuery * *inQuery*)

10.346.3.23 void gdcm::network::ULConnectionManager::SendNSet (const BaseQuery * *inQuery*, ULConnectionCallback * *inCallback*)

10.346.3.24 std::vector<DataSet> gdcm::network::ULConnectionManager::SendStore (const File & *file*, std::istream * *pStream* = NULL, std::streampos *dataSetOffset* = 0)

10.346.3.25 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 ULConnection* gdcm::network::ULConnectionManager::mConnection [protected]

10.346.4.2 ULConnection* gdcm::network::ULConnectionManager::mSecondaryConnection [protected]

10.346.4.3 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 base class for network events.

`#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 gdcm::network::ULEvent::ULEvent (const EEventID & *inEventID*, std::vector< BasePDU * > *inBasePDU*, std::istream * *iStream* = NULL, std::streampos *posDataSet* = 0) [inline]

10.347.2.2 gdcm::network::ULEvent::ULEvent (const EEventID & *inEventID*, BasePDU * *inBasePDU*, std::istream * *iStream* = NULL, std::streampos *posDataSet* = 0) [inline]

10.347.2.3 gdcm::network::ULEvent::~ULEvent ( ) [inline]

10.347.3 Member Function Documentation

10.347.3.1 std::streampos gdcm::network::ULEvent::GetDataSetPos ( ) const [inline]

10.347.3.2 EEventID gdcm::network::ULEvent::GetEvent ( ) const [inline]

10.347.3.3 std::istream* gdcm::network::ULEvent::GetIStream ( ) const [inline]

10.347.3.4 std::vector<BasePDU*> const& gdcm::network::ULEvent::GetPDUs ( ) const [inline]

10.347.3.5 void gdcm::network::ULEvent::SetEvent (const EEventID & *inEvent*) [inline]

10.347.3.6 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 gdcm::network::ULTransitionTable::ULTransitionTable ( )

10.348.3 Member Function Documentation

10.348.3.1 void gdcm::network::ULTransitionTable::HandleEvent (Subject * *s*, ULEvent & *inEvent*, ULConnection & *inConnection*, bool & *outWaitingForEvent*, EEventID & *outRaisedEvent*) const

10.348.3.2 void gdcm::network::ULTransitionTable::PrintTable ( ) const

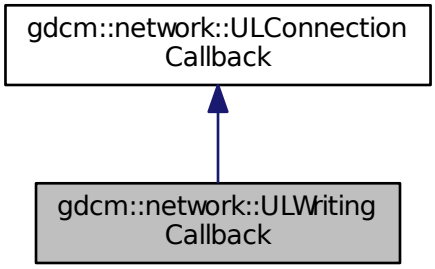
The documentation for this class was generated from the following file:

- gdcmULTransitionTable.h

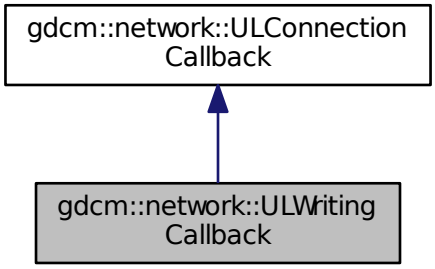
10.349 gdcm::network::ULWritingCallback Class Reference

`#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 gdcm::network::ULWritingCallback::ULWritingCallback ( ) [inline]

10.349.1.2 virtual gdcm::network::ULWritingCallback::~ULWritingCallback ( ) [inline], [virtual]

10.349.2 Member Function Documentation

10.349.2.1 virtual void gdcm::network::ULWritingCallback::HandleDataSet (const DataSet & *inDataSet*) [virtual]

Implements gdcm::network::ULConnectionCallback.

10.349.2.2 virtual void gdcm::network::ULWritingCallback::HandleResponse (const DataSet & *inDataSet*) [virtual]

Implements gdcm::network::ULConnectionCallback.

10.349.2.3 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:

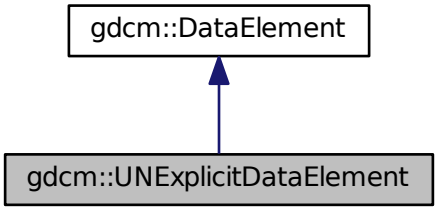
- gdcmULWritingCallback.h

10.350 gdcm::UNExplicitDataElement Class Reference

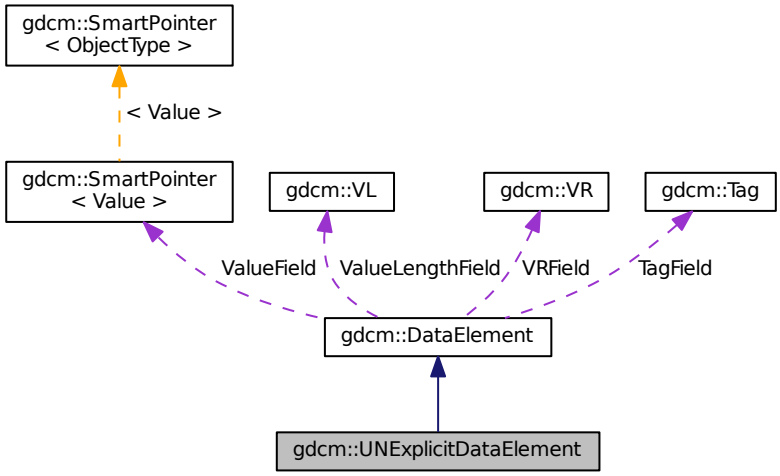
Class to read/write a DataElement as UNExplicit Data Element.

`#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 VL gdcm::UNExplicitDataElement::GetLength ( ) const

10.350.2.2 template<typename TSwap > std::istream& gdcm::UNExplicitDataElement::Read (std::istream & *is*)

10.350.2.3 template<typename TSwap > std::istream& gdcm::UNExplicitDataElement::ReadPreValue (std::istream & *is*)

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

10.350.2.5 template<typename TSwap > std::istream& gdcm::UNExplicitDataElement::ReadWithLength (std::istream & *is,* VL & *length*)

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

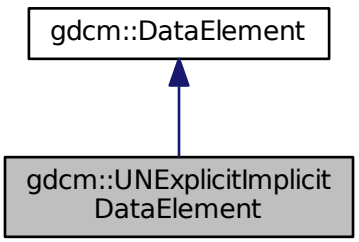
- gdcmUNExplicitDataElement.h

10.351 gdcm::UNExplicitImplicitDataElement Class Reference

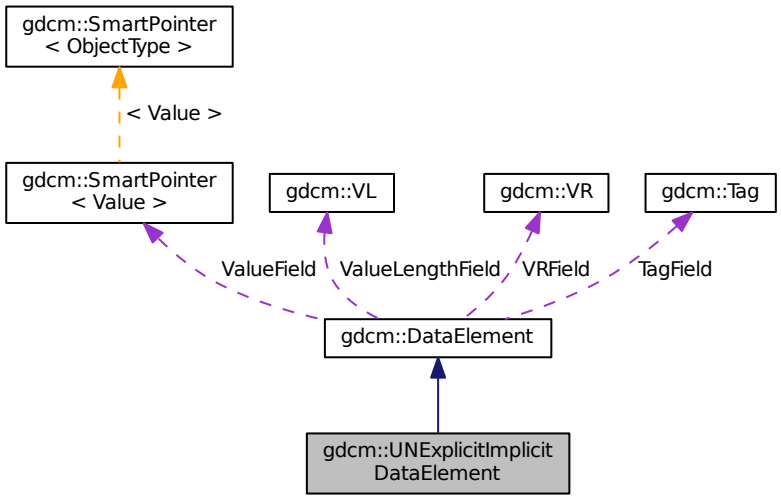
Class to read/write a DataElement as ExplicitImplicit Data Element This class gather two known bugs:

```
#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 VL gdcm::UNExplicitImplicitDataElement::GetLength ( ) const

10.351.2.2 template<typename TSwap > std::istream& gdcm::UNExplicitImplicitDataElement::Read (std::istream & *is*)

10.351.2.3 template<typename TSwap > std::istream& gdcm::UNExplicitImplicitDataElement::ReadPreValue (std::istream & *is*)

10.351.2.4 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 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 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 enum gdcm::Usage::UsageType

Enumerator

Mandatory

Conditional

UserOption

Invalid

10.353.3 Constructor & Destructor Documentation

10.353.3.1 gdcm::Usage::Usage (UsageType *type* = Invalid) [inline]

10.353.4 Member Function Documentation

10.353.4.1 static const char* gdcm::Usage::GetUsageString (UsageType *type*) [static]

Referenced by gdcm::operator<<().

10.353.4.2 static UsageType gdcm::Usage::GetUsageType (const char * *type*) [static]

10.353.4.3 gdcm::Usage::operator UsageType ( ) const [inline]

References gdcm::operator<<().

10.353.5 Friends And Related Function Documentation

10.353.5.1 std::ostream& operator<< (std::ostream & *os*, const Usage & *vr*) [friend]

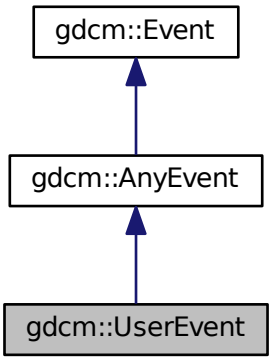
The documentation for this class was generated from the following file:

- gdcmUsage.h

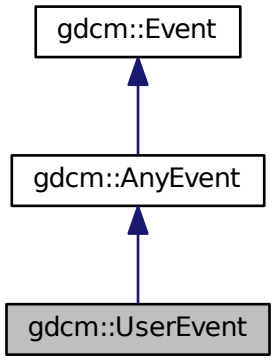
10.354 gdcm::UserEvent Class Reference

`#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 Table 9-16 USER INFORMATION ITEM FIELDS.

`#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 gdcm::network::UserInformation::UserInformation ( )

Referenced by GetMaximumLengthSub().

10.355.2.2 gdcm::network::UserInformation::~UserInformation ( )

10.355.3 Member Function Documentation

10.355.3.1 void gdcm::network::UserInformation::AddRoleSelectionSub (RoleSelectionSub const & *r*)

Referenced by GetMaximumLengthSub().

10.355.3.2 void gdcm::network::UserInformation::AddSOPClassExtendedNegociationSub (SOPClassExtendedNegociation↩ Sub const & *s*)

Referenced by GetMaximumLengthSub().

10.355.3.3 const MaximumLengthSub& gdcm::network::UserInformation::GetMaximumLengthSub ( ) const [inline]

10.355.3.4 MaximumLengthSub& gdcm::network::UserInformation::GetMaximumLengthSub ( ) [inline]

References AddRoleSelectionSub(), AddSOPClassExtendedNegociationSub(), operator=(), and UserInformation().

10.355.3.5 UserInformation& gdcm::network::UserInformation::operator= (const UserInformation &)

Referenced by GetMaximumLengthSub().

10.355.3.6 void gdcm::network::UserInformation::Print (std::ostream & *os*) const

10.355.3.7 std::istream& gdcm::network::UserInformation::Read (std::istream & *is*)

10.355.3.8 size_t gdcm::network::UserInformation::Size ( ) const

10.355.3.9 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 generate DCE 1.1 uid.

`#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 const char* gdcm::UUIDGenerator::Generate ( )

Return the generated uuid NOT THREAD SAFE

10.356.2.2 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:

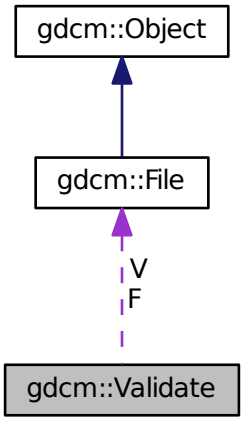
- gdcmUUIDGenerator.h

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 gdcm::Validate::Validate ( )

10.357.2.2 gdcm::Validate::∼Validate ( )

10.357.3 Member Function Documentation

10.357.3.1 const File& gdcm::Validate::GetValidatedFile ( ) [inline]

10.357.3.2 void gdcm::Validate::SetFile (File const & *f*) [inline]

10.357.3.3 void gdcm::Validate::Validation ( )

10.357.4 Member Data Documentation

10.357.4.1 const File∗ gdcm::Validate::F [protected]

10.357.4.2 File gdcm::Validate::V [protected]

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

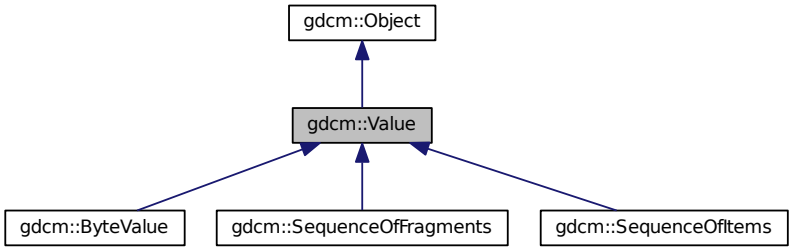
- gdcmValidate.h

10.358 gdcm::Value Class Reference

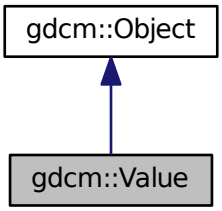
Class to represent the value of a Data Element.

`#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 gdcm::Value::Value ( ) [inline]

10.358.2.2 gdcm::Value::~Value ( ) [inline]

References gdcm::operator==().

10.358.3 Member Function Documentation

10.358.3.1 virtual void gdcm::Value::Clear ( ) [pure virtual]

Implemented in gdcm::ByteValue, gdcm::SequenceOfItems, and gdcm::SequenceOfFragments.

10.358.3.2 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 virtual bool gdcm::Value::operator== (const Value & *val*) const [pure virtual]

Implemented in gdcm::SequenceOfFragments, gdcm::SequenceOfItems, and gdcm::ByteValue.

10.358.3.4 virtual void gdcm::Value::SetLength (VL *l*) [pure virtual]

Implemented in gdcm::ByteValue, gdcm::SequenceOfItems, and gdcm::SequenceOfFragments.

10.358.3.5 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 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 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 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 gdcm::Version::Version ( ) [inline]

10.360.2.2 gdcm::Version::~Version ( ) [inline]

10.360.3 Member Function Documentation

10.360.3.1 static int gdcm::Version::GetBuildVersion ( ) [static]

10.360.3.2 static int gdcm::Version::GetMajorVersion ( ) [static]

10.360.3.3 static int gdcm::Version::GetMinorVersion ( ) [static]

10.360.3.4 static const char* gdcm::Version::GetVersion ( ) [static]

10.360.3.5 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 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 typedef uint32_t gdcm::VL::Type

10.361.3 Constructor & Destructor Documentation

10.361.3.1 gdcm::VL::VL (uint32_t *vl* = 0) [inline]

10.361.4 Member Function Documentation

10.361.4.1 VL gdcm::VL::GetLength ( ) const [inline]

References gdcm::operator<<().

Referenced by gdcm::FileMetaInformation::GetFullLength(), and gdcm::Item::Write().

10.361.4.2 static uint16_t gdcm::VL::GetVL16Max ( ) [inline], [static]

10.361.4.3 static uint32_t gdcm::VL::GetVL32Max ( ) [inline], [static]

10.361.4.4 bool gdcm::VL::IsOdd ( ) const [inline]

Return whether or not the VL is odd or not.

10.361.4.5 bool gdcm::VL::IsUndefined ( ) const [inline]

10.361.4.6 gdcm::VL::operator uint32_t ( ) const [inline]

10.361.4.7 VL& gdcm::VL::operator++ ( ) [inline]

10.361.4.8 VL gdcm::VL::operator++ (int) [inline]

10.361.4.9 VL& gdcm::VL::operator+= (VL const & *vl*) [inline]

+= operator

10.361.4.10 template<typename TSwap > std::istream& gdcm::VL::Read (std::istream & *is*) [inline]

10.361.4.11 template<typename TSwap > std::istream& gdcm::VL::Read16 (std::istream & *is*) [inline]

10.361.4.12 void gdcm::VL::SetToUndefined ( ) [inline]

10.361.4.13 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::SequenceOfFragments::Write().

10.361.4.14 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 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 enum gdcm::VM::VMType

Enumerator

VM0

VM1

VM2

VM3

VM4

VM5

VM6

VM8

VM9

VM10

VM12

VM16

VM18

VM24

VM28

VM32

VM35

VM99

VM256

VM1_2

VM1_3

VM1_4

VM1_5

VM1_8

VM1_32

VM1_99

VM1_n

VM2_2n

VM2_n

VM3_4

VM3_3n

VM3_n

VM4_4n

VM6_6n

VM7_7n

VM30_30n

VM47_47n

VM_END

10.362.3 Constructor & Destructor Documentation

10.362.3.1 gdcm::VM::VM (VMType *type* = VM0) [inline]

10.362.4 Member Function Documentation

10.362.4.1 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 static unsigned int gdcm::VM::GetIndex (VMType *vm*) [static], [protected]

10.362.4.3 unsigned int gdcm::VM::GetLength ( ) const

10.362.4.4 static unsigned int gdcm::VM::GetNumberOfElementsFromArray (const char * *array*, unsigned int *length*) [static]

10.362.4.5 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 static VMType gdcm::VM::GetVMType (const char * *vm*) [static]

10.362.4.7 static VMType gdcm::VM::GetVMTypeFromLength (unsigned int *length*, unsigned int *size*) [static]

10.362.4.8 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 gdcm::VM::operator VMType ( ) const [inline]

References gdcm::operator<<().

10.362.5 Friends And Related Function Documentation

10.362.5.1 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 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.

```
#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 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
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 gdcm::VR::VR (VRType *vr* = INVALID) [inline]

10.364.4 Member Function Documentation

10.364.4.1 static bool gdcm::VR::CanDisplay (VRType *vr*) [static]

10.364.4.2 bool gdcm::VR::Compatible (VR const & *vr*) const

10.364.4.3 int gdcm::VR::GetLength ( ) const [inline]

10.364.4.4 static uint32_t gdcm::VR::GetLength (VRType *vr*) [inline], [static]

10.364.4.5 unsigned int gdcm::VR::GetSize ( ) const [inline]

References US_SS, and VRTypeTemplateCase.

10.364.4.6 unsigned int gdcm::VR::GetSizeof ( ) const

10.364.4.7 static const char* gdcm::VR::GetVRString (VRType *vr*) [static]

Referenced by gdcm::operator<<().

10.364.4.8 static const char* gdcm::VR::GetVRStringFromFile (VRType *vr*) [static]

10.364.4.9 static VRType gdcm::VR::GetVRType (const char * *vr*) [static]

10.364.4.10 static VRType gdcm::VR::GetVRTypeFromFile (const char * *vr*) [static]

10.364.4.11 static bool gdcm::VR::IsASCII (VRType *vr*) [static]

10.364.4.12 static bool gdcm::VR::IsASCII2 (VRType *vr*) [static]

10.364.4.13 static bool gdcm::VR::IsBinary (VRType *vr*) [static]

10.364.4.14 static bool gdcm::VR::IsBinary2 (VRType *vr*) [static]

10.364.4.15 bool gdcm::VR::IsDual ( ) const

10.364.4.16 static bool gdcm::VR::IsSwap (const char * *vr*) [static]

10.364.4.17 static bool gdcm::VR::IsValid (const char * *vr*) [static]

10.364.4.18 static bool gdcm::VR::IsValid (const char * *vr1*, VRType *vr2*) [static]

10.364.4.19 bool gdcm::VR::IsVRFile ( ) const

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

10.364.4.20 gdcm::VR::operator VRType ( ) const [inline]

10.364.4.21 std::istream& gdcm::VR::Read (std::istream & *is*) [inline]

References gdcmDebugMacro, INVALID, and VR_END.

10.364.4.22 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 std::ostream& operator<< (std::ostream & *os,* const VR & *vr*) [friend]

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

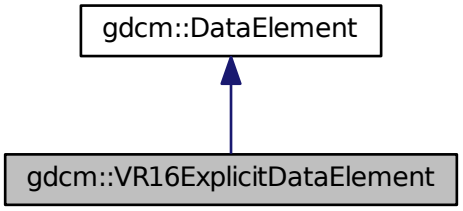
- gdcmVR.h

10.365 gdcm::VR16ExplicitDataElement Class Reference

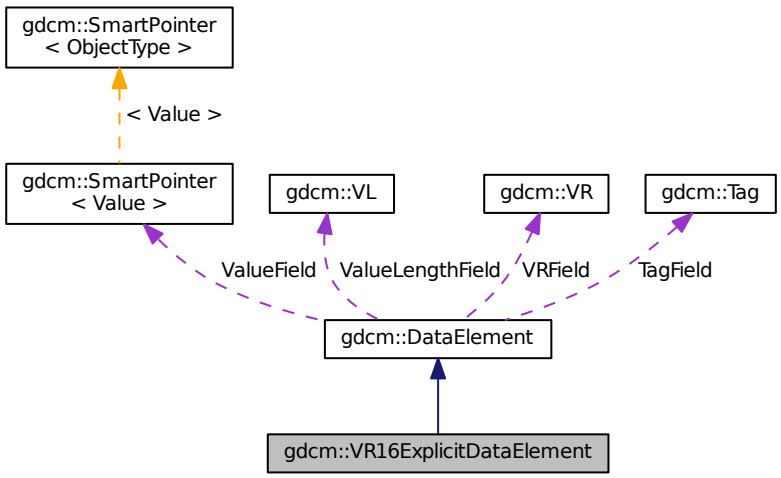
Class to read/write a DataElement as Explicit Data Element.

`#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 VL gdcm::VR16ExplicitDataElement::GetLength ( ) const

10.365.2.2 template<typename TSwap > std::istream& gdcm::VR16ExplicitDataElement::Read (std::istream & *is*)

10.365.2.3 template<typename TSwap > std::istream& gdcm::VR16ExplicitDataElement::ReadPreValue (std::istream & *is*)

10.365.2.4 template<typename TSwap > std::istream& gdcm::VR16ExplicitDataElement::ReadValue (std::istream & *is*, bool *readvalues* = `true`)

10.365.2.5 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>`

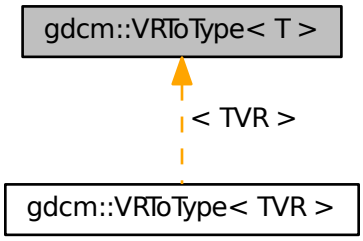
The documentation for this struct was generated from the following file:

- 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 static uint16_t gdcm::VRVLSize< 0 >::Read (std::istream & *_is*) [inline], [static]

10.369.1.2 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 static uint32_t gdcm::VRVLSize< 1 >::Read (std::istream & *_is*) [inline], [static]

10.370.1.2 static void gdcm::VRVLSize< 1 >::Write (std::ostream & *os*) [inline], [static]

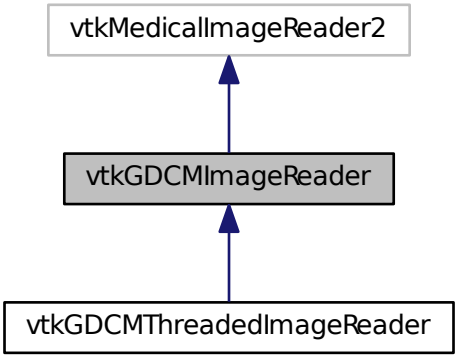
The documentation for this class was generated from the following file:

- gdcmAttribute.h

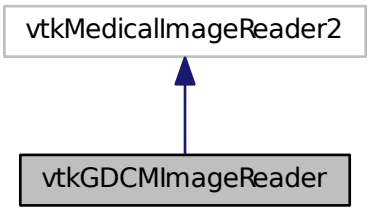
10.371 vtkGDCMImageReader Class Reference

`#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)
- vtkTypeMacro (vtkGDCMImageReader, vtkMedicalImageReader2)

Static Public Member Functions

- static vtkGDCMImageReader * New ()

Protected Member Functions

- vtkGDCMImageReader ()
- ~vtkGDCMImageReader ()
- void ExecuteData (vtkDataObject *out)
- void ExecuteInformation ()
- void FillMedicalImageInformation (const gdcm::ImageReader &reader)
- int LoadSingleFile (const char *filename, char *pointer, unsigned long &outlen)
- int RequestDataCompat ()
- int RequestInformationCompat ()
- void SetFilePattern (const char *)
- void SetFilePrefix (const char *)
- vtkGetStringMacro (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 ( ) [protected]

Examples:

HelloActiviz2.cs.

10.371.2.2 vtkGDCMImageReader::~vtkGDCMImageReader ( ) [protected]

10.371.3 Member Function Documentation

10.371.3.1 virtual int vtkGDCMImageReader::CanReadFile (const char * *fname*) [virtual]

Examples:

MetaImageMD5Activiz.cs.

10.371.3.2 void vtkGDCMImageReader::ExecuteData (vtkDataObject * *out*) [protected]

10.371.3.3 void vtkGDCMImageReader::ExecuteInformation ( ) [protected]

10.371.3.4 void vtkGDCMImageReader::FillMedicalImageInformation (const gdcm::ImageReader & *reader*) [protected]

10.371.3.5 virtual const char* vtkGDCMImageReader::GetDescriptiveName ( ) [inline], [virtual]

10.371.3.6 virtual const char* vtkGDCMImageReader::GetFileExtensions ( ) [inline], [virtual]

10.371.3.7 vtkImageData* vtkGDCMImageReader::GetIconImage ( )

10.371.3.8 vtkImageData* vtkGDCMImageReader::GetOverlay (int *i*)

10.371.3.9 int vtkGDCMImageReader::LoadSingleFile (const char * *filename,* char * *pointer,* unsigned long & *outlen*) [protected]

10.371.3.10 static vtkGDCMImageReader* vtkGDCMImageReader::New ( ) [static]

Examples:

Convert16BitsTo8Bits.cxx, ConvertMultiFrameToSingleFrame.cxx, ConvertRGBToLuminance.cxx, Convert↩ SingleBitTo8Bits.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 virtual void vtkGDCMImageReader::PrintSelf (ostream & *os*, vtkIndent *indent*) [virtual]

Reimplemented in vtkGDCMThreadedImageReader.

10.371.3.12 int vtkGDCMImageReader::RequestDataCompat ( ) [protected]

10.371.3.13 int vtkGDCMImageReader::RequestInformationCompat ( ) [protected]

10.371.3.14 virtual void vtkGDCMImageReader::SetCurve (vtkPolyData * *pd*) [virtual]

10.371.3.15 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 void vtkGDCMImageReader::SetFilePattern (const char *) [inline], [protected]

10.371.3.17 void vtkGDCMImageReader::SetFilePrefix (const char *) [inline], [protected]

10.371.3.18 virtual void vtkGDCMImageReader::SetMedicalImageProperties (vtkMedicalImageProperties * *pd*) [virtual]

10.371.3.19 vtkGDCMImageReader::vtkBooleanMacro (**LoadOverlays**, int)

10.371.3.20 vtkGDCMImageReader::vtkBooleanMacro (**LoadIconImage**, int)

10.371.3.21 vtkGDCMImageReader::vtkBooleanMacro (**LossyFlag**, int)

10.371.3.22 vtkGDCMImageReader::vtkBooleanMacro (**ApplyLookupTable**, int)

10.371.3.23 int vtkGDCMImageReader::vtkBooleanMacro (**ApplyYBRToRGB**, int)

10.371.3.24 vtkGDCMImageReader::vtkGetMacro (**LoadOverlays**, int)

10.371.3.25 vtkGDCMImageReader::vtkGetMacro (**LoadIconImage**, int)

10.371.3.26 vtkGDCMImageReader::vtkGetMacro (**LossyFlag**, int)

10.371.3.27 vtkGDCMImageReader::vtkGetMacro (**NumberOfOverlays**, int)

10.371.3.28 vtkGDCMImageReader::vtkGetMacro (NumberOfIconImages , int)

10.371.3.29 vtkGDCMImageReader::vtkGetMacro (ApplyLookupTable , int)

10.371.3.30 vtkGDCMImageReader::vtkGetMacro (ApplyYBRToRGB , int)

10.371.3.31 vtkGDCMImageReader::vtkGetMacro (ImageFormat , int)

10.371.3.32 vtkGDCMImageReader::vtkGetMacro (PlanarConfiguration , int)

10.371.3.33 vtkGDCMImageReader::vtkGetMacro (Shift , double)

10.371.3.34 vtkGDCMImageReader::vtkGetMacro (Scale , double)

10.371.3.35 vtkGDCMImageReader::vtkGetObjectMacro (DirectionCosines , vtkMatrix4x4)

10.371.3.36 vtkGDCMImageReader::vtkGetObjectMacro (MedicalImageProperties , vtkMedicalImageProperties)

10.371.3.37 vtkGDCMImageReader::vtkGetObjectMacro (FileNames , vtkStringArray)

10.371.3.38 vtkGDCMImageReader::vtkGetObjectMacro (Curve , vtkPolyData)

10.371.3.39 vtkGDCMImageReader::vtkGetStringMacro (FilePrefix) [protected]

10.371.3.40 vtkGDCMImageReader::vtkGetStringMacro (FilePattern) [protected]

10.371.3.41 vtkGDCMImageReader::vtkGetVector3Macro (ImagePositionPatient , double)

10.371.3.42 vtkGDCMImageReader::vtkGetVector6Macro (ImageOrientationPatient , double)

10.371.3.43 vtkGDCMImageReader::vtkSetMacro (LoadOverlays , int)

10.371.3.44 vtkGDCMImageReader::vtkSetMacro (LoadIconImage , int)

10.371.3.45 vtkGDCMImageReader::vtkSetMacro (LossyFlag , int)

10.371.3.46 vtkGDCMImageReader::vtkSetMacro (ApplyLookupTable , int)

10.371.3.47 vtkGDCMImageReader::vtkSetVector6Macro (ImageOrientationPatient , double) [protected]

10.371.3.48 vtkGDCMImageReader::vtkTypeMacro (vtkGDCMImageReader , vtkMedicalImageReader2)

10.371.4 Member Data Documentation

10.371.4.1 int vtkGDCMImageReader::ApplyInverseVideo [protected]

10.371.4.2 int vtkGDCMImageReader::ApplyLookupTable [protected]

10.371.4.3 int vtkGDCMImageReader::ApplyPlanarConfiguration [protected]

10.371.4.4 int vtkGDCMImageReader::ApplyShiftScale [protected]

10.371.4.5 int vtkGDCMImageReader::ApplyYBRToRGB [protected]

10.371.4.6 vtkPolyData* vtkGDCMImageReader::Curve [protected]

10.371.4.7 vtkMatrix4x4* vtkGDCMImageReader::DirectionCosines [protected]

10.371.4.8 vtkStringArray* vtkGDCMImageReader::FileNames [protected]

10.371.4.9 int vtkGDCMImageReader::ForceRescale [protected]

10.371.4.10 int vtkGDCMImageReader::IconDataScalarType [protected]

10.371.4.11 int vtkGDCMImageReader::IconImageDataExtent[6] [protected]

10.371.4.12 int vtkGDCMImageReader::IconNumberOfScalarComponents [protected]

10.371.4.13 int vtkGDCMImageReader::ImageFormat [protected]

10.371.4.14 double vtkGDCMImageReader::ImageOrientationPatient[6] [protected]

10.371.4.15 double vtkGDCMImageReader::ImagePositionPatient[3] [protected]

10.371.4.16 int vtkGDCMImageReader::LoadIconImage [protected]

10.371.4.17 int vtkGDCMImageReader::LoadOverlays [protected]

10.371.4.18 int vtkGDCMImageReader::LossyFlag [protected]

10.371.4.19 vtkMedicalImageProperties* vtkGDCMImageReader::MedicalImageProperties [protected]

10.371.4.20 int vtkGDCMImageReader::NumberOfIconImages [protected]

10.371.4.21 int vtkGDCMImageReader::NumberOfOverlays [protected]

10.371.4.22 int vtkGDCMImageReader::PlanarConfiguration [protected]

10.371.4.23 double vtkGDCMImageReader::Scale [protected]

10.371.4.24 double vtkGDCMImageReader::Shift [protected]

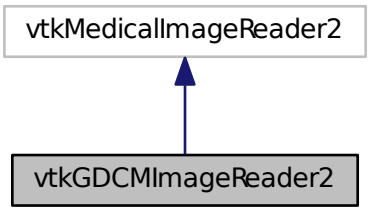
The documentation for this class was generated from the following file:

- vtkGDCMImageReader.h

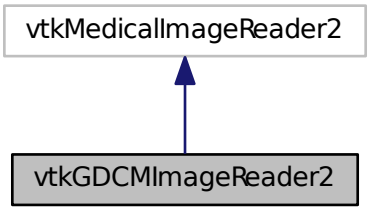
10.372 vtkGDCMImageReader2 Class Reference

`#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)
- vtkTypeMacro (vtkGDCMImageReader2, vtkMedicalImageReader2)

Static Public Member Functions

- static vtkGDCMImageReader2 * New ()

Protected Member Functions

- vtkGDCMImageReader2 ()
- ~vtkGDCMImageReader2 ()
- void FillMedicalImageInformation (const gdcm::ImageReader &reader)
- int LoadSingleFile (const char *filename, char *pointer, unsigned long &outlen)
- int ProcessRequest (vtkInformation *request, vtkInformationVector **inputVector, vtkInformationVector *output↩ Vector)
- int RequestData (vtkInformation *request, vtkInformationVector **inputVector, vtkInformationVector *output↩ Vector)
- int RequestDataCompat ()
- int RequestInformation (vtkInformation *request, vtkInformationVector **inputVector, vtkInformationVector *outputVector)
- int RequestInformationCompat ()
- void SetFilePattern (const char *)
- void SetFilePrefix (const char *)
- vtkGetStringMacro (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 ( ) [protected]

10.372.2.2 vtkGDCMImageReader2::~vtkGDCMImageReader2 ( ) [protected]

10.372.3 Member Function Documentation

10.372.3.1 virtual int vtkGDCMImageReader2::CanReadFile (const char * *fname*) [virtual]

10.372.3.2 void vtkGDCMImageReader2::FillMedicalImageInformation (const gdcm::ImageReader & *reader*) [protected]

10.372.3.3 virtual const char* vtkGDCMImageReader2::GetDescriptiveName ( ) [inline], [virtual]

10.372.3.4 virtual const char* vtkGDCMImageReader2::GetFileExtensions ( ) [inline], [virtual]

10.372.3.5 vtkImageData* vtkGDCMImageReader2::GetIconImage ( )

10.372.3.6 vtkAlgorithmOutput* vtkGDCMImageReader2::GetIconImagePort ( )

10.372.3.7 vtkImageData* vtkGDCMImageReader2::GetOverlay (int *i*)

10.372.3.8 vtkAlgorithmOutput* vtkGDCMImageReader2::GetOverlayPort (int *index*)

10.372.3.9 int vtkGDCMImageReader2::LoadSingleFile (const char * *filename,* char * *pointer,* unsigned long & *outlen*) [protected]

10.372.3.10 static vtkGDCMImageReader2* vtkGDCMImageReader2::New ( ) [static]

Examples:

Compute3DSpacing.cxx.

10.372.3.11 virtual void vtkGDCMImageReader2::PrintSelf (ostream & *os,* vtkIndent *indent*) [virtual]

10.372.3.12 int vtkGDCMImageReader2::ProcessRequest (vtkInformation * *request,* vtkInformationVector ** *inputVector,* vtkInformationVector * *outputVector*) [protected]

10.372.3.13 int vtkGDCMImageReader2::RequestData (vtkInformation * *request,* vtkInformationVector ** *inputVector,* vtkInformationVector * *outputVector*) [protected]

10.372.3.14 int vtkGDCMImageReader2::RequestDataCompat ( ) [protected]

10.372.3.15 int vtkGDCMImageReader2::RequestInformation (vtkInformation * *request,* vtkInformationVector ** *inputVector,* vtkInformationVector * *outputVector*) [protected]

10.372.3.16 int vtkGDCMImageReader2::RequestInformationCompat ( ) [protected]

10.372.3.17 virtual void vtkGDCMImageReader2::SetCurve (vtkPolyData * *pd*) [virtual]

10.372.3.18 void vtkGDCMImageReader2::SetFilePattern (const char *) [inline], [protected]

10.372.3.19 void vtkGDCMImageReader2::SetFilePrefix (const char *) [inline], [protected]

10.372.3.20 virtual void vtkGDCMImageReader2::SetMedicalImageProperties (vtkMedicalImageProperties * *pd*) [virtual]

10.372.3.21 vtkGDCMImageReader2::vtkBooleanMacro (LoadOverlays , int)

10.372.3.22 vtkGDCMImageReader2::vtkBooleanMacro (**LoadIconImage** , int)

10.372.3.23 vtkGDCMImageReader2::vtkBooleanMacro (**LossyFlag** , int)

10.372.3.24 vtkGDCMImageReader2::vtkBooleanMacro (**ApplyLookupTable** , int)

10.372.3.25 int vtkGDCMImageReader2::vtkBooleanMacro (**ApplyYBRToRGB** , int)

10.372.3.26 vtkGDCMImageReader2::vtkGetMacro (**LoadOverlays** , int)

10.372.3.27 vtkGDCMImageReader2::vtkGetMacro (**LoadIconImage** , int)

10.372.3.28 vtkGDCMImageReader2::vtkGetMacro (**LossyFlag** , int)

10.372.3.29 vtkGDCMImageReader2::vtkGetMacro (**NumberOfOverlays** , int)

10.372.3.30 vtkGDCMImageReader2::vtkGetMacro (**NumberOfIconImages** , int)

10.372.3.31 vtkGDCMImageReader2::vtkGetMacro (**ApplyLookupTable** , int)

10.372.3.32 vtkGDCMImageReader2::vtkGetMacro (**ApplyYBRToRGB** , int)

10.372.3.33 vtkGDCMImageReader2::vtkGetMacro (**ImageFormat** , int)

10.372.3.34 vtkGDCMImageReader2::vtkGetMacro (**PlanarConfiguration** , int)

10.372.3.35 vtkGDCMImageReader2::vtkGetMacro (**Shift** , double)

10.372.3.36 vtkGDCMImageReader2::vtkGetMacro (**Scale** , double)

10.372.3.37 vtkGDCMImageReader2::vtkGetObjectMacro (**DirectionCosines** , vtkMatrix4x4)

10.372.3.38 vtkGDCMImageReader2::vtkGetObjectMacro (**Curve** , vtkPolyData)

10.372.3.39 vtkGDCMImageReader2::vtkGetStringMacro (**FilePrefix**) [protected]

10.372.3.40 vtkGDCMImageReader2::vtkGetStringMacro (**FilePattern**) [protected]

10.372.3.41 vtkGDCMImageReader2::vtkGetVector3Macro (**ImagePositionPatient** , double)

10.372.3.42 vtkGDCMImageReader2::vtkGetVector6Macro (**ImageOrientationPatient** , double)

10.372.3.43 vtkGDCMImageReader2::vtkSetMacro (LoadOverlays , int)

10.372.3.44 vtkGDCMImageReader2::vtkSetMacro (LoadIconImage , int)

10.372.3.45 vtkGDCMImageReader2::vtkSetMacro (LossyFlag , int)

10.372.3.46 vtkGDCMImageReader2::vtkSetMacro (ApplyLookupTable , int)

10.372.3.47 vtkGDCMImageReader2::vtkSetVector6Macro (ImageOrientationPatient , double) [protected]

10.372.3.48 vtkGDCMImageReader2::vtkTypeMacro (vtkGDCMImageReader2 , vtkMedicalImageReader2)

10.372.4 Member Data Documentation

10.372.4.1 int vtkGDCMImageReader2::ApplyInverseVideo [protected]

10.372.4.2 int vtkGDCMImageReader2::ApplyLookupTable [protected]

10.372.4.3 int vtkGDCMImageReader2::ApplyPlanarConfiguration [protected]

10.372.4.4 int vtkGDCMImageReader2::ApplyShiftScale [protected]

10.372.4.5 int vtkGDCMImageReader2::ApplyYBRToRGB [protected]

10.372.4.6 vtkPolyData* vtkGDCMImageReader2::Curve [protected]

10.372.4.7 vtkMatrix4x4* vtkGDCMImageReader2::DirectionCosines [protected]

10.372.4.8 int vtkGDCMImageReader2::ForceRescale [protected]

10.372.4.9 int vtkGDCMImageReader2::IconDataScalarType [protected]

10.372.4.10 int vtkGDCMImageReader2::IconImageDataExtent[6] [protected]

10.372.4.11 int vtkGDCMImageReader2::IconNumberOfScalarComponents [protected]

10.372.4.12 int vtkGDCMImageReader2::ImageFormat [protected]

10.372.4.13 double vtkGDCMImageReader2::ImageOrientationPatient[6] [protected]

10.372.4.14 double vtkGDCMImageReader2::ImagePositionPatient[3] [protected]

10.372.4.15 int vtkGDCMImageReader2::LoadIconImage [protected]

10.372.4.16 int vtkGDCMImageReader2::LoadOverlays [protected]

10.372.4.17 int vtkGDCMImageReader2::LossyFlag [protected]

10.372.4.18 int vtkGDCMImageReader2::NumberOfIconImages [protected]

10.372.4.19 int vtkGDCMImageReader2::NumberOfOverlays [protected]

10.372.4.20 int vtkGDCMImageReader2::PlanarConfiguration [protected]

10.372.4.21 double vtkGDCMImageReader2::Scale [protected]

10.372.4.22 double vtkGDCMImageReader2::Shift [protected]

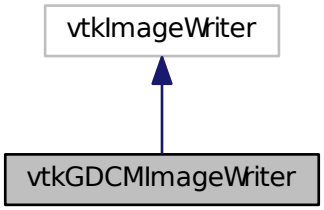
The documentation for this class was generated from the following file:

- vtkGDCMImageReader2.h

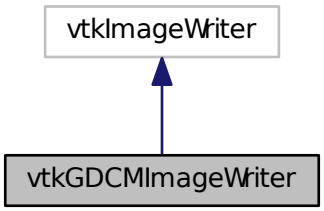
10.373 vtkGDCMImageWriter Class Reference

`#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)
- vtkTypeMacro (vtkGDCMImageWriter, vtkImageWriter)
- virtual void Write ()

Static Public Member Functions

- static vtkGDCMImageWriter * New ()

Protected Member Functions

- vtkGDCMImageWriter ()
- ~vtkGDCMImageWriter ()
- virtual char * GetFileName ()
- int WriteGDCMData (vtkImageData *data, int timeStep)
- void WriteSlice (vtkImageData *data)

10.373.1 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 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 ( ) [protected]

10.373.3.2 vtkGDCMImageWriter::~vtkGDCMImageWriter ( ) [protected]

10.373.4 Member Function Documentation

10.373.4.1 virtual const char* vtkGDCMImageWriter::GetDescriptiveName ( ) [inline], [virtual]

10.373.4.2 virtual const char* vtkGDCMImageWriter::GetFileExtensions ( ) [inline], [virtual]

10.373.4.3 virtual char* vtkGDCMImageWriter::GetFileName ( ) [protected], [virtual]

10.373.4.4 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 virtual void vtkGDCMImageWriter::PrintSelf (ostream & *os,* vtkIndent *indent*) [virtual]

10.373.4.6 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 virtual void vtkGDCMImageWriter::SetDirectionCosinesFromImageOrientationPatient (const double *dircos[6]*) [virtual]

10.373.4.8 virtual void vtkGDCMImageWriter::SetFileNames (vtkStringArray *) [virtual]

Examples:

ConvertMultiFrameToSingleFrame.cxx, and CreateFakePET.cxx.

10.373.4.9 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 vtkGDCMImageWriter::vtkBooleanMacro (LossyFlag , int)

10.373.4.11 vtkGDCMImageWriter::vtkBooleanMacro (FileLowerLeft , int)

10.373.4.12 vtkGDCMImageWriter::vtkGetMacro (LossyFlag , int)

10.373.4.13 vtkGDCMImageWriter::vtkGetMacro (Shift , double)

10.373.4.14 vtkGDCMImageWriter::vtkGetMacro (Scale , double)

10.373.4.15 vtkGDCMImageWriter::vtkGetMacro (ImageFormat , int)

10.373.4.16 vtkGDCMImageWriter::vtkGetMacro (FileLowerLeft , int)

10.373.4.17 vtkGDCMImageWriter::vtkGetMacro (PlanarConfiguration , int)

10.373.4.18 vtkGDCMImageWriter::vtkGetMacro (CompressionType , int)

10.373.4.19 vtkGDCMImageWriter::vtkGetObjectMacro (MedicalImageProperties , vtkMedicalImageProperties)

10.373.4.20 vtkGDCMImageWriter::vtkGetObjectMacro (FileNames , vtkStringArray)

10.373.4.21 vtkGDCMImageWriter::vtkGetObjectMacro (DirectionCosines , vtkMatrix4x4)

10.373.4.22 vtkGDCMImageWriter::vtkGetStringMacro (StudyUID)

10.373.4.23 vtkGDCMImageWriter::vtkGetStringMacro (SeriesUID)

10.373.4.24 vtkGDCMImageWriter::vtkSetMacro (LossyFlag , int)

10.373.4.25 vtkGDCMImageWriter::vtkSetMacro (Shift , double)

10.373.4.26 vtkGDCMImageWriter::vtkSetMacro (Scale , double)

10.373.4.27 vtkGDCMImageWriter::vtkSetMacro (ImageFormat , int)

10.373.4.28 vtkGDCMImageWriter::vtkSetMacro (FileLowerLeft , int)

10.373.4.29 vtkGDCMImageWriter::vtkSetMacro (PlanarConfiguration , int)

10.373.4.30 vtkGDCMImageWriter::vtkSetMacro (CompressionType , int)

10.373.4.31 vtkGDCMImageWriter::vtkSetStringMacro (StudyUID)

10.373.4.32 vtkGDCMImageWriter::vtkSetStringMacro (SeriesUID)

10.373.4.33 vtkGDCMImageWriter::vtkTypeMacro (vtkGDCMImageWriter , vtkImageWriter)

10.373.4.34 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 int vtkGDCMImageWriter::WriteGDCMData (vtkImageData * *data*, int *timeStep*) [protected]

10.373.4.36 void vtkGDCMImageWriter::WriteSlice (vtkImageData * *data*) [protected]

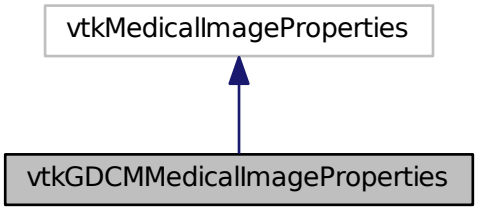
The documentation for this class was generated from the following file:

- vtkGDCMImageWriter.h

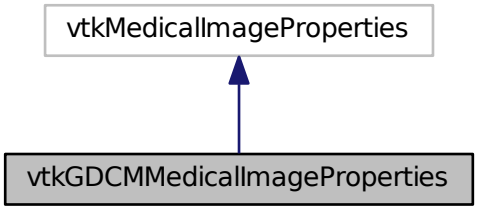
10.374 vtkGDCMMedicalImageProperties Class Reference

`#include <vtkGDCMMedicalImageProperties.h>`

Inheritance diagram for vtkGDCMMedicalImageProperties:



Collaboration diagram for vtkGDCMMedicalImageProperties:



Public Member Functions

- virtual void Clear ()
- void PrintSelf (ostream &os, vtkIndent indent)
- vtkTypeMacro (vtkGDCMMedicalImageProperties, vtkMedicalImageProperties)

Static Public Member Functions

- static vtkGDCMMedicalImageProperties * New ()

Protected Member Functions

- vtkGDCMMedicalImageProperties ()
- ~vtkGDCMMedicalImageProperties ()
- gdcm::File const & GetFile (unsigned int t)
- void PushBackFile (gdcm::File const &f)

Friends

- class vtkGDCMImageReader
- class vtkGDCMImageReader2
- class vtkGDCMImageWriter

10.374.1 Constructor & Destructor Documentation

10.374.1.1 vtkGDCMMedicalImageProperties::vtkGDCMMedicalImageProperties ( ) [protected]

10.374.1.2 vtkGDCMMedicalImageProperties::~vtkGDCMMedicalImageProperties ( ) [protected]

10.374.2 Member Function Documentation

10.374.2.1 virtual void vtkGDCMMedicalImageProperties::Clear ( ) [virtual]

10.374.2.2 gdcm::File const& vtkGDCMMedicalImageProperties::GetFile (unsigned int *t*) [protected]

10.374.2.3 static vtkGDCMMedicalImageProperties* vtkGDCMMedicalImageProperties::New ( ) [static]

10.374.2.4 void vtkGDCMMedicalImageProperties::PrintSelf (ostream & *os*, vtkIndent *indent*)

10.374.2.5 void vtkGDCMMedicalImageProperties::PushBackFile (gdcm::File const & *f*) [protected]

10.374.2.6 vtkGDCMMedicalImageProperties::vtkTypeMacro (vtkGDCMMedicalImageProperties , vtkMedicalImageProperties)

10.374.3 Friends And Related Function Documentation

10.374.3.1 friend class vtkGDCMImageReader [friend]

10.374.3.2 friend class vtkGDCMImageReader2 [friend]

10.374.3.3 friend class vtkGDCMImageWriter [friend]

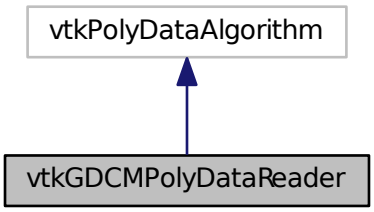
The documentation for this class was generated from the following file:

- vtkGDCMMedicalImageProperties.h

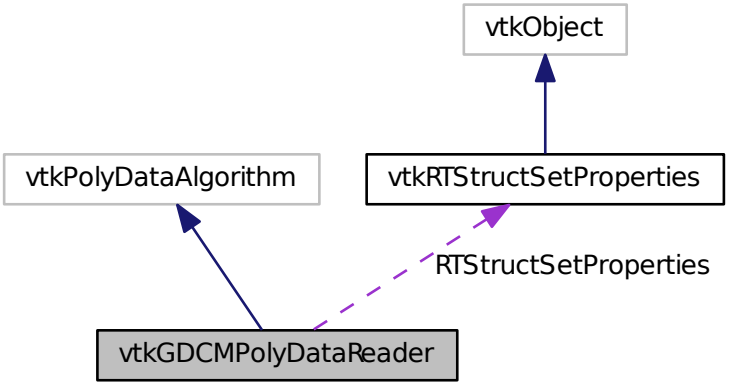
10.375 vtkGDCMPolyDataReader Class Reference

`#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)
- vtkTypeMacro (vtkGDCMPolyDataReader, vtkPolyDataAlgorithm)

Static Public Member Functions

- static vtkGDCMPolyDataReader * New ()

Protected Member Functions

- vtkGDCMPolyDataReader ()
- ~vtkGDCMPolyDataReader ()
- void FillMedicalImageInformation (const gdcm::Reader &reader)
- int RequestData (vtkInformation *, vtkInformationVector **, vtkInformationVector *)
- int RequestData_HemodynamicWaveformStorage (gdcm::Reader const &reader, vtkInformationVector *output↩ Vector)
- int RequestData_RTStructureSetStorage (gdcm::Reader const &reader, vtkInformationVector *outputVector)
- int RequestInformation (vtkInformation *vtkNotUsed(request), vtkInformationVector **vtkNotUsed(inputVector), vtkInformationVector *outputVector)
- int RequestInformation_HemodynamicWaveformStorage (gdcm::Reader const &reader)
- int RequestInformation_RTStructureSetStorage (gdcm::Reader const &reader)

Protected Attributes

- char * FileName
- vtkMedicalImageProperties * MedicalImageProperties
- vtkRTStructSetProperties * RTStructSetProperties

10.375.1 Detailed Description

Examples:

gdcmscene.cxx, GenerateRTSTRUCT.cxx, and rtstructapp.cxx.

10.375.2 Constructor & Destructor Documentation

10.375.2.1 vtkGDCMPolyDataReader::vtkGDCMPolyDataReader ( ) [protected]

10.375.2.2 vtkGDCMPolyDataReader::~vtkGDCMPolyDataReader ( ) [protected]

10.375.3 Member Function Documentation

10.375.3.1 void vtkGDCMPolyDataReader::FillMedicalImageInformation (const gdcm::Reader & *reader*) [protected]

10.375.3.2 static vtkGDCMPolyDataReader* vtkGDCMPolyDataReader::New ( ) [static]

Examples:

gdcmscene.cxx, GenerateRTSTRUCT.cxx, and rtstructapp.cxx.

10.375.3.3 virtual void vtkGDCMPolyDataReader::PrintSelf (ostream & *os,* vtkIndent *indent*) [virtual]

10.375.3.4 int vtkGDCMPolyDataReader::RequestData (vtkInformation * , vtkInformationVector ** , vtkInformationVector *) [protected]

10.375.3.5 int vtkGDCMPolyDataReader::RequestData_HemodynamicWaveformStorage (gdcm::Reader const & *reader,* vtkInformationVector * *outputVector*) [protected]

10.375.3.6 int vtkGDCMPolyDataReader::RequestData_RTStructureSetStorage (gdcm::Reader const & *reader,* vtkInformationVector * *outputVector*) [protected]

10.375.3.7 int vtkGDCMPolyDataReader::RequestInformation (vtkInformation * *vtkNotUsedrequest,* vtkInformationVector ** *vtkNotUsedinputVector,* vtkInformationVector * *outputVector*) [protected]

10.375.3.8 int vtkGDCMPolyDataReader::RequestInformation_HemodynamicWaveformStorage (gdcm::Reader const & *reader*) [protected]

10.375.3.9 int vtkGDCMPolyDataReader::RequestInformation_RTStructureSetStorage (gdcm::Reader const & *reader*) [protected]

10.375.3.10 vtkGDCMPolyDataReader::vtkGetObjectMacro (MedicalImageProperties , vtkMedicalImageProperties)

10.375.3.11 vtkGDCMPolyDataReader::vtkGetObjectMacro (RTStructSetProperties , vtkRTStructSetProperties)

10.375.3.12 vtkGDCMPolyDataReader::vtkGetStringMacro (FileName)

10.375.3.13 vtkGDCMPolyDataReader::vtkSetStringMacro (FileName)

10.375.3.14 vtkGDCMPolyDataReader::vtkTypeMacro (vtkGDCMPolyDataReader , vtkPolyDataAlgorithm)

10.375.4 Member Data Documentation

10.375.4.1 char* vtkGDCMPolyDataReader::FileName [protected]

10.375.4.2 vtkMedicalImageProperties* vtkGDCMPolyDataReader::MedicalImageProperties [protected]

10.375.4.3 vtkRTStructSetProperties* vtkGDCMPolyDataReader::RTStructSetProperties [protected]

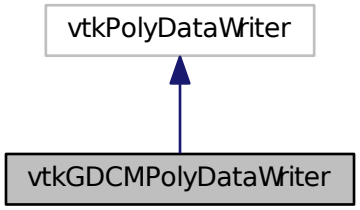
The documentation for this class was generated from the following file:

- vtkGDCMPolyDataReader.h

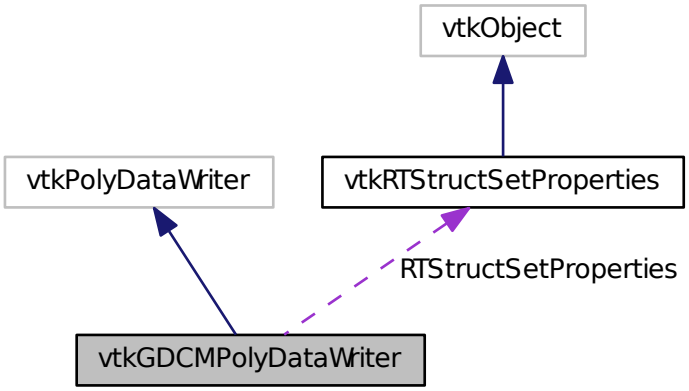
10.376 vtkGDCMPolyDataWriter Class Reference

`#include <vtkGDCMPolyDataWriter.h>`

Inheritance diagram for vtkGDCMPolyDataWriter:



Collaboration diagram for vtkGDCMPolyDataWriter:



Public Member Functions

- void InitializeRTStructSet (vtkStdString inDirectory, vtkStdString inStructLabel, vtkStdString inStructName, vtk↩ StringArray ∗inROINames, vtkStringArray ∗inROIAlgorithmName, vtkStringArray ∗inROIType)
- virtual void PrintSelf (ostream &os, vtkIndent indent)
- virtual void SetMedicalImageProperties (vtkMedicalImageProperties ∗pd)
- void SetNumberOfInputPorts (int n)
- virtual void SetRTStructSetProperties (vtkRTStructSetProperties ∗pd)
- vtkTypeMacro (vtkGDCMPolyDataWriter, vtkPolyDataWriter)

Static Public Member Functions

- static vtkGDCMPolyDataWriter * New ()

Protected Member Functions

- vtkGDCMPolyDataWriter ()
- ~vtkGDCMPolyDataWriter ()
- void WriteData ()
- void WriteRTSTRUCTData (gdcm::File &file, int num)
- void WriteRTSTRUCTInfo (gdcm::File &file)

Protected Attributes

- vtkMedicalImageProperties * MedicalImageProperties
- vtkRTStructSetProperties * RTStructSetProperties

10.376.1 Detailed Description

Examples:

GenerateRTSTRUCT.cxx, and rtstructapp.cxx.

10.376.2 Constructor & Destructor Documentation

10.376.2.1 vtkGDCMPolyDataWriter::vtkGDCMPolyDataWriter ( ) [protected]

10.376.2.2 vtkGDCMPolyDataWriter::~vtkGDCMPolyDataWriter ( ) [protected]

10.376.3 Member Function Documentation

10.376.3.1 void vtkGDCMPolyDataWriter::InitializeRTStructSet (vtkStdString *inDirectory,* vtkStdString *inStructLabel,* vtkStdString *inStructName,* vtkStringArray * *inROINames,* vtkStringArray * *inROIAlgorithmName,* vtkStringArray * *inROIType*)

Examples:

GenerateRTSTRUCT.cxx.

10.376.3.2 static vtkGDCMPolyDataWriter* vtkGDCMPolyDataWriter::New ( ) [static]

Examples:

GenerateRTSTRUCT.cxx, and rtstructapp.cxx.

10.376.3.3 virtual void vtkGDCMPolyDataWriter::PrintSelf (ostream & *os,* vtkIndent *indent*) [virtual]

10.376.3.4 virtual void vtkGDCMPolyDataWriter::SetMedicalImageProperties (vtkMedicalImageProperties * *pd*) [virtual]

Examples:

GenerateRTSTRUCT.cxx, and rtstructapp.cxx.

10.376.3.5 void vtkGDCMPolyDataWriter::SetNumberOfInputPorts (int *n*)

Examples:

GenerateRTSTRUCT.cxx, and rtstructapp.cxx.

10.376.3.6 virtual void vtkGDCMPolyDataWriter::SetRTStructSetProperties (**vtkRTStructSetProperties** * *pd*) [virtual]

Examples:

GenerateRTSTRUCT.cxx, and rtstructapp.cxx.

10.376.3.7 vtkGDCMPolyDataWriter::vtkTypeMacro (**vtkGDCMPolyDataWriter** , vtkPolyDataWriter)

10.376.3.8 void vtkGDCMPolyDataWriter::WriteData ( ) [protected]

10.376.3.9 void vtkGDCMPolyDataWriter::WriteRTSTRUCTData (**gdcm::File** & *file,* int *num*) [protected]

10.376.3.10 void vtkGDCMPolyDataWriter::WriteRTSTRUCTInfo (**gdcm::File** & *file*) [protected]

10.376.4 Member Data Documentation

10.376.4.1 vtkMedicalImageProperties* vtkGDCMPolyDataWriter::MedicalImageProperties [protected]

10.376.4.2 **vtkRTStructSetProperties*** vtkGDCMPolyDataWriter::RTStructSetProperties [protected]

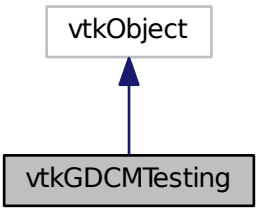
The documentation for this class was generated from the following file:

- vtkGDCMPolyDataWriter.h

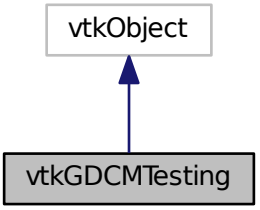
10.377 vtkGDCMTesting Class Reference

`#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)
- vtkTypeMacro (vtkGDCMTesting, vtkObject)

Static Public Member Functions

- static const char * GetGDCMDataRoot ()
- static const char *const * GetMD5MetaImage (unsigned int file)
- static const char * GetMHDMD5FromFile (const char *filepath)
- static unsigned int GetNumberOfMD5MetaImages ()
- static const char * GetRAWMD5FromFile (const char *filepath)
- static const char * GetVTKDataRoot ()
- static vtkGDCMTesting * New ()

Protected Member Functions

- vtkGDCMTesting ()
- ~vtkGDCMTesting ()

10.377.1 Detailed Description

Examples:

HelloActiviz5.cs, HelloVTKWorld2.cs, MetaImageMD5Activiz.cs, ReadSeriesIntoVTK.java, and RefCounting.cs.

10.377.2 Member Typedef Documentation

10.377.2.1 typedef const char* const(* vtkGDCMTesting::MD5MetaImagesType)[3]

10.377.3 Constructor & Destructor Documentation

10.377.3.1 vtkGDCMTesting::vtkGDCMTesting ( ) [protected]

10.377.3.2 vtkGDCMTesting::~vtkGDCMTesting ( ) [protected]

10.377.4 Member Function Documentation

10.377.4.1 static const char* vtkGDCMTesting::GetGDCMDataRoot ( ) [static]

Examples:

HelloActiviz5.cs, and ReadSeriesIntoVTK.java.

10.377.4.2 static const char* const* vtkGDCMTesting::GetMD5MetaImage (unsigned int *file*) [static]

10.377.4.3 static const char* vtkGDCMTesting::GetMHDMD5FromFile (const char * *filepath*) [static]

Examples:

MetaImageMD5Activiz.cs.

10.377.4.4 static unsigned int vtkGDCMTesting::GetNumberOfMD5MetaImages ( ) [static]

10.377.4.5 static const char* vtkGDCMTesting::GetRAWMD5FromFile (const char * *filepath*) [static]

Examples:

MetaImageMD5Activiz.cs.

10.377.4.6 static const char* vtkGDCMTesting::GetVTKDataRoot ( ) [static]

Examples:

HelloActiviz5.cs, and HelloVTKWorld2.cs.

10.377.4.7 static vtkGDCMTesting* vtkGDCMTesting::New ( ) [static]

Examples:

RefCounting.cs.

10.377.4.8 void vtkGDCMTesting::PrintSelf (ostream & *os,* vtkIndent *indent*)

10.377.4.9 vtkGDCMTesting::vtkTypeMacro (vtkGDCMTesting , vtkObject)

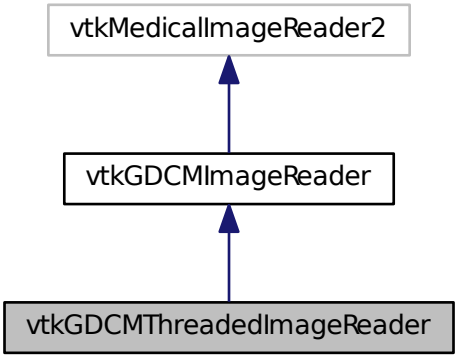
The documentation for this class was generated from the following file:

- vtkGDCMTesting.h

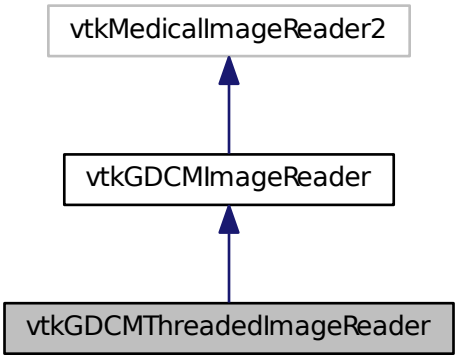
10.378 vtkGDCMThreadedImageReader Class Reference

`#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)
- vtkTypeMacro (vtkGDCMThreadedImageReader, vtkGDCMImageReader)

Static Public Member Functions

- static vtkGDCMThreadedImageReader * New ()

Protected Member Functions

- vtkGDCMThreadedImageReader ()
- ~vtkGDCMThreadedImageReader ()
- void ExecuteData (vtkDataObject *out)
- void ExecuteInformation ()
- void ReadFiles (unsigned int nfiles, const char *filenames[])
- void RequestDataCompat ()

Additional Inherited Members

10.378.1 Constructor & Destructor Documentation

10.378.1.1 vtkGDCMThreadedImageReader::vtkGDCMThreadedImageReader ( ) [protected]

10.378.1.2 vtkGDCMThreadedImageReader::∼vtkGDCMThreadedImageReader ( ) [protected]

10.378.2 Member Function Documentation

10.378.2.1 void vtkGDCMThreadedImageReader::ExecuteData (vtkDataObject * *out*) [protected]

10.378.2.2 void vtkGDCMThreadedImageReader::ExecuteInformation ( ) [protected]

10.378.2.3 static vtkGDCMThreadedImageReader* vtkGDCMThreadedImageReader::New ( ) [static]

10.378.2.4 virtual void vtkGDCMThreadedImageReader::PrintSelf (ostream & *os,* vtkIndent *indent*) [virtual]

Reimplemented from vtkGDCMImageReader.

10.378.2.5 void vtkGDCMThreadedImageReader::ReadFiles (unsigned int *nfiles,* const char * *filenames[ ]*) [protected]

10.378.2.6 void vtkGDCMThreadedImageReader::RequestDataCompat ( ) [protected]

10.378.2.7 vtkGDCMThreadedImageReader::vtkBooleanMacro (UseShiftScale , int)

10.378.2.8 vtkGDCMThreadedImageReader::vtkGetMacro (UseShiftScale , int)

10.378.2.9 vtkGDCMThreadedImageReader::vtkSetMacro (Shift , double)

10.378.2.10 vtkGDCMThreadedImageReader::vtkSetMacro (Scale , double)

10.378.2.11 vtkGDCMThreadedImageReader::vtkSetMacro (UseShiftScale , int)

10.378.2.12 vtkGDCMThreadedImageReader::vtkTypeMacro (vtkGDCMThreadedImageReader , vtkGDCMImageReader)

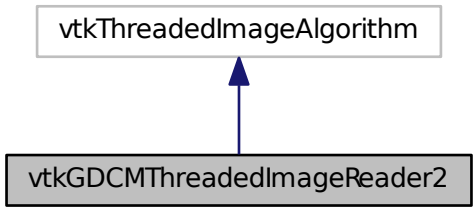
The documentation for this class was generated from the following file:

- vtkGDCMThreadedImageReader.h

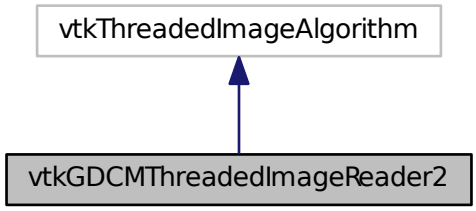
10.379 vtkGDCMThreadedImageReader2 Class Reference

`#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)
- vtkTypeMacro (vtkGDCMThreadedImageReader2, vtkThreadedImageAlgorithm)

Static Public Member Functions

- static vtkGDCMThreadedImageReader2 * New ()

Protected Member Functions

- vtkGDCMThreadedImageReader2 ()
- ~vtkGDCMThreadedImageReader2 ()
- int RequestInformation (vtkInformation *request, vtkInformationVector **inputVector, vtkInformationVector *outputVector)
- void ThreadedRequestData (vtkInformation *request, vtkInformationVector **inputVector, vtkInformationVector *outputVector, vtkImageData ***inData, vtkImageData **outData, int outExt[6], int id)

10.379.1 Constructor & Destructor Documentation

10.379.1.1 vtkGDCMThreadedImageReader2::vtkGDCMThreadedImageReader2 ( ) [protected]

10.379.1.2 vtkGDCMThreadedImageReader2::~vtkGDCMThreadedImageReader2 ( ) [protected]

10.379.2 Member Function Documentation

10.379.2.1 virtual const char* vtkGDCMThreadedImageReader2::GetFileName (int *i* = 0) [virtual]

10.379.2.2 static vtkGDCMThreadedImageReader2* vtkGDCMThreadedImageReader2::New ( ) [static]

10.379.2.3 virtual void vtkGDCMThreadedImageReader2::PrintSelf (ostream & *os,* vtkIndent *indent*) [virtual]

10.379.2.4 int vtkGDCMThreadedImageReader2::RequestInformation (vtkInformation * *request,* vtkInformationVector ** *inputVector,* vtkInformationVector * *outputVector*) [protected]

10.379.2.5 virtual void vtkGDCMThreadedImageReader2::SetFileName (const char * *filename*) [virtual]

10.379.2.6 virtual void vtkGDCMThreadedImageReader2::SetFileNames (vtkStringArray *) [virtual]

10.379.2.7 int vtkGDCMThreadedImageReader2::SplitExtent (int *splitExt[6],* int *startExt[6],* int *num,* int *total*)

10.379.2.8 void vtkGDCMThreadedImageReader2::ThreadedRequestData (vtkInformation * *request,* vtkInformationVector ** *inputVector,* vtkInformationVector * *outputVector,* vtkImageData *** *inData,* vtkImageData ** *outData,* int *outExt[6],* int *id*) [protected]

10.379.2.9 vtkGDCMThreadedImageReader2::vtkBooleanMacro (FileLowerLeft , int)

10.379.2.10 vtkGDCMThreadedImageReader2::vtkBooleanMacro (LoadOverlays , int)

10.379.2.11 vtkGDCMThreadedImageReader2::vtkBooleanMacro (UseShiftScale , int)

10.379.2.12 vtkGDCMThreadedImageReader2::vtkGetMacro (FileLowerLeft , int)

10.379.2.13 vtkGDCMThreadedImageReader2::vtkGetMacro (NumberOfOverlays , int)

10.379.2.14 vtkGDCMThreadedImageReader2::vtkGetMacro (DataScalarType , int)

10.379.2.15 vtkGDCMThreadedImageReader2::vtkGetMacro (NumberOfScalarComponents , int)

10.379.2.16 vtkGDCMThreadedImageReader2::vtkGetMacro (LoadOverlays , int)

10.379.2.17 vtkGDCMThreadedImageReader2::vtkGetMacro (Shift , double)

10.379.2.18 vtkGDCMThreadedImageReader2::vtkGetMacro (Scale , double)

10.379.2.19 vtkGDCMThreadedImageReader2::vtkGetMacro (UseShiftScale , int)

10.379.2.20 vtkGDCMThreadedImageReader2::vtkGetObjectMacro (FileNames , vtkStringArray)

10.379.2.21 vtkGDCMThreadedImageReader2::vtkGetVector3Macro (DataOrigin , double)

10.379.2.22 vtkGDCMThreadedImageReader2::vtkGetVector3Macro (DataSpacing , double)

10.379.2.23 vtkGDCMThreadedImageReader2::vtkGetVector6Macro (DataExtent , int)

10.379.2.24 vtkGDCMThreadedImageReader2::vtkSetMacro (FileLowerLeft , int)

10.379.2.25 vtkGDCMThreadedImageReader2::vtkSetMacro (DataScalarType , int)

10.379.2.26 vtkGDCMThreadedImageReader2::vtkSetMacro (NumberOfScalarComponents , int)

10.379.2.27 vtkGDCMThreadedImageReader2::vtkSetMacro (LoadOverlays , int)

10.379.2.28 vtkGDCMThreadedImageReader2::vtkSetMacro (Shift , double)

10.379.2.29 vtkGDCMThreadedImageReader2::vtkSetMacro (Scale , double)

10.379.2.30 vtkGDCMThreadedImageReader2::vtkSetMacro (UseShiftScale , int)

10.379.2.31 vtkGDCMThreadedImageReader2::vtkSetVector3Macro (DataOrigin , double)

10.379.2.32 vtkGDCMThreadedImageReader2::vtkSetVector3Macro (DataSpacing , double)

10.379.2.33 vtkGDCMThreadedImageReader2::vtkSetVector6Macro (DataExtent , int)

10.379.2.34 vtkGDCMThreadedImageReader2::vtkTypeMacro (vtkGDCMThreadedImageReader2 , vtkThreadedImageAlgorithm)

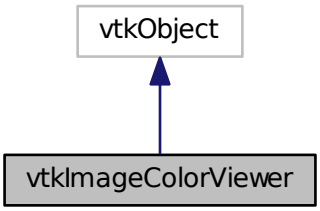
The documentation for this class was generated from the following file:

- vtkGDCMThreadedImageReader2.h

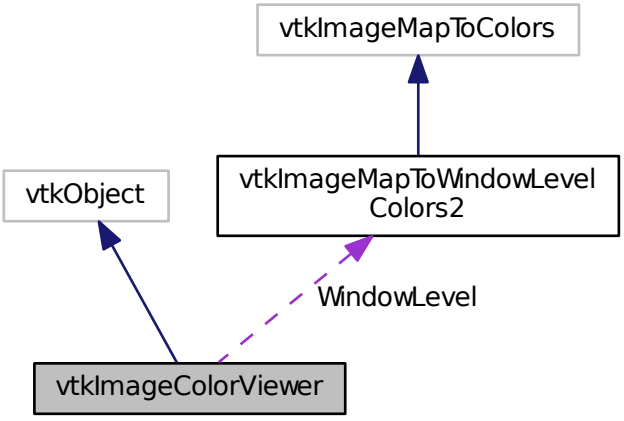
10.380 vtkImageColorViewer Class Reference

`#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)
- vtkTypeMacro (vtkImageColorViewer, vtkObject)

Static Public Member Functions

- static vtkImageColorViewer * New ()

Protected Member Functions

- vtkImageColorViewer ()
- ~vtkImageColorViewer ()
- virtual void InstallPipeline ()
- virtual void UnInstallPipeline ()
- virtual void UpdateOrientation ()

Protected Attributes

- int FirstRender
- vtkImageActor * ImageActor
- vtkRenderWindowInteractor * Interactor
- vtkInteractorStyleImage * InteractorStyle
- vtkImageActor * OverlayImageActor
- vtkRenderer * Renderer
- vtkRenderWindow * RenderWindow
- int Slice
- int SliceOrientation
- vtkImageMapToWindowLevelColors2 * WindowLevel

Friends

- class vtkImageColorViewerCallback

10.380.1 Detailed Description

Examples:

gdcmrtionplan.cxx, and gdcmrtplan.cxx.

10.380.2 Member Enumeration Documentation

10.380.2.1 anonymous enum

Enumerator

SLICE_ORIENTATION_YZ

SLICE_ORIENTATION_XZ

SLICE_ORIENTATION_XY

10.380.3 Constructor & Destructor Documentation

10.380.3.1 vtkImageColorViewer::vtkImageColorViewer ( ) [protected]

10.380.3.2 vtkImageColorViewer::~vtkImageColorViewer ( ) [protected]

10.380.4 Member Function Documentation

10.380.4.1 virtual void vtkImageColorViewer::AddInput (vtkImageData * *input*) [virtual]

10.380.4.2 virtual void vtkImageColorViewer::AddInputConnection (vtkAlgorithmOutput * *input*) [virtual]

10.380.4.3 virtual double vtkImageColorViewer::GetColorLevel ( ) [virtual]

10.380.4.4 virtual double vtkImageColorViewer::GetColorWindow ( ) [virtual]

10.380.4.5 virtual vtkImageData* vtkImageColorViewer::GetInput ( ) [virtual]

10.380.4.6 virtual int vtkImageColorViewer::GetOffScreenRendering ( ) [virtual]

10.380.4.7 double vtkImageColorViewer::GetOverlayVisibility ( )

10.380.4.8 virtual int* vtkImageColorViewer::GetPosition ( ) [virtual]

10.380.4.9 virtual int* vtkImageColorViewer::GetSize ( ) [virtual]

10.380.4.10 virtual int vtkImageColorViewer::GetSliceMax ( ) [virtual]

10.380.4.11 virtual int vtkImageColorViewer::GetSliceMin ( ) [virtual]

10.380.4.12 virtual void vtkImageColorViewer::GetSliceRange (int *range[2]*) [inline], [virtual]

10.380.4.13 virtual void vtkImageColorViewer::GetSliceRange (int & *min,* int & *max*) [virtual]

10.380.4.14 virtual int* vtkImageColorViewer::GetSliceRange ( ) [virtual]

10.380.4.15 virtual const char* vtkImageColorViewer::GetWindowName ( ) [virtual]

10.380.4.16 virtual void vtkImageColorViewer::InstallPipeline ( ) [protected], [virtual]

10.380.4.17 static vtkImageColorViewer* vtkImageColorViewer::New ( ) [static]

Examples:

gdcmrtionplan.cxx, and gdcmrtplan.cxx.

10.380.4.18 void vtkImageColorViewer::PrintSelf (ostream & *os,* vtkIndent *indent*)

10.380.4.19 virtual void vtkImageColorViewer::Render (void) [virtual]

Examples:

gdcmrtionplan.cxx, and gdcmrtplan.cxx.

10.380.4.20 virtual void vtkImageColorViewer::SetColorLevel (double *s*) [virtual]

10.380.4.21 virtual void vtkImageColorViewer::SetColorWindow (double *s*) [virtual]

10.380.4.22 virtual void vtkImageColorViewer::SetDisplayId (void * *a*) [virtual]

10.380.4.23 virtual void vtkImageColorViewer::SetInput (vtkImageData * *in*) [virtual]

Examples:

gdcmrtionplan.cxx, and gdcmrtplan.cxx.

10.380.4.24 virtual void vtkImageColorViewer::SetInputConnection (vtkAlgorithmOutput * *input*) [virtual]

10.380.4.25 virtual void vtkImageColorViewer::SetOffScreenRendering (int) [virtual]

10.380.4.26 void vtkImageColorViewer::SetOverlayVisibility (double *vis*)

10.380.4.27 virtual void vtkImageColorViewer::SetParentId (void * *a*) [virtual]

10.380.4.28 virtual void vtkImageColorViewer::SetPosition (int *a,* int *b*) [virtual]

10.380.4.29 virtual void vtkImageColorViewer::SetPosition (int *a[2]*) [inline], [virtual]

References SetPosition().

Referenced by SetPosition().

10.380.4.30 virtual void vtkImageColorViewer::SetRenderer (vtkRenderer * *arg*) [virtual]

10.380.4.31 virtual void vtkImageColorViewer::SetRenderWindow (vtkRenderWindow * *arg*) [virtual]

10.380.4.32 virtual void vtkImageColorViewer::SetSize (int *a,* int *b*) [virtual]

Examples:

gdcmrtionplan.cxx, and gdcmrtplan.cxx.

10.380.4.33 virtual void vtkImageColorViewer::SetSize (int *a[2]*) [inline], [virtual]

References SetSize().

Referenced by SetSize().

10.380.4.34 virtual void vtkImageColorViewer::SetSlice (int *s*) [virtual]

10.380.4.35 virtual void vtkImageColorViewer::SetSliceOrientation (int *orientation*) [virtual]

10.380.4.36 virtual void vtkImageColorViewer::SetSliceOrientationToXY ( ) [inline], [virtual]

References SLICE_ORIENTATION_XY.

10.380.4.37 virtual void vtkImageColorViewer::SetSliceOrientationToXZ ( ) [inline], [virtual]

References SLICE_ORIENTATION_XZ.

10.380.4.38 virtual void vtkImageColorViewer::SetSliceOrientationToYZ ( ) [inline], [virtual]

References SLICE_ORIENTATION_YZ.

10.380.4.39 virtual void vtkImageColorViewer::SetupInteractor (vtkRenderWindowInteractor *) [virtual]

Examples:

gdcmrtionplan.cxx, and gdcmrtplan.cxx.

10.380.4.40 virtual void vtkImageColorViewer::SetWindowId (void * *a*) [virtual]

10.380.4.41 virtual void vtkImageColorViewer::UnInstallPipeline ( ) [protected], [virtual]

10.380.4.42 virtual void vtkImageColorViewer::UpdateDisplayExtent ( ) [virtual]

10.380.4.43 virtual void vtkImageColorViewer::UpdateOrientation ( ) [protected], [virtual]

10.380.4.44 vtkImageColorViewer::VTK_LEGACY (int *GetWholeZMin()*)

10.380.4.45 vtkImageColorViewer::VTK_LEGACY (int *GetWholeZMax()*)

10.380.4.46 vtkImageColorViewer::VTK_LEGACY (int *GetZSlice()*)

10.380.4.47 vtkImageColorViewer::VTK_LEGACY (void *SetZSliceint*)

10.380.4.48 vtkImageColorViewer::vtkBooleanMacro (OffScreenRendering , int)

10.380.4.49 vtkImageColorViewer::vtkGetMacro (SliceOrientation , int)

10.380.4.50 vtkImageColorViewer::vtkGetMacro (Slice , int)

10.380.4.51 vtkImageColorViewer::vtkGetObjectMacro (RenderWindow , vtkRenderWindow)

10.380.4.52 vtkImageColorViewer::vtkGetObjectMacro (Renderer , vtkRenderer)

10.380.4.53 vtkImageColorViewer::vtkGetObjectMacro (ImageActor , vtkImageActor)

10.380.4.54 vtkImageColorViewer::vtkGetObjectMacro (WindowLevel , vtkImageMapToWindowLevelColors2)

10.380.4.55 vtkImageColorViewer::vtkGetObjectMacro (InteractorStyle , vtkInteractorStyleImage)

10.380.4.56 vtkImageColorViewer::vtkTypeMacro (vtkImageColorViewer , vtkObject)

10.380.5 Friends And Related Function Documentation

10.380.5.1 friend class vtkImageColorViewerCallback [friend]

10.380.6 Member Data Documentation

10.380.6.1 int vtkImageColorViewer::FirstRender [protected]

10.380.6.2 vtkImageActor* vtkImageColorViewer::ImageActor [protected]

10.380.6.3 vtkRenderWindowInteractor* vtkImageColorViewer::Interactor [protected]

10.380.6.4 vtkInteractorStyleImage* vtkImageColorViewer::InteractorStyle [protected]

10.380.6.5 vtkImageActor* vtkImageColorViewer::OverlayImageActor [protected]

10.380.6.6 vtkRenderer* vtkImageColorViewer::Renderer [protected]

10.380.6.7 vtkRenderWindow* vtkImageColorViewer::RenderWindow [protected]

10.380.6.8 int vtkImageColorViewer::Slice [protected]

10.380.6.9 int vtkImageColorViewer::SliceOrientation [protected]

10.380.6.10 vtkImageMapToWindowLevelColors2* vtkImageColorViewer::WindowLevel [protected]

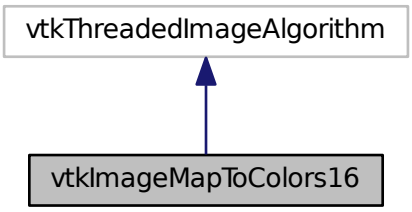
The documentation for this class was generated from the following file:

- vtkImageColorViewer.h

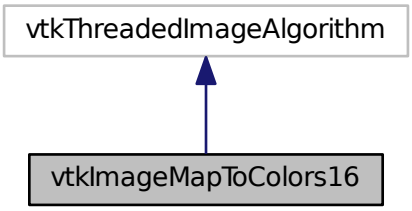
10.381 vtkImageMapToColors16 Class Reference

`#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)
- vtkTypeMacro (vtkImageMapToColors16, vtkThreadedImageAlgorithm)

Static Public Member Functions

- static vtkImageMapToColors16 * New ()

Protected Member Functions

- vtkImageMapToColors16 ()
- ~vtkImageMapToColors16 ()
- virtual int RequestData (vtkInformation *request, vtkInformationVector **inputVector, vtkInformationVector *outputVector)
- virtual int RequestInformation (vtkInformation *, vtkInformationVector **, vtkInformationVector *)
- void ThreadedRequestData (vtkInformation *request, vtkInformationVector **inputVector, vtkInformationVector *outputVector, vtkImageData ***inData, vtkImageData **outData, int extent[6], int id)

Protected Attributes

- int ActiveComponent
- int DataWasPassed
- vtkScalarsToColors * LookupTable
- int OutputFormat
- int PassAlphaToOutput

10.381.1 Constructor & Destructor Documentation

10.381.1.1 vtkImageMapToColors16::vtkImageMapToColors16 ( ) [protected]

10.381.1.2 vtkImageMapToColors16::~vtkImageMapToColors16 ( ) [protected]

10.381.2 Member Function Documentation

10.381.2.1 virtual unsigned long vtkImageMapToColors16::GetMTime ( ) [virtual]

10.381.2.2 static vtkImageMapToColors16* vtkImageMapToColors16::New ( ) [static]

10.381.2.3 void vtkImageMapToColors16::PrintSelf (ostream & *os*, vtkIndent *indent*)

10.381.2.4 virtual int vtkImageMapToColors16::RequestData (vtkInformation * *request,* vtkInformationVector ** *inputVector,* vtkInformationVector * *outputVector*) [protected], [virtual]

10.381.2.5 virtual int vtkImageMapToColors16::RequestInformation (vtkInformation * , vtkInformationVector ** , vtkInformationVector *) [protected], [virtual]

10.381.2.6 virtual void vtkImageMapToColors16::SetLookupTable (vtkScalarsToColors *) [virtual]

10.381.2.7 void vtkImageMapToColors16::SetOutputFormatToLuminance ( ) [inline]

10.381.2.8 void vtkImageMapToColors16::SetOutputFormatToLuminanceAlpha ( ) [inline]

10.381.2.9 void vtkImageMapToColors16::SetOutputFormatToRGB ( ) [inline]

10.381.2.10 void vtkImageMapToColors16::SetOutputFormatToRGBA ( ) [inline]

10.381.2.11 void vtkImageMapToColors16::ThreadedRequestData (vtkInformation * *request,* vtkInformationVector ** *inputVector,* vtkInformationVector * *outputVector,* vtkImageData *** *inData,* vtkImageData ** *outData,* int *extent[6],* int *id*) [protected]

10.381.2.12 vtkImageMapToColors16::vtkBooleanMacro (**PassAlphaToOutput** , int)

10.381.2.13 vtkImageMapToColors16::vtkGetMacro (**OutputFormat** , int)

10.381.2.14 vtkImageMapToColors16::vtkGetMacro (**ActiveComponent** , int)

10.381.2.15 vtkImageMapToColors16::vtkGetMacro (**PassAlphaToOutput** , int)

10.381.2.16 vtkImageMapToColors16::vtkGetObjectMacro (**LookupTable** , vtkScalarsToColors)

10.381.2.17 vtkImageMapToColors16::vtkSetMacro (**OutputFormat** , int)

10.381.2.18 vtkImageMapToColors16::vtkSetMacro (**ActiveComponent** , int)

10.381.2.19 vtkImageMapToColors16::vtkSetMacro (**PassAlphaToOutput** , int)

10.381.2.20 vtkImageMapToColors16::vtkTypeMacro (**vtkImageMapToColors16** , vtkThreadedImageAlgorithm)

10.381.3 Member Data Documentation

10.381.3.1 int vtkImageMapToColors16::ActiveComponent [protected]

10.381.3.2 int vtkImageMapToColors16::DataWasPassed [protected]

10.381.3.3 vtkScalarsToColors* vtkImageMapToColors16::LookupTable [protected]

10.381.3.4 int vtkImageMapToColors16::OutputFormat [protected]

10.381.3.5 int vtkImageMapToColors16::PassAlphaToOutput [protected]

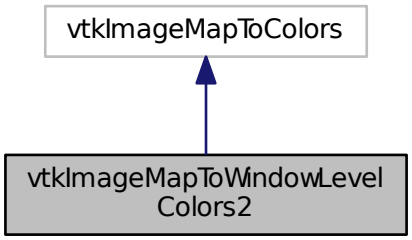
The documentation for this class was generated from the following file:

- vtkImageMapToColors16.h

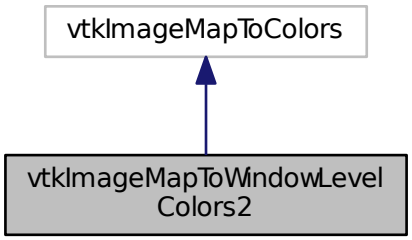
10.382 vtkImageMapToWindowLevelColors2 Class Reference

`#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)
- vtkTypeMacro (vtkImageMapToWindowLevelColors2, vtkImageMapToColors)

Static Public Member Functions

- static vtkImageMapToWindowLevelColors2 * New ()

Protected Member Functions

- vtkImageMapToWindowLevelColors2 ()
- ~vtkImageMapToWindowLevelColors2 ()
- virtual int RequestData (vtkInformation *request, vtkInformationVector **inputVector, vtkInformationVector *outputVector)
- virtual int RequestInformation (vtkInformation *, vtkInformationVector **, vtkInformationVector *)
- void ThreadedRequestData (vtkInformation *request, vtkInformationVector **inputVector, vtkInformationVector *outputVector, vtkImageData ***inData, vtkImageData **outData, int extent[6], int id)

Protected Attributes

- double Level
- double Window

10.382.1 Constructor & Destructor Documentation

10.382.1.1 vtkImageMapToWindowLevelColors2::vtkImageMapToWindowLevelColors2 ( ) [protected]

10.382.1.2 vtkImageMapToWindowLevelColors2::~vtkImageMapToWindowLevelColors2 ( ) [protected]

10.382.2 Member Function Documentation

10.382.2.1 static vtkImageMapToWindowLevelColors2* vtkImageMapToWindowLevelColors2::New ( ) [static]

10.382.2.2 void vtkImageMapToWindowLevelColors2::PrintSelf (ostream & *os,* vtkIndent *indent*)

10.382.2.3 virtual int vtkImageMapToWindowLevelColors2::RequestData (vtkInformation * *request,* vtkInformationVector ** *inputVector,* vtkInformationVector * *outputVector*) [protected], [virtual]

10.382.2.4 virtual int vtkImageMapToWindowLevelColors2::RequestInformation (vtkInformation * , vtkInformationVector ** , vtkInformationVector *) [protected], [virtual]

10.382.2.5 void vtkImageMapToWindowLevelColors2::ThreadedRequestData (vtkInformation * *request,* vtkInformationVector ** *inputVector,* vtkInformationVector * *outputVector,* vtkImageData *** *inData,* vtkImageData ** *outData,* int *extent[6],* int *id*) [protected]

10.382.2.6 vtkImageMapToWindowLevelColors2::vtkGetMacro (Window , double)

10.382.2.7 vtkImageMapToWindowLevelColors2::vtkGetMacro (Level , double)

10.382.2.8 vtkImageMapToWindowLevelColors2::vtkSetMacro (Window , double)

10.382.2.9 vtkImageMapToWindowLevelColors2::vtkSetMacro (Level , double)

10.382.2.10 vtkImageMapToWindowLevelColors2::vtkTypeMacro (vtkImageMapToWindowLevelColors2 , vtkImageMapToColors)

10.382.3 Member Data Documentation

10.382.3.1 double vtkImageMapToWindowLevelColors2::Level [protected]

10.382.3.2 double vtkImageMapToWindowLevelColors2::Window [protected]

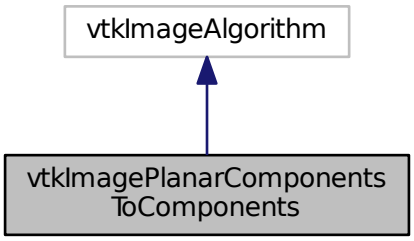
The documentation for this class was generated from the following file:

- vtkImageMapToWindowLevelColors2.h

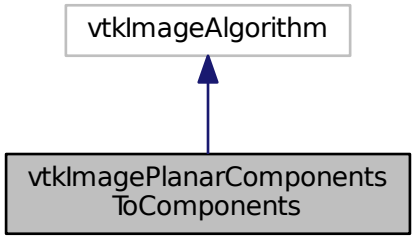
10.383 vtkImagePlanarComponentsToComponents Class Reference

`#include <vtkImagePlanarComponentsToComponents.h>`

Inheritance diagram for vtkImagePlanarComponentsToComponents:



Collaboration diagram for vtkImagePlanarComponentsToComponents:



Public Member Functions

- void PrintSelf (ostream &os, vtkIndent indent)
- vtkTypeMacro (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 ( ) [protected]

10.383.1.2 vtkImagePlanarComponentsToComponents::~vtkImagePlanarComponentsToComponents ( ) [inline], [protected]

10.383.2 Member Function Documentation

10.383.2.1 static vtkImagePlanarComponentsToComponents* vtkImagePlanarComponentsToComponents::New ( ) [static]

10.383.2.2 void vtkImagePlanarComponentsToComponents::PrintSelf (ostream & *os,* vtkIndent *indent*)

10.383.2.3 virtual int vtkImagePlanarComponentsToComponents::RequestData (vtkInformation * , vtkInformationVector ** , vtkInformationVector *) [protected], [virtual]

10.383.2.4 vtkImagePlanarComponentsToComponents::vtkTypeMacro (vtkImagePlanarComponentsToComponents , vtkImageAlgorithm)

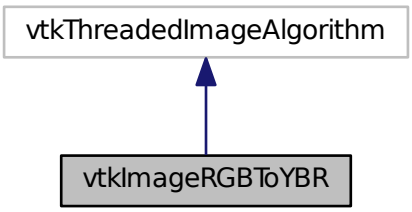
The documentation for this class was generated from the following file:

- vtkImagePlanarComponentsToComponents.h

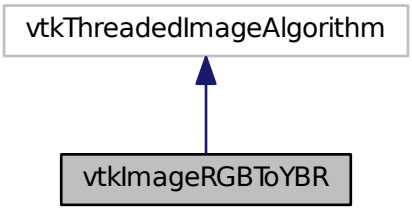
10.384 vtkImageRGBToYBR Class Reference

`#include <vtkImageRGBToYBR.h>`

Inheritance diagram for vtkImageRGBToYBR:



Collaboration diagram for vtkImageRGBToYBR:



Public Member Functions

- void PrintSelf (ostream &os, vtkIndent indent)
- vtkTypeMacro (vtkImageRGBToYBR, vtkThreadedImageAlgorithm)

Static Public Member Functions

- static vtkImageRGBToYBR * New ()

Protected Member Functions

- vtkImageRGBToYBR ()
- ~vtkImageRGBToYBR ()
- void ThreadedExecute (vtkImageData *inData, vtkImageData *outData, int ext[6], int id)

10.384.1 Constructor & Destructor Documentation

10.384.1.1 vtkImageRGBToYBR::vtkImageRGBToYBR ( ) [protected]

10.384.1.2 vtkImageRGBToYBR::~vtkImageRGBToYBR ( ) [inline], [protected]

10.384.2 Member Function Documentation

10.384.2.1 static vtkImageRGBToYBR* vtkImageRGBToYBR::New ( ) [static]

10.384.2.2 void vtkImageRGBToYBR::PrintSelf (ostream & *os,* vtkIndent *indent*)

10.384.2.3 void vtkImageRGBToYBR::ThreadedExecute (vtkImageData * *inData,* vtkImageData * *outData,* int *ext[6],* int *id*) [protected]

10.384.2.4 vtkImageRGBToYBR::vtkTypeMacro (vtkImageRGBToYBR , vtkThreadedImageAlgorithm)

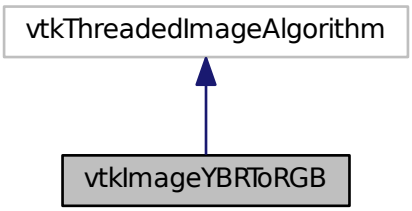
The documentation for this class was generated from the following file:

- vtkImageRGBToYBR.h

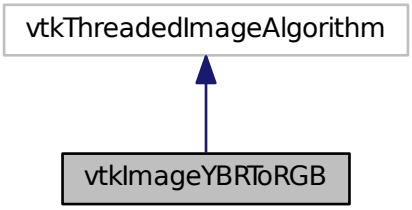
10.385 vtkImageYBRToRGB Class Reference

`#include <vtkImageYBRToRGB.h>`

Inheritance diagram for vtkImageYBRToRGB:



Collaboration diagram for vtkImageYBRToRGB:



Public Member Functions

- void PrintSelf (ostream &os, vtkIndent indent)
- vtkTypeMacro (vtkImageYBRToRGB, vtkThreadedImageAlgorithm)

Static Public Member Functions

- static vtkImageYBRToRGB * New ()

Protected Member Functions

- vtkImageYBRToRGB ()
- ∼vtkImageYBRToRGB ()
- void ThreadedExecute (vtkImageData ∗inData, vtkImageData ∗outData, int ext[6], int id)

10.385.1 Constructor & Destructor Documentation

10.385.1.1 vtkImageYBRToRGB::vtkImageYBRToRGB ( ) [protected]

10.385.1.2 vtkImageYBRToRGB::∼vtkImageYBRToRGB ( ) [inline],[protected]

10.385.2 Member Function Documentation

10.385.2.1 static vtkImageYBRToRGB∗ vtkImageYBRToRGB::New ( ) [static]

10.385.2.2 void vtkImageYBRToRGB::PrintSelf (ostream & *os,* vtkIndent *indent*)

10.385.2.3 void vtkImageYBRToRGB::ThreadedExecute (vtkImageData ∗ *inData,* vtkImageData ∗ *outData,* int *ext[6],* int *id*) [protected]

10.385.2.4 vtkImageYBRToRGB::vtkTypeMacro (vtkImageYBRToRGB , vtkThreadedImageAlgorithm)

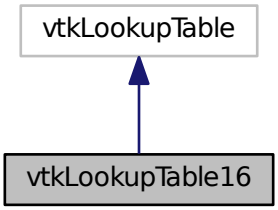
The documentation for this class was generated from the following file:

- vtkImageYBRToRGB.h

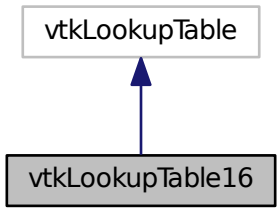
10.386 vtkLookupTable16 Class Reference

`#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)
- vtkTypeMacro (vtkLookupTable16, vtkLookupTable)
- unsigned char * WritePointer (const vtkIdType id, const int number)

Static Public Member Functions

- static vtkLookupTable16 * New ()

Protected Member Functions

- vtkLookupTable16 (int sze=256, int ext=256)
- ~vtkLookupTable16 ()
- void MapScalarsThroughTable2 (void *input, unsigned char *output, int inputDataType, int numberOfValues, int inputIncrement, int outputFormat)

Protected Attributes

- vtkUnsignedShortArray * Table16

10.386.1 Constructor & Destructor Documentation

10.386.1.1 vtkLookupTable16::vtkLookupTable16 (int *sze* = 256, int *ext* = 256) [protected]

10.386.1.2 vtkLookupTable16::~vtkLookupTable16 ( ) [protected]

10.386.2 Member Function Documentation

10.386.2.1 void vtkLookupTable16::Build ( )

10.386.2.2 unsigned short* vtkLookupTable16::GetPointer (const vtkIdType *id*) [inline]

10.386.2.3 void vtkLookupTable16::MapScalarsThroughTable2 (void * *input*, unsigned char * *output*, int *inputDataType*, int *numberOfValues*, int *inputIncrement*, int *outputFormat*) [protected]

10.386.2.4 static vtkLookupTable16* vtkLookupTable16::New ( ) [static]

10.386.2.5 void vtkLookupTable16::PrintSelf (ostream & *os*, vtkIndent *indent*)

10.386.2.6 void vtkLookupTable16::SetNumberOfTableValues (vtkIdType *number*)

10.386.2.7 vtkLookupTable16::vtkTypeMacro (vtkLookupTable16 , vtkLookupTable)

10.386.2.8 unsigned char * vtkLookupTable16::WritePointer (const vtkIdType *id*, const int *number*) [inline]

10.386.3 Member Data Documentation

10.386.3.1 vtkUnsignedShortArray* vtkLookupTable16::Table16 [protected]

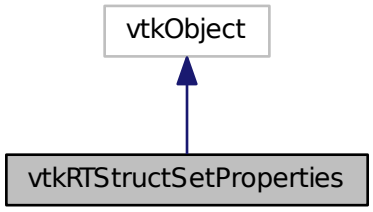
The documentation for this class was generated from the following file:

- vtkLookupTable16.h

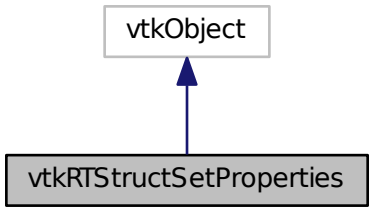
10.387 vtkRTStructSetProperties Class Reference

`#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 *reframerefuid, 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)
- vtkTypeMacro (vtkRTStructSetProperties, vtkObject)

Static Public Member Functions

- static vtkRTStructSetProperties * New ()

Protected Member Functions

- vtkRTStructSetProperties ()
- ~vtkRTStructSetProperties ()

Protected Attributes

- vtkRTStructSetPropertiesInternals * Internals
- char * ReferenceFrameOfReferenceUID
- char * ReferenceSeriesInstanceUID
- char * SeriesInstanceUID
- char * SOPInstanceUID
- char * StructureSetDate
- char * StructureSetLabel
- char * StructureSetName
- char * StructureSetTime
- char * StudyInstanceUID

10.387.1 Detailed Description

Examples:

GenerateRTSTRUCT.cxx.

10.387.2 Constructor & Destructor Documentation

10.387.2.1 vtkRTStructSetProperties::vtkRTStructSetProperties ( ) [protected]

10.387.2.2 vtkRTStructSetProperties::~vtkRTStructSetProperties ( ) [protected]

10.387.3 Member Function Documentation

10.387.3.1 void vtkRTStructSetProperties::AddContourReferencedFrameOfReference (vtkIdType *pdnum*, const char * *classuid*, const char * *instanceuid*)

10.387.3.2 void vtkRTStructSetProperties::AddReferencedFrameOfReference (const char * *classuid*, const char * *instanceuid*)

10.387.3.3 void vtkRTStructSetProperties::AddStructureSetROI (int *roinumber*, const char * *reframerefuid*, const char * *roiname*, const char * *ROIGenerationAlgorithm*, const char * *ROIDescription* = 0)

10.387.3.4 void vtkRTStructSetProperties::AddStructureSetROIObservation (int *refnumber*, int *observationnumber*, const char * *rtroiinterpretedtype*, const char * *roiinterpreter*, const char * *roiobservationlabel* = 0)

10.387.3.5 virtual void vtkRTStructSetProperties::Clear ( ) [virtual]

10.387.3.6 virtual void vtkRTStructSetProperties::DeepCopy (vtkRTStructSetProperties * *p*) [virtual]

10.387.3.7 const char* vtkRTStructSetProperties::GetContourReferencedFrameOfReferenceClassUID (vtkIdType *pdnum*, vtkIdType *id*)

10.387.3.8 const char* vtkRTStructSetProperties::GetContourReferencedFrameOfReferenceInstanceUID (vtkIdType *pdnum,* vtkIdType *id*)

10.387.3.9 vtkIdType vtkRTStructSetProperties::GetNumberOfContourReferencedFrameOfReferences ( )

10.387.3.10 vtkIdType vtkRTStructSetProperties::GetNumberOfContourReferencedFrameOfReferences (vtkIdType *pdnum*)

10.387.3.11 vtkIdType vtkRTStructSetProperties::GetNumberOfReferencedFrameOfReferences ( )

10.387.3.12 vtkIdType vtkRTStructSetProperties::GetNumberOfStructureSetROIs ( )

10.387.3.13 const char* vtkRTStructSetProperties::GetReferencedFrameOfReferenceClassUID (vtkIdType *id*)

10.387.3.14 const char* vtkRTStructSetProperties::GetReferencedFrameOfReferenceInstanceUID (vtkIdType *id*)

10.387.3.15 int vtkRTStructSetProperties::GetStructureSetObservationNumber (vtkIdType *id*)

10.387.3.16 const char* vtkRTStructSetProperties::GetStructureSetROIDescription (vtkIdType *id*)

10.387.3.17 const char* vtkRTStructSetProperties::GetStructureSetROIGenerationAlgorithm (vtkIdType)

10.387.3.18 const char* vtkRTStructSetProperties::GetStructureSetROIName (vtkIdType)

10.387.3.19 int vtkRTStructSetProperties::GetStructureSetROINumber (vtkIdType *id*)

10.387.3.20 const char* vtkRTStructSetProperties::GetStructureSetROIObservationLabel (vtkIdType *id*)

10.387.3.21 const char* vtkRTStructSetProperties::GetStructureSetROIRefFrameRefUID (vtkIdType)

10.387.3.22 const char* vtkRTStructSetProperties::GetStructureSetRTROIInterpretedType (vtkIdType *id*)

10.387.3.23 static **vtkRTStructSetProperties*** vtkRTStructSetProperties::New ( ) [static]

Examples:

GenerateRTSTRUCT.cxx.

10.387.3.24 void vtkRTStructSetProperties::PrintSelf (ostream & *os,* vtkIndent *indent*)

10.387.3.25 vtkRTStructSetProperties::vtkGetStringMacro (**StructureSetLabel**)

10.387.3.26 vtkRTStructSetProperties::vtkGetStringMacro (**StructureSetName**)

10.387.3.27 vtkRTStructSetProperties::vtkGetStringMacro (**StructureSetDate**)

10.387.3.28 vtkRTStructSetProperties::vtkGetStringMacro (**StructureSetTime**)

10.387.3.29 vtkRTStructSetProperties::vtkGetStringMacro (**SOPInstanceUID**)

10.387.3.30 vtkRTStructSetProperties::vtkGetStringMacro (**StudyInstanceUID**)

10.387.3.31 vtkRTStructSetProperties::vtkGetStringMacro (**SeriesInstanceUID**)

10.387.3.32 vtkRTStructSetProperties::vtkGetStringMacro (**ReferenceSeriesInstanceUID**)

10.387.3.33 vtkRTStructSetProperties::vtkGetStringMacro (**ReferenceFrameOfReferenceUID**)

10.387.3.34 vtkRTStructSetProperties::vtkSetStringMacro (**StructureSetLabel**)

10.387.3.35 vtkRTStructSetProperties::vtkSetStringMacro (**StructureSetName**)

10.387.3.36 vtkRTStructSetProperties::vtkSetStringMacro (**StructureSetDate**)

10.387.3.37 vtkRTStructSetProperties::vtkSetStringMacro (**StructureSetTime**)

10.387.3.38 vtkRTStructSetProperties::vtkSetStringMacro (**SOPInstanceUID**)

10.387.3.39 vtkRTStructSetProperties::vtkSetStringMacro (**StudyInstanceUID**)

10.387.3.40 vtkRTStructSetProperties::vtkSetStringMacro (**SeriesInstanceUID**)

10.387.3.41 vtkRTStructSetProperties::vtkSetStringMacro (**ReferenceSeriesInstanceUID**)

10.387.3.42 vtkRTStructSetProperties::vtkSetStringMacro (**ReferenceFrameOfReferenceUID**)

10.387.3.43 vtkRTStructSetProperties::vtkTypeMacro (**vtkRTStructSetProperties** , vtkObject)

10.387.4 Member Data Documentation

10.387.4.1 vtkRTStructSetPropertiesInternals* vtkRTStructSetProperties::Internals [protected]

10.387.4.2 char* vtkRTStructSetProperties::ReferenceFrameOfReferenceUID [protected]

10.387.4.3 char* vtkRTStructSetProperties::ReferenceSeriesInstanceUID [protected]

10.387.4.4 char* vtkRTStructSetProperties::SeriesInstanceUID [protected]

10.387.4.5 char* vtkRTStructSetProperties::SOPInstanceUID [protected]

10.387.4.6 char* vtkRTStructSetProperties::StructureSetDate [protected]

10.387.4.7 char* vtkRTStructSetProperties::StructureSetLabel [protected]

10.387.4.8 char* vtkRTStructSetProperties::StructureSetName [protected]

10.387.4.9 char* vtkRTStructSetProperties::StructureSetTime [protected]

10.387.4.10 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 gdcm::Waveform::Waveform ( ) [inline]

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

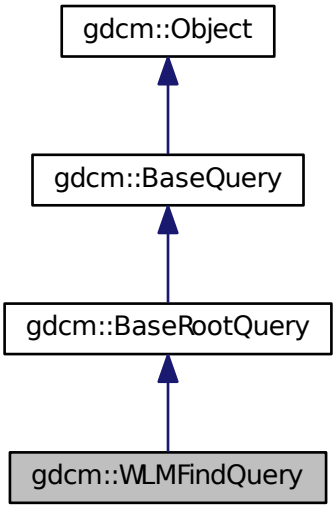
- gdcmWaveform.h

10.389 gdcm::WLMFindQuery Class Reference

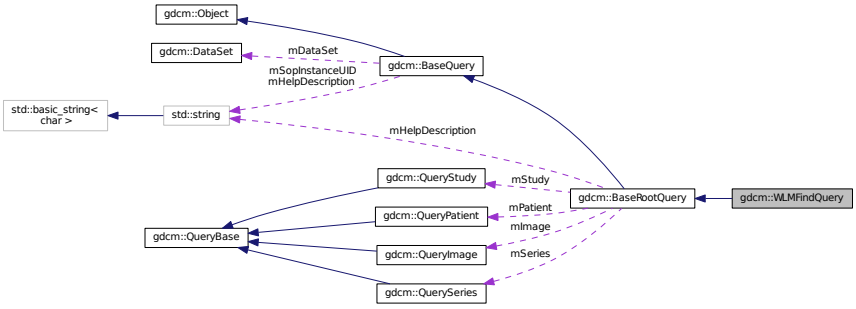
PatientRootQuery contains: the class which will produce a dataset for c-find with patient root.

```
#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 gdcm::WLMFindQuery::WLMFindQuery ( )

10.389.3 Member Function Documentation

10.389.3.1 UIDs::TSName gdcm::WLMFindQuery::GetAbstractSyntaxUID ( ) const [virtual]

Implements gdcm::BaseQuery.

10.389.3.2 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 DataSet gdcm::WLMFindQuery::GetValidDataSet ( ) const [protected]

10.389.3.4 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 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 friend class QueryFactory [friend]

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

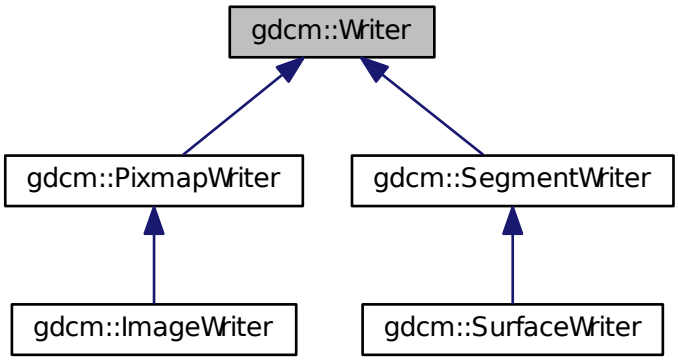
- gdcmWLMFindQuery.h

10.390 gdcm::Writer Class Reference

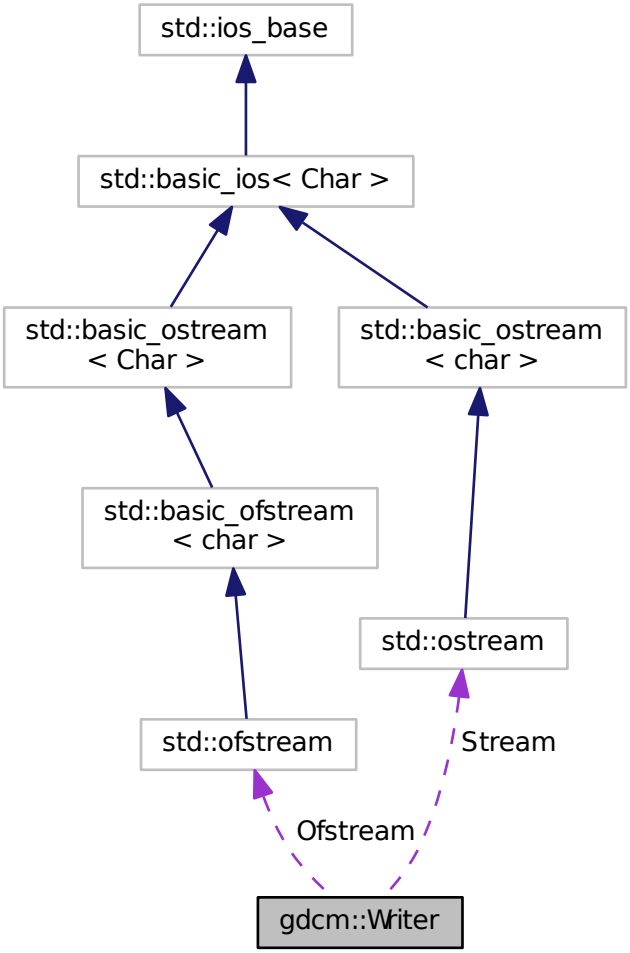
Writer ala DOM (Document Object Model) This class is a non-validating writer, it will only performs well- formedness check only.

```
#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

- 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 gdcm::Writer::Writer ( )

10.390.2.2 virtual gdcm::Writer::~Writer ( ) [virtual]

10.390.3 Member Function Documentation

10.390.3.1 void gdcm::Writer::CheckFileMetaInformationOff ( ) [inline]

Examples:

CreateFakeRTDOSE.cxx, FixBrokenJ2K.cxx, and HelloWorld.cxx.

10.390.3.2 void gdcm::Writer::CheckFileMetaInformationOn ( ) [inline]

10.390.3.3 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.

10.390.3.4 std::ostream* gdcm::Writer::GetStreamPtr ( ) const [inline], [protected]

10.390.3.5 void gdcm::Writer::SetCheckFileMetaInformation (bool *b*) [inline]

Undocumented function, do not use (= leave default)

Examples:

GenAllVR.cxx, GenFakeIdentifyFile.cxx, and PatchFile.cxx.

10.390.3.6 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.

10.390.3.7 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.

10.390.3.8 void gdcm::Writer::SetStream (std::ostream & *output_stream*) [inline]

Set user ostream buffer.

10.390.3.9 void gdcm::Writer::SetWriteDataSetOnly (bool *b*) [inline], [protected]

10.390.3.10 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 friend class StreamImageWriter [friend]

10.390.5 Member Data Documentation

10.390.5.1 std::ofstream* gdcm::Writer::Ofstream [protected]

10.390.5.2 std::ostream* gdcm::Writer::Stream [protected]

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

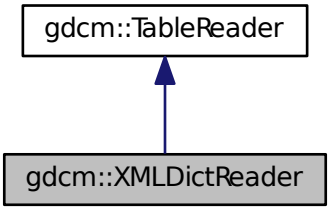
- gdcmWriter.h

10.391 gdcm::XMLDictReader Class Reference

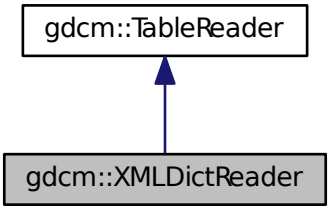
Class for representing a XMLDictReader.

`#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 gdcm::XMLDictReader::XMLDictReader ( )

10.391.2.2 gdcm::XMLDictReader::~XMLDictReader ( ) [inline]

10.391.3 Member Function Documentation

10.391.3.1 void gdcm::XMLDictReader::CharacterDataHandler (const char * *data*, int *length*) [virtual]

Reimplemented from gdcm::TableReader.

10.391.3.2 void gdcm::XMLDictReader::EndElement (const char * *name*) [virtual]

Reimplemented from gdcm::TableReader.

10.391.3.3 const Dict& gdcm::XMLDictReader::GetDict ( ) [inline]

10.391.3.4 void gdcm::XMLDictReader::HandleDescription (const char ** *atts*) [protected]

10.391.3.5 void gdcm::XMLDictReader::HandleEntry (const char ** *atts*) [protected]

10.391.3.6 void gdcm::XMLDictReader::StartElement (const char * *name*, const char ** *atts*) [virtual]

Reimplemented from gdcm::TableReader.

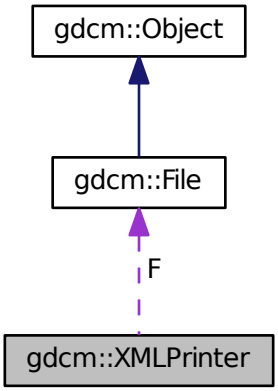
The documentation for this class was generated from the following file:

- 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 enum gdcm::XMLPrinter::PrintStyles

Enumerator

OnlyUUID

LOADBULKDATA

10.392.2 Constructor & Destructor Documentation

10.392.2.1 gdcm::XMLPrinter::XMLPrinter ( )

10.392.2.2 virtual gdcm::XMLPrinter::~XMLPrinter ( ) [virtual]

10.392.3 Member Function Documentation

10.392.3.1 PrintStyles gdcm::XMLPrinter::GetPrintStyle ( ) const [inline]

10.392.3.2 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 void gdcm::XMLPrinter::Print (std::ostream & *os*)

10.392.3.4 VR gdcm::XMLPrinter::PrintDataElement (std::ostream & *os*, const Dicts & *dicts*, const DataSet & *ds*, const DataElement & *de*, const TransferSyntax & *ts*) [protected]

10.392.3.5 void gdcm::XMLPrinter::PrintDataSet (const DataSet & *ds*, const TransferSyntax & *ts*, std::ostream & *os*)

10.392.3.6 void gdcm::XMLPrinter::PrintSQ (const SequenceOfItems * *sqi*, const TransferSyntax & *ts*, std::ostream & *os*) [protected]

10.392.3.7 void gdcm::XMLPrinter::SetFile (File const & *f*) [inline]

10.392.3.8 void gdcm::XMLPrinter::SetStyle (PrintStyles *ps*) [inline]

10.392.4 Member Data Documentation

10.392.4.1 const File* gdcm::XMLPrinter::F [protected]

10.392.4.2 PrintStyles gdcm::XMLPrinter::PrintStyle [protected]

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

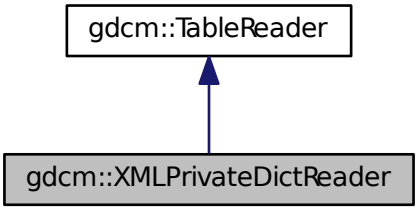
- gdcmXMLPrinter.h

10.393 gdcm::XMLPrivateDictReader Class Reference

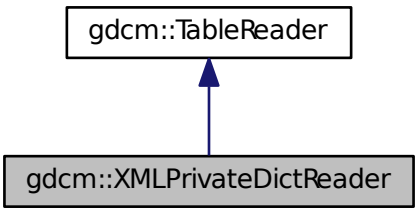
Class for representing a XMLPrivateDictReader.

`#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 gdcm::XMLPrivateDictReader::XMLPrivateDictReader ( )

10.393.2.2 gdcm::XMLPrivateDictReader::~XMLPrivateDictReader ( ) [inline]

10.393.3 Member Function Documentation

10.393.3.1 void gdcm::XMLPrivateDictReader::CharacterDataHandler (const char * *data*, int *length*) [virtual]

Reimplemented from gdcm::TableReader.

10.393.3.2 void gdcm::XMLPrivateDictReader::EndElement (const char * *name*) [virtual]

Reimplemented from gdcm::TableReader.

10.393.3.3 const PrivateDict& gdcm::XMLPrivateDictReader::GetPrivateDict ( ) [inline]

10.393.3.4 void gdcm::XMLPrivateDictReader::HandleDescription (const char ** *atts*) [protected]

10.393.3.5 void gdcm::XMLPrivateDictReader::HandleEntry (const char ** *atts*) [protected]

10.393.3.6 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

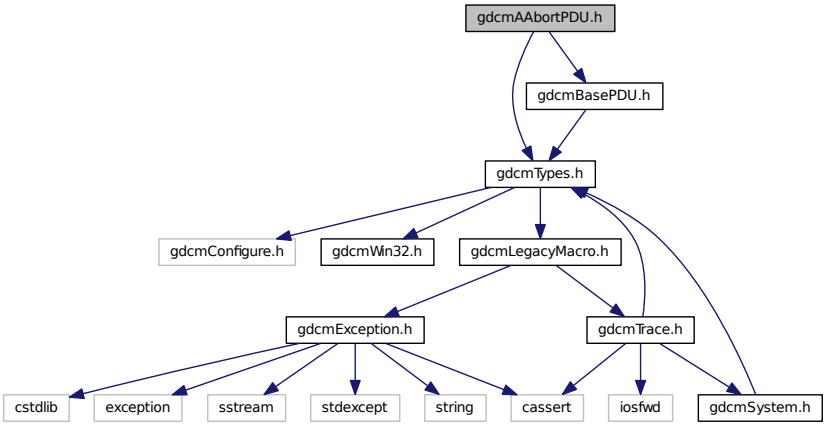
Chapter 11

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 Table 9-26 A-ABORT PDU FIELDS.

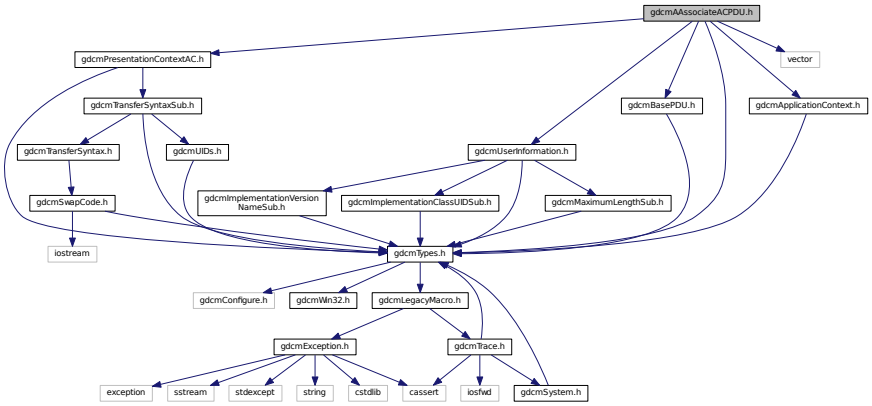
Namespaces

- 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 Table 9-17 ASSOCIATE-AC PDU fields.

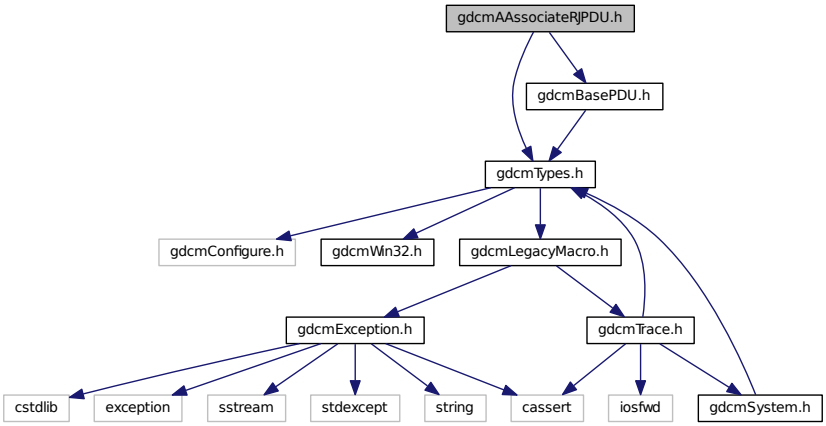
Namespaces

- 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 Table 9-21 ASSOCIATE-RJ PDU FIELDS.

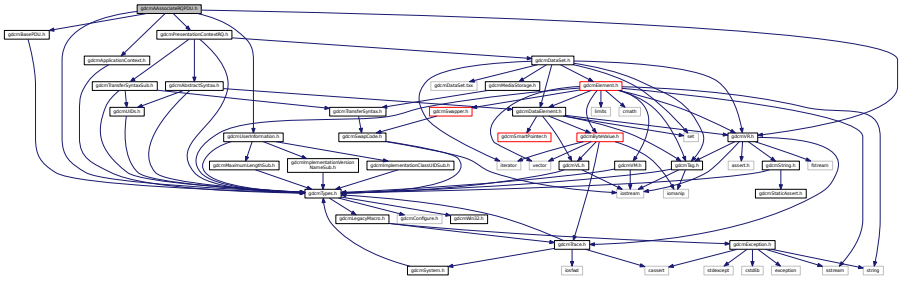
Namespaces

- 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 Table 9-11 ASSOCIATE-RQ PDU fields.

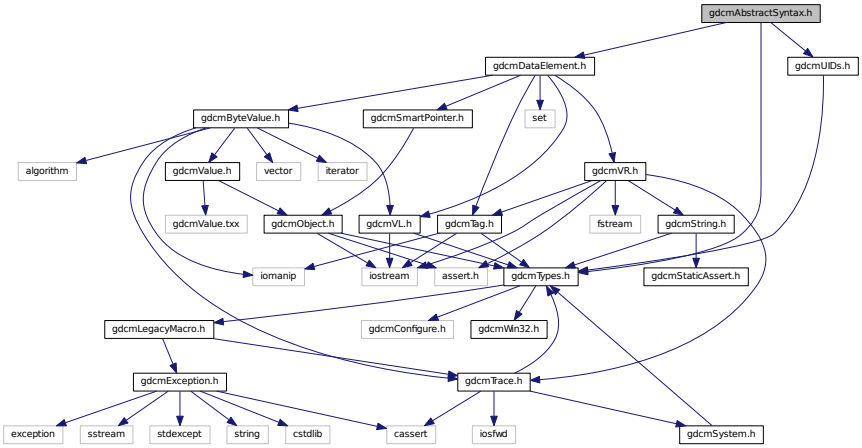
Namespaces

- gdcm
- gdcm::network

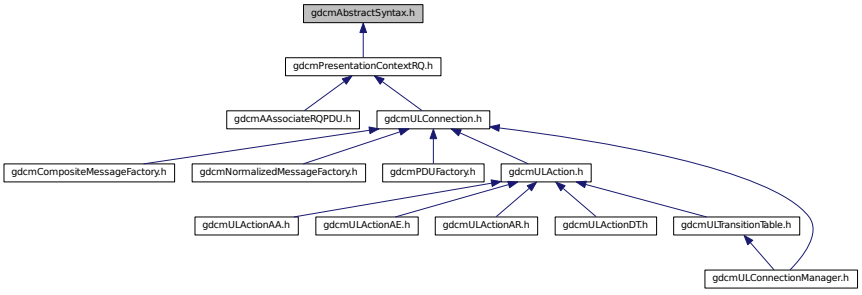
11.5 gdcmAbstractSyntax.h File Reference

```
#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 Table 9-14 ABSTRACT SYNTAX SUB-ITEM FIELDS.

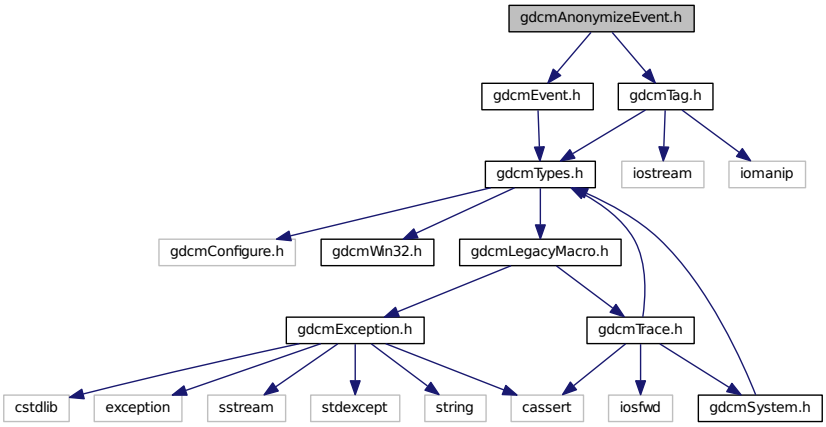
Namespaces

- gdcm
- gdcm::network

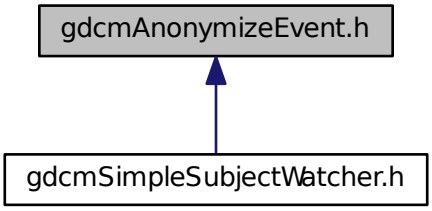
11.6 gdcmAnonymizeEvent.h File Reference

```
#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 Special type of event triggered during the Anonymization process.

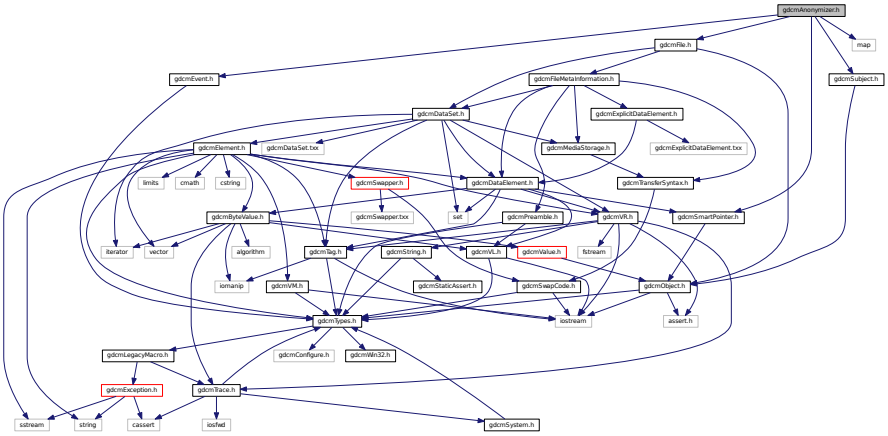
Namespaces

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

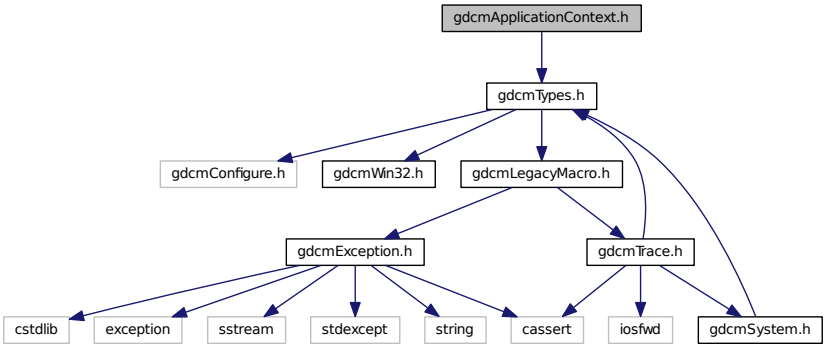
Namespaces

- gdcm

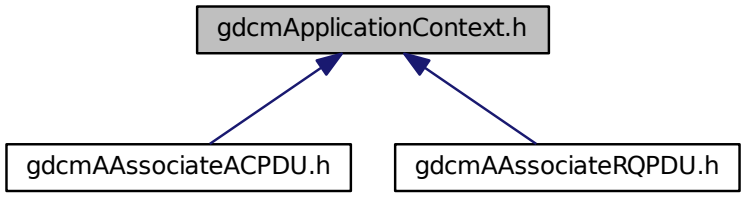
11.8 gdcmApplicationContext.h File Reference

```
#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 Table 9-12 APPLICATION CONTEXT ITEM FIELDS.

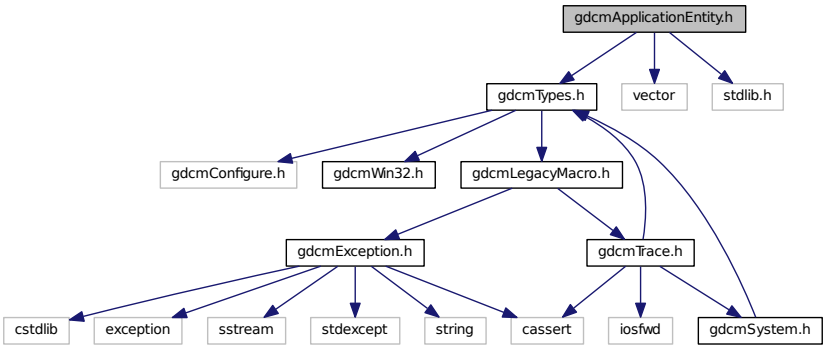
Namespaces

- 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.

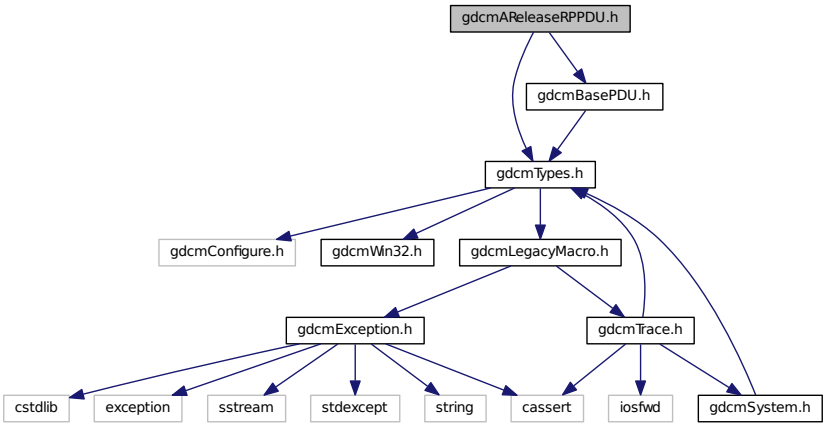
Namespaces

- 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 Table 9-25 A-RELEASE-RP PDU fields.

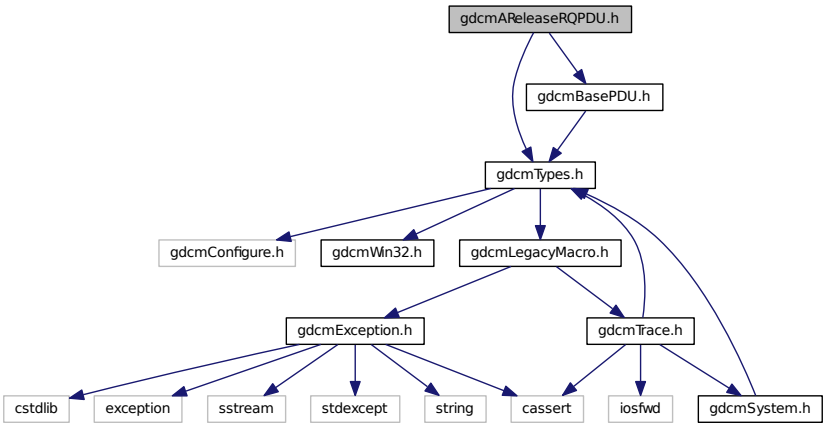
Namespaces

- 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

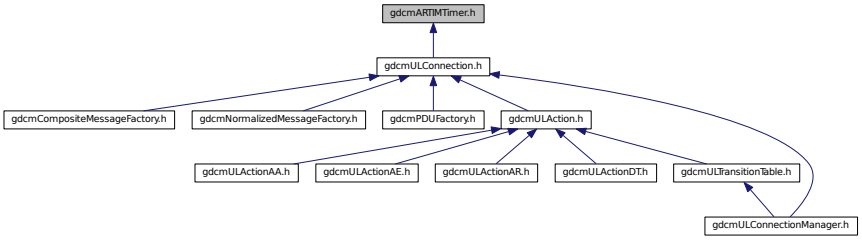
  AReleaseRQPDU Table 9-24 A-RELEASE-RQ PDU FIELDS.

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 This file contains the code for the ARTIM timer.

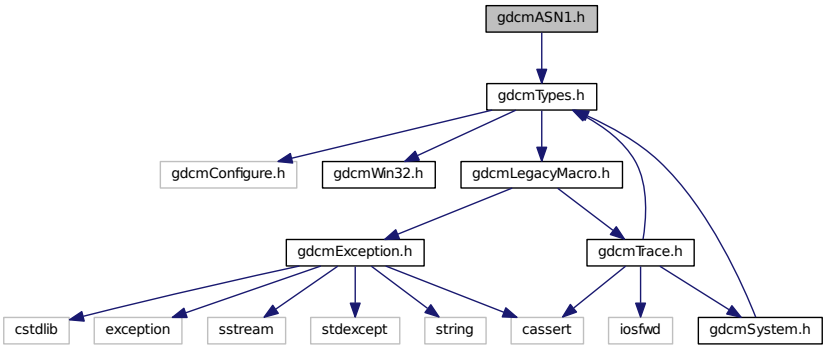
Namespaces

- 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.

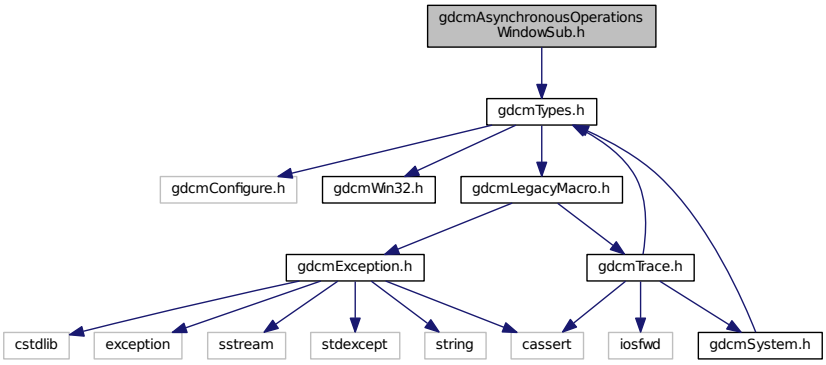
Namespaces

- gdcm

11.14 gdcmAsynchronousOperationsWindowSub.h File Reference

```
#include "gdcmTypes.h"
```

Include dependency graph for gdcmAsynchronousOperationsWindowSub.h:



Classes

- class gdcm::network::AsynchronousOperationsWindowSub

  AsynchronousOperationsWindowSub PS 3.7 Table D.3-7 ASYNCHRONOUS OPERATIONS WINDOW SUB-ITEM FIE↩ LDS (A-ASSOCIATE-RQ)

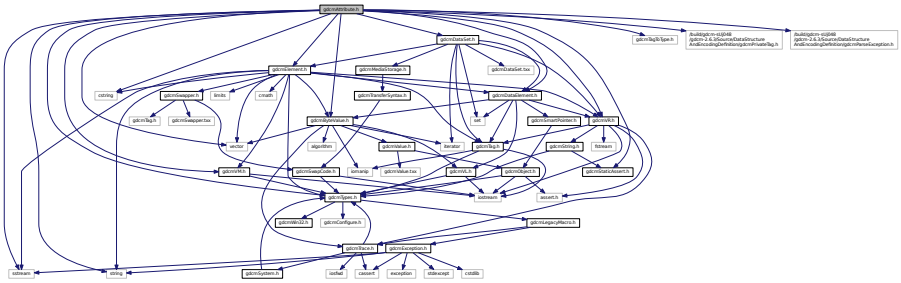
Namespaces

- gdcm
- gdcm::network

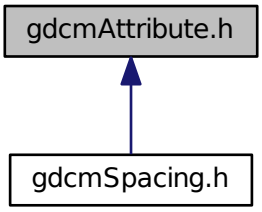
11.15 gdcmAttribute.h File Reference

```
#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 >

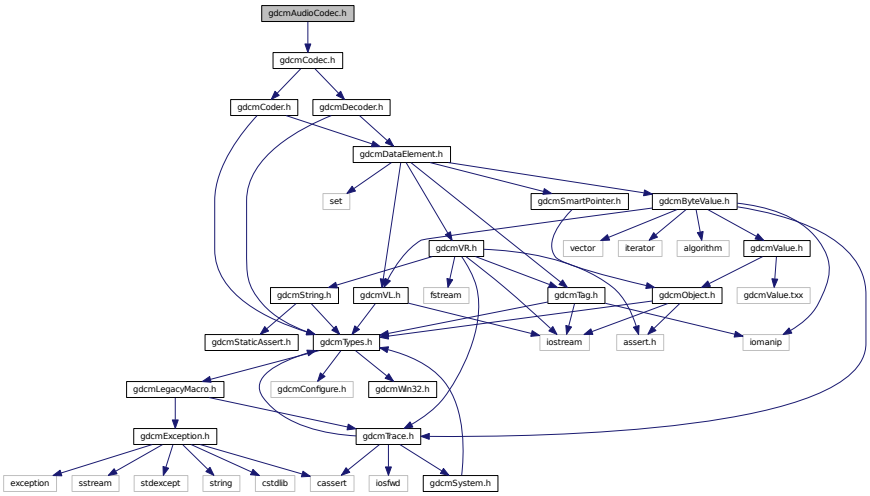
Namespaces

- gdcm

11.16 gdcmAudioCodec.h File Reference

```
#include "gdcmCodec.h"
```

Include dependency graph for gdcmAudioCodec.h:



Classes

- class gdcm::AudioCodec

  AudioCodec.

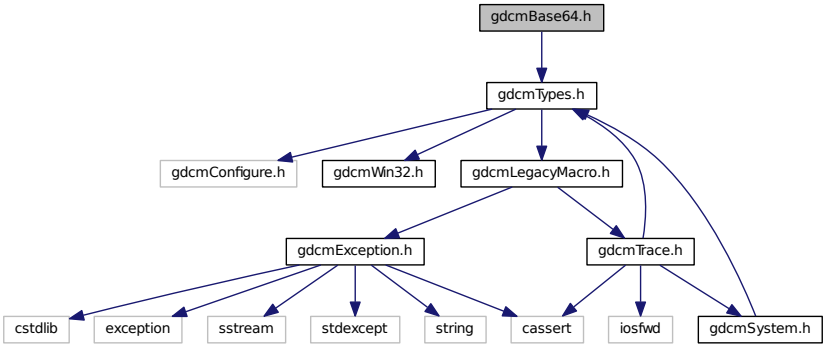
Namespaces

- gdcm

11.17 gdcmBase64.h File Reference

```
#include "gdcmTypes.h"
```

Include dependency graph for gdcmBase64.h:



Classes

- class gdcm::Base64

  Class for Base64.

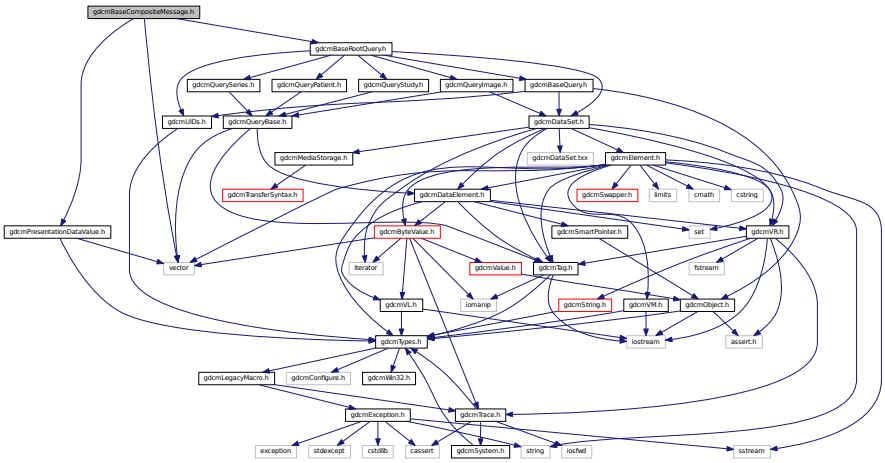
Namespaces

- gdcm

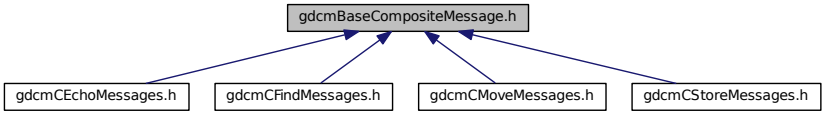
11.18 gdcmBaseCompositeMessage.h File Reference

```
#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 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.

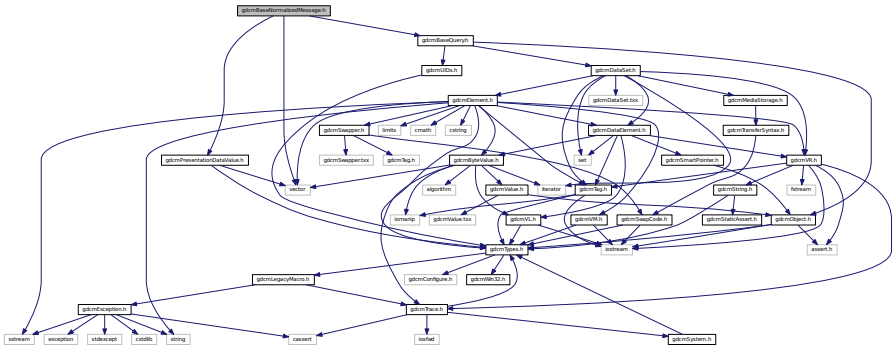
Namespaces

- gdcm
- gdcm::network

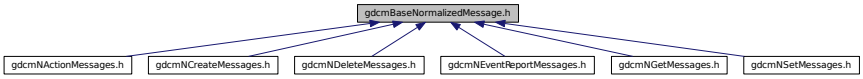
11.19 gdcmBaseNormalizedMessage.h File Reference

```
#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 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.

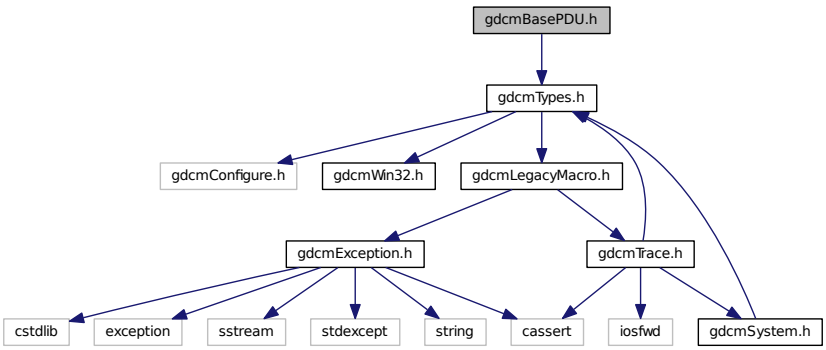
Namespaces

- gdcm
- gdcm::network

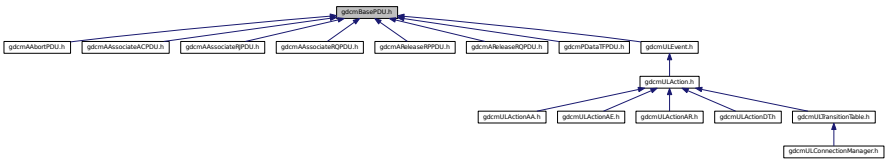
11.20 gdcmBasePDU.h File Reference

```
#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 base class for PDUs.

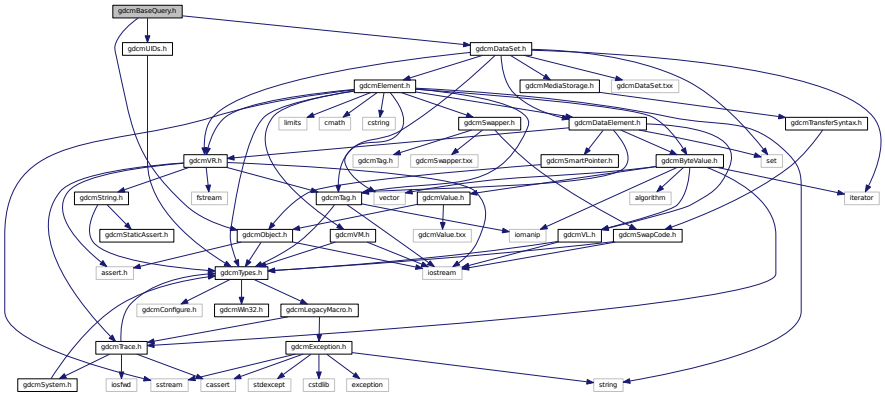
Namespaces

- 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 contains: a baseclass which will produce a dataset for all dimse messages.

Namespaces

- gdcm

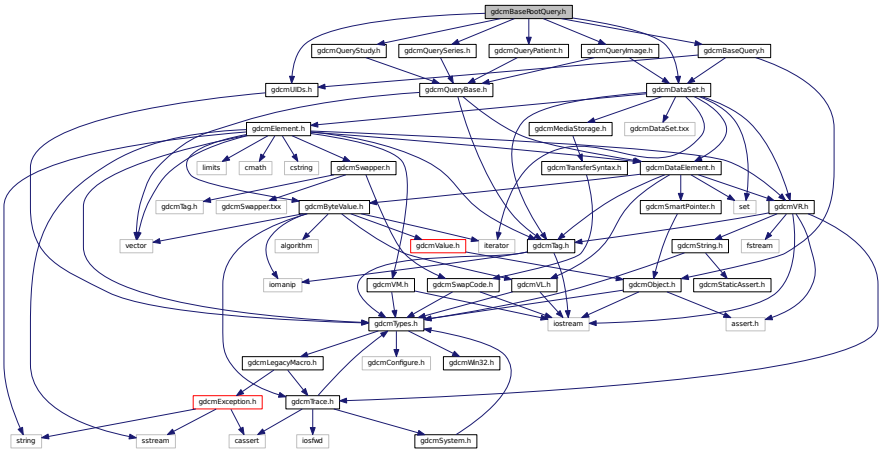
Enumerations

- 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 contains: a baseclass which will produce a dataset for c-find and c-move with patient/study root.

Namespaces

- gdcm

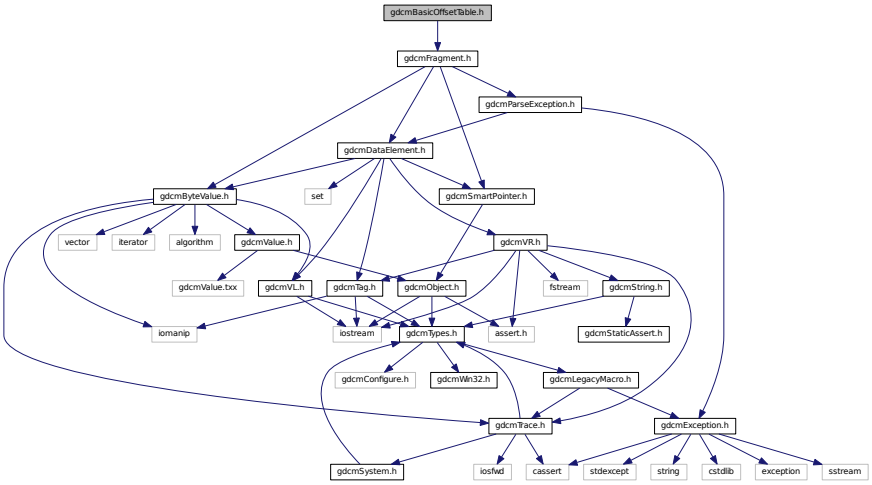
Enumerations

- 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 }

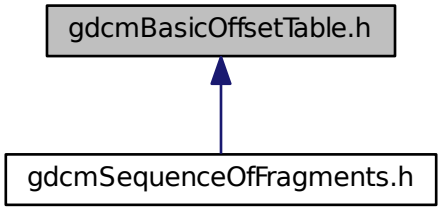
11.23 gdcmBasicOffsetTable.h File Reference

```
#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

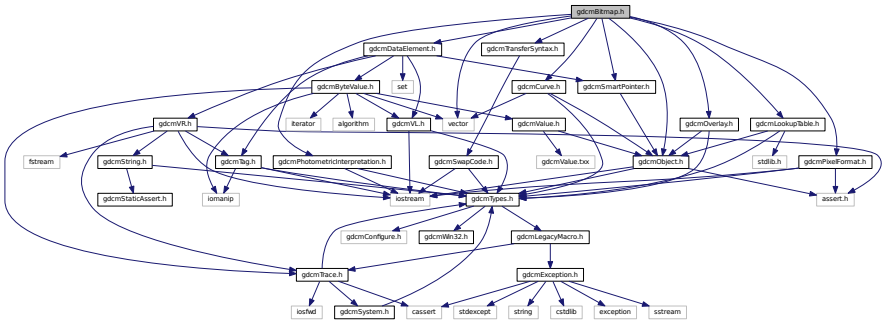
Functions

- std::ostream & gdcm::operator<< (std::ostream &os, const BasicOffsetTable &val)

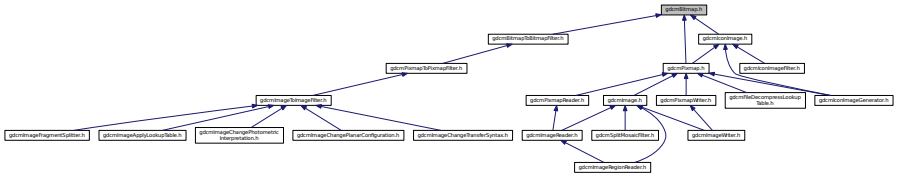
11.24 gdcmBitmap.h File Reference

```
#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 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)

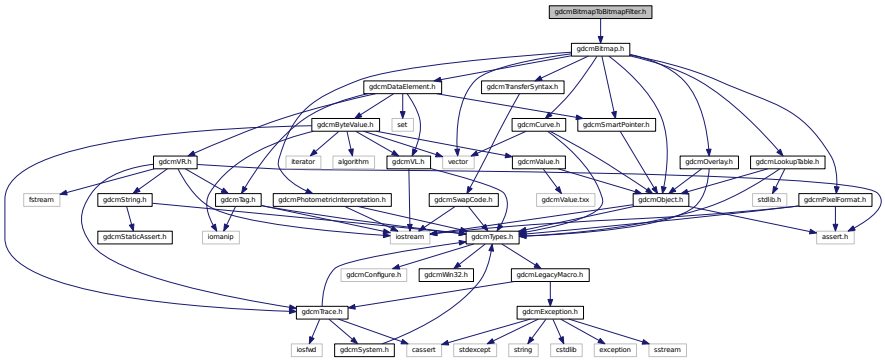
Namespaces

- gdcm

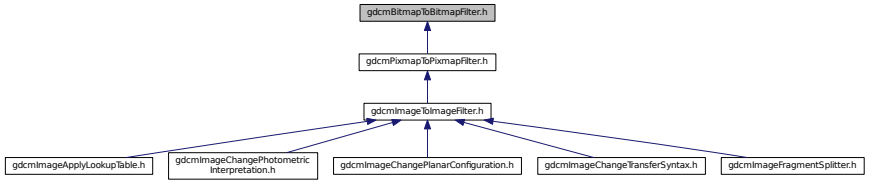
11.25 gdcmBitmapToBitmapFilter.h File Reference

```
#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 Super class for all filter taking an image and producing an output image.

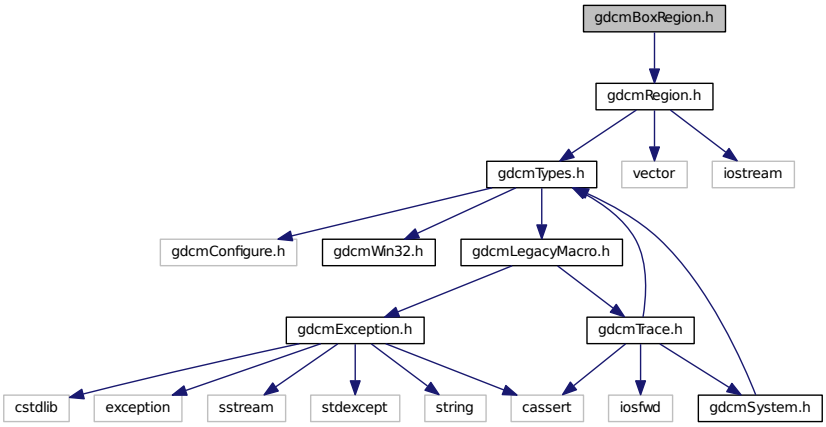
Namespaces

- 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 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)

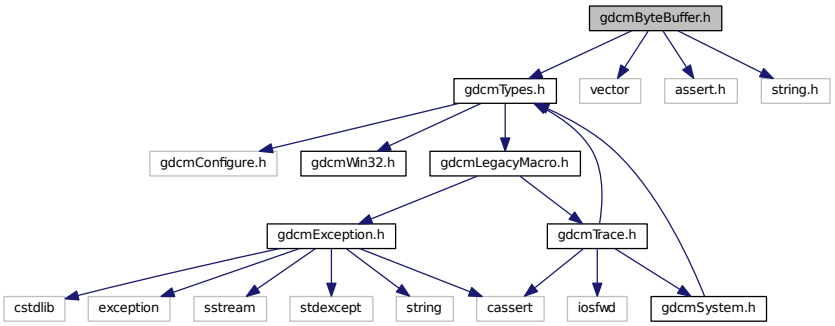
Namespaces

- gdcm

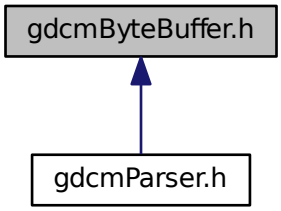
11.27 gdcmByteBuffer.h File Reference

```
#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.

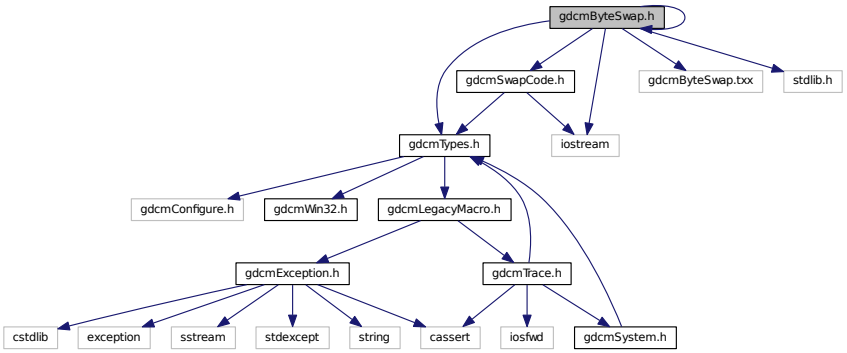
Namespaces

- 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.

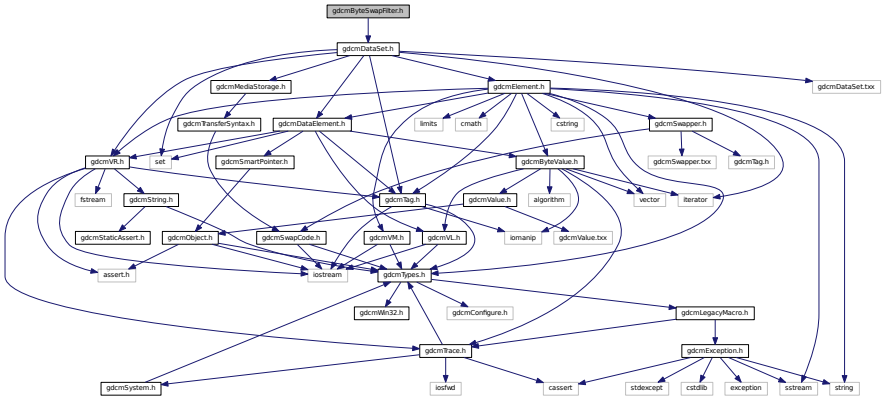
Namespaces

- gdcm

11.29 gdcmByteSwapFilter.h File Reference

```
#include "gdcmDataSet.h"
```

Include dependency graph for gdcmByteSwapFilter.h:



Classes

- class gdcm::ByteSwapFilter

  ByteSwapFilter *In place byte-swapping of a dataset FIXME: FL status ??*

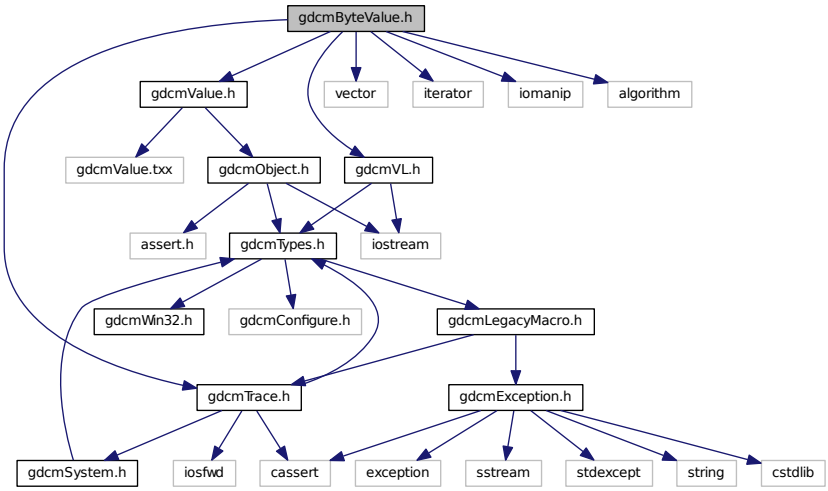
Namespaces

- gdcm

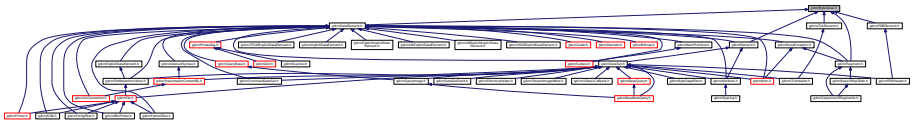
11.30 gdcmByteValue.h File Reference

```
#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)

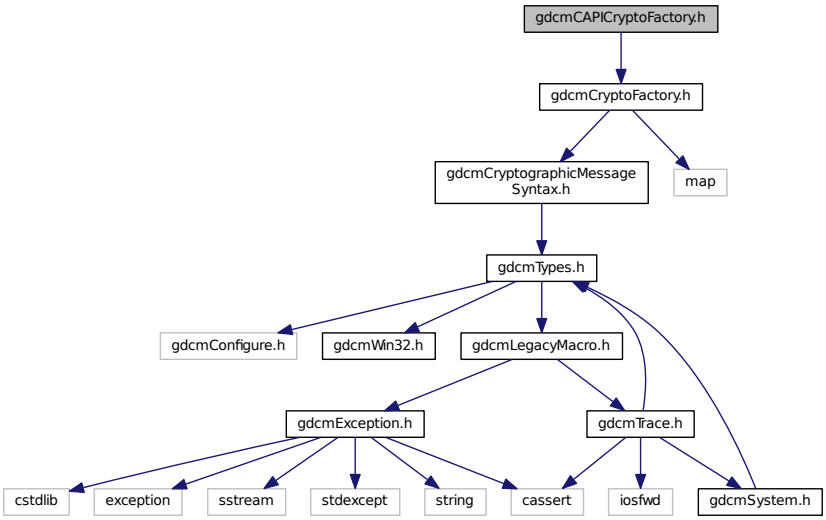
Namespaces

- gdcm

11.31 gdcmCAPICryptoFactory.h File Reference

```
#include "gdcmCryptoFactory.h"
```

Include dependency graph for gdcmCAPICryptoFactory.h:



Classes

- class gdcm::CAPICryptoFactory

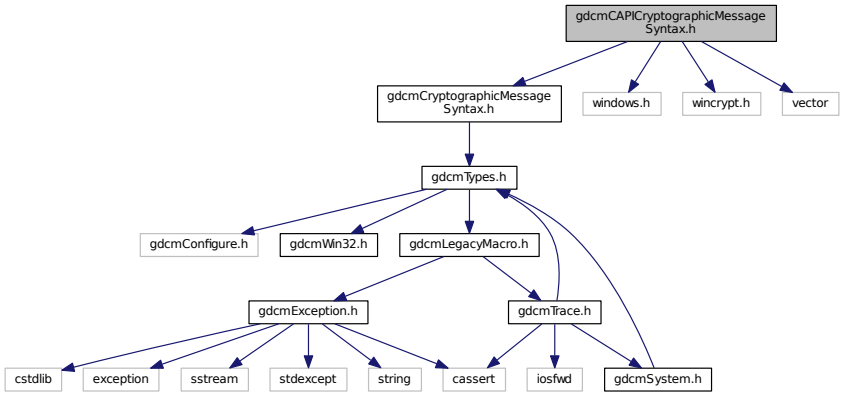
Namespaces

- 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

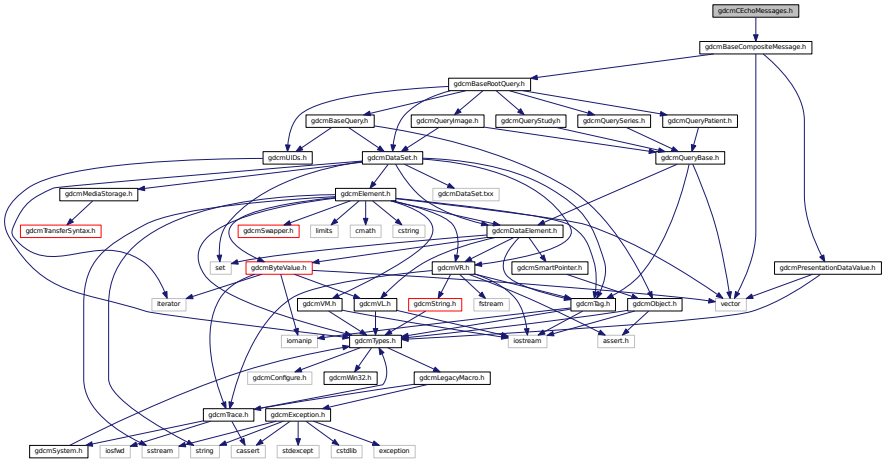
Namespaces

- gdcm

11.33 gdcmCEchoMessages.h File Reference

```
#include "gdcmBaseCompositeMessage.h"
```

Include dependency graph for gdcmCEchoMessages.h:



Classes

- class gdcm::network::CEchoRQ

  CEchoRQ this file defines the messages for the cecho action.

- class gdcm::network::CEchoRSP

  CEchoRSP this file defines the messages for the cecho action.

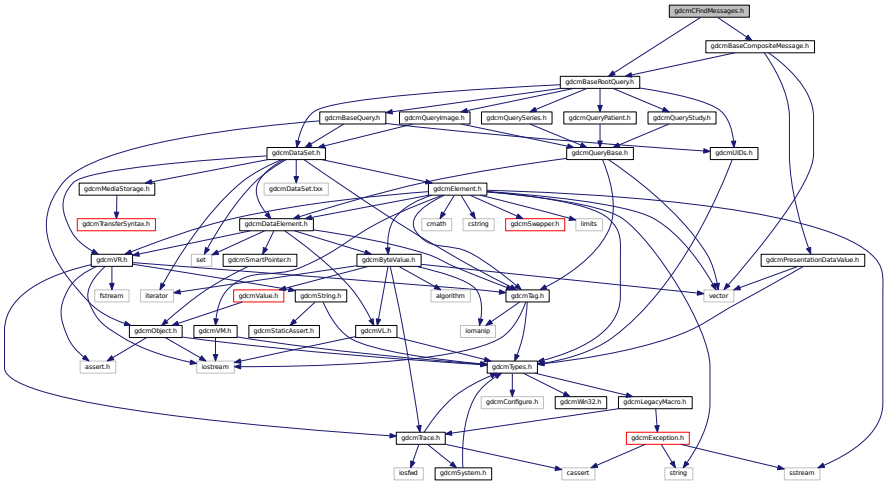
Namespaces

- 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 this file defines the messages for the cfind action.
- class gdcm::network::CFindRSP

  CFindRSP this file defines the messages for the cfind action.

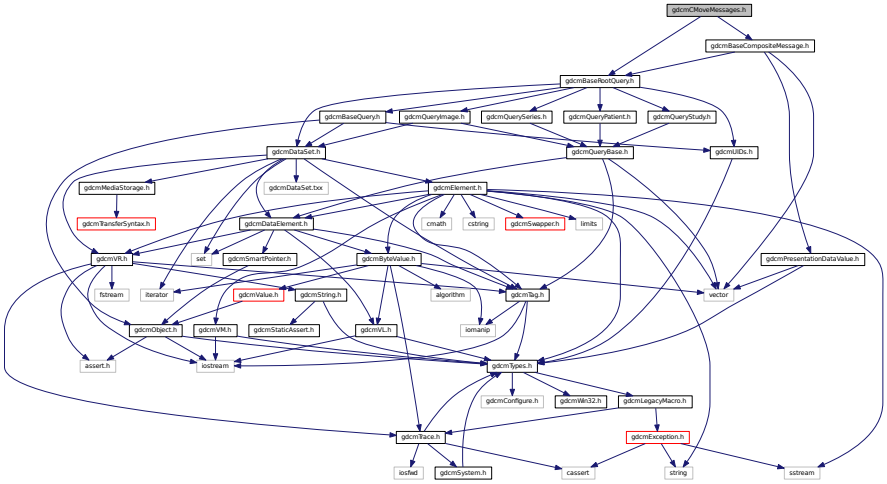
Namespaces

- 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 this file defines the messages for the cmove action.
- class gdcm::network::CMoveRSP

  CMoveRSP this file defines the messages for the cmove action.

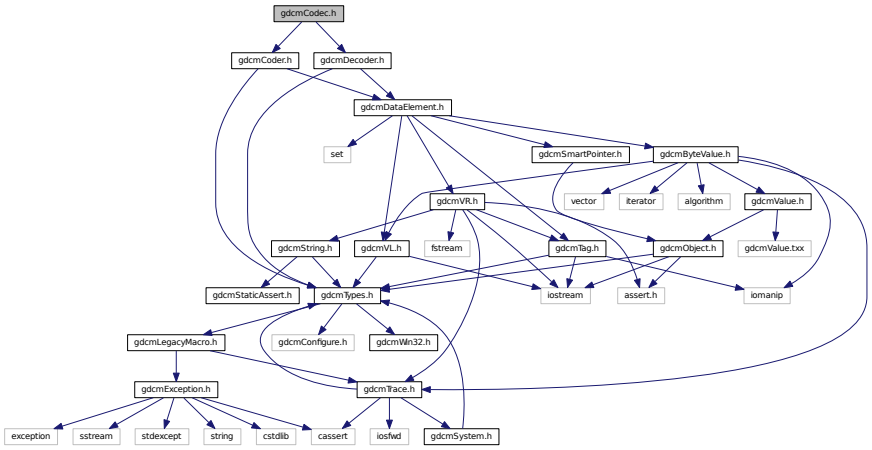
Namespaces

- gdcm
- gdcm::network

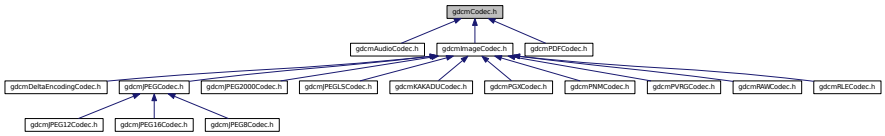
11.36 gdcmCodec.h File Reference

```
#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.

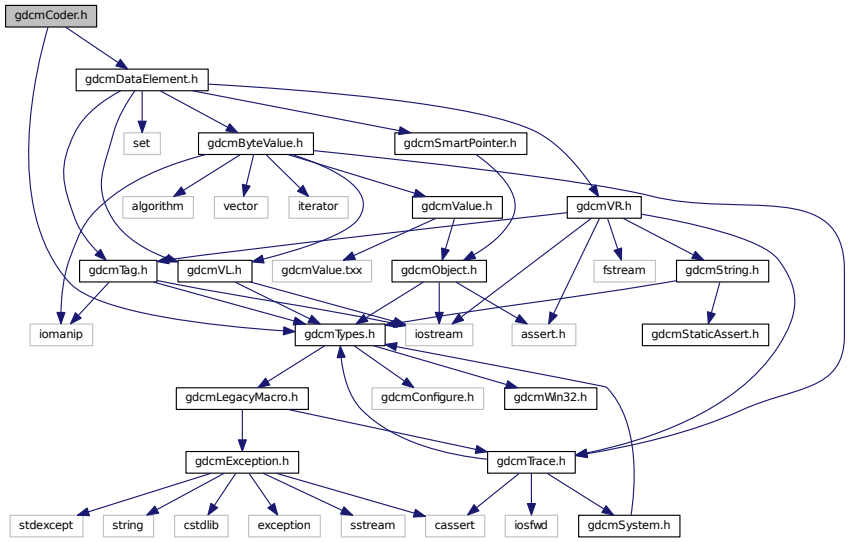
Namespaces

- gdcm

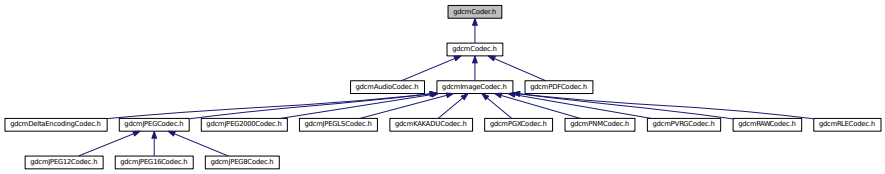
11.37 gdcmCoder.h File Reference

```
#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.

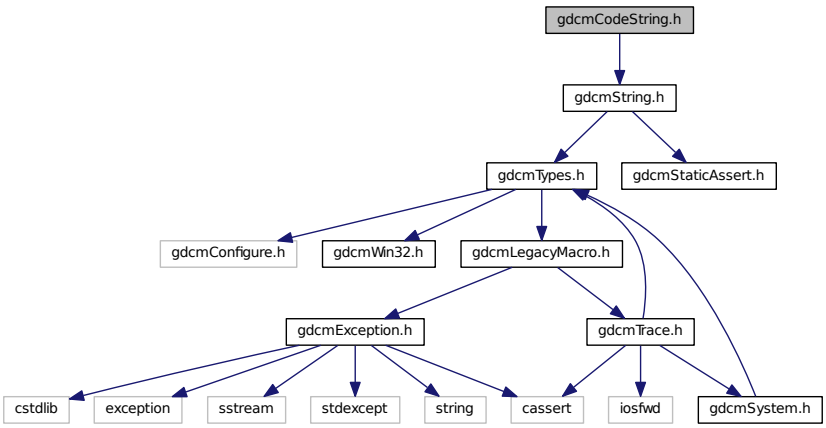
Namespaces

- gdcm

11.38 gdcmCodeString.h File Reference

```
#include "gdcmString.h"
```

Include dependency graph for gdcmCodeString.h:



Classes

- class gdcm::CodeString

  CodeString This is an implementation of DICOM VR: CS The cstor will properly Trim so that operator== is correct.

Namespaces

- gdcm

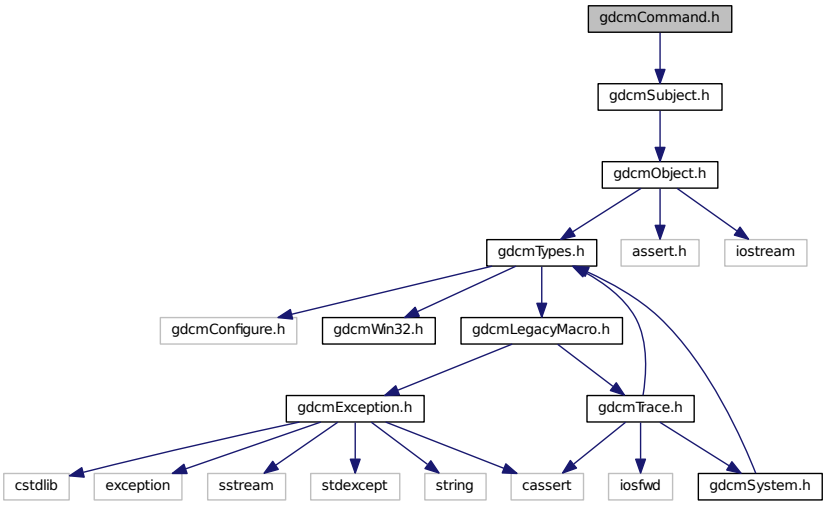
Functions

- 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)

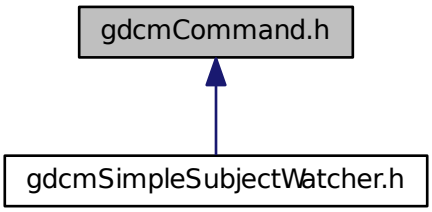
11.39 gdcmCommand.h File Reference

```
#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.

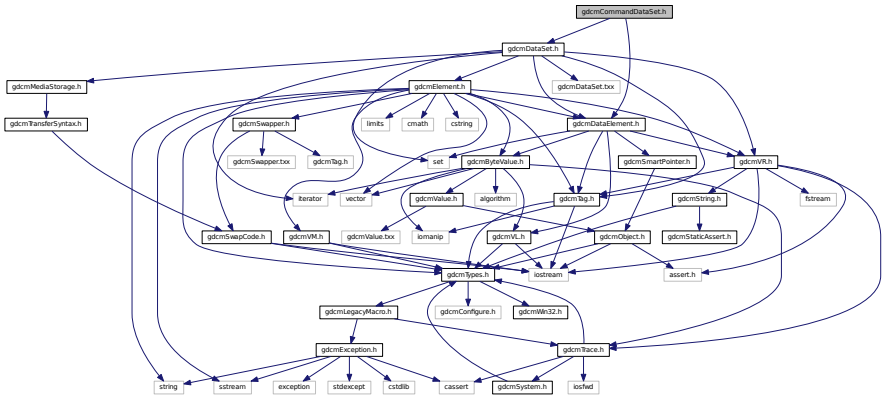
Namespaces

- 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

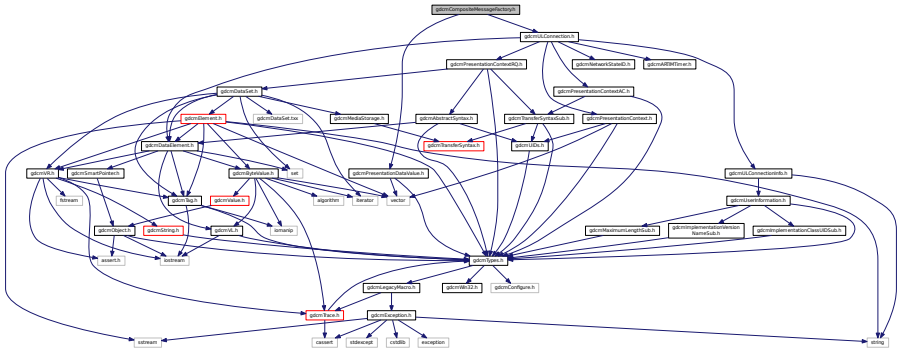
Functions

- 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 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).

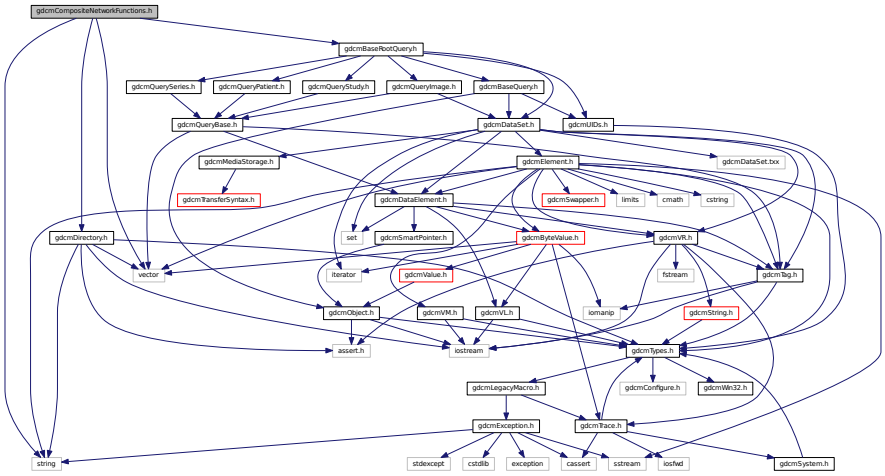
Namespaces

- 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 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:

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.

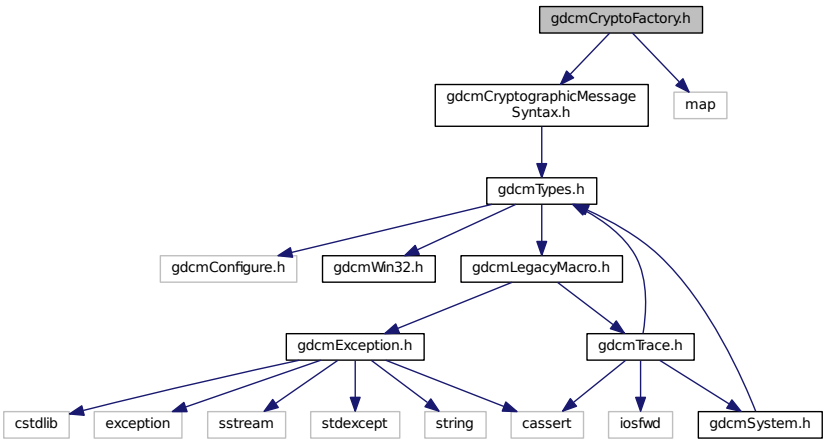
Namespaces

- gdcm

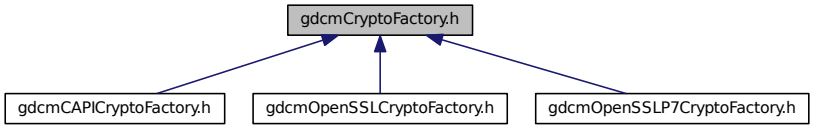
11.45 gdcmCryptoFactory.h File Reference

```
#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.

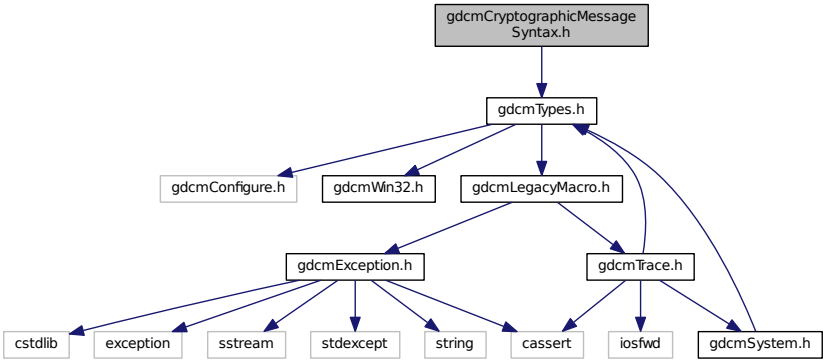
Namespaces

- gdcm

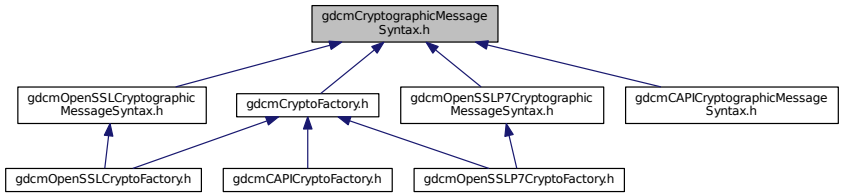
11.46 gdcmCryptographicMessageSyntax.h File Reference

```
#include "gdcmTypes.h"
```

Include dependency graph for gdcmCryptographicMessageSyntax.h:



This graph shows which files directly or indirectly include this file:



Classes

- class gdcm::CryptographicMessageSyntax

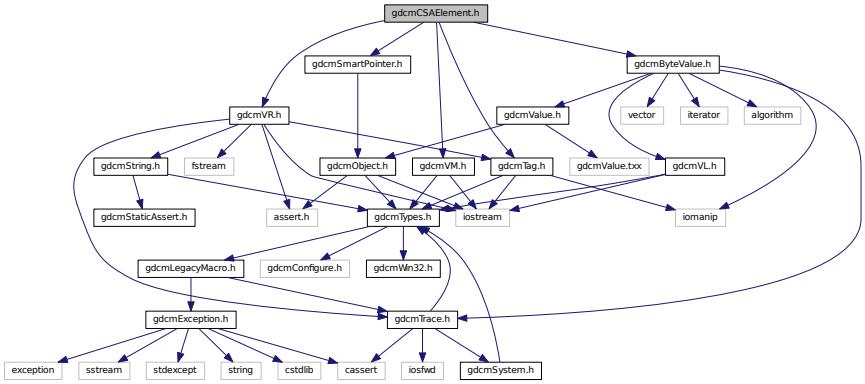
Namespaces

- gdcm

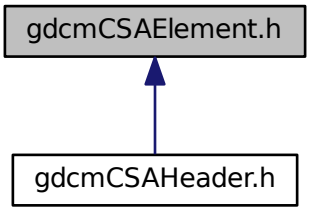
11.47 gdcmCSAElement.h File Reference

```
#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

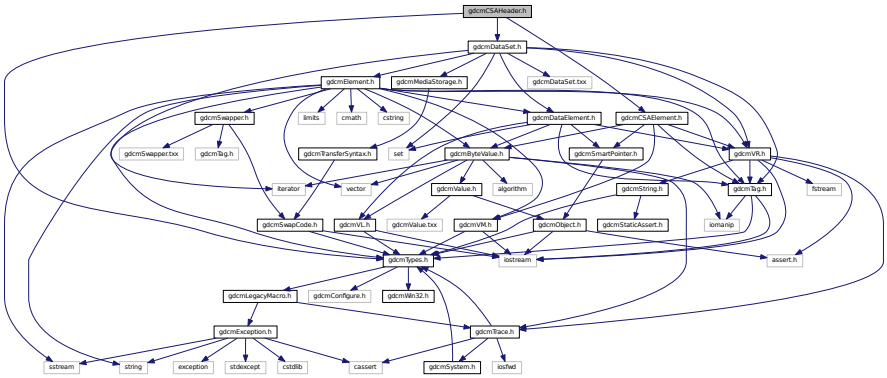
Functions

- 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

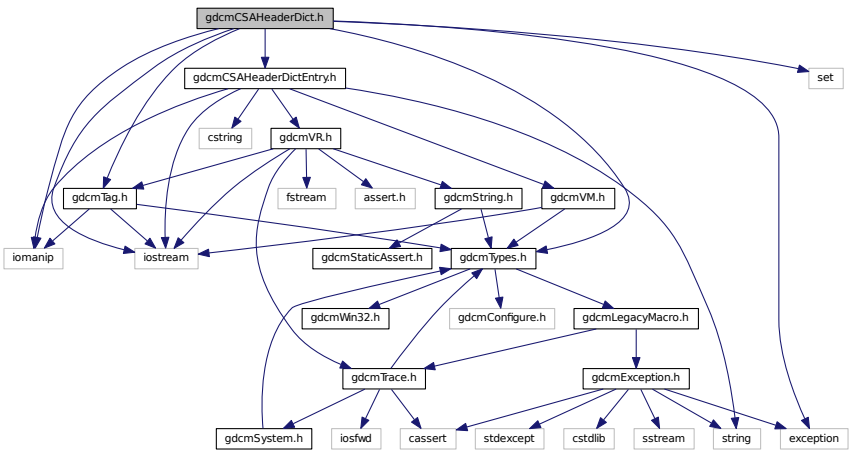
Functions

- std::ostream & gdcm::operator<< (std::ostream &os, const CSAHeader &d)

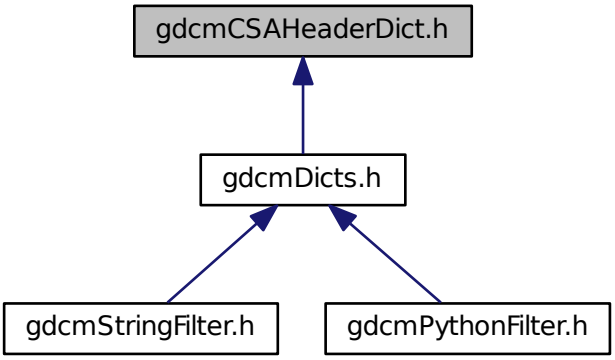
11.49 gdcmCSAHeaderDict.h File Reference

```
#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

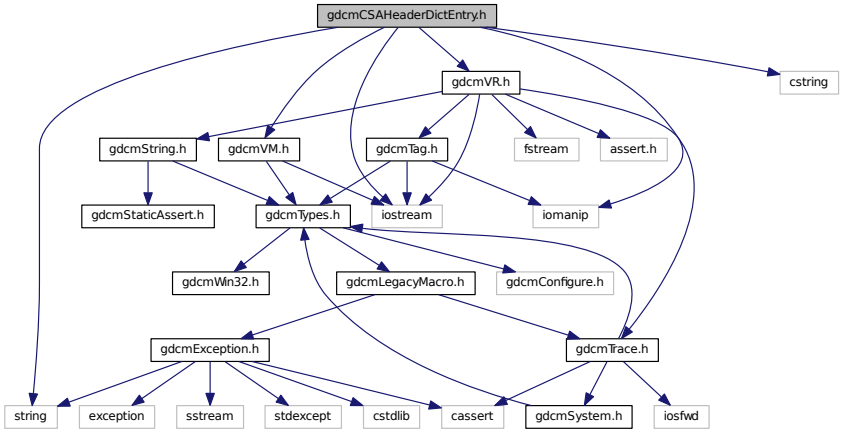
Functions

- std::ostream & gdcm::operator<< (std::ostream &os, const CSAHeaderDict &val)

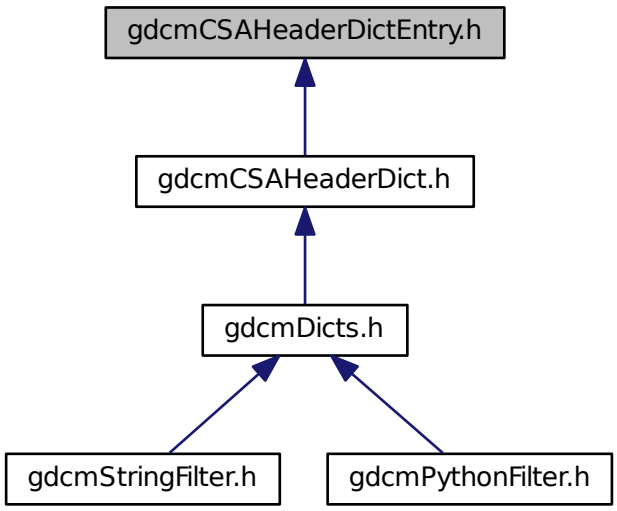
11.50 gdcmCSAHeaderDictEntry.h File Reference

```
#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 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.

Namespaces

- gdcm

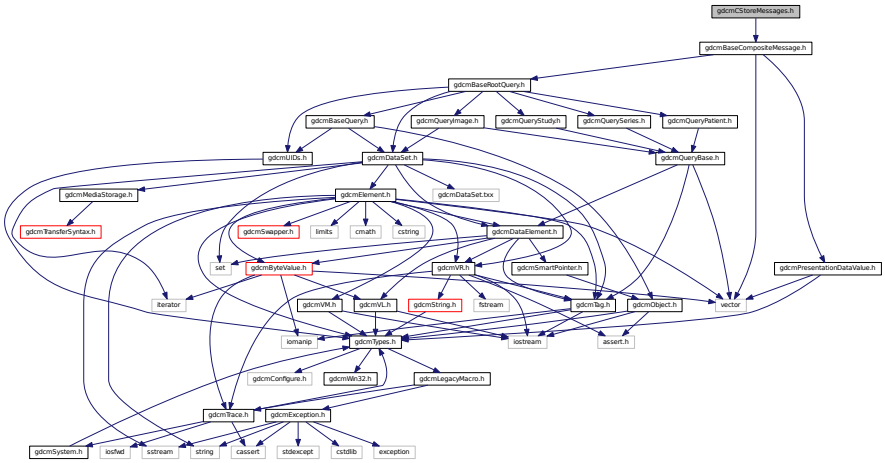
Functions

- 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 this file defines the messages for the cecho action.
- class gdcm::network::CStoreRSP

  CStoreRSP this file defines the messages for the cecho action.

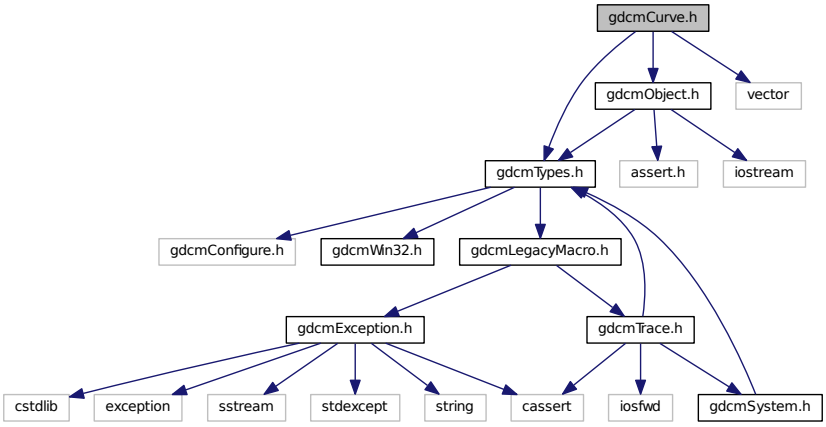
Namespaces

- gdcm
- gdcm::network

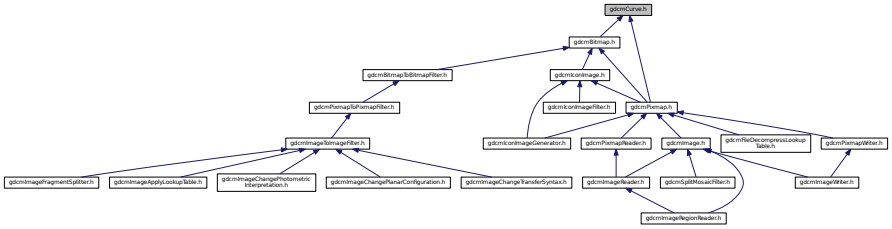
11.52 gdcmCurve.h File Reference

```
#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 WARNING: This is deprecated and lastly defined in PS 3.3 - 2004.

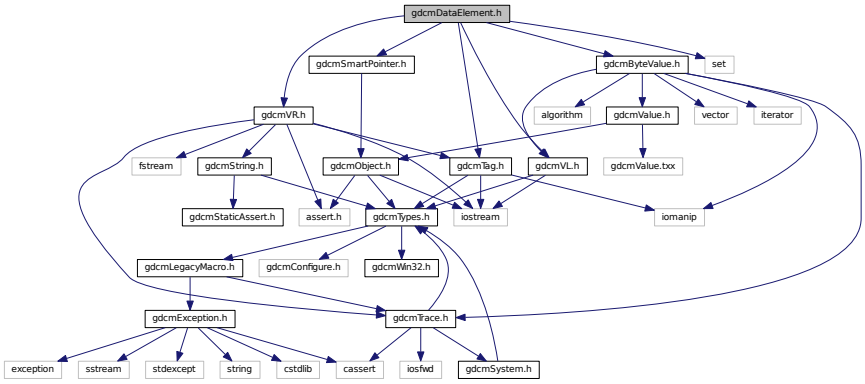
Namespaces

- gdcm

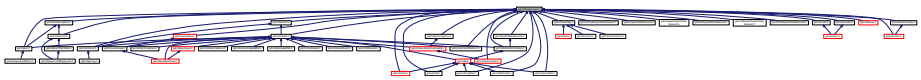
11.53 gdcmDataElement.h File Reference

```
#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

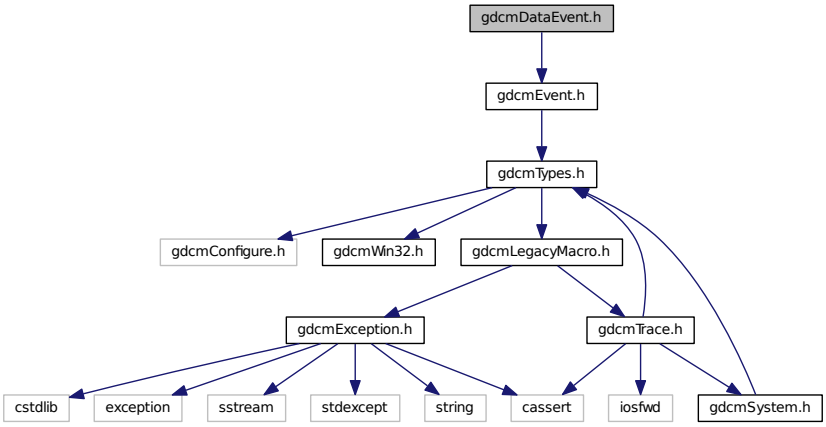
Functions

- bool gdcm::operator!= (const DataElement &lhs, const DataElement &rhs)
- std::ostream & gdcm::operator<< (std::ostream &os, const DataElement &val)

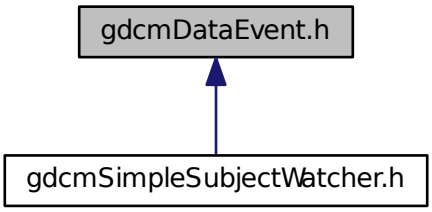
11.54 gdcmDataEvent.h File Reference

```
#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.

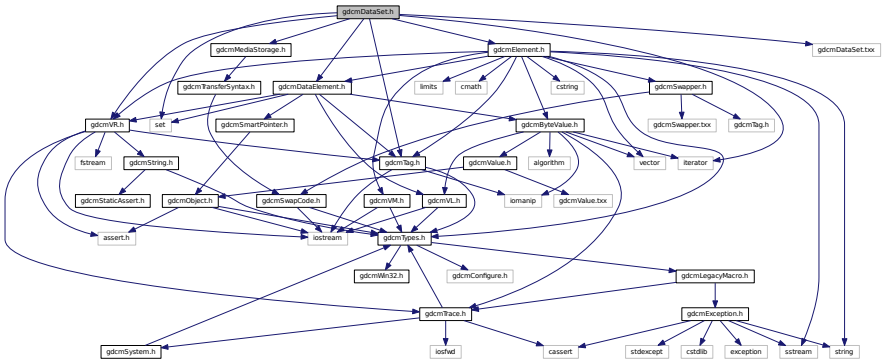
Namespaces

- gdcm

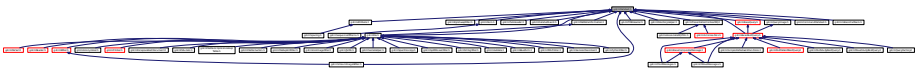
11.55 gdcmDataSet.h File Reference

```
#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) A Data Set represents an instance of a real world Information Object.

Namespaces

- gdcm

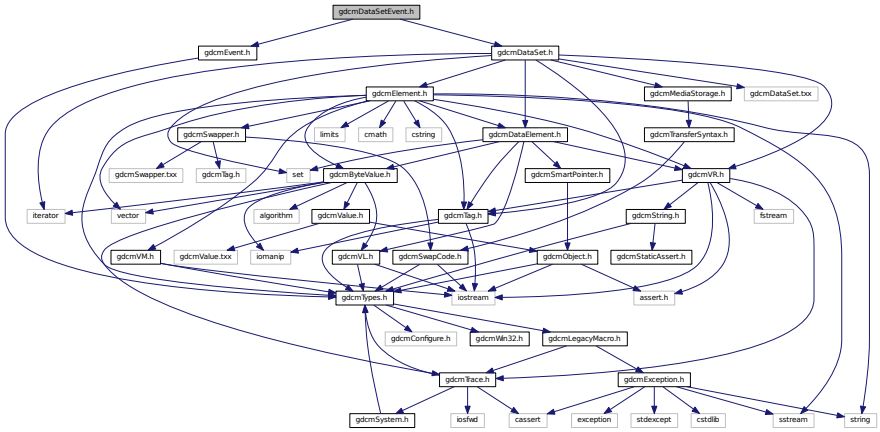
Functions

- 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 Special type of event triggered during the DataSet store/move process.

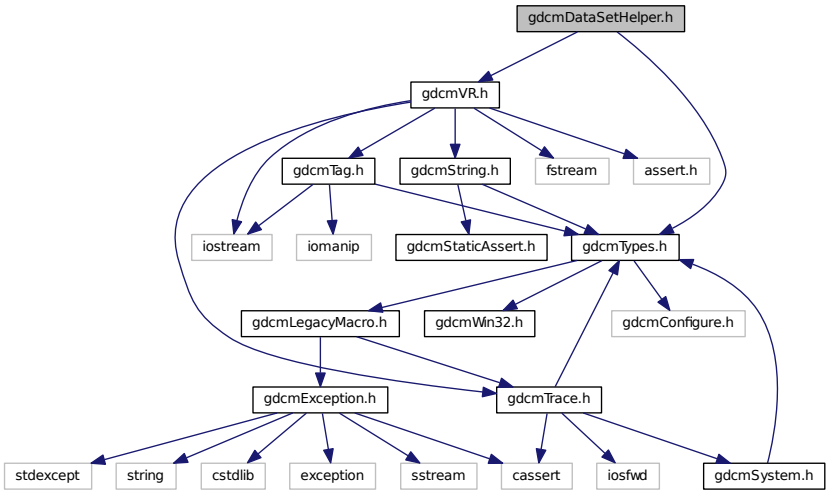
Namespaces

- 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)

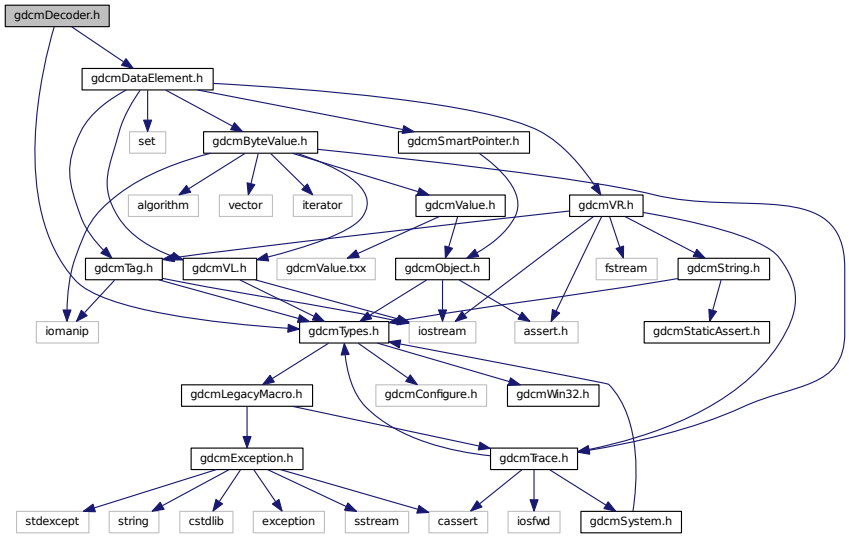
Namespaces

- gdcm

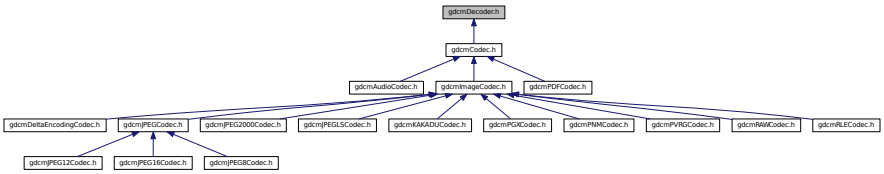
11.58 gdcmDecoder.h File Reference

```
#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.

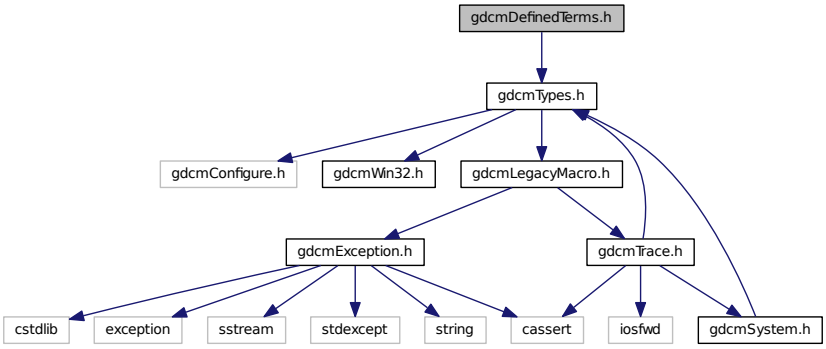
Namespaces

- 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.

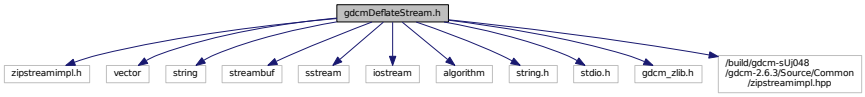
Namespaces

- gdcm

11.60 gdcmDeflateStream.h File Reference

```
#include "zipstreamimpl.h"
```

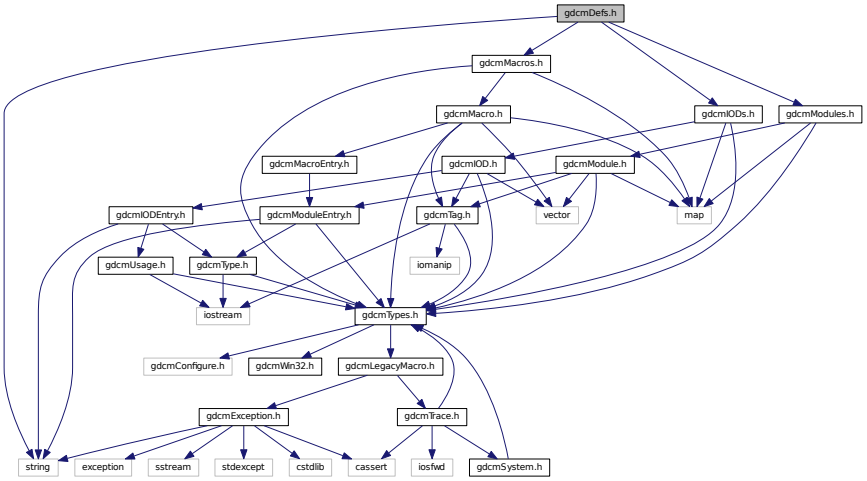
Include dependency graph for gdcmDeflateStream.h:



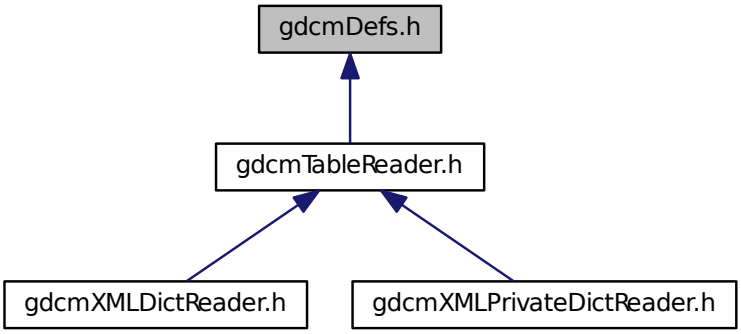
11.61 gdcmDefs.h File Reference

```
#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'.

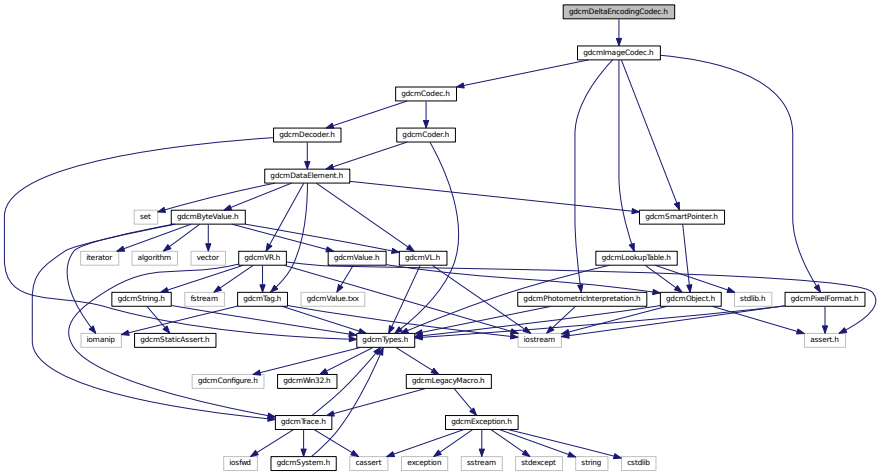
Namespaces

- 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.

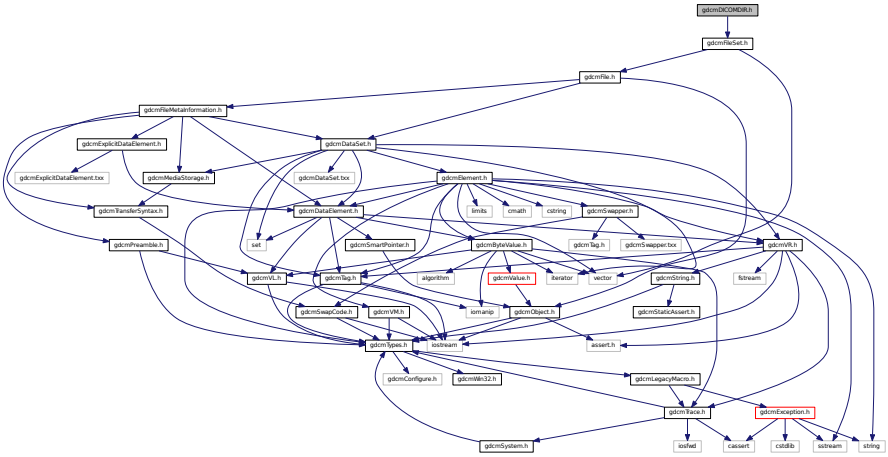
Namespaces

- gdcm

11.63 gdcmDICOMDIR.h File Reference

```
#include "gdcmFileSet.h"
```

Include dependency graph for gdcmDICOMDIR.h:



Classes

- class gdcm::DICOMDIR

  DICOMDIR class.

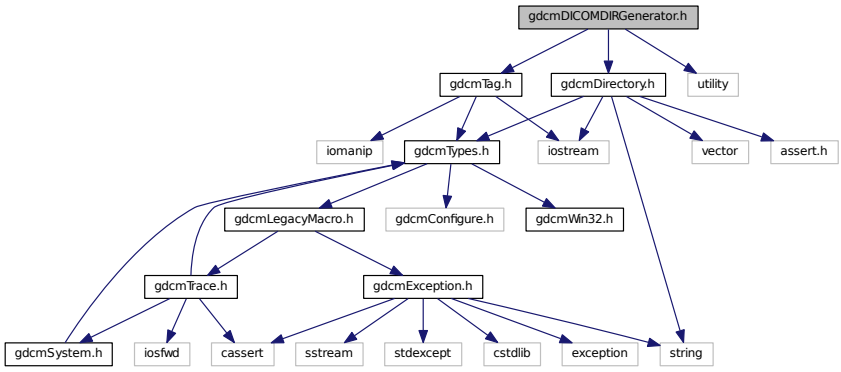
Namespaces

- 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 This is a STD-GEN-CD DICOMDIR generator. ref: PS 3.11-2008 Annex D (Normative) - General Purpose CD-R and DVD Interchange Profiles.

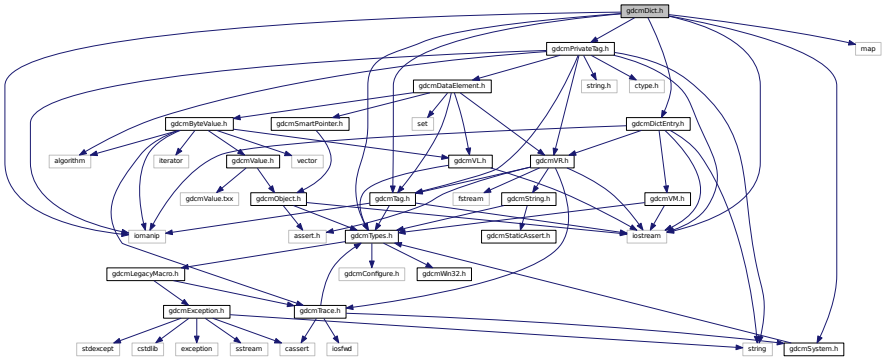
Namespaces

- gdcm

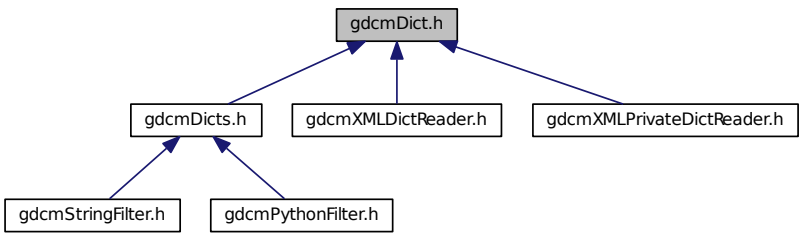
11.65 gdcmDict.h File Reference

```
#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

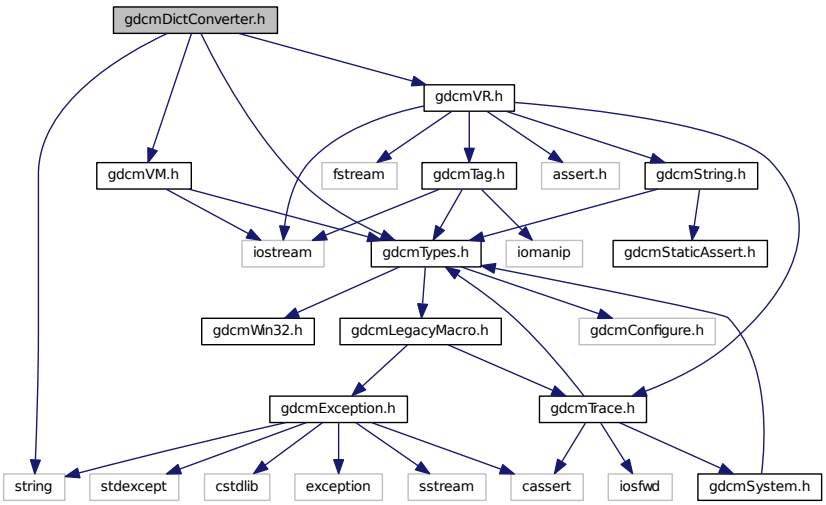
Functions

- 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:

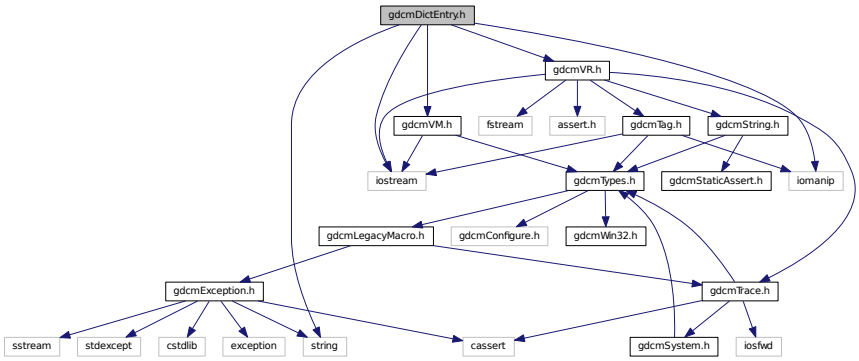
Namespaces

- gdcm

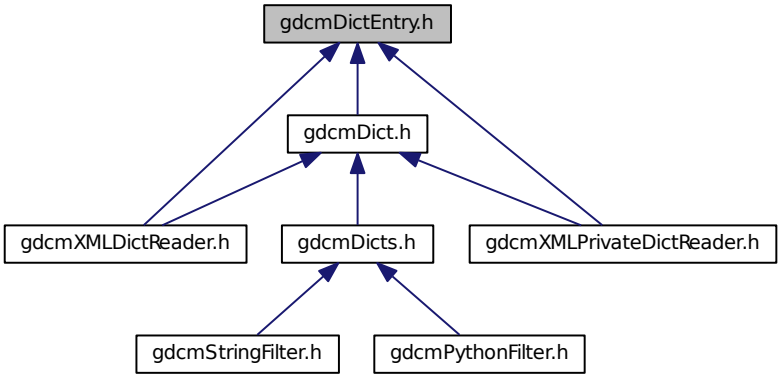
11.67 gdcmDictEntry.h File Reference

```
#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 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.

Namespaces

- gdcm

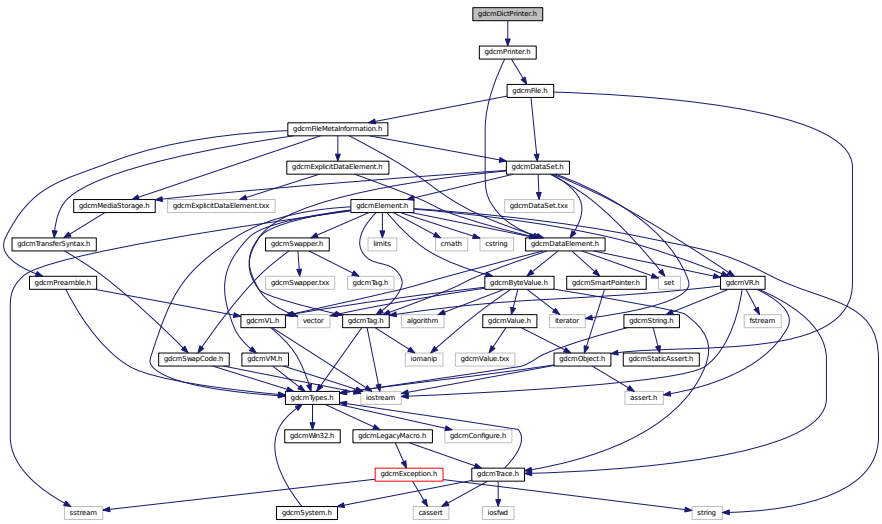
Functions

- 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.

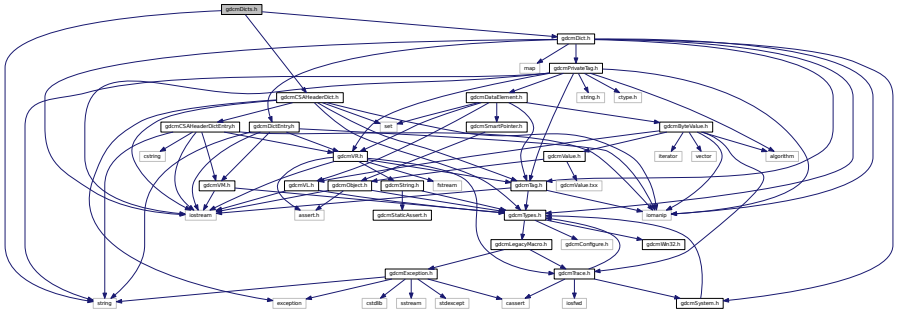
Namespaces

- gdcm

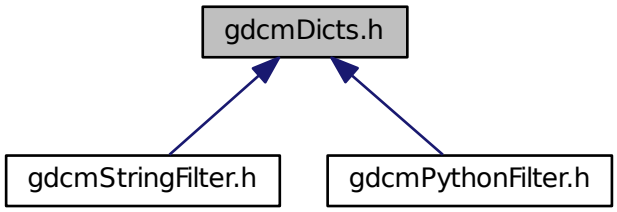
11.69 gdcmDicts.h File Reference

```
#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

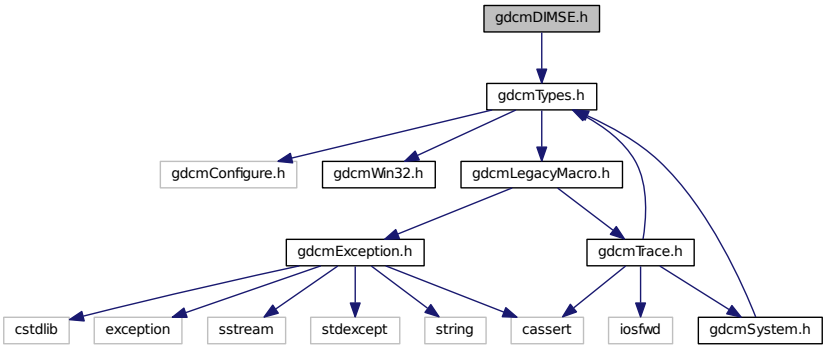
Functions

- 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 this file defines the messages for the cecho action.
- class gdcm::network::CEchoRSP

  CEchoRSP this file defines the messages for the cecho action.
- class gdcm::network::CFind
- class gdcm::network::DIMSE

  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)

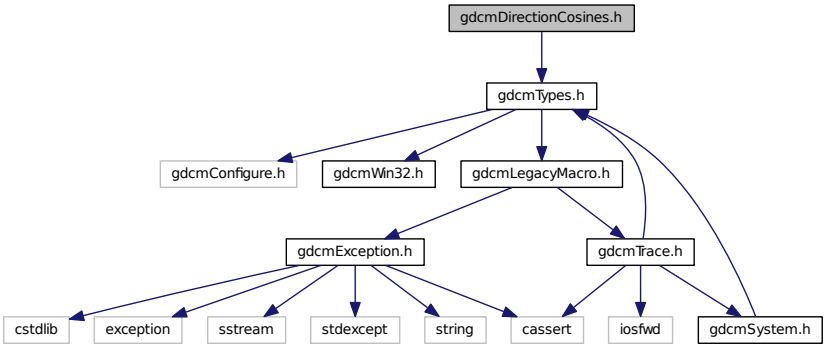
Namespaces

- 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

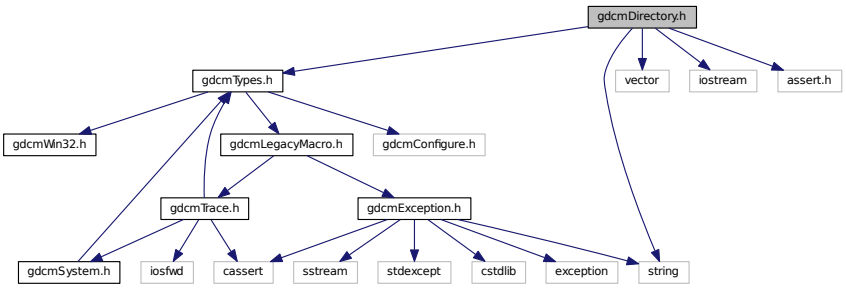
Namespaces

- gdcm

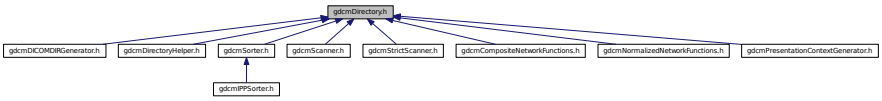
11.72 gdcmDirectory.h File Reference

```
#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

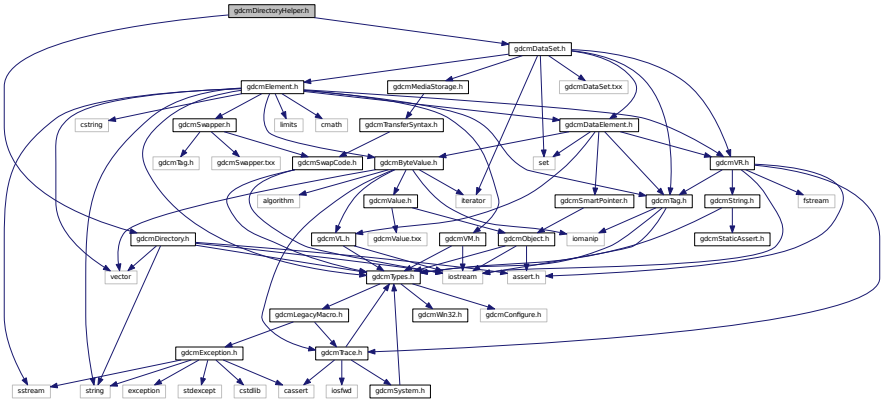
Functions

- 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 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.

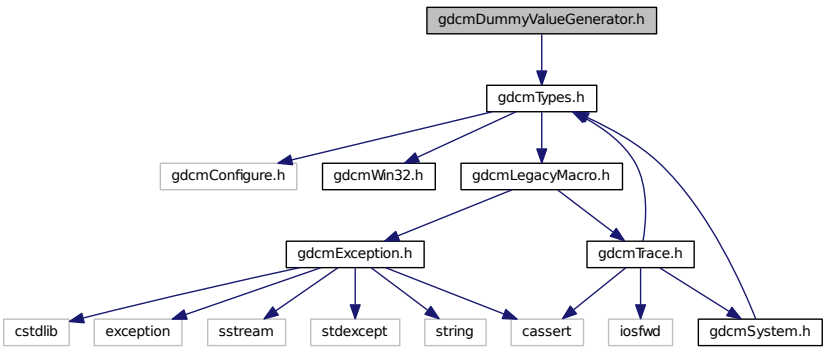
Namespaces

- 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.

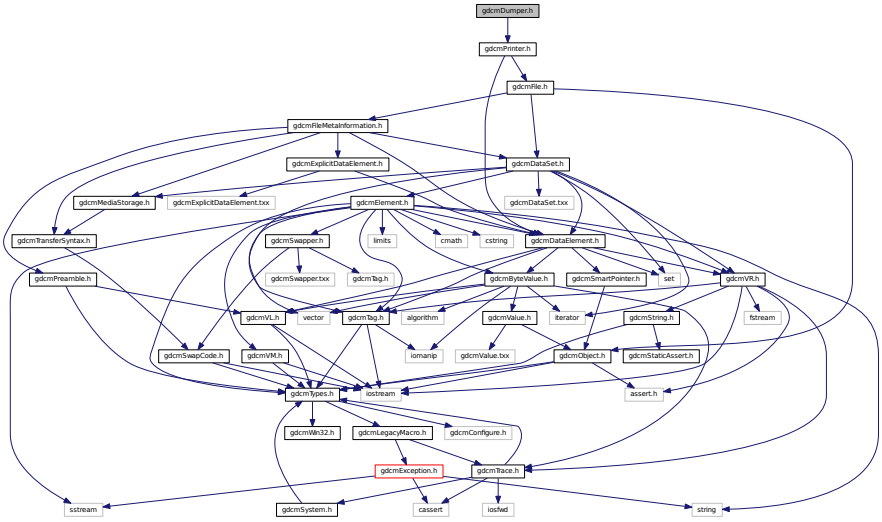
Namespaces

- gdcm

11.75 gdcmDumper.h File Reference

```
#include "gdcmPrinter.h"
```

Include dependency graph for gdcmDumper.h:



Classes

- class gdcm::Dumper

  Codec class.

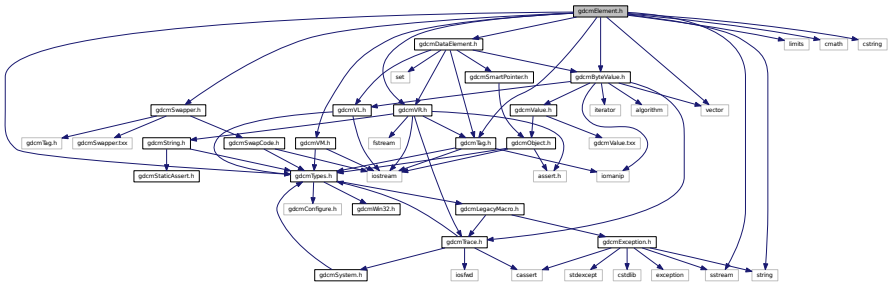
Namespaces

- 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)

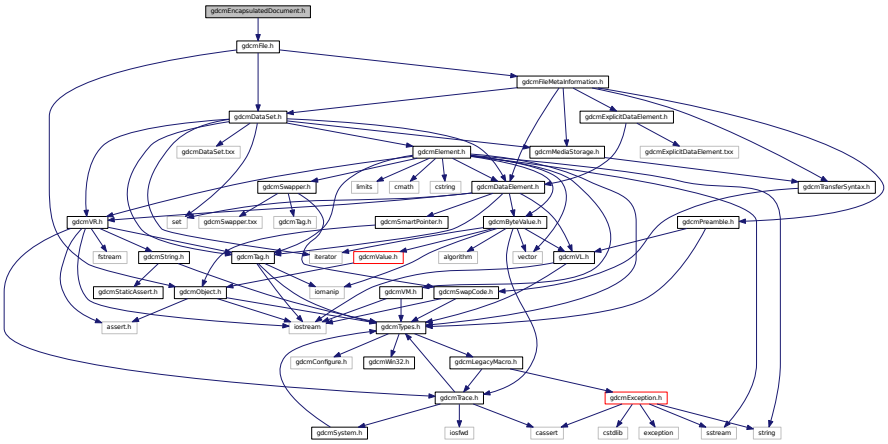
11.76.1 Macro Definition Documentation

11.76.1.1 #define VRDS16ILLEGAL

11.77 gdcmEncapsulatedDocument.h File Reference

```
#include "gdcmFile.h"
```

Include dependency graph for gdcmEncapsulatedDocument.h:



Classes

- class gdcm::EncapsulatedDocument
  EncapsulatedDocument.

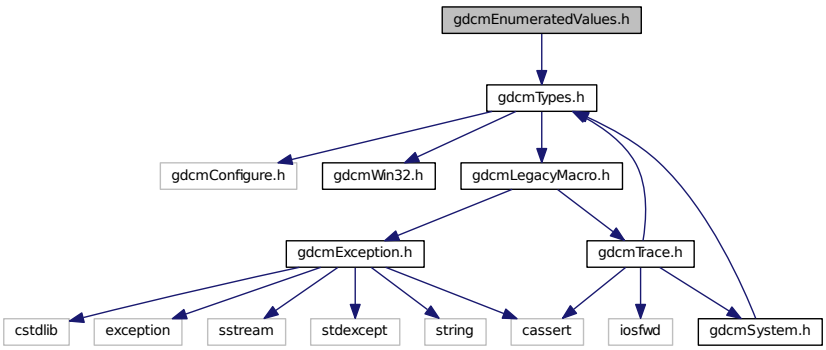
Namespaces

- 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:

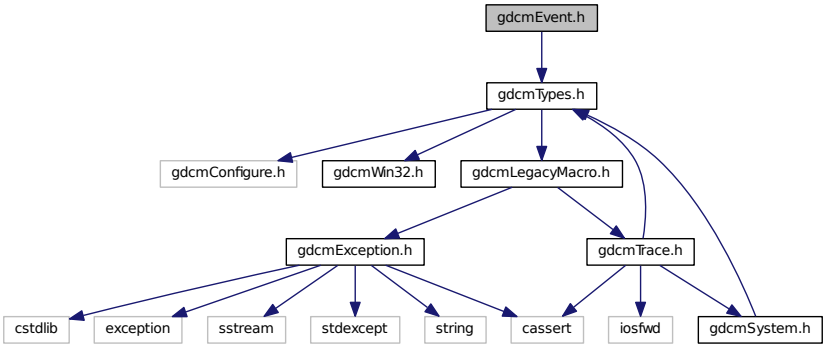
Namespaces

- gdcm

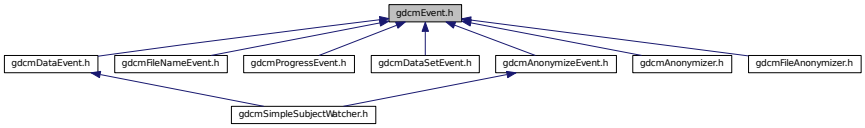
11.79 gdcmEvent.h File Reference

```
#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 #define gdcmEventMacro(*classname*, *super*)

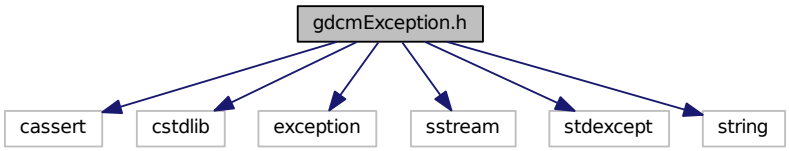
Value:

```
 \
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.

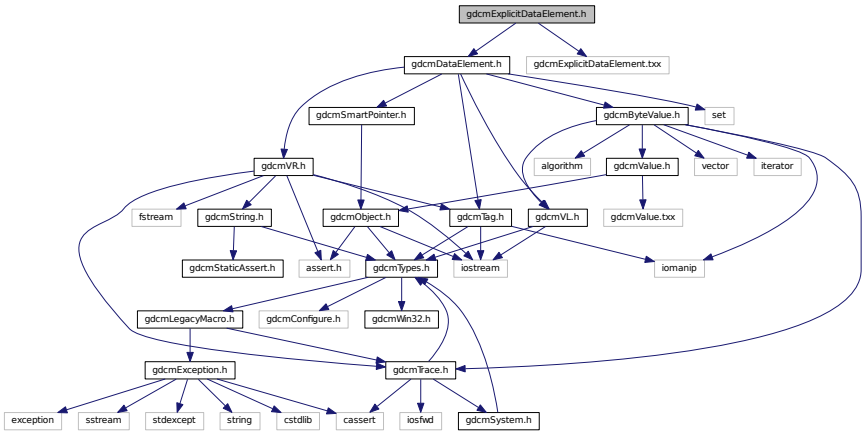
Namespaces

- gdcm

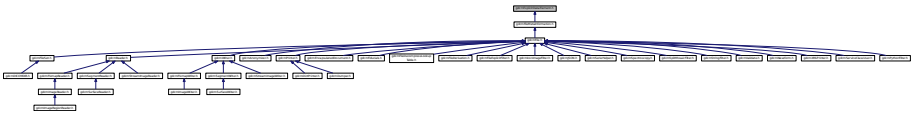
11.81 gdcmExplicitDataElement.h File Reference

```
#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.

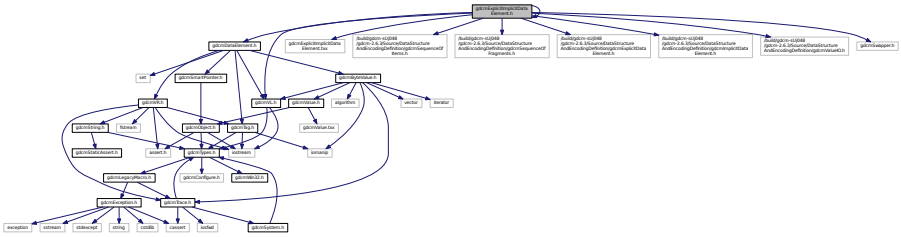
Namespaces

- 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.

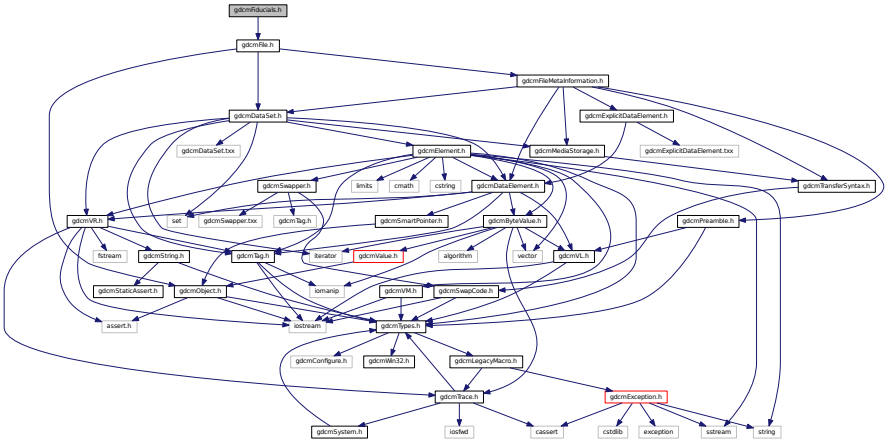
Namespaces

- gdcm

11.83 gdcmFiducials.h File Reference

```
#include "gdcmFile.h"
```

Include dependency graph for gdcmFiducials.h:



Classes

- class gdcm::Fiducials

  Fiducials.

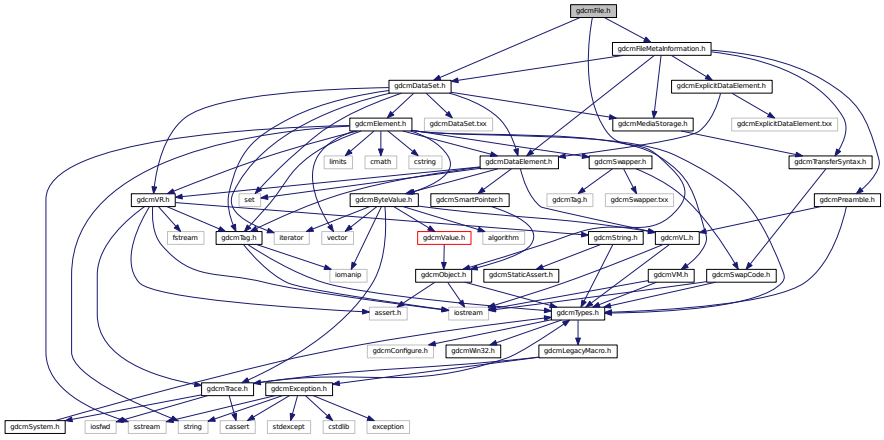
Namespaces

- 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 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.

Namespaces

- gdcm

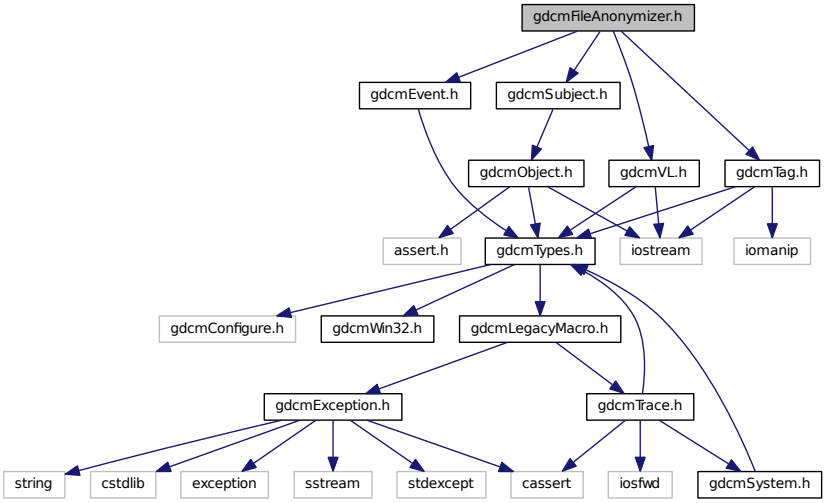
Functions

- 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.

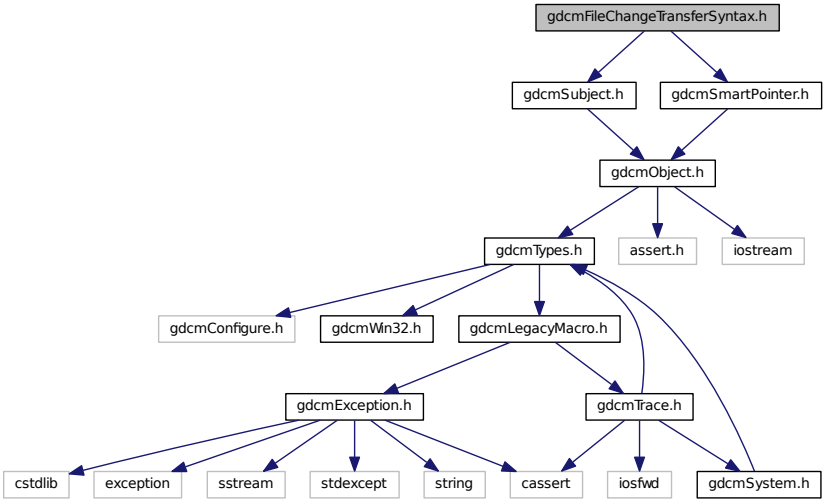
Namespaces

- gdcm

11.86 gdcmFileChangeTransferSyntax.h File Reference

```
#include "gdcmSubject.h"
#include "gdcmSmartPointer.h"
```

Include dependency graph for gdcmFileChangeTransferSyntax.h:



Classes

- class gdcm::FileChangeTransferSyntax

  FileChangeTransferSyntax.

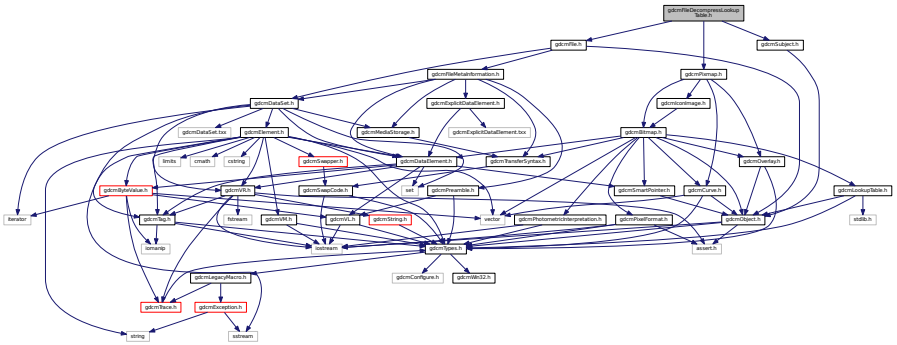
Namespaces

- 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 It decompress the segmented LUT into linearized one (only PALETTE_COLOR images) Output will be a PhotometricInterpretation=RGB image.

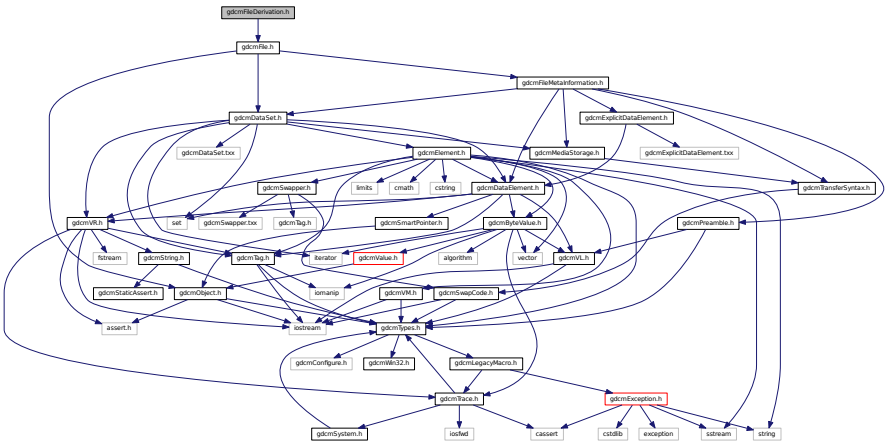
Namespaces

- gdcm

11.88 gdcmFileDerivation.h File Reference

```
#include "gdcmFile.h"
```

Include dependency graph for gdcmFileDerivation.h:



Classes

- class gdcm::FileDerivation

  FileDerivation class See PS 3.16 - 2008 For the list of Code Value that can be used for in Derivation Code Sequence.

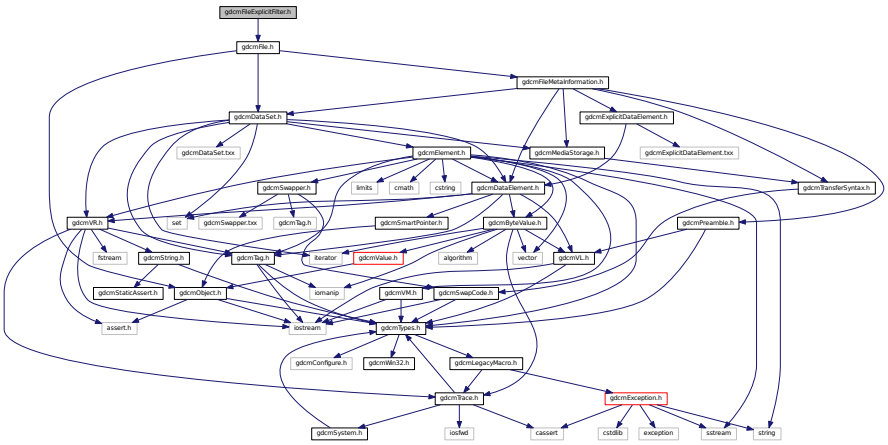
Namespaces

- gdcm

11.89 gdcmFileExplicitFilter.h File Reference

```
#include "gdcmFile.h"
```

Include dependency graph for gdcmFileExplicitFilter.h:



Classes

- class gdcm::FileExplicitFilter

  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.

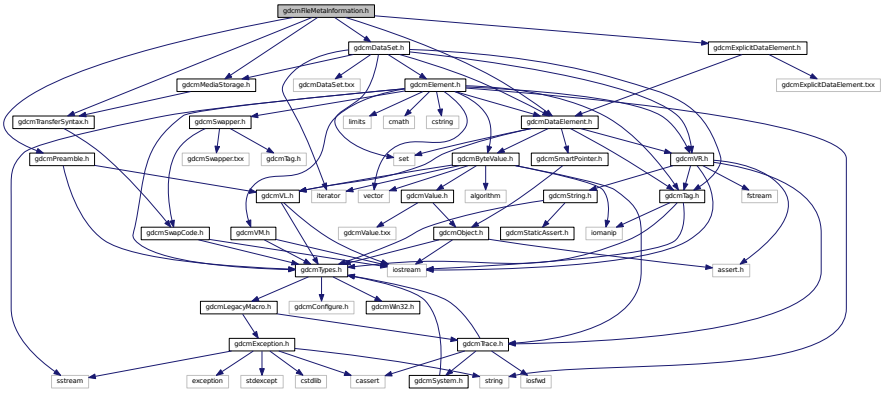
Namespaces

- gdcm

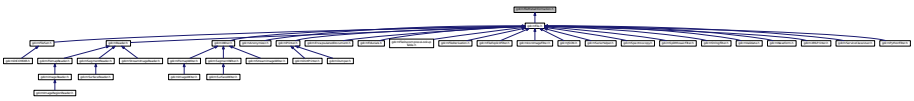
11.90 gdcmFileMetaInformation.h File Reference

```
#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

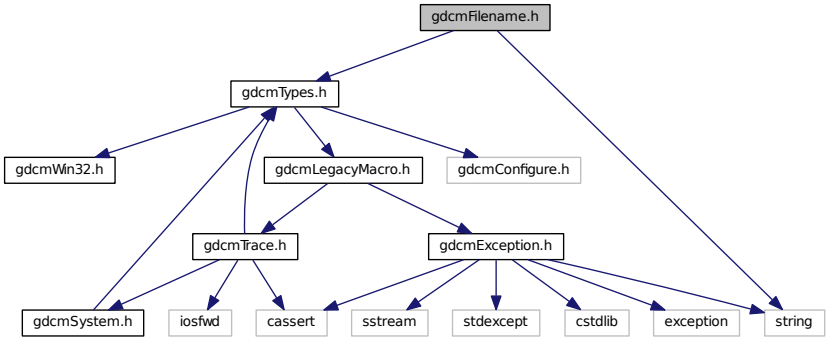
Functions

- 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.

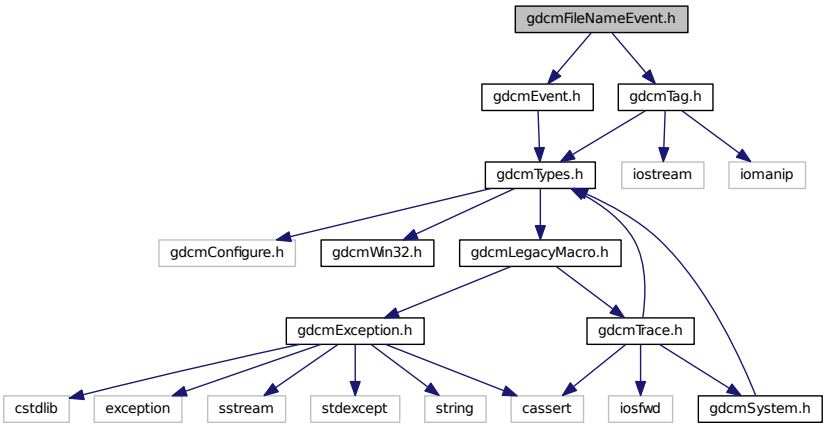
Namespaces

- gdcm

11.92 gdcmFileNameEvent.h File Reference

```
#include "gdcmEvent.h"
#include "gdcmTag.h"
```

Include dependency graph for gdcmFileNameEvent.h:



Classes

- class gdcm::FileNameEvent

  FileNameEvent Special type of event triggered during processing of FileSet.

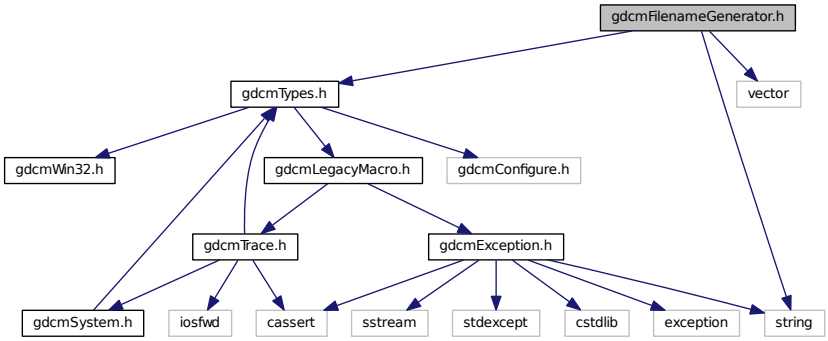
Namespaces

- 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.

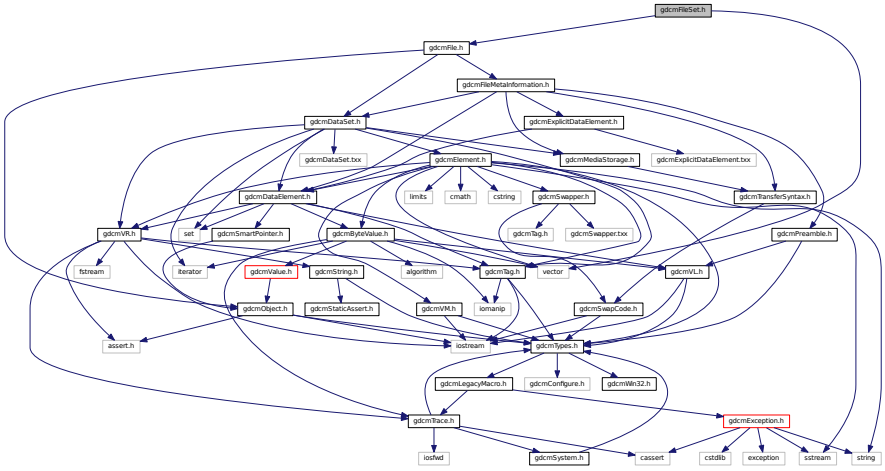
Namespaces

- gdcm

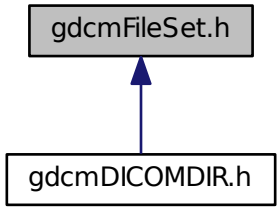
11.94 gdcmFileSet.h File Reference

```
#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

  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.

Namespaces

- gdcm

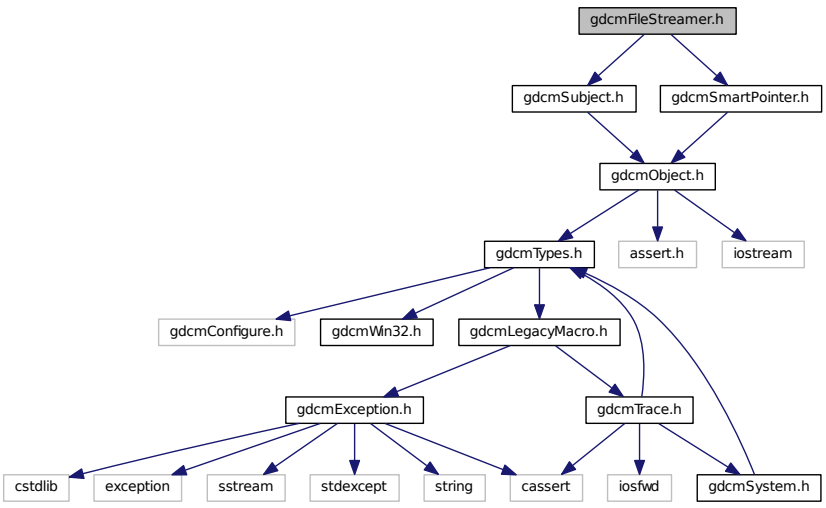
Functions

- 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 This class let a user create a massive DICOM DataSet from a template DICOM file, by appending chunks of data.

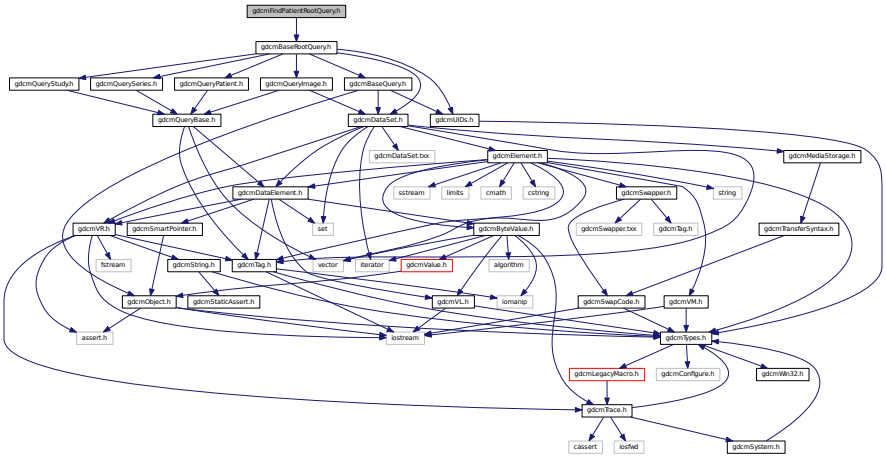
Namespaces

- gdcm

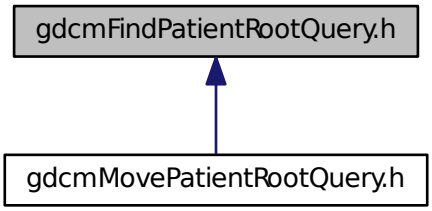
11.96 gdcmFindPatientRootQuery.h File Reference

```
#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 contains: the class which will produce a dataset for c-find with patient root.

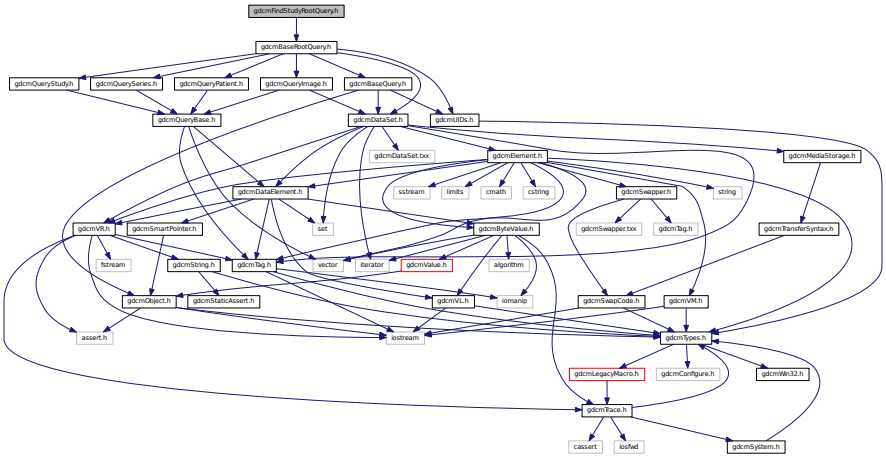
Namespaces

- gdcm

11.97 gdcmFindStudyRootQuery.h File Reference

```
#include "gdcmBaseRootQuery.h"
```

Include dependency graph for gdcmFindStudyRootQuery.h:



Classes

- class gdcm::FindStudyRootQuery

  FindStudyRootQuery contains: the class which will produce a dataset for C-FIND with study root.

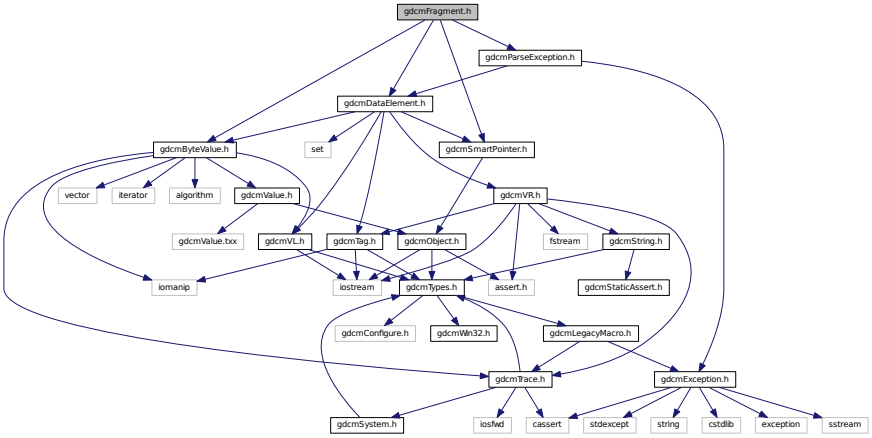
Namespaces

- gdcm

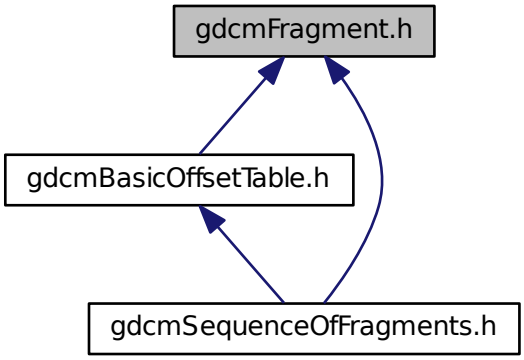
11.98 gdcmFragment.h File Reference

```
#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

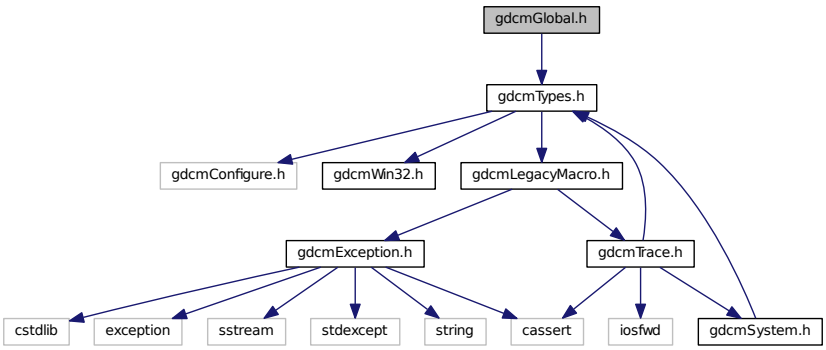
Functions

- 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)

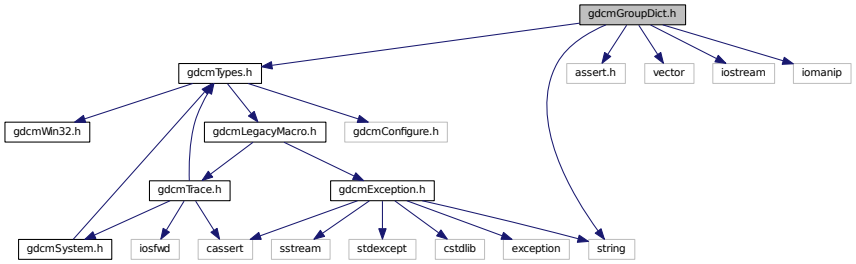
Variables

- 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

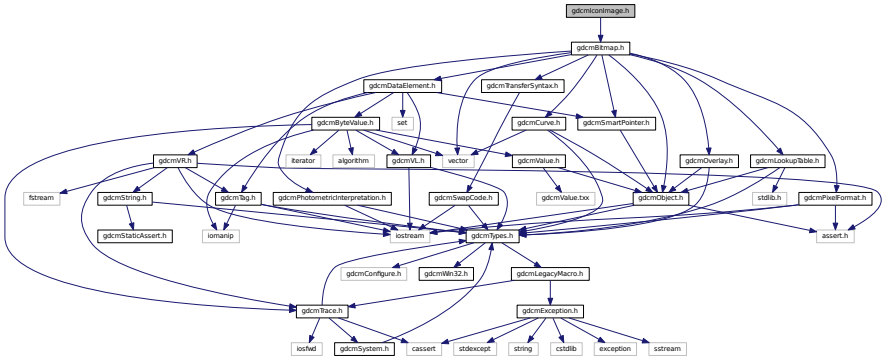
Functions

- std::ostream & gdcm::operator<< (std::ostream &_os, const GroupDict &_val)

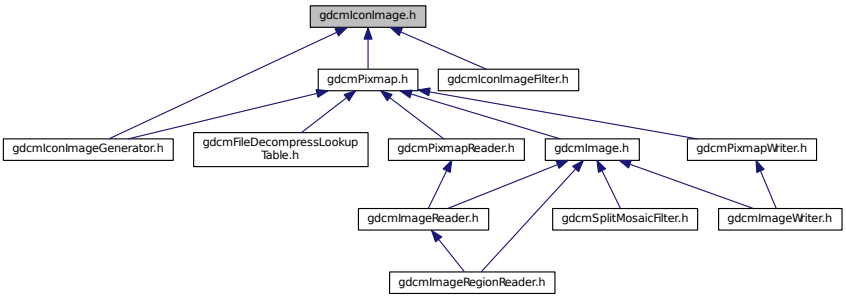
11.101 gdcmIconImage.h File Reference

```
#include "gdcmBitmap.h"
```

Include dependency graph for gdcmIconImage.h:



This graph shows which files directly or indirectly include this file:



Namespaces

- gdcm

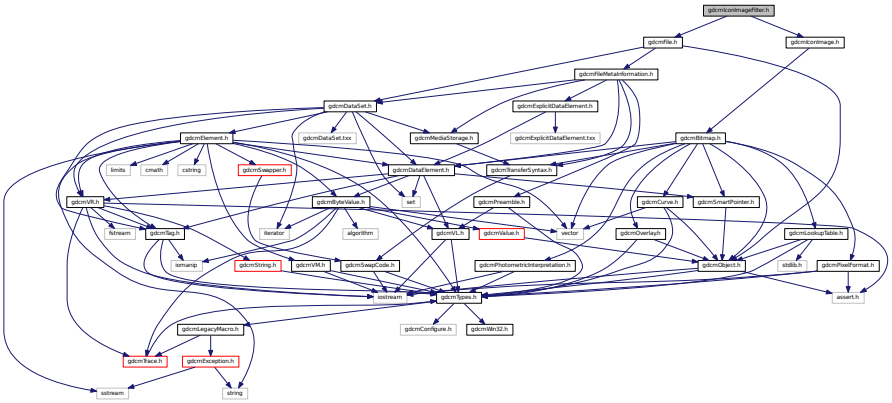
Typedefs

- 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 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.

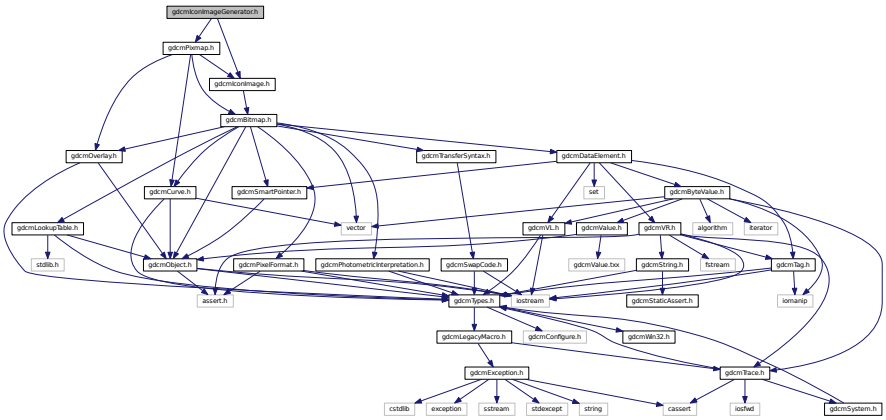
Namespaces

- gdcm

11.103 gdcmIconImageGenerator.h File Reference

```
#include "gdcmPixmap.h"
#include "gdcmIconImage.h"
```

Include dependency graph for gdcmIconImageGenerator.h:



Classes

- class gdcm::IconImageGenerator

  IconImageGenerator 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:

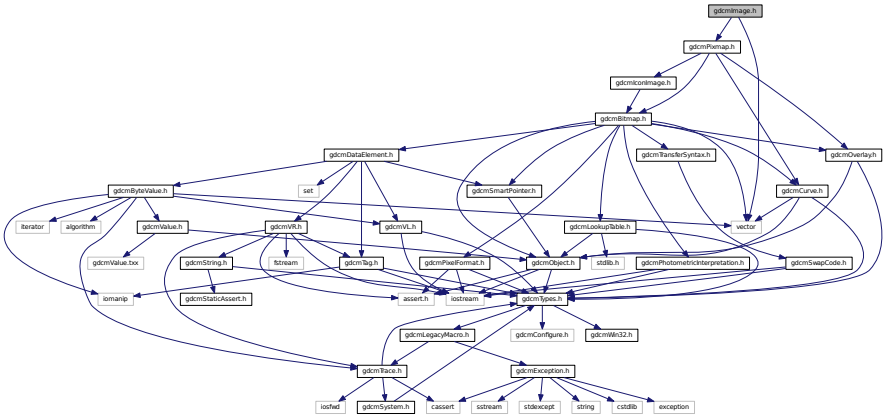
Namespaces

- gdcm

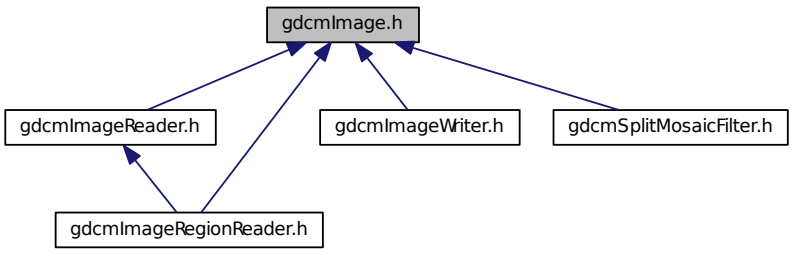
11.104 gdcmImage.h File Reference

```
#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 This is the container for an Image in the general sense. From this container you should be able to request information like:

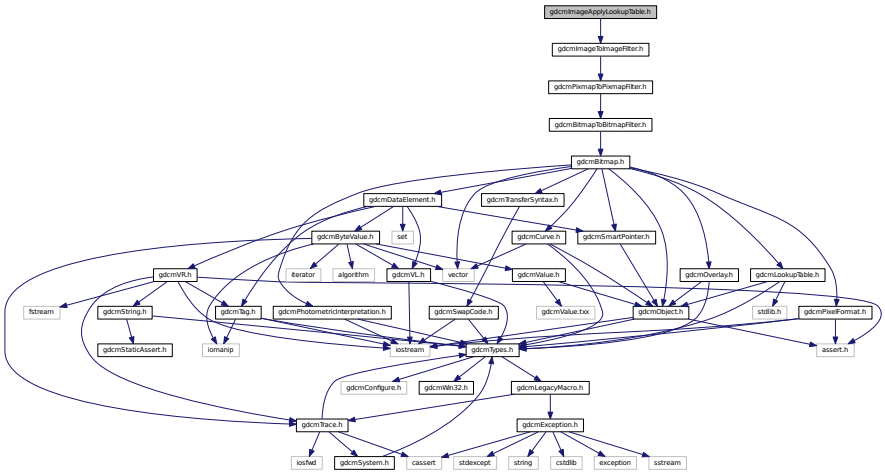
Namespaces

- gdcm

11.105 gdcmImageApplyLookupTable.h File Reference

```
#include "gdcmImageToImageFilter.h"
```

Include dependency graph for gdcmImageApplyLookupTable.h:



Classes

- class gdcm::ImageApplyLookupTable

  ImageApplyLookupTable class It applies the LUT the PixelData (only PALETTE_COLOR images) Output will be a PhotometricInterpretation=RGB image.

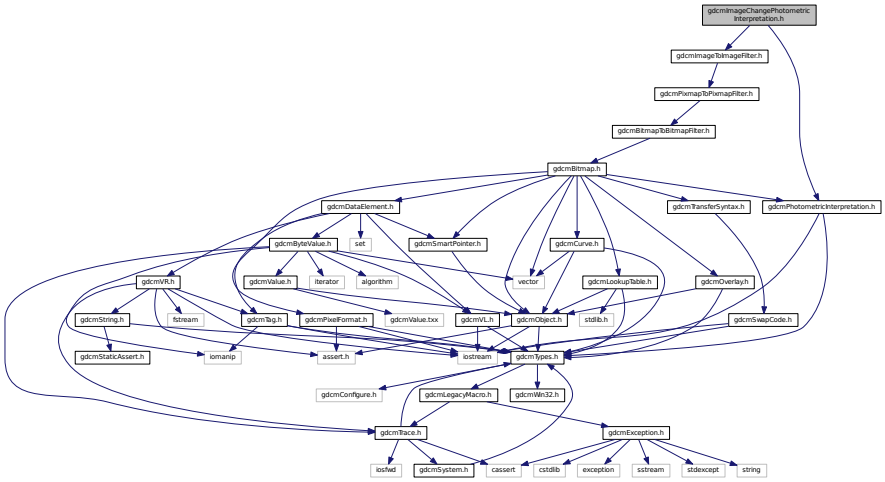
Namespaces

- 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 Class to change the Photometric Interpetation of an input DICOM.

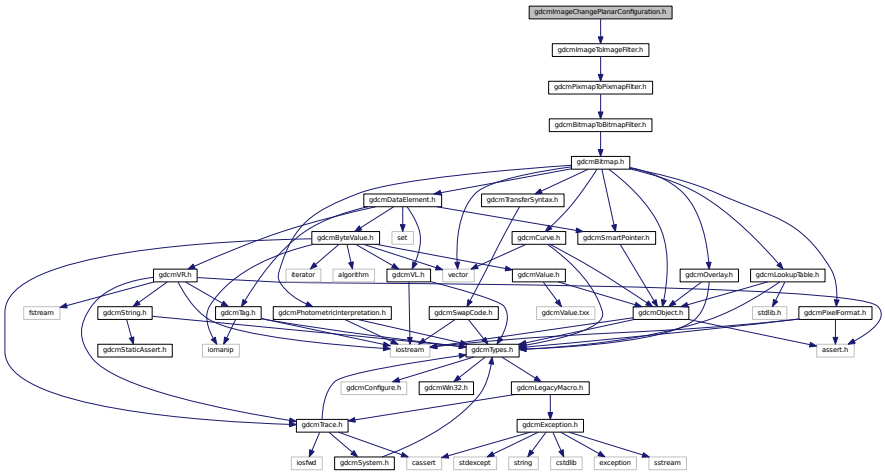
Namespaces

- gdcm

11.107 gdcmImageChangePlanarConfiguration.h File Reference

```
#include "gdcmImageToImageFilter.h"
```

Include dependency graph for gdcmImageChangePlanarConfiguration.h:



Classes

- class gdcm::ImageChangePlanarConfiguration

  ImageChangePlanarConfiguration class Class to change the Planar configuration of an input DICOM By default it will change into the more usual reprensentation: PlanarConfiguration = 0.

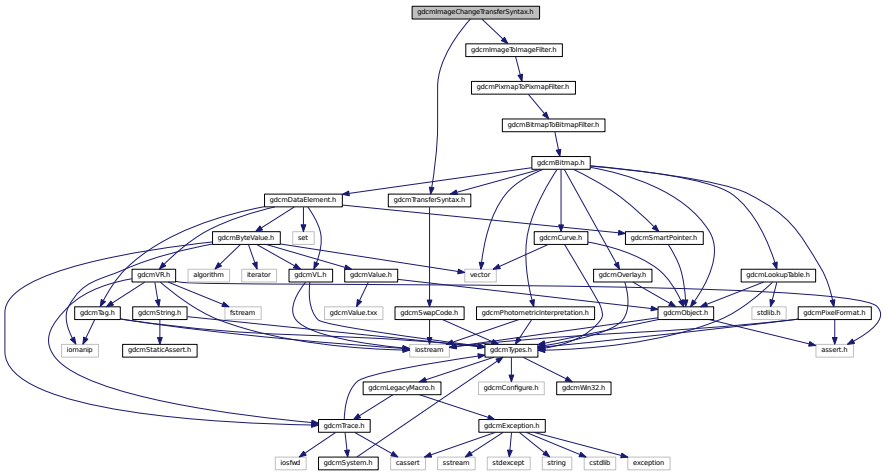
Namespaces

- 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 Class to change the transfer syntax of an input DICOM.

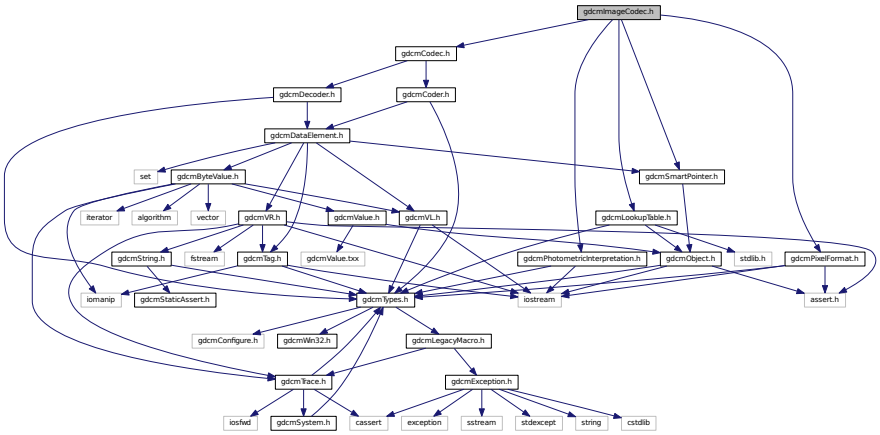
Namespaces

- 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.

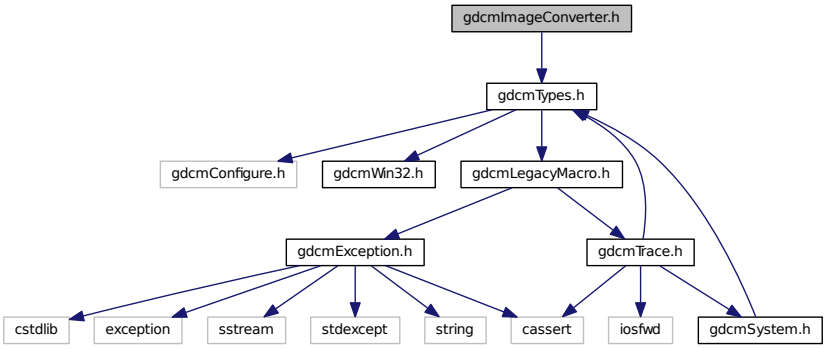
Namespaces

- gdcm

11.110 gdcmImageConverter.h File Reference

```
#include "gdcmTypes.h"
```

Include dependency graph for gdcmImageConverter.h:



Classes

- class gdcm::ImageConverter

  Image Converter.

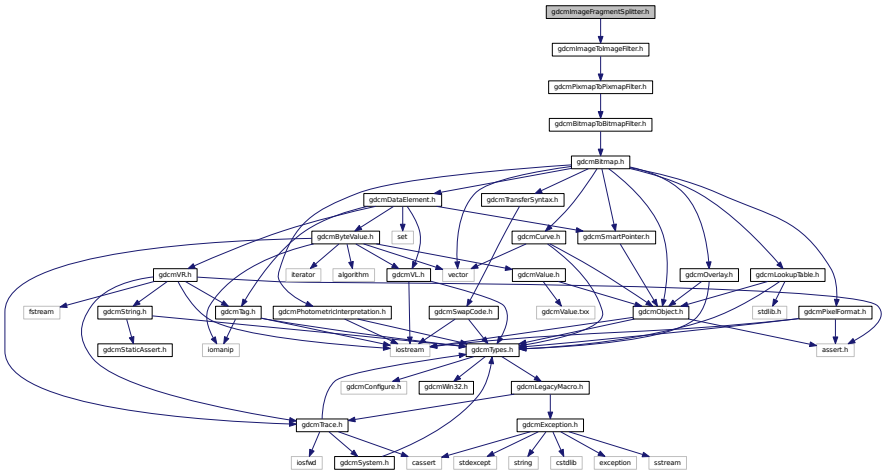
Namespaces

- gdcm

11.111 gdcmImageFragmentSplitter.h File Reference

```
#include "gdcmImageToImageFilter.h"
```

Include dependency graph for gdcmImageFragmentSplitter.h:



Classes

- class gdcm::ImageFragmentSplitter

  ImageFragmentSplitter class For single frame image, DICOM standard allow splitting the frame into multiple fragments.

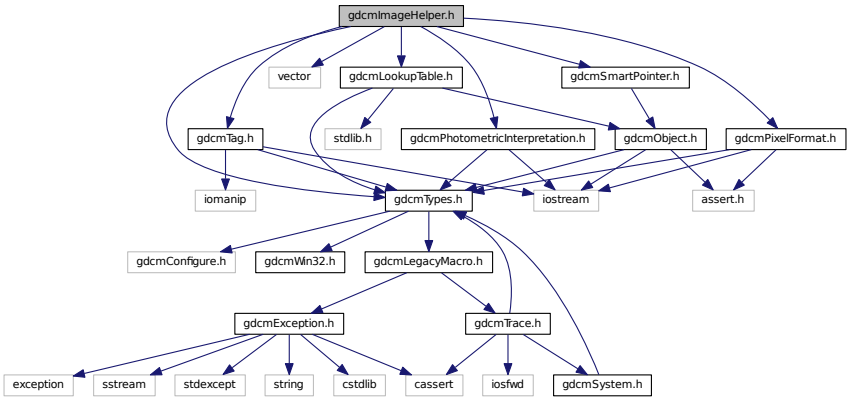
Namespaces

- 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

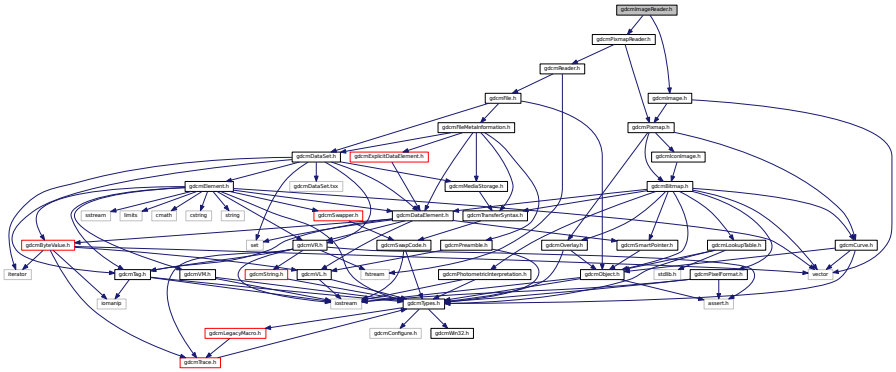
Namespaces

- gdcm

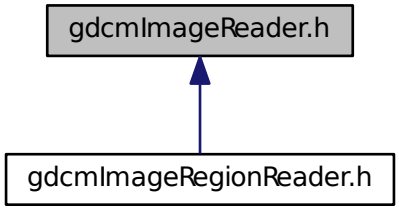
11.113 gdcmImageReader.h File Reference

```
#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.

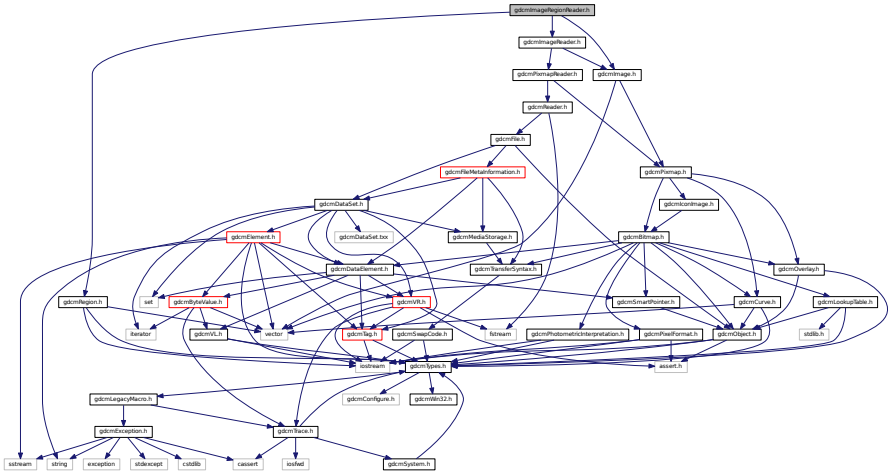
Namespaces

- 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.

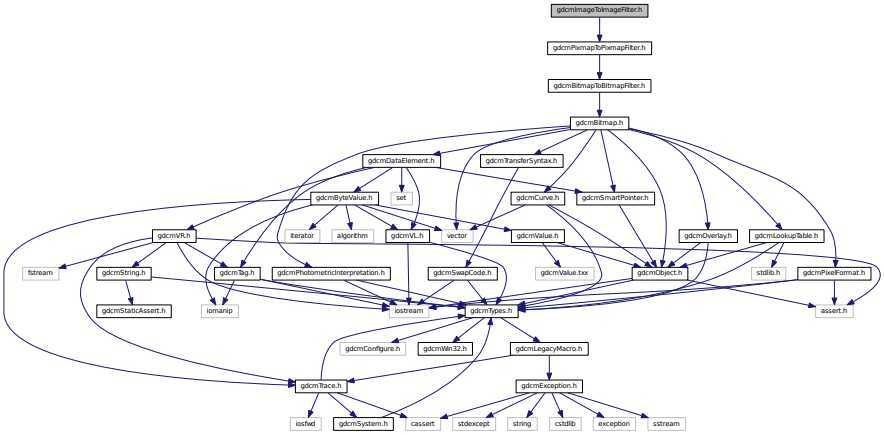
Namespaces

- gdcm

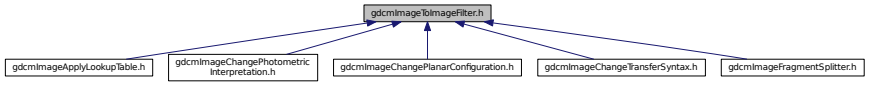
11.115 gdcmImageToImageFilter.h File Reference

```
#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 Super class for all filter taking an image and producing an output image.

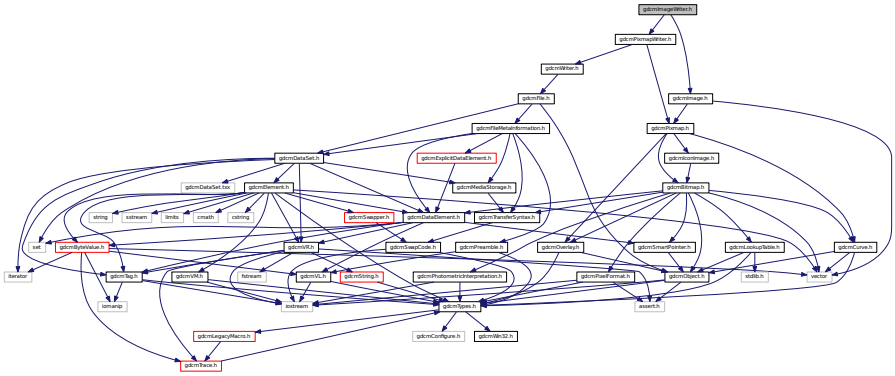
Namespaces

- gdcm

11.116 gdcmImageWriter.h File Reference

```
#include "gdcmPixmapWriter.h"
#include "gdcmImage.h"
```

Include dependency graph for gdcmImageWriter.h:



Classes

- class gdcm::ImageWriter

  ImageWriter.

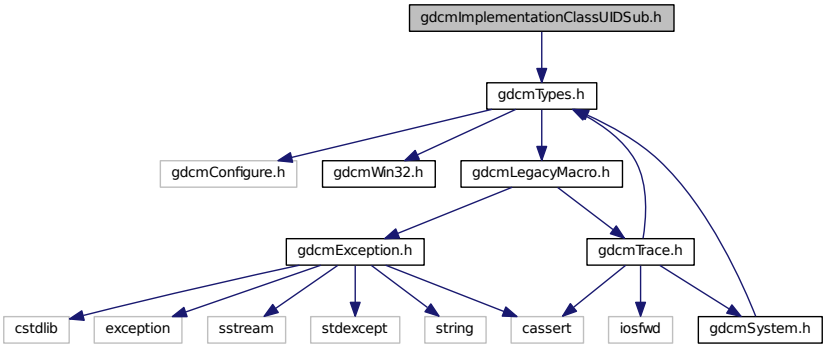
Namespaces

- gdcm

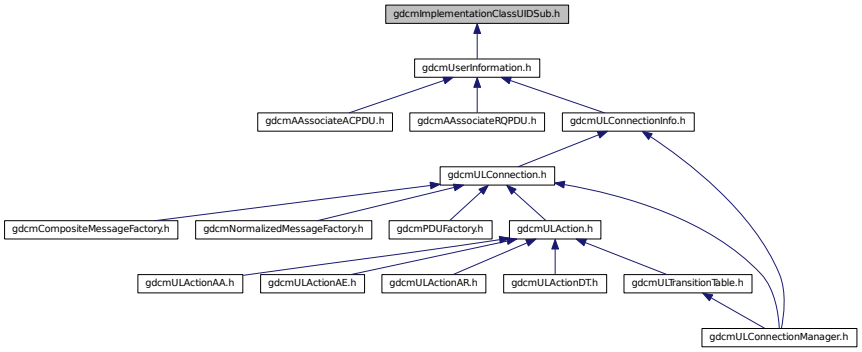
11.117 gdcmImplementationClassUIDSub.h File Reference

```
#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

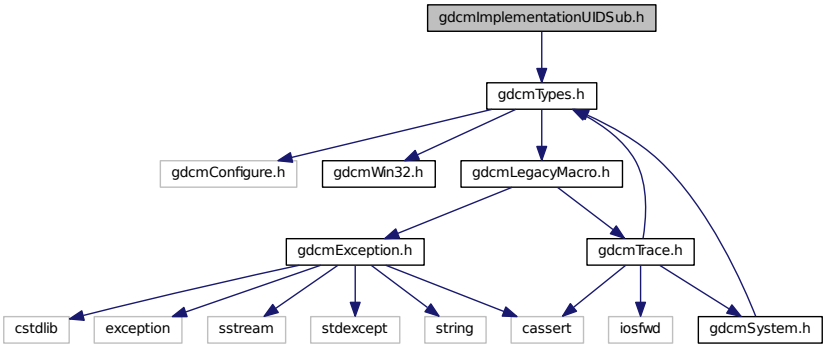
  ImplementationClassUIDSub PS 3.7 Table D.3-1 IMPLEMENTATION CLASS UID SUB-ITEM FIELDS (A-ASSOCIAT↩ E-RQ)

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 Table D.3-2 IMPLEMENTATION UID SUB-ITEM FIELDS (A-ASSOCIATE-AC)

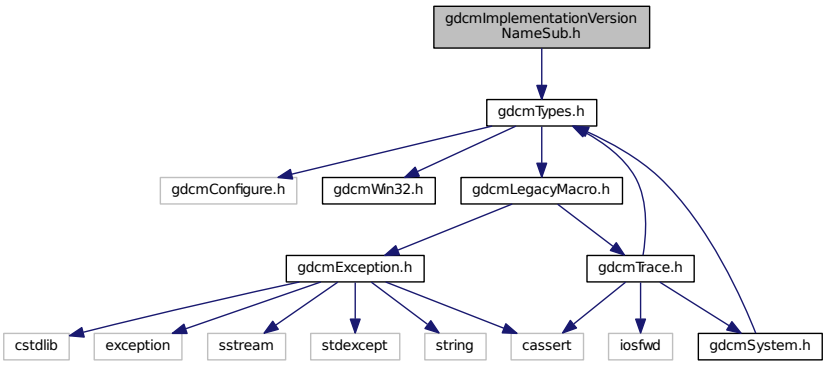
Namespaces

- gdcm
- gdcm::network

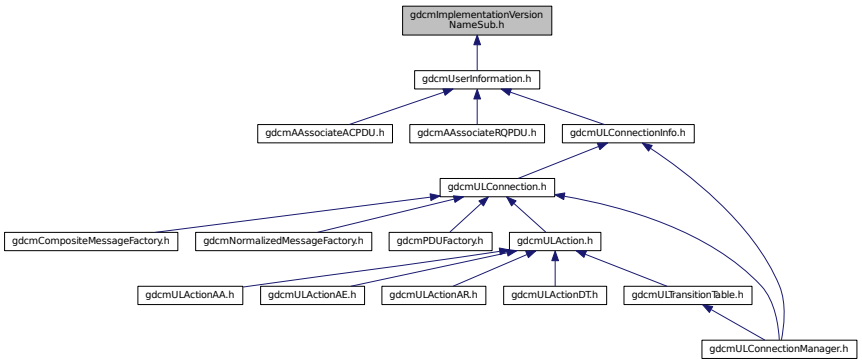
11.119 gdcmImplementationVersionNameSub.h File Reference

```
#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 Table D.3-3 IMPLEMENTATION VERSION NAME SUB-ITEM FIELDS (A-ASSOCIATE-RQ)

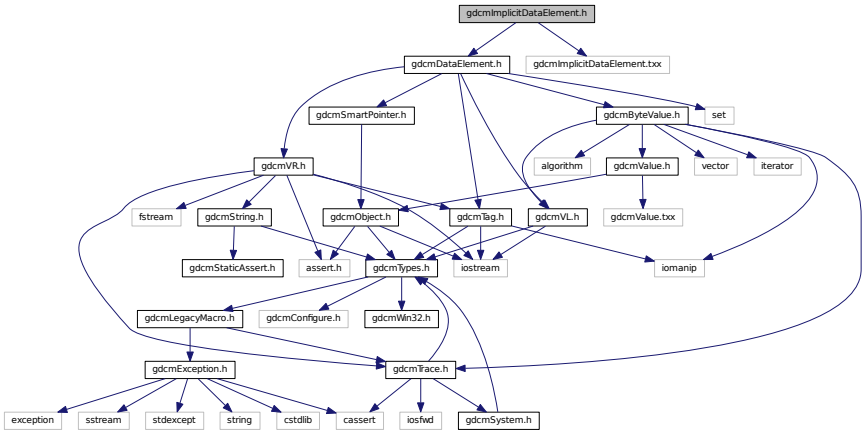
Namespaces

- 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.

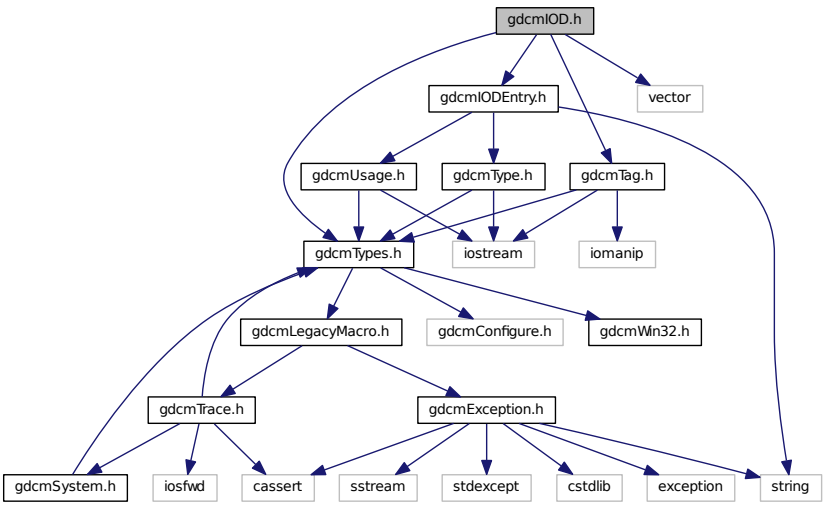
Namespaces

- gdcm

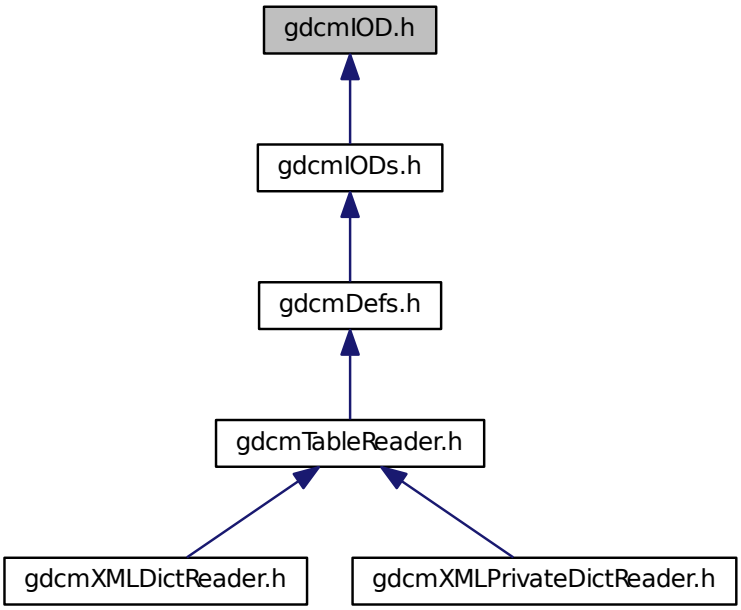
11.121 gdcmIOD.h File Reference

```
#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

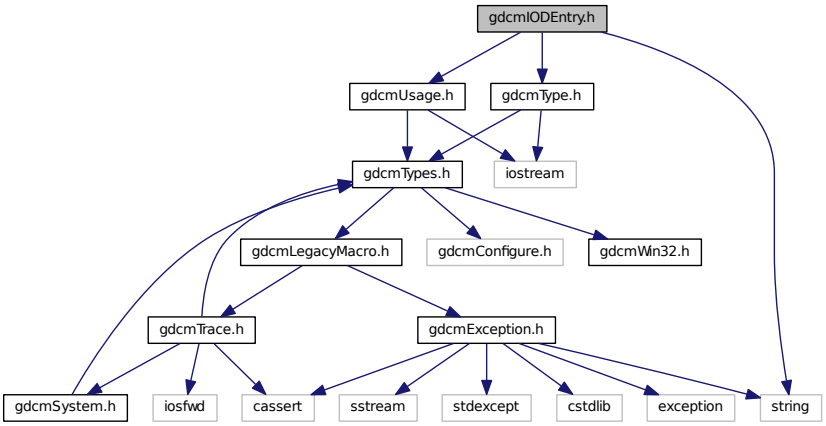
Functions

- std::ostream & gdcm::operator<< (std::ostream &_os, const IOD &_val)

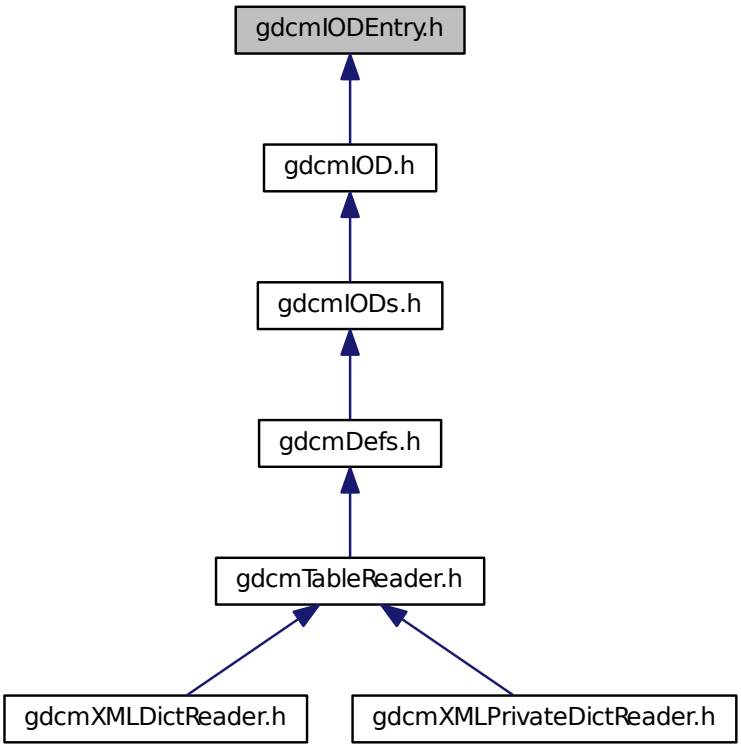
11.122 gdcmIODEntry.h File Reference

```
#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

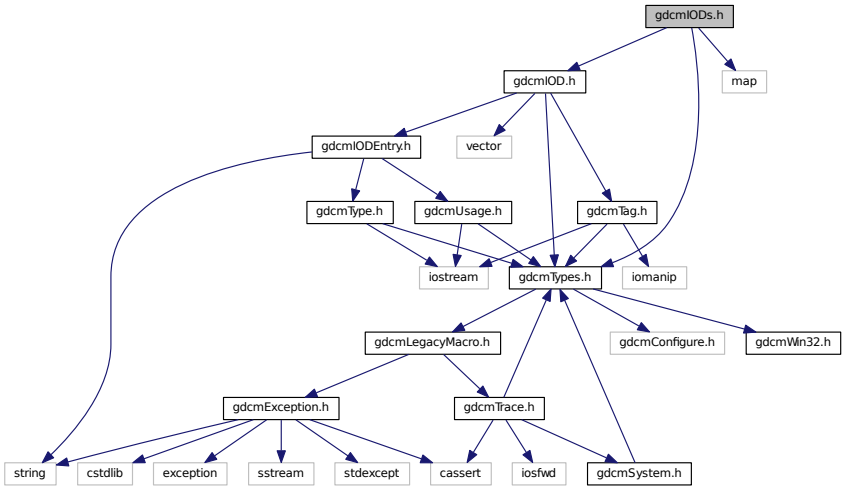
Functions

- std::ostream & gdcm::operator<< (std::ostream &_os, const IODEntry &_val)

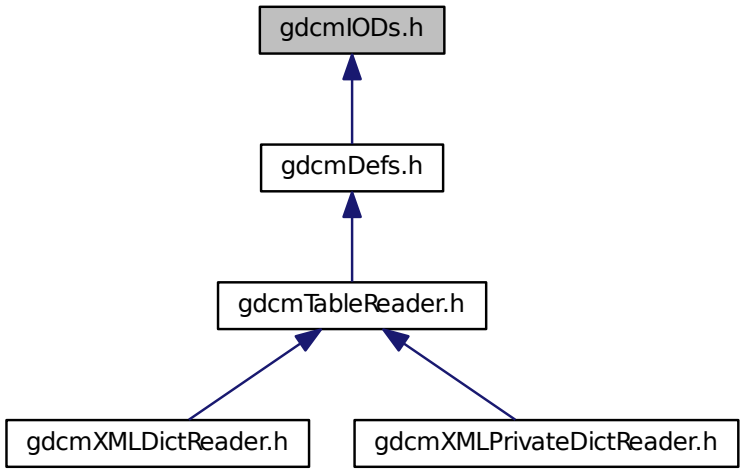
11.123 gdcmIODs.h File Reference

```
#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

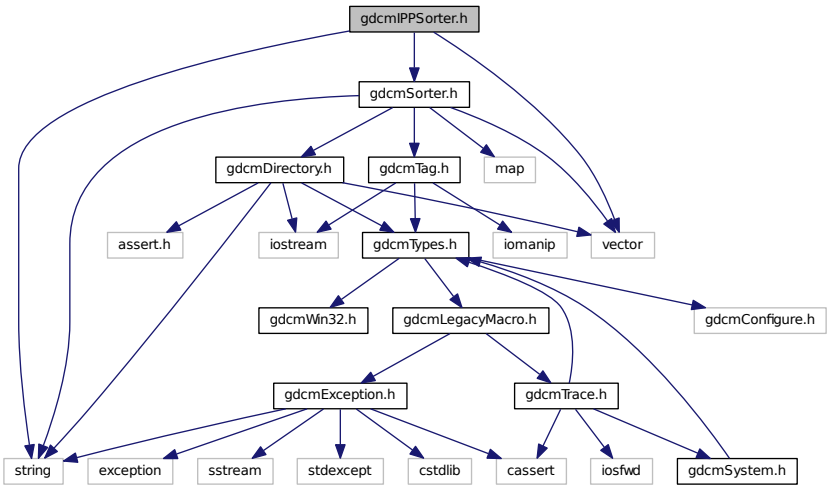
Functions

- 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 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.

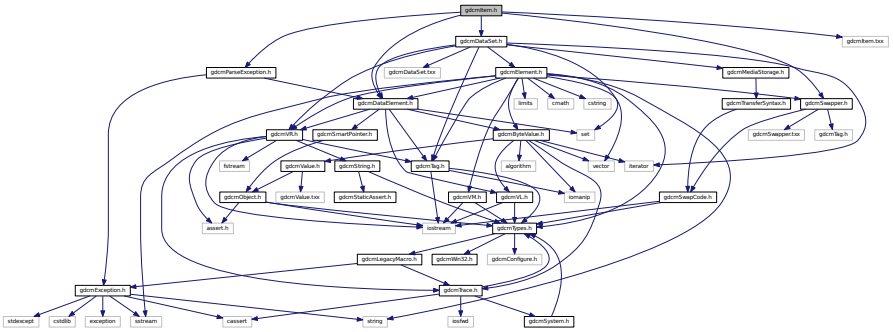
Namespaces

- gdcm

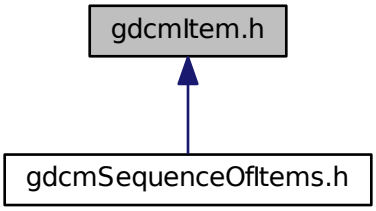
11.125 gdcmItem.h File Reference

```
#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 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.

Namespaces

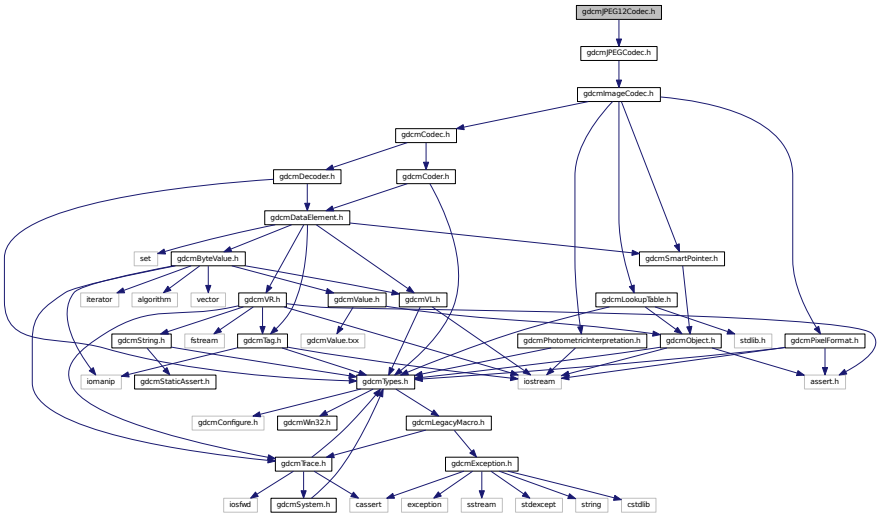
- gdcm

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)

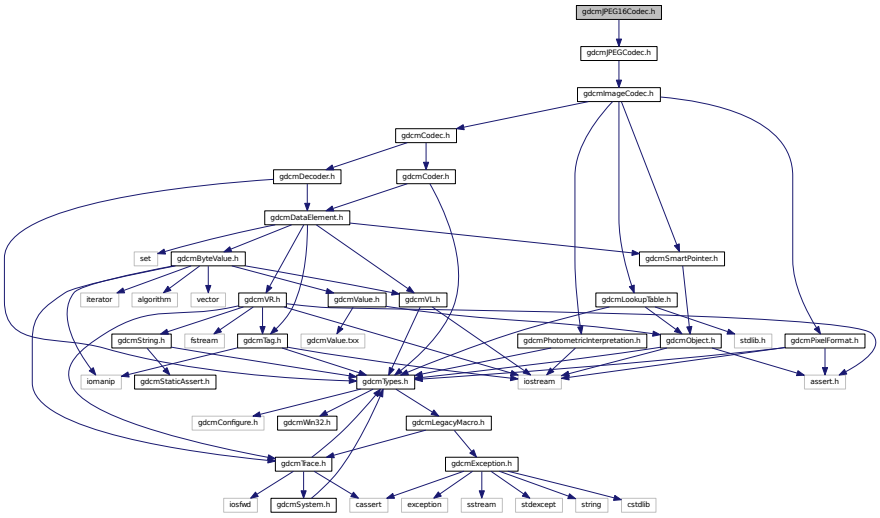
Namespaces

- 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)

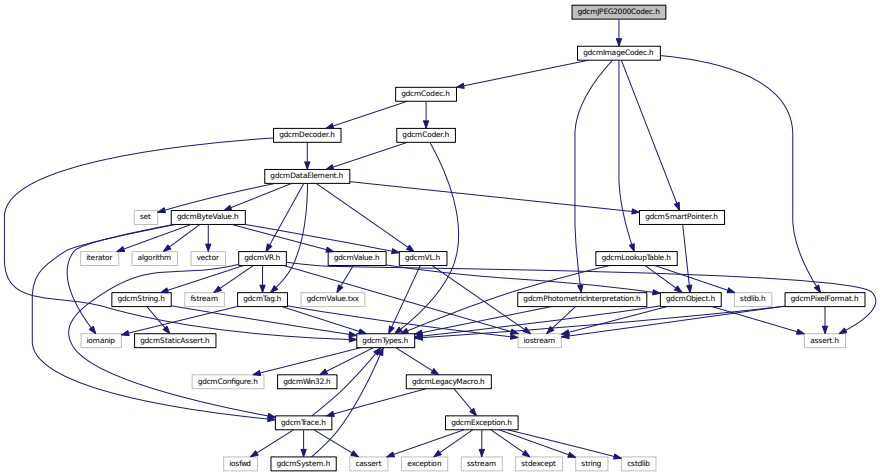
Namespaces

- 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.

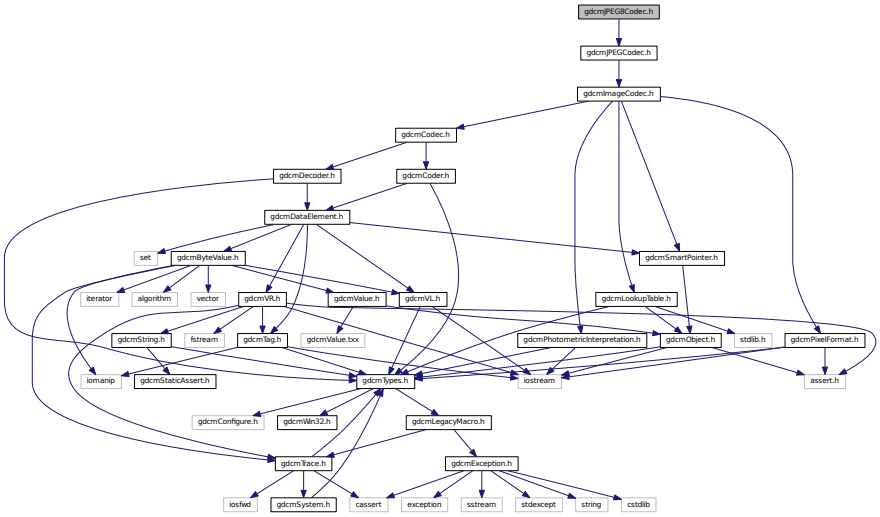
Namespaces

- 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)

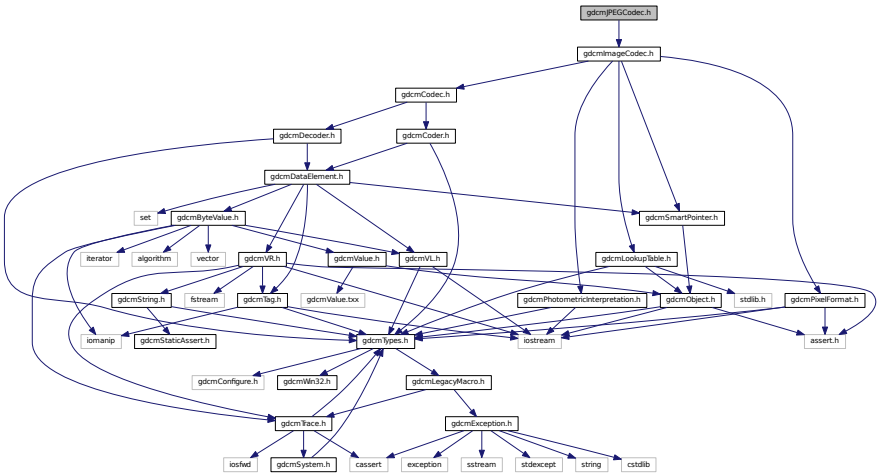
Namespaces

- gdcm

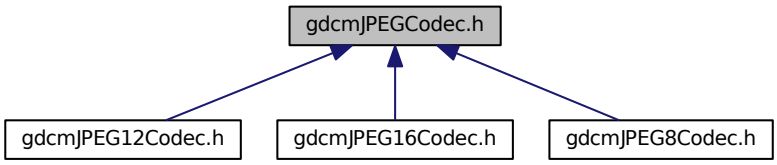
11.130 gdcmJPEGCodec.h File Reference

```
#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 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.

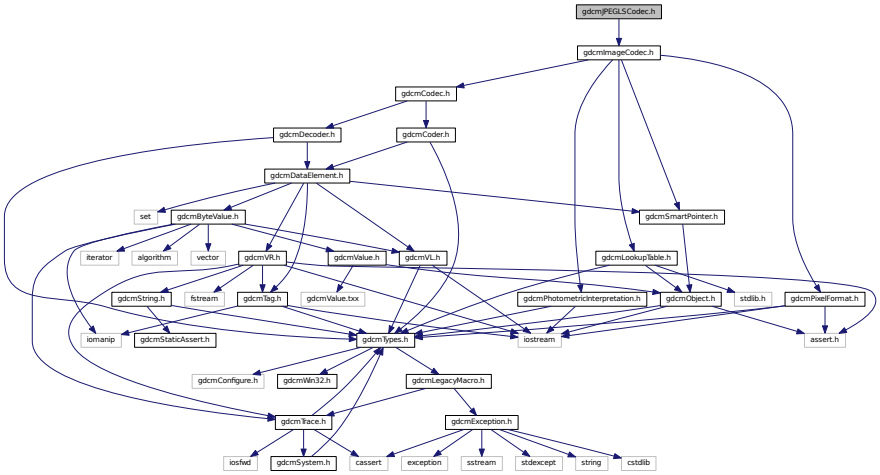
Namespaces

- gdcm

11.131 gdcmJPEGLSCodec.h File Reference

```
#include "gdcmImageCodec.h"
```

Include dependency graph for gdcmJPEGLSCodec.h:



Classes

- class gdcm::JPEGLSCodec

  JPEG-LS.

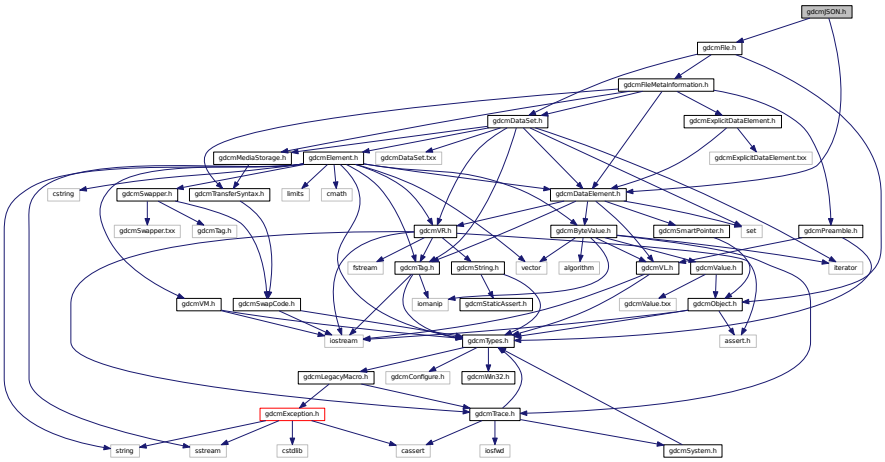
Namespaces

- gdcm

11.132 gdcmJSON.h File Reference

```
#include "gdcmFile.h"
#include "gdcmDataElement.h"
```

Include dependency graph for gdcmJSON.h:



Classes

- class gdcm::JSON

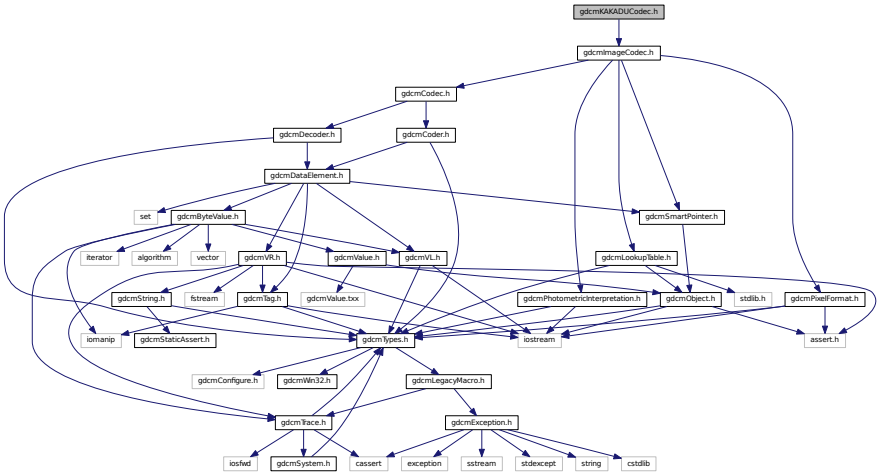
Namespaces

- gdcm

11.133 gdcmKAKADUCodec.h File Reference

```
#include "gdcmImageCodec.h"
```

Include dependency graph for gdcmKAKADUCodec.h:



Classes

- class gdcm::KAKADUCodec

  KAKADUCodec.

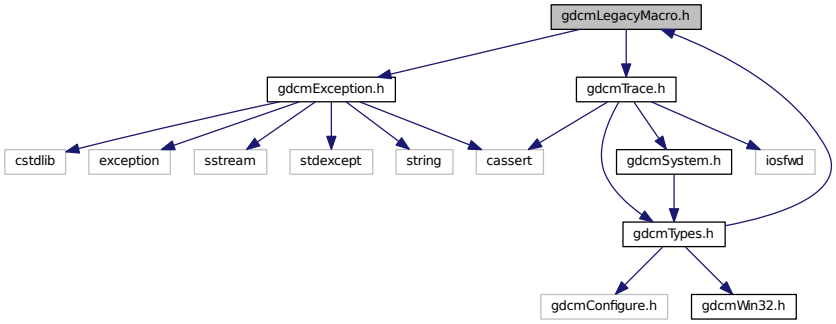
Namespaces

- gdcm

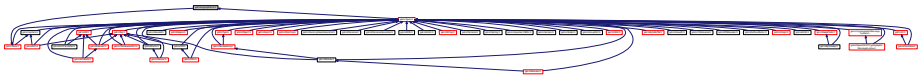
11.134 gdcmLegacyMacro.h File Reference

```
#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 #define GDCM_LEGACY(*method*) method;

Referenced by gdcm::StringFilter::UseDictAlways().

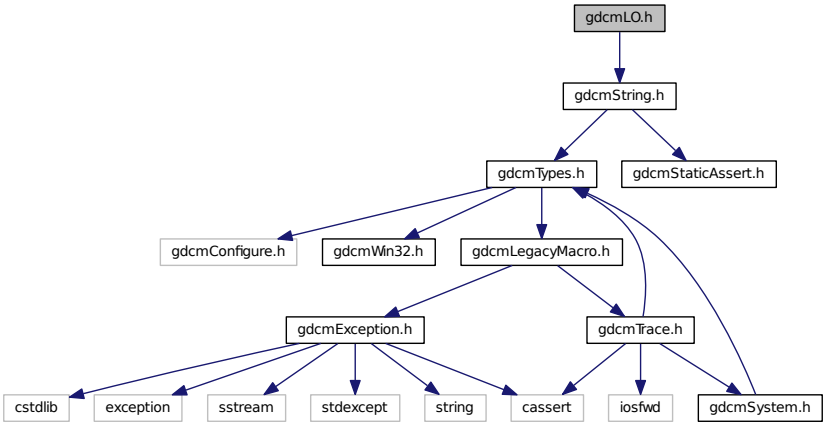
11.134.1.2 #define GDCM_LEGACY_BODY(*method, version*) gdcmWarningMacro(#method " was deprecated for " version " and will be removed in a future version.")

11.134.1.3 #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.

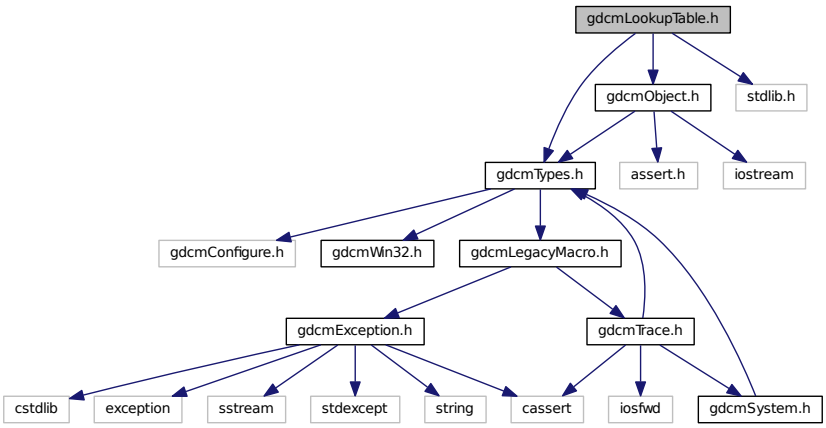
Namespaces

- gdcm

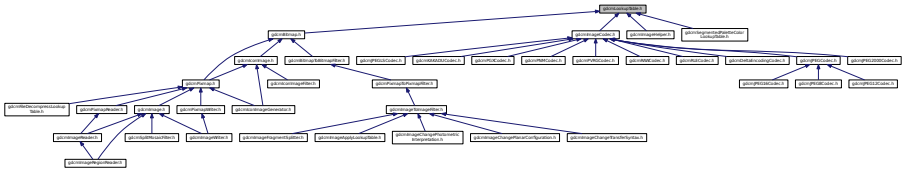
11.136 gdcmLookupTable.h File Reference

```
#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.

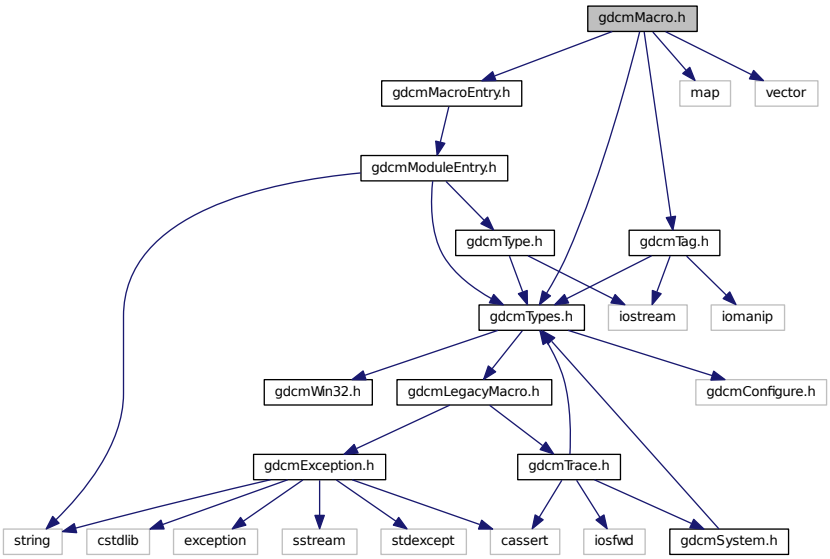
Namespaces

- gdcm

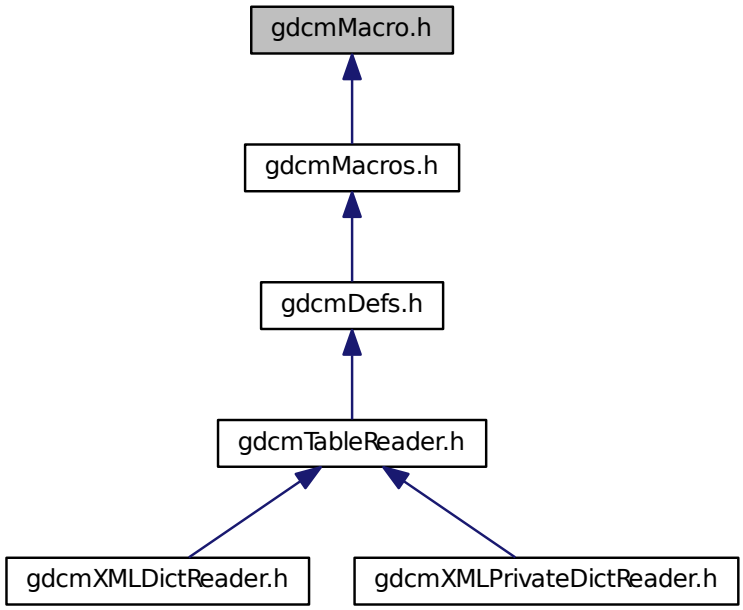
11.137 gdcmMacro.h File Reference

```
#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

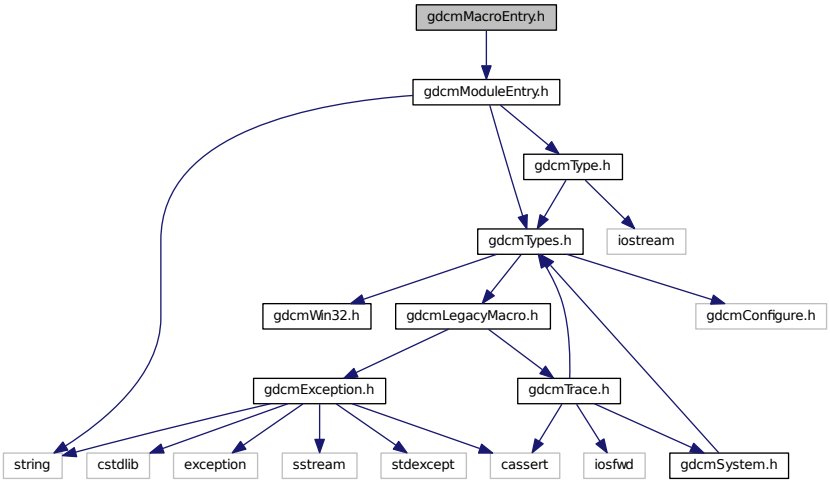
Functions

- std::ostream & gdcm::operator<< (std::ostream &_os, const Macro &_val)

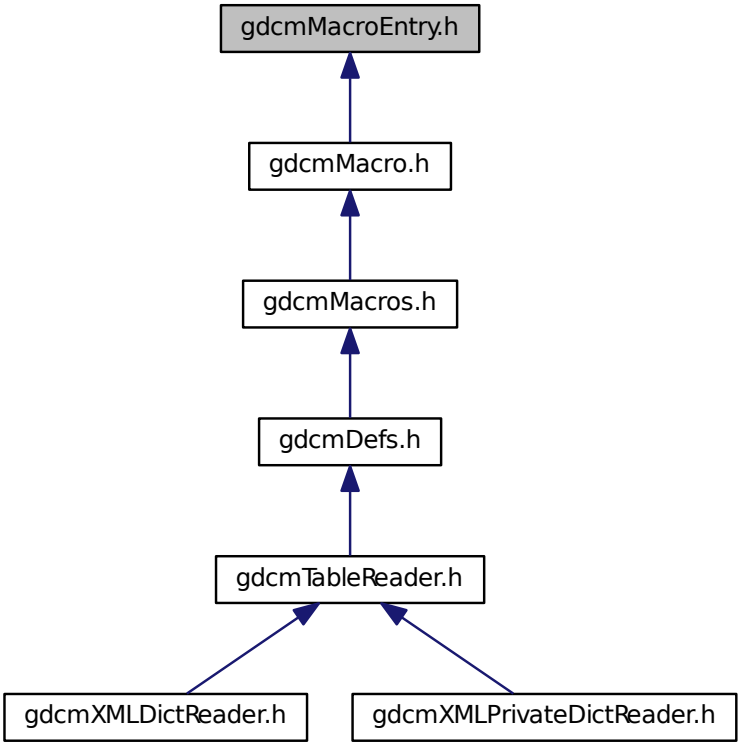
11.138 gdcmMacroEntry.h File Reference

```
#include "gdcmModuleEntry.h"
```

Include dependency graph for gdcmMacroEntry.h:



This graph shows which files directly or indirectly include this file:



Macros

- #define GDCMMACROENTRY_H

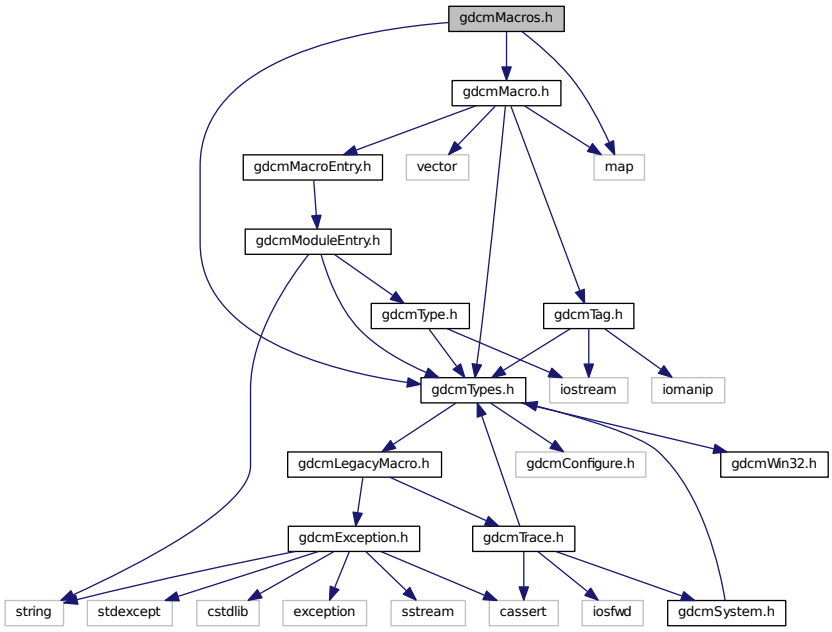
11.138.1 Macro Definition Documentation

11.138.1.1 #define GDCMMACROENTRY_H

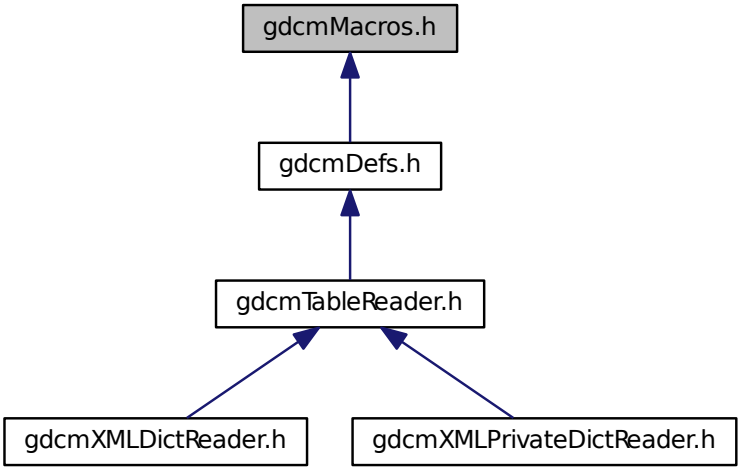
11.139 gdcmMacros.h File Reference

```
#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

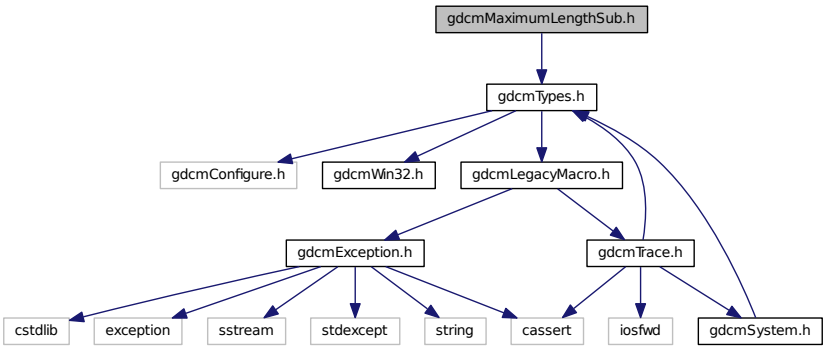
Functions

- std::ostream & gdcm::operator<< (std::ostream &_os, const Macros &_val)

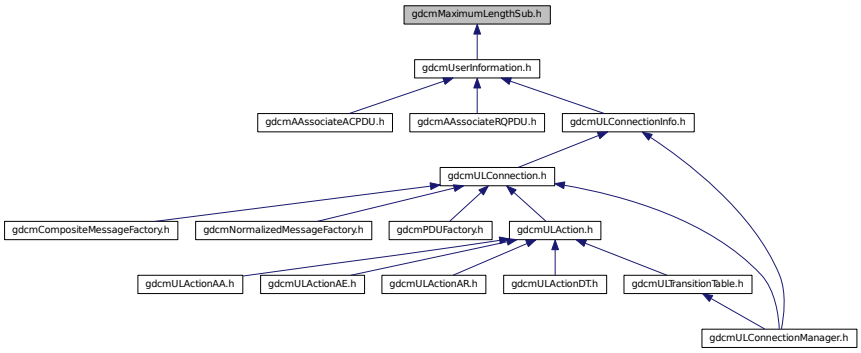
11.140 gdcmMaximumLengthSub.h File Reference

```
#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 Annex D Table D.1-1 MAXIMUM LENGTH SUB-ITEM FIELDS (A-ASSOCIATE-RQ)

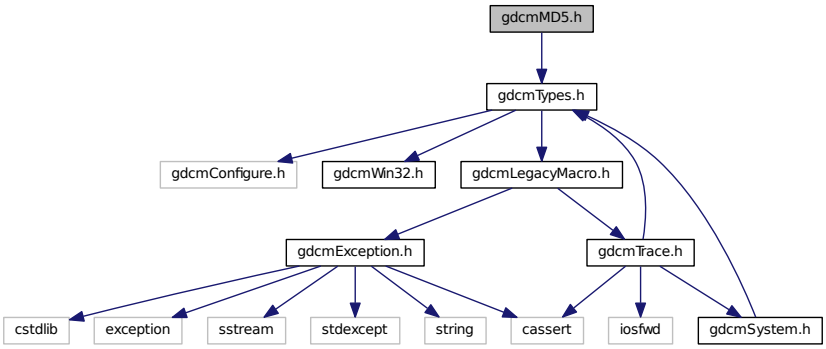
Namespaces

- 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.

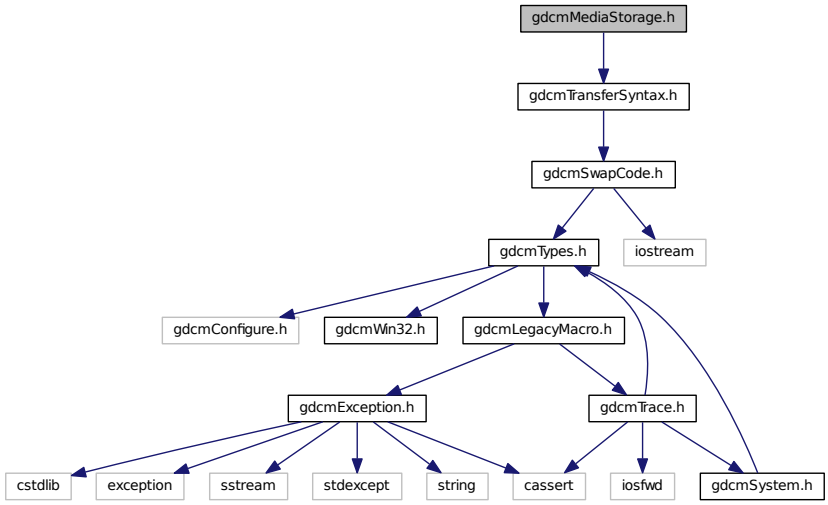
Namespaces

- gdcm

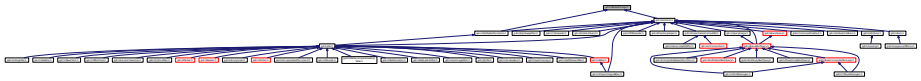
11.142 gdcmMediaStorage.h File Reference

```
#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

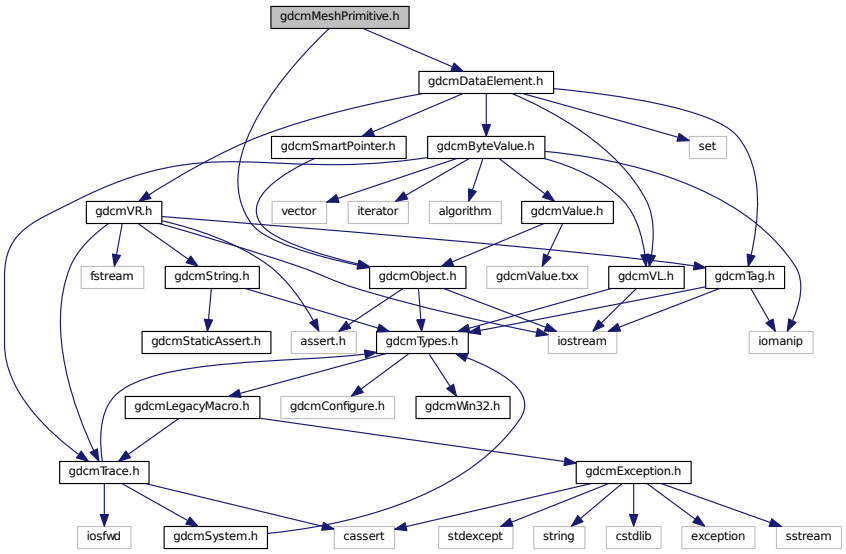
Functions

- std::ostream & gdcm::operator<< (std::ostream &_os, const MediaStorage &ms)

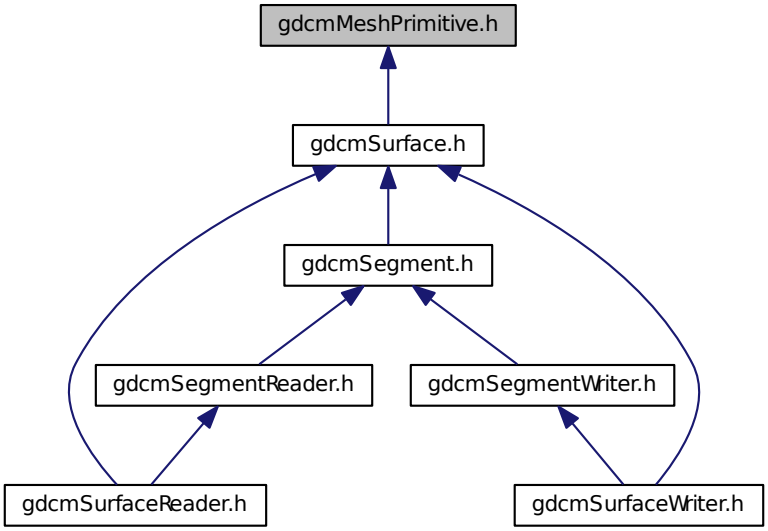
11.143 gdcmMeshPrimitive.h File Reference

```
#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. It is designed from surface mesh primitives macro.

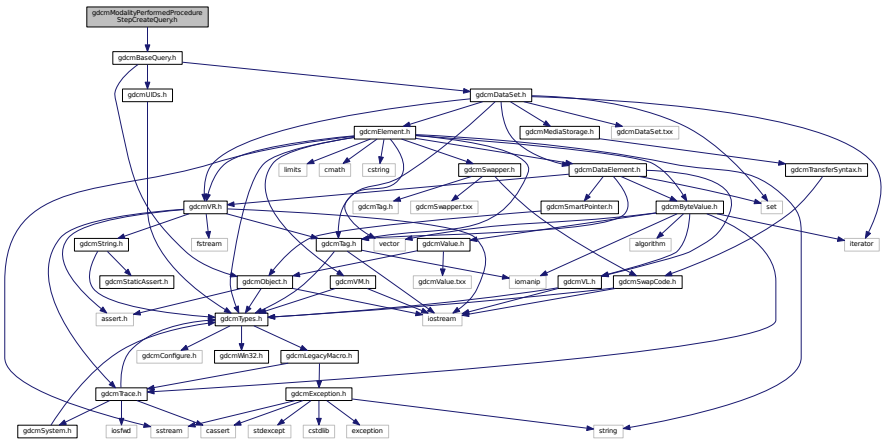
Namespaces

- gdcm

11.144 gdcmModalityPerformedProcedureStepCreateQuery.h File Reference

```
#include "gdcmBaseQuery.h"
```

Include dependency graph for gdcmModalityPerformedProcedureStepCreateQuery.h:



Classes

- class gdcm::ModalityPerformedProcedureStepCreateQuery

  ModalityPerformedProcedureStepCreateQuery contains: the class which will produce a dataset for n-create for Modality Performed Procedure Step sop class.

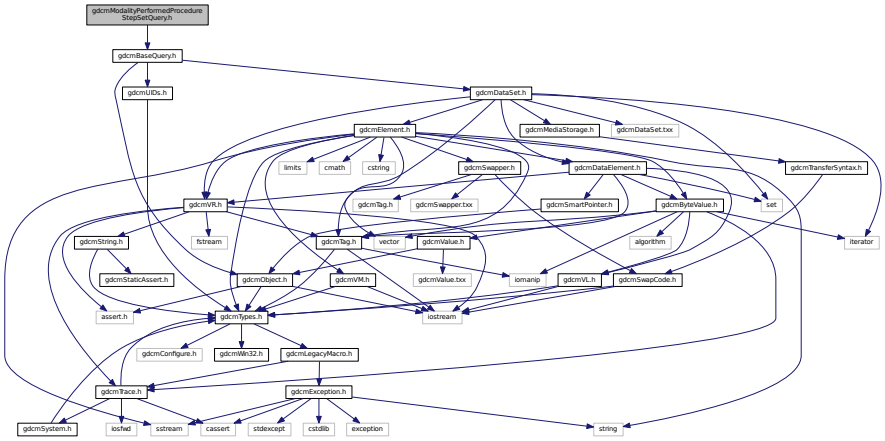
Namespaces

- gdcm

11.145 gdcmModalityPerformedProcedureStepSetQuery.h File Reference

```
#include "gdcmBaseQuery.h"
```

Include dependency graph for gdcmModalityPerformedProcedureStepSetQuery.h:



Classes

- class gdcm::ModalityPerformedProcedureStepSetQuery

  ModalityPerformedProcedureStepSetQuery contains: the class which will produce a dataset for n-set for Modality Performed Procedure Step sop class.

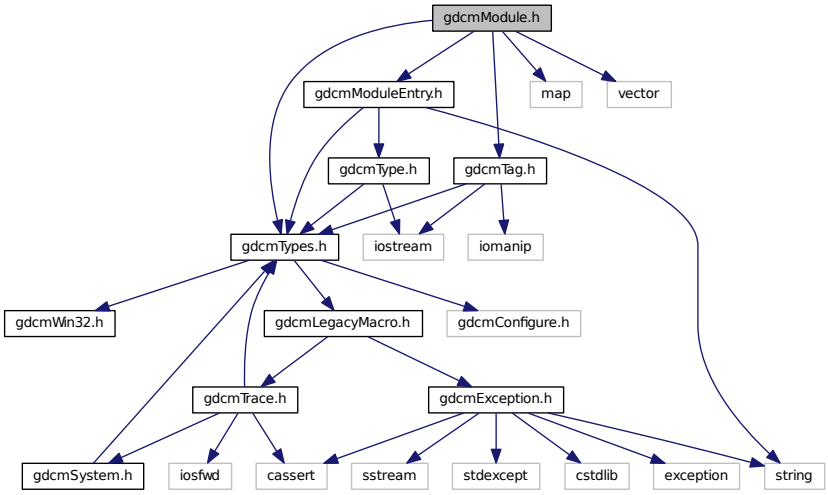
Namespaces

- gdcm

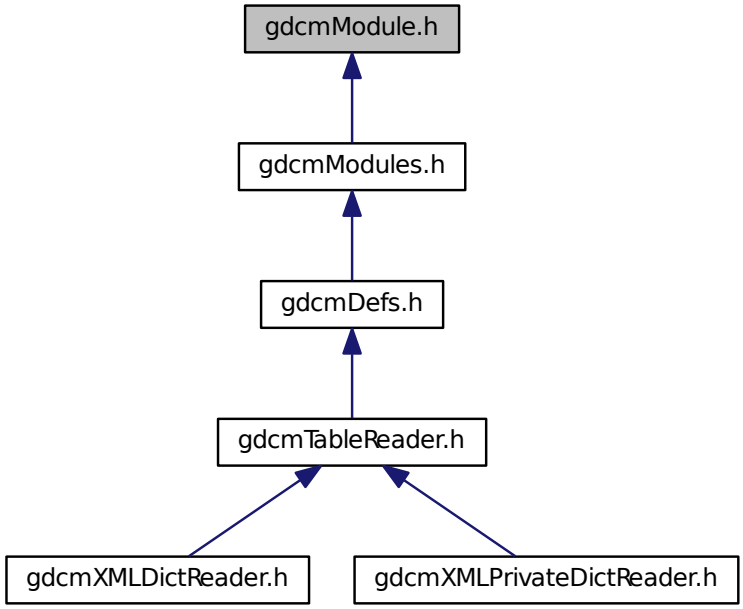
11.146 gdcmModule.h File Reference

```
#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

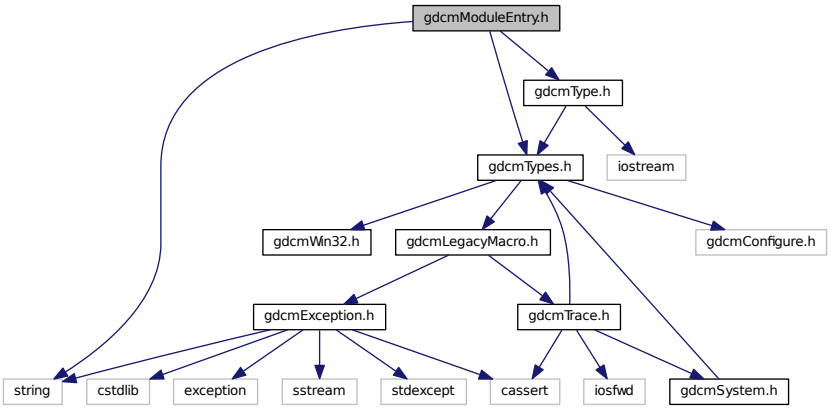
Functions

- std::ostream & gdcm::operator<< (std::ostream &_os, const Module &_val)

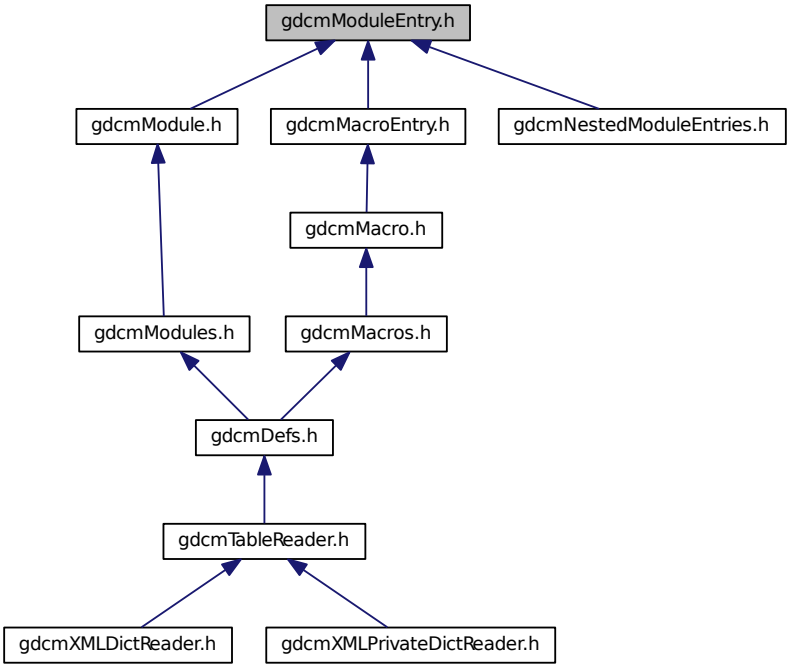
11.147 gdcmModuleEntry.h File Reference

```
#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

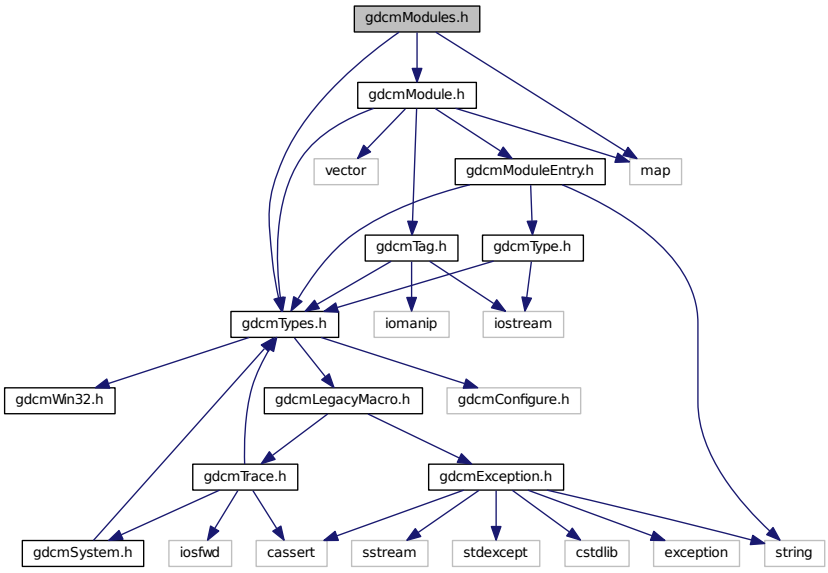
Functions

- std::ostream & gdcm::operator<< (std::ostream &_os, const ModuleEntry &_val)

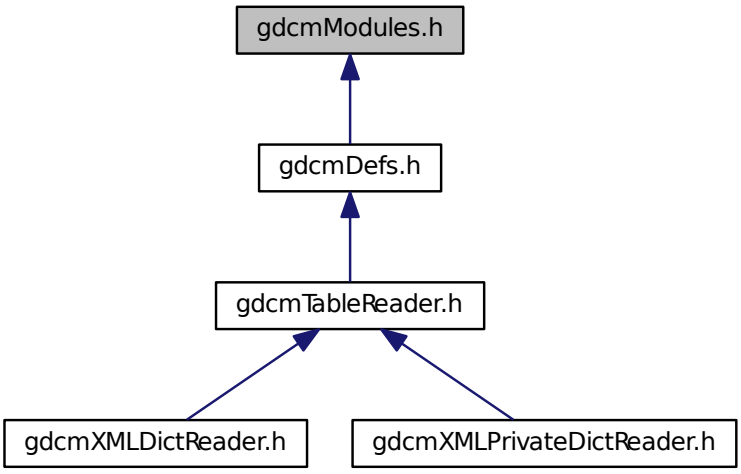
11.148 gdcmModules.h File Reference

```
#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

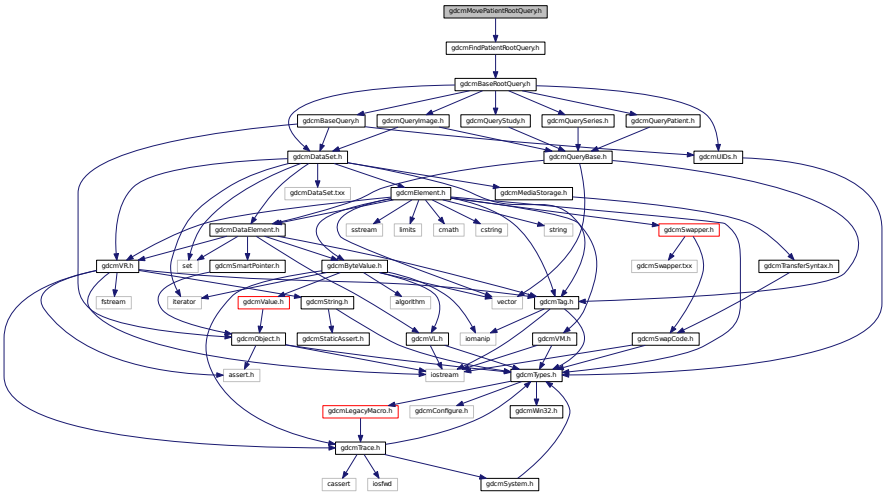
Functions

- 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

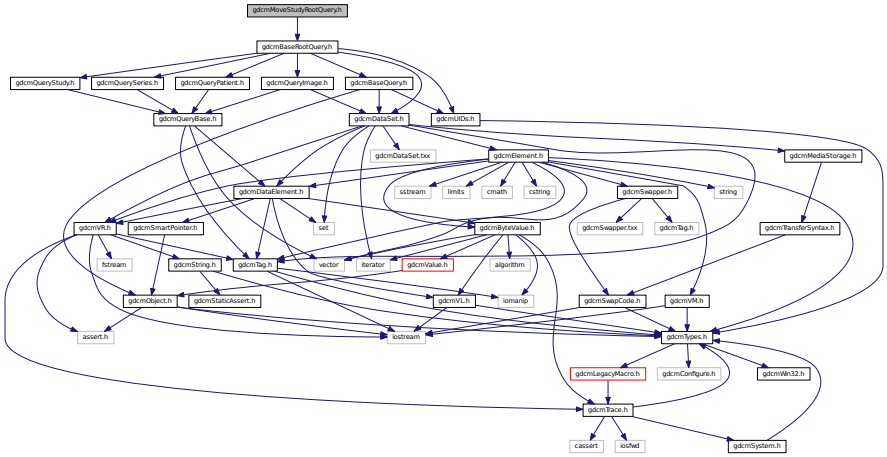
  MovePatientRootQuery contains: the class which will produce a dataset for c-move with patient root.

Namespaces

- gdcm

11.150 gdcmMoveStudyRootQuery.h File Reference

`#include "gdcmBaseRootQuery.h"`
Include dependency graph for gdcmMoveStudyRootQuery.h:



Classes

- class gdcm::MoveStudyRootQuery

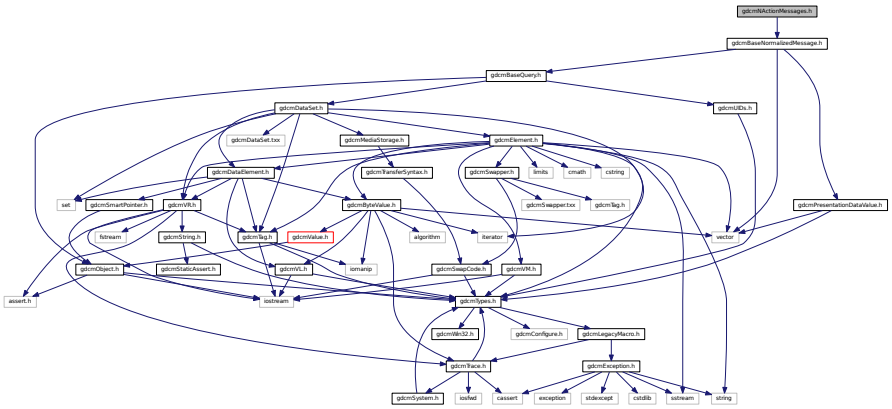
  MoveStudyRootQuery contains: the class which will produce a dataset for C-MOVE with study root.

Namespaces

- gdcm

11.151 gdcmNActionMessages.h File Reference

`#include "gdcmBaseNormalizedMessage.h"`
Include dependency graph for gdcmNActionMessages.h:



Classes

- class gdcm::network::NActionRQ

  NActionRQ this file defines the messages for the NAction action.
- class gdcm::network::NActionRSP

  NActionRSP this file defines the messages for the NAction action.

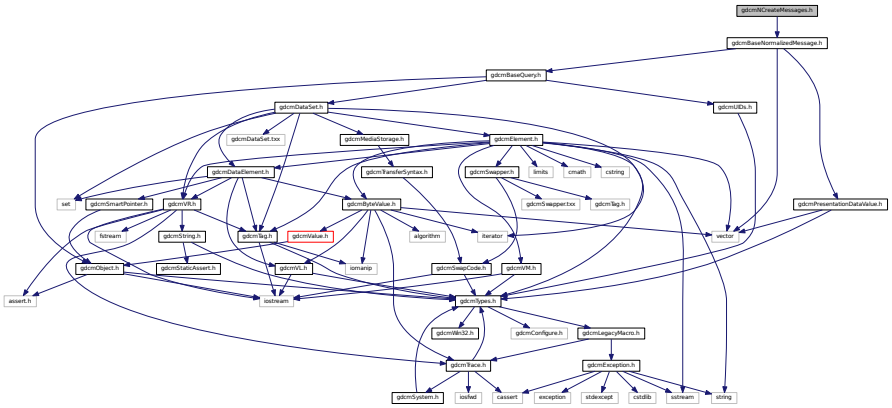
Namespaces

- gdcm
- gdcm::network

11.152 gdcmNCreateMessages.h File Reference

```
#include "gdcmBaseNormalizedMessage.h"
```

Include dependency graph for gdcmNCreateMessages.h:



Classes

- class gdcm::network::NCreateRQ

  NCreateRQ this file defines the messages for the ncreate action.
- class gdcm::network::NCreateRSP

  NCreateRSP this file defines the messages for the ncreate action.

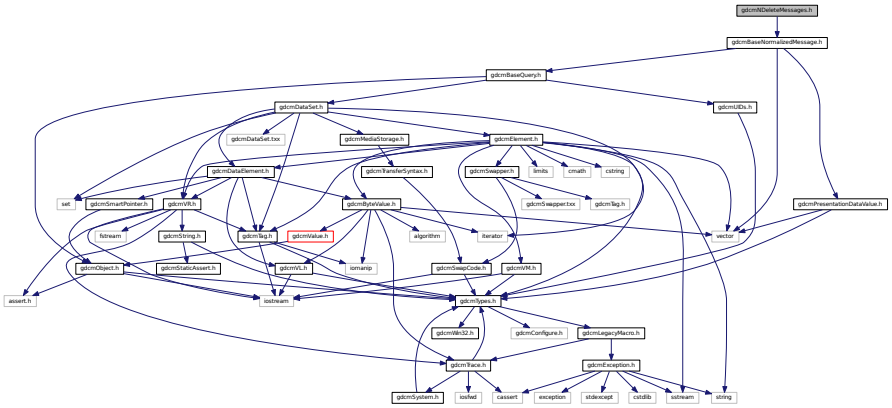
Namespaces

- gdcm
- gdcm::network

11.153 gdcmNDeleteMessages.h File Reference

```
#include "gdcmBaseNormalizedMessage.h"
```

Include dependency graph for gdcmNDeleteMessages.h:



Classes

- class gdcm::network::NDeleteRQ

  NDeleteRQ this file defines the messages for the ndelete action.

- class gdcm::network::NDeleteRSP

  NDeleteRSP this file defines the messages for the ndelete action.

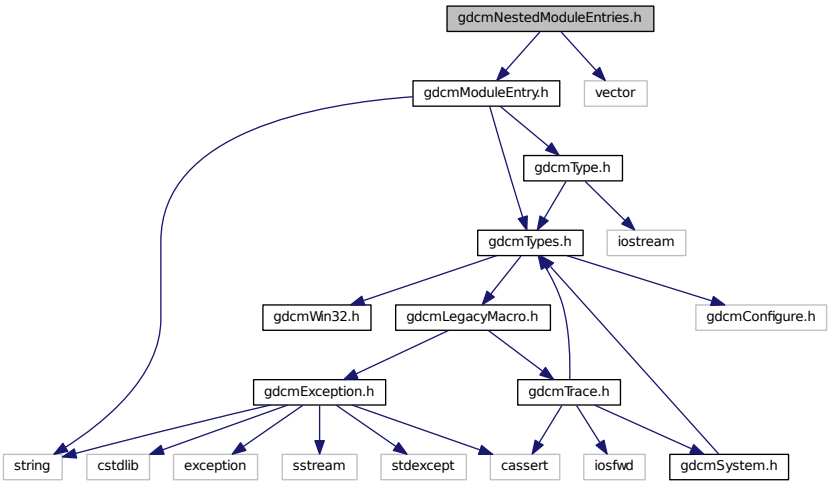
Namespaces

- 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

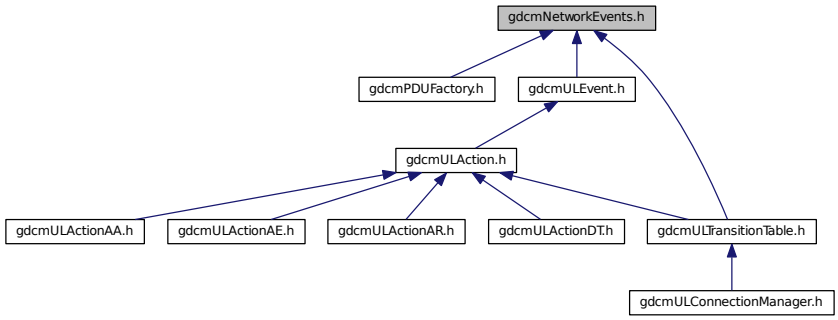
- typedef NestedModuleEntries gdcm::NestedMacroEntries

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

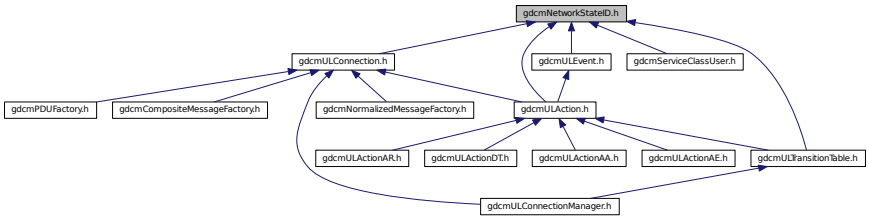
- enum gdcm::network::EEventID {
  gdcm::network::eAASSOCIATERequestLocalUser = 0,
  gdcm::network::eTransportConnConfirmLocal,
  gdcm::network::eASSOCIATE_ACPDUreceived,
  gdcm::network::eASSOCIATE_RJPDUreceived,
  gdcm::network::eTransportConnIndicLocal,
  gdcm::network::eAASSOCIATE_RQPDUreceived,
  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)

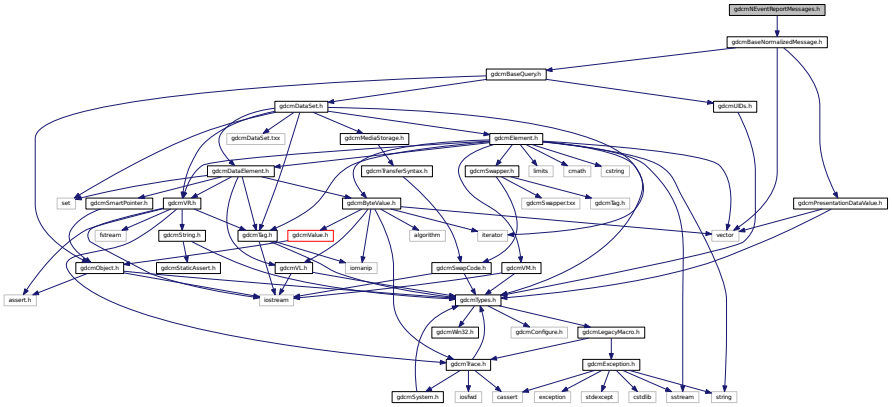
Variables

- 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 this file defines the messages for the neventreport action.

- class gdcm::network::NEventReportRSP

  NEventReportRSP this file defines the messages for the neventreport action.

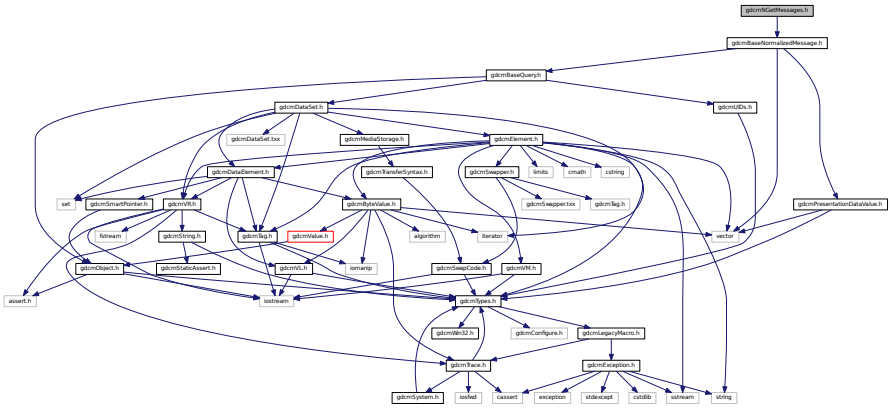
Namespaces

- gdcm
- gdcm::network

11.158 gdcmNGetMessages.h File Reference

```
#include "gdcmBaseNormalizedMessage.h"
```

Include dependency graph for gdcmNGetMessages.h:



Classes

- class gdcm::network::NGetRQ

  NGetRQ this file defines the messages for the nget action.
- class gdcm::network::NGetRSP

  NGetRSP this file defines the messages for the nget action.

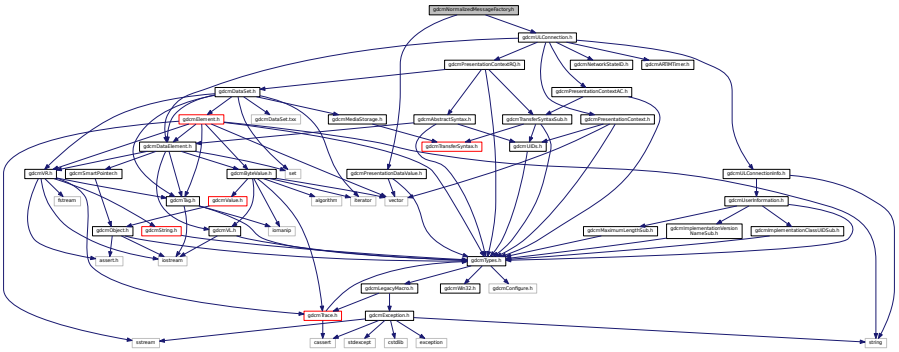
Namespaces

- 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

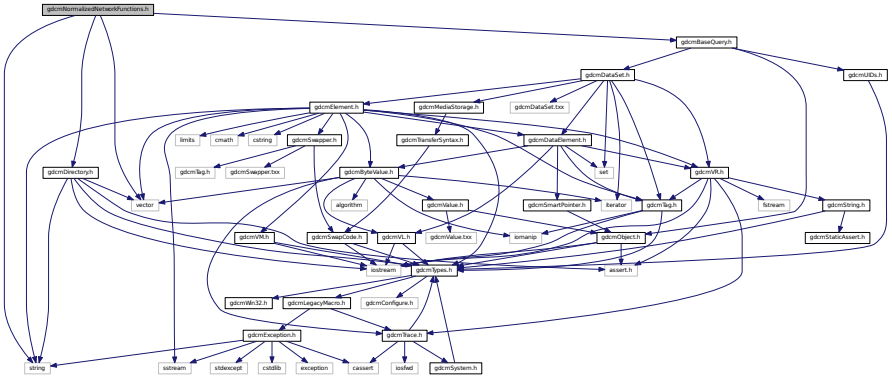
Namespaces

- 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 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:

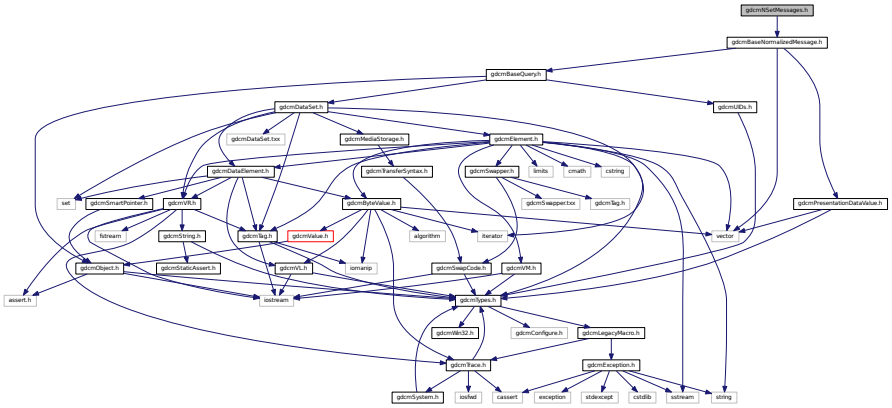
Namespaces

- gdcm

11.161 gdcmNSetMessages.h File Reference

```
#include "gdcmBaseNormalizedMessage.h"
```

Include dependency graph for gdcmNSetMessages.h:



Classes

- class gdcm::network::NSetRQ

  NSetRQ this file defines the messages for the nset action.

- class gdcm::network::NSetRSP

  NSetRSP this file defines the messages for the nset action.

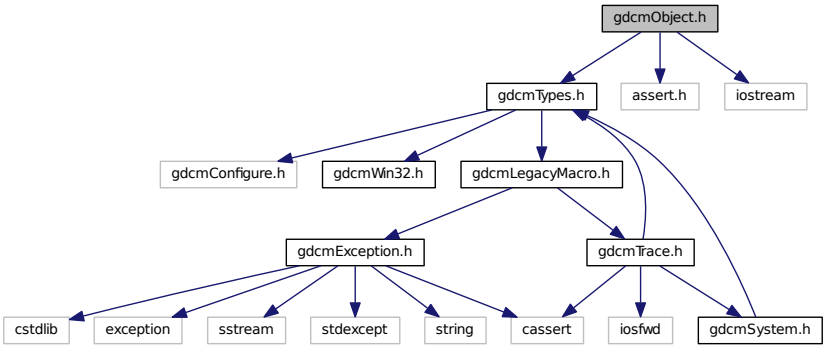
Namespaces

- 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

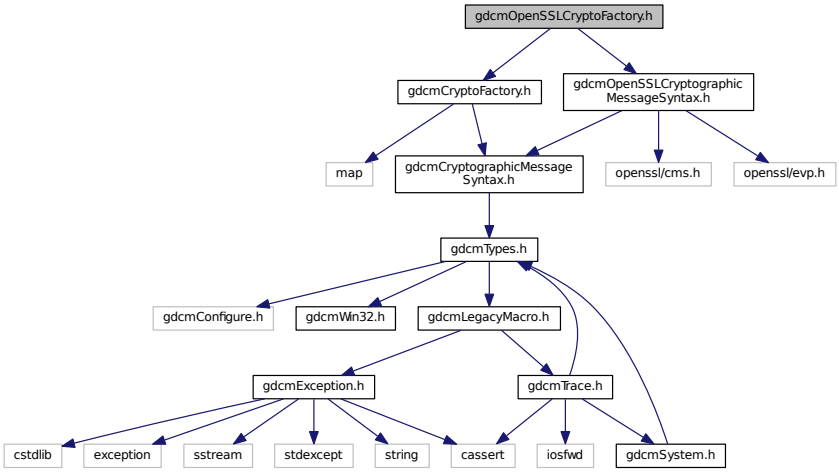
Functions

- 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

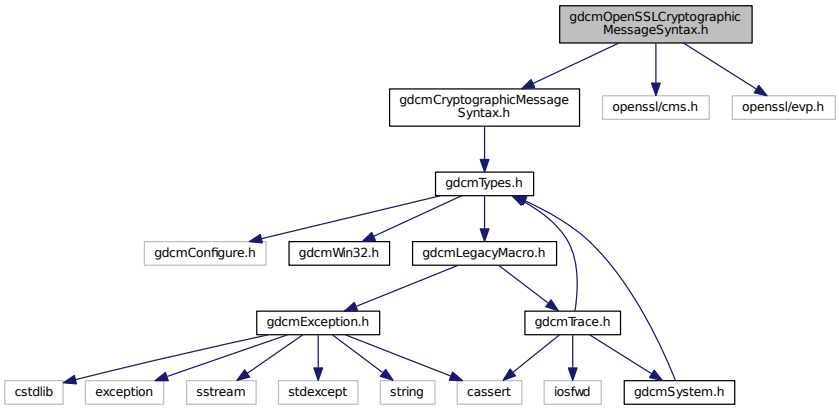
Namespaces

- gdcm

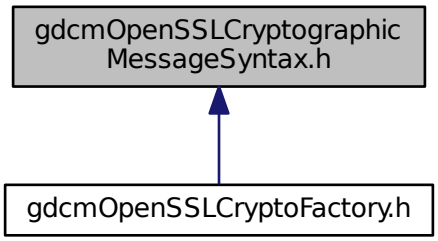
11.164 gdcmOpenSSLCryptographicMessageSyntax.h File Reference

```
#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

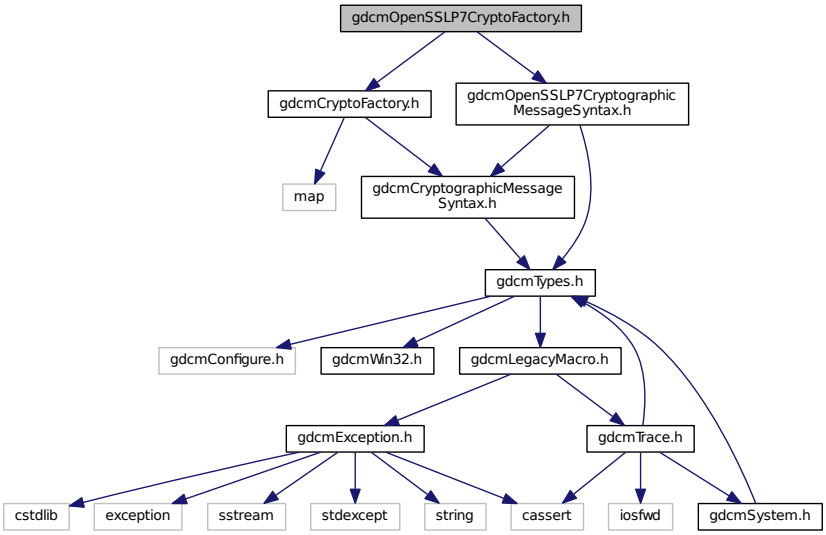
Namespaces

- gdcm

11.165 gdcmOpenSSLP7CryptoFactory.h File Reference

```
#include "gdcmCryptoFactory.h"
#include "gdcmOpenSSLP7CryptographicMessageSyntax.h"
```

Include dependency graph for gdcmOpenSSLP7CryptoFactory.h:



Classes

- class gdcm::OpenSSLP7CryptoFactory

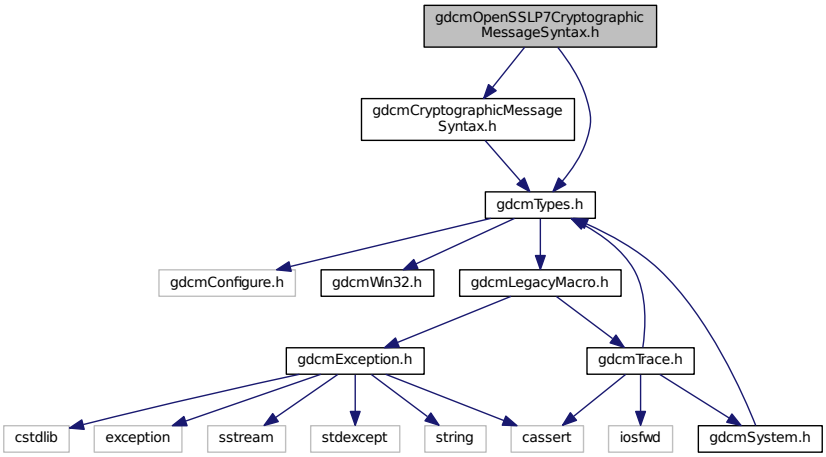
Namespaces

- gdcm

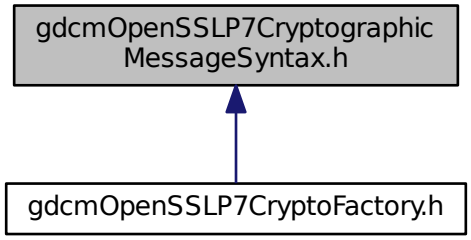
11.166 gdcmOpenSSLP7CryptographicMessageSyntax.h File Reference

```
#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

  Class for CryptographicMessageSyntax encryption. This is just a simple wrapper around openssl PKCS7_encrypt functionalities.

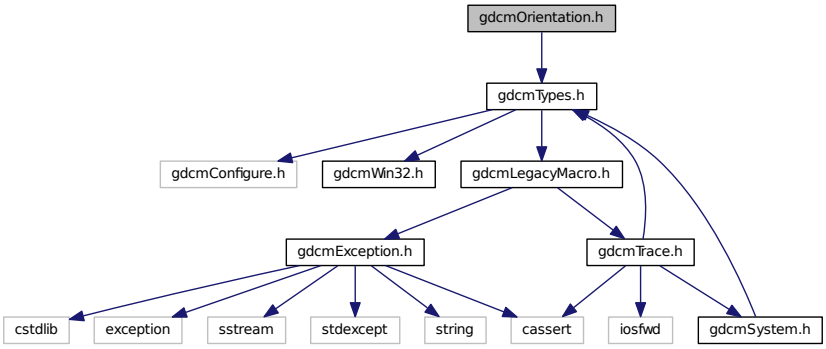
Namespaces

- 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

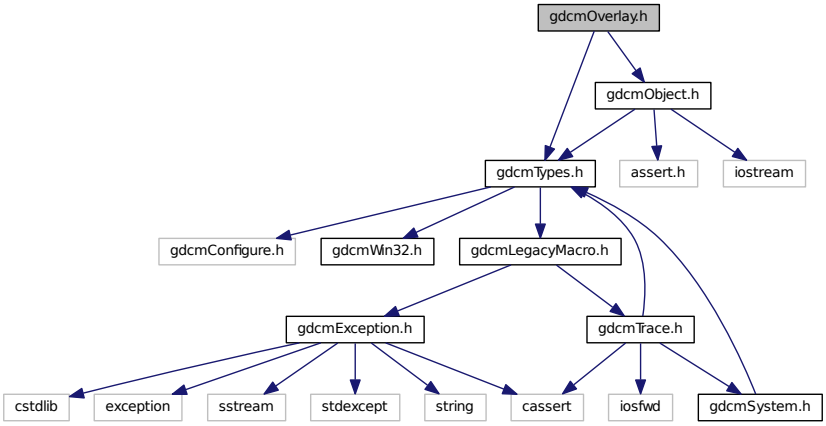
Functions

- std::ostream & gdcm::operator<< (std::ostream &os, const Orientation &o)

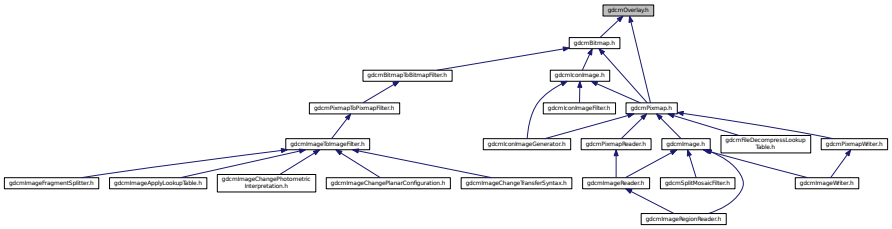
11.168 gdcmOverlay.h File Reference

```
#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.

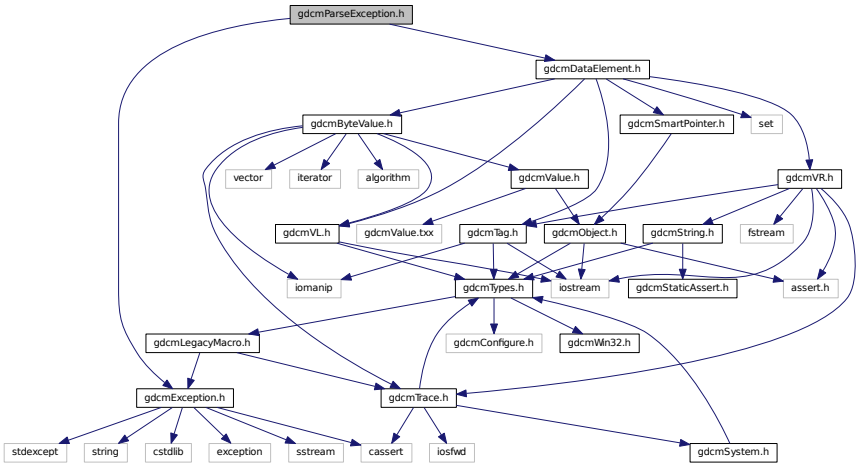
Namespaces

- gdcm

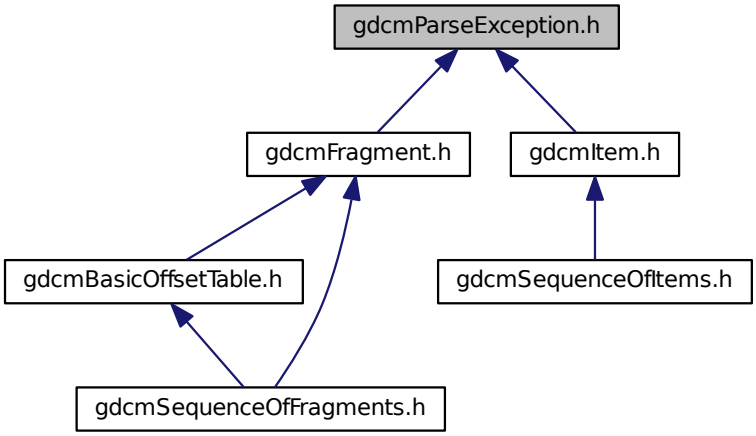
11.169 gdcmParseException.h File Reference

```
#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.

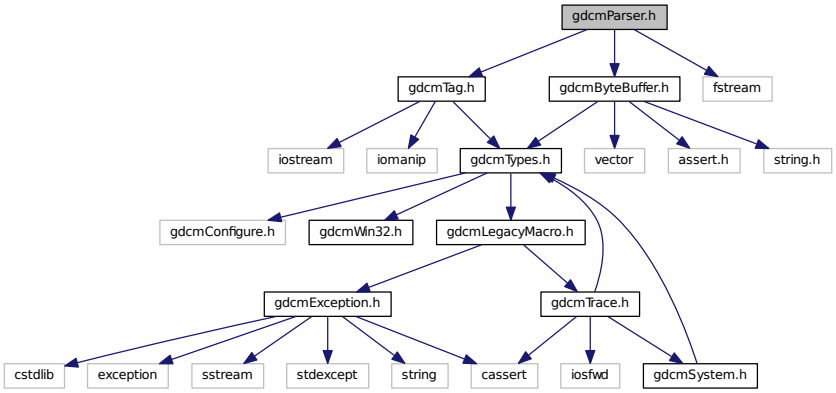
Namespaces

- 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)

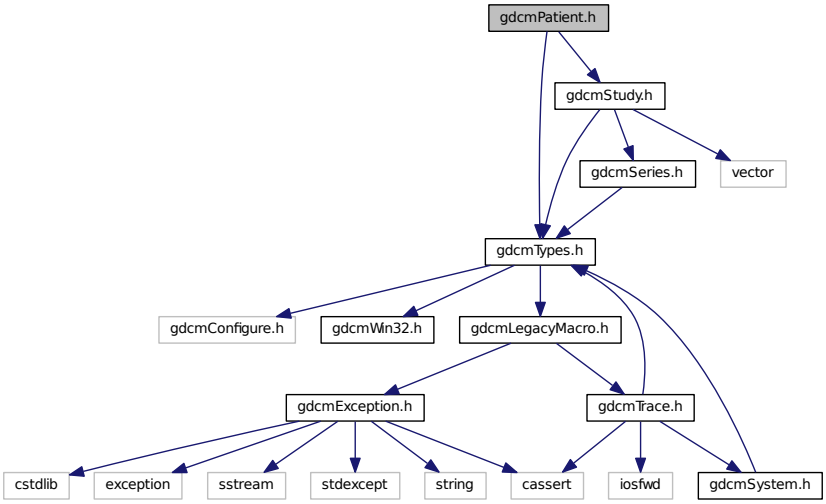
Namespaces

- 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.

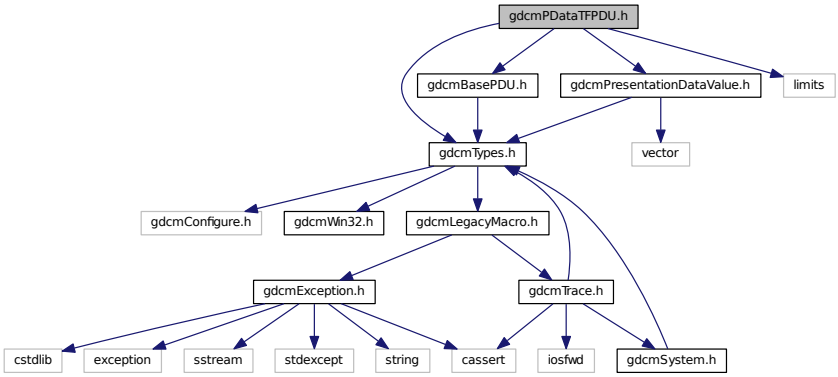
Namespaces

- 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 Table 9-22 P-DATA-TF PDU FIELDS.

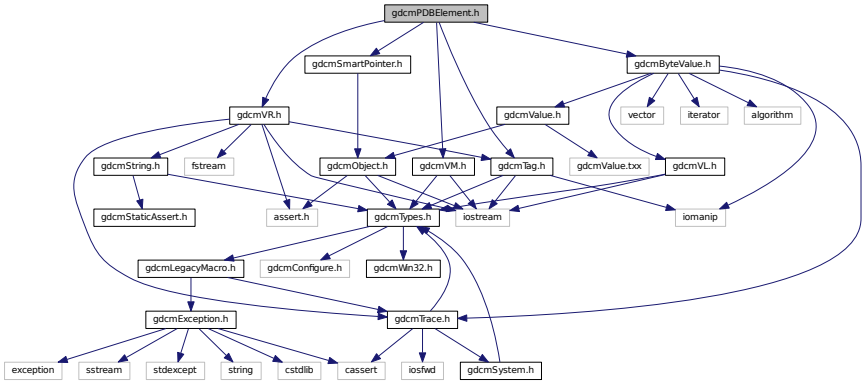
Namespaces

- gdcm
- gdcm::network

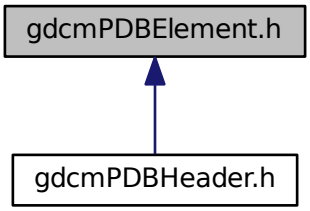
11.173 gdcmPDBElement.h File Reference

```
#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

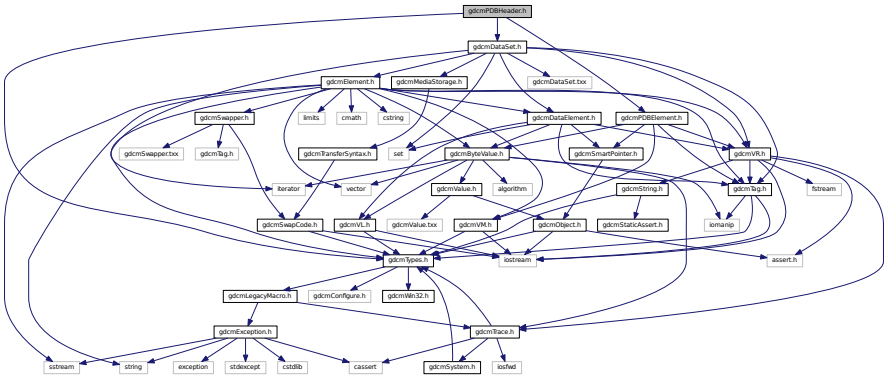
Functions

- 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

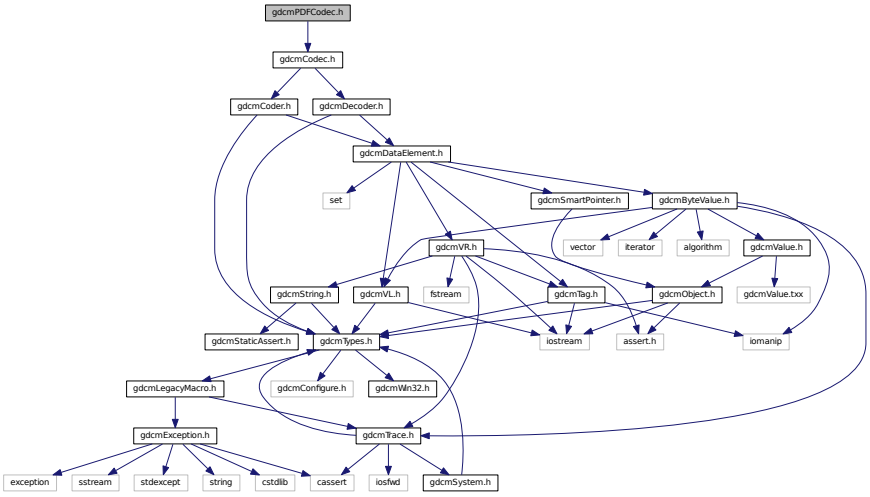
Functions

- 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.

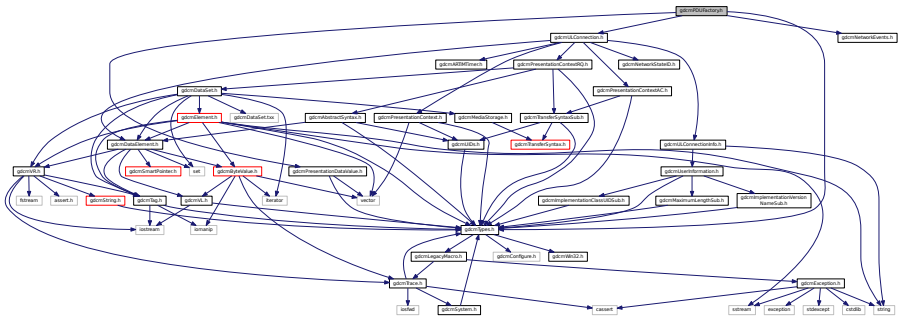
Namespaces

- 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 appropriate PDU. This way, the event loop doesn't have to know about all the different PDU types.

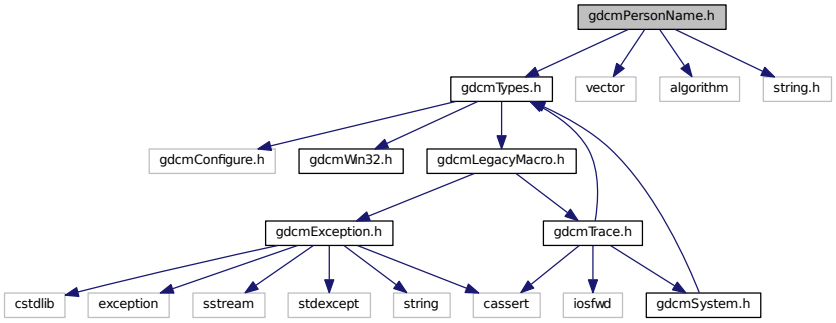
Namespaces

- 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.

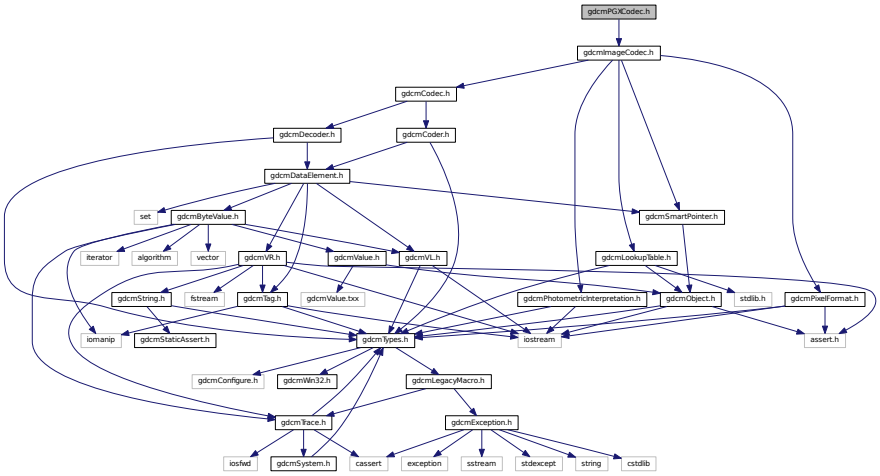
Namespaces

- gdcm

11.178 gdcmPGXCodec.h File Reference

```
#include "gdcmImageCodec.h"
```

Include dependency graph for gdcmPGXCodec.h:



Classes

- class gdcm::PGXCodec

  Class to do PGX See PGX as used in JPEG 2000 implementation and reference images.

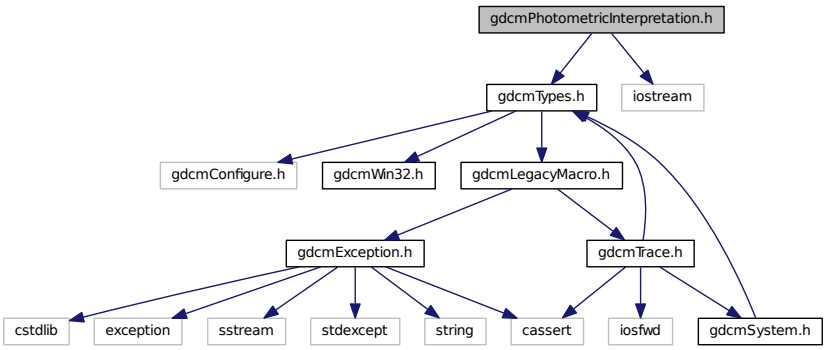
Namespaces

- gdcm

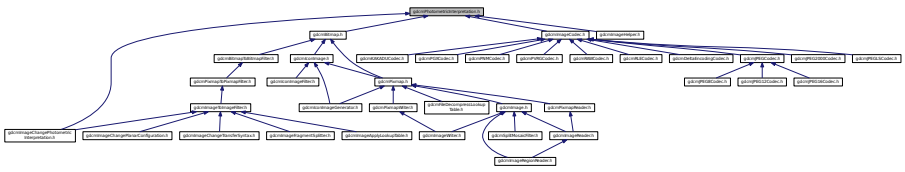
11.179 gdcmPhotometricInterpretation.h File Reference

```
#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

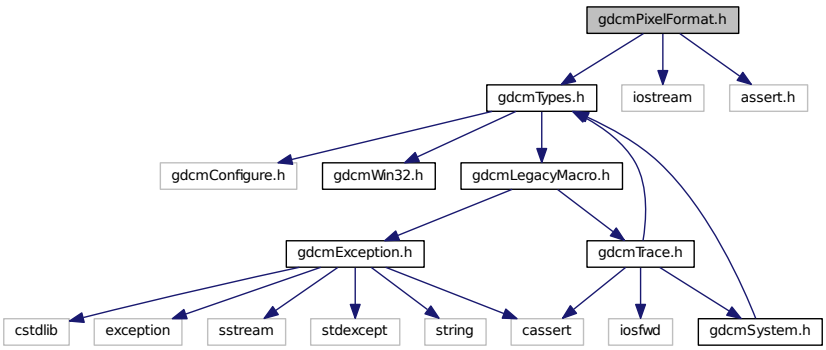
Functions

- std::ostream & gdcm::operator<< (std::ostream &os, const PhotometricInterpretation &val)

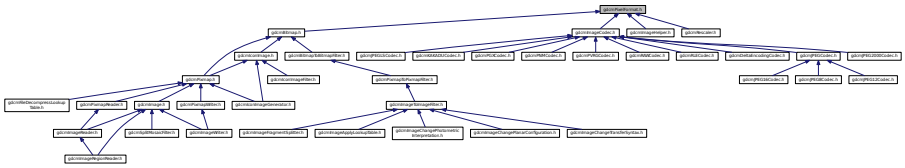
11.180 gdcmPixelFormat.h File Reference

```
#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

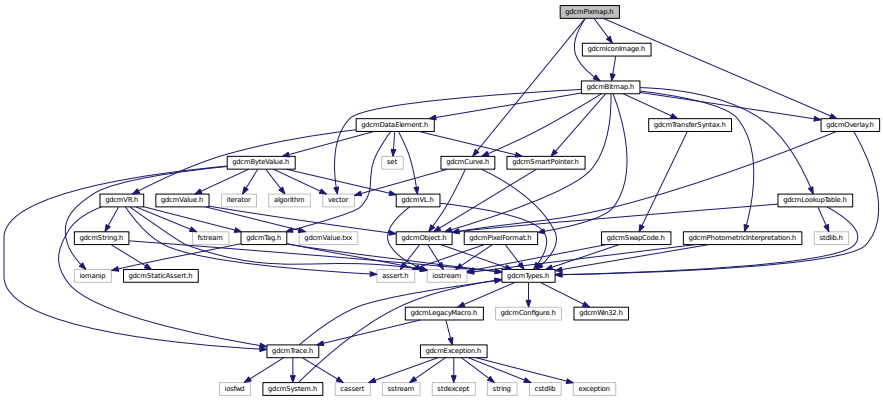
Functions

- std::ostream & gdcm::operator<< (std::ostream &os, const PixelFormat &pf)

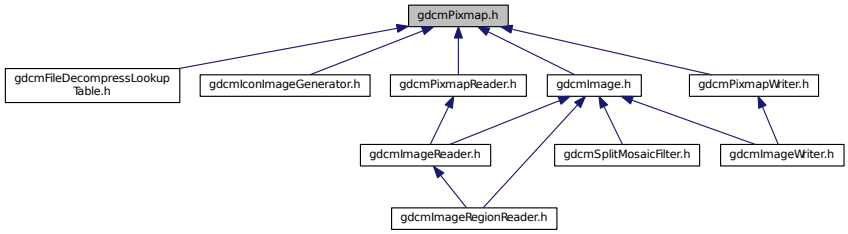
11.181 gdcmPixmap.h File Reference

```
#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 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)

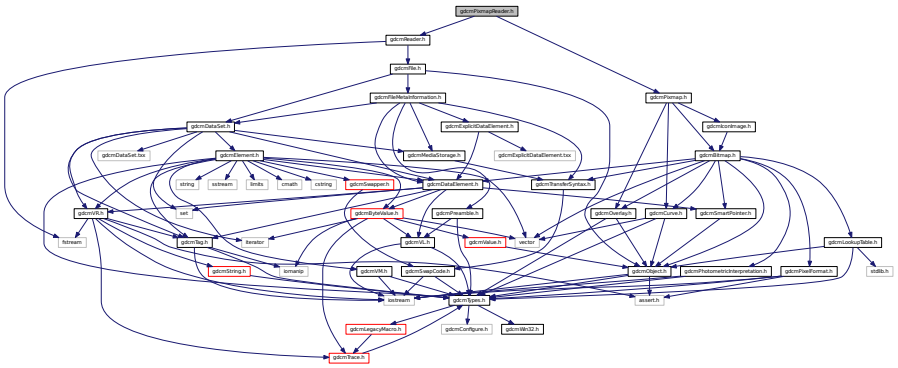
Namespaces

- gdcm

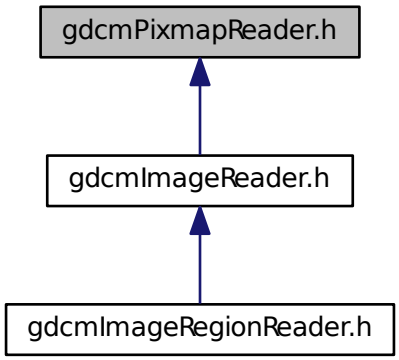
11.182 gdcmPixmapReader.h File Reference

```
#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.

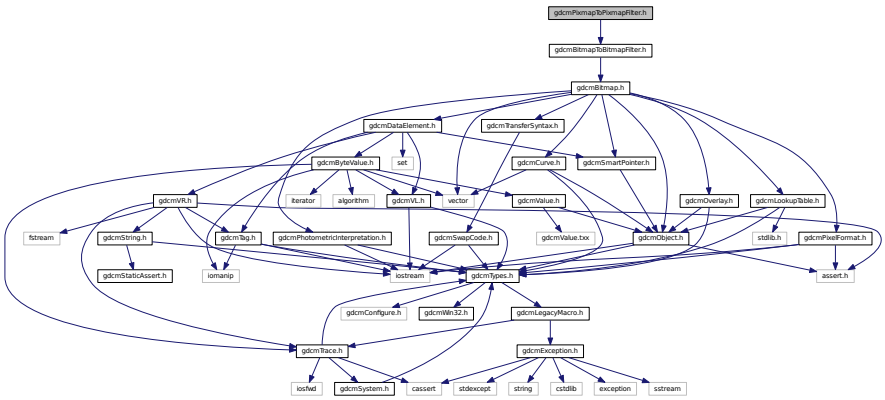
Namespaces

- gdcm

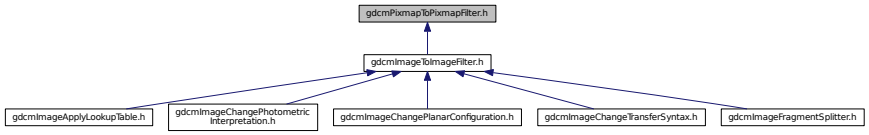
11.183 gdcmPixmapToPixmapFilter.h File Reference

```
#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 Super class for all filter taking an image and producing an output image.

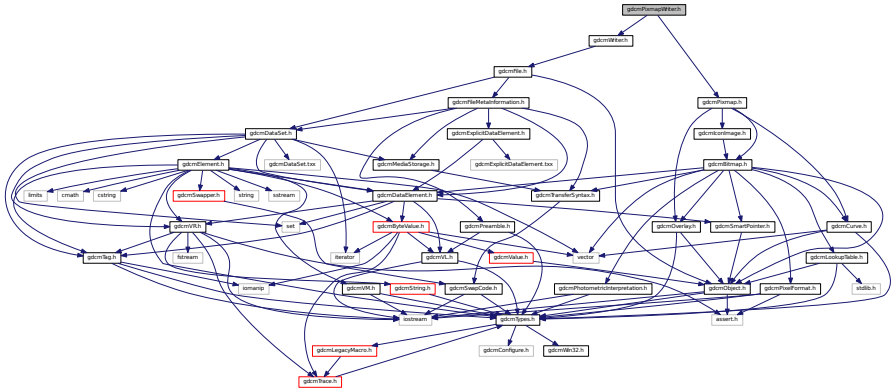
Namespaces

- gdcm

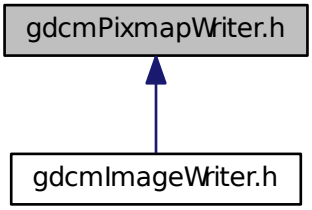
11.184 gdcmPixmapWriter.h File Reference

```
#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 This class will takes two inputs:

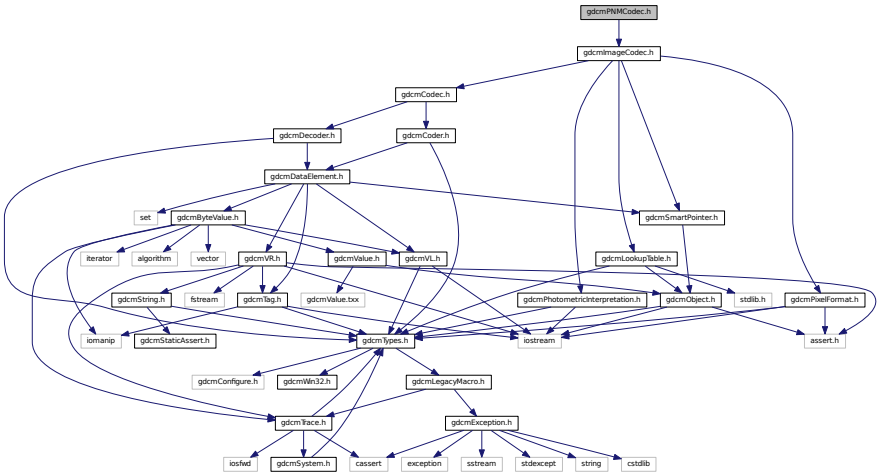
Namespaces

- gdcm

11.185 gdcmPNMCodec.h File Reference

```
#include "gdcmImageCodec.h"
```

Include dependency graph for gdcmPNMCodec.h:



Classes

- class gdcm::PNMCodec

  Class to do PNM PNM is the Portable anymap file format. The main web page can be found at: `http://netpbm.↩ sourceforge.net/`.

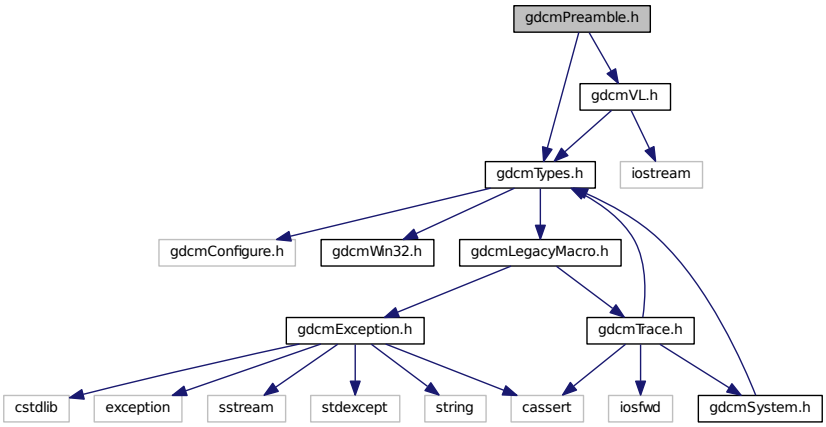
Namespaces

- gdcm

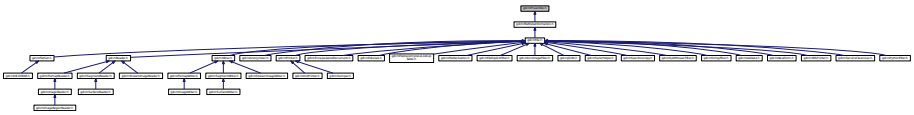
11.186 gdcmPreamble.h File Reference

```
#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

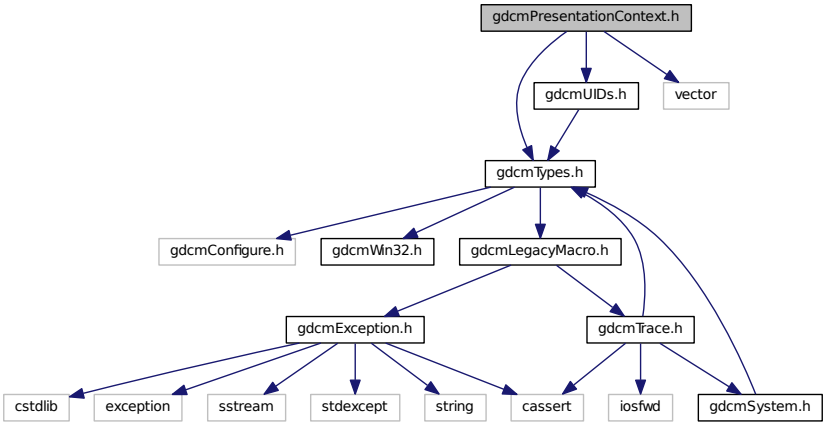
Functions

- std::ostream & gdcm::operator<< (std::ostream &os, const Preamble &val)

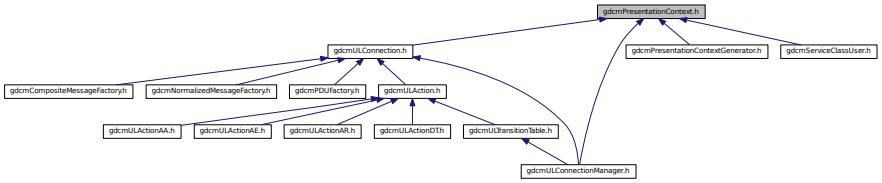
11.187 gdcmPresentationContext.h File Reference

```
#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.

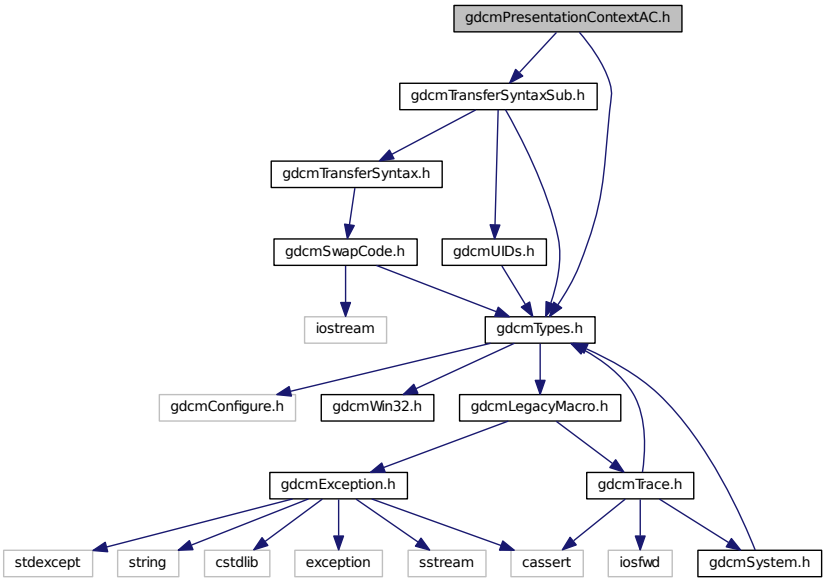
Namespaces

- gdcm

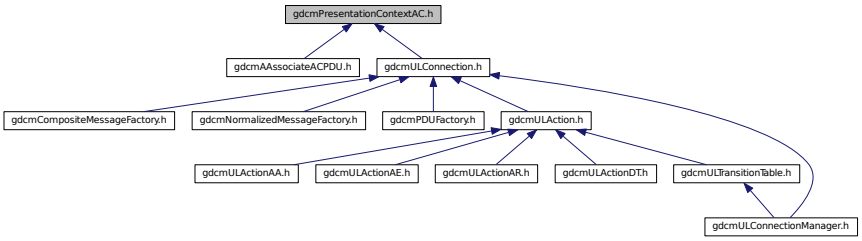
11.188 gdcmPresentationContextAC.h File Reference

```
#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 Table 9-18 PRESENTATION CONTEXT ITEM FIELDS.

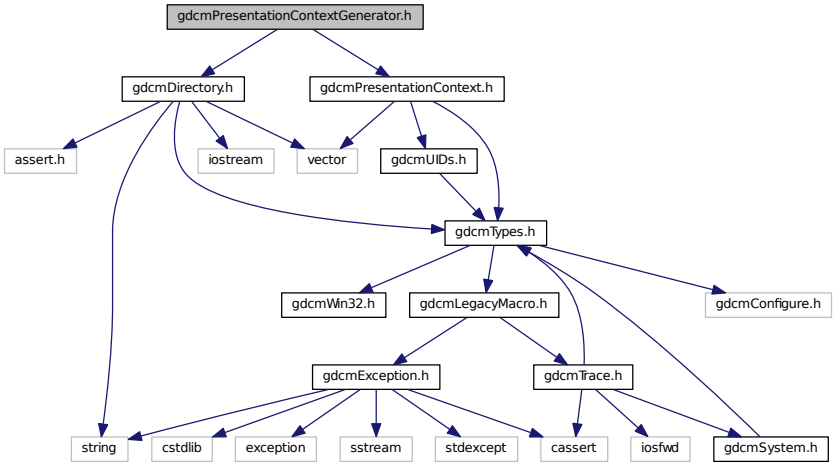
Namespaces

- 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 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.

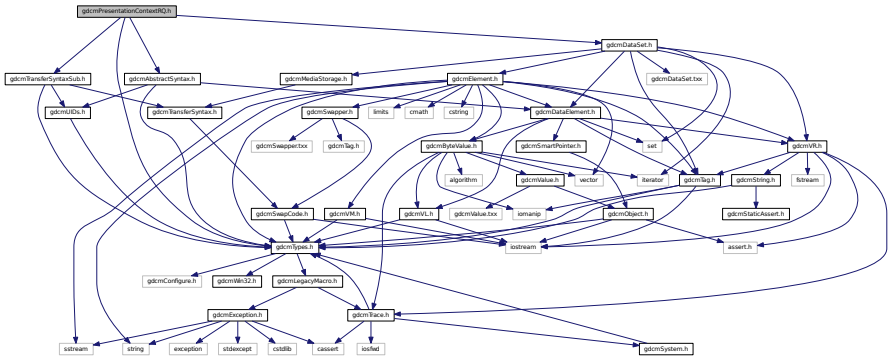
Namespaces

- gdcm

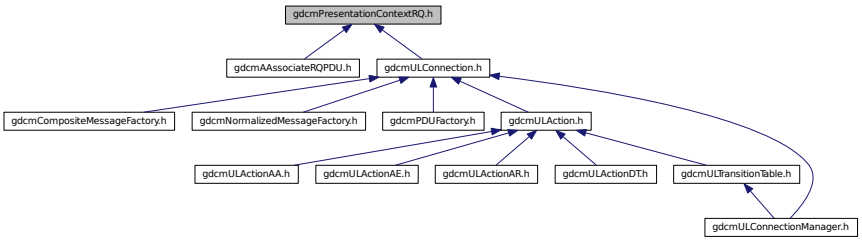
11.190 gdcmPresentationContextRQ.h File Reference

```
#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 Table 9-13 PRESENTATION CONTEXT ITEM FIELDS.

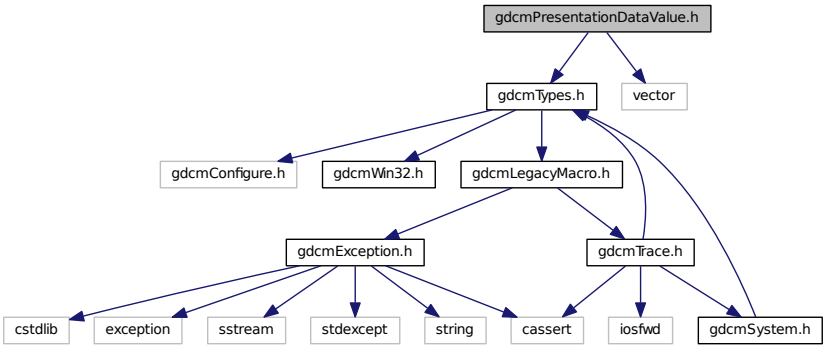
Namespaces

- gdcm
- gdcm::network

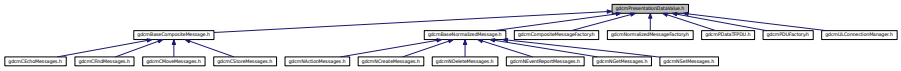
11.191 gdcmPresentationDataValue.h File Reference

```
#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 Table 9-23 PRESENTATION-DATA-VALUE ITEM FIELDS.

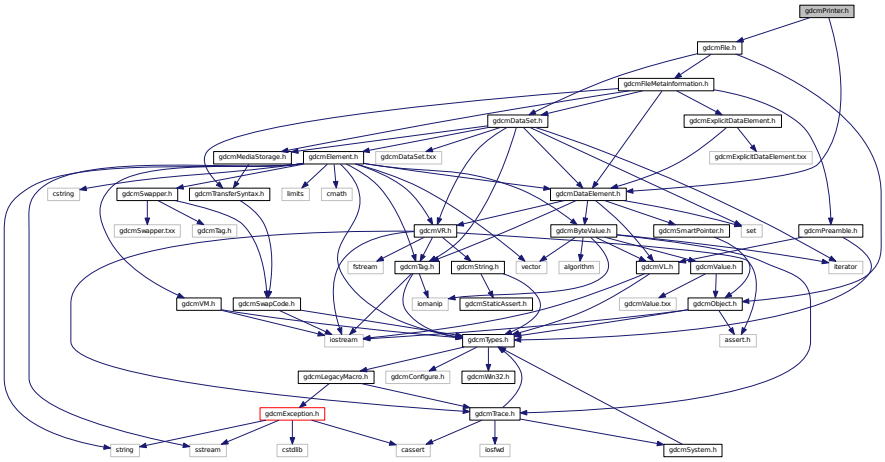
Namespaces

- gdcm
- gdcm::network

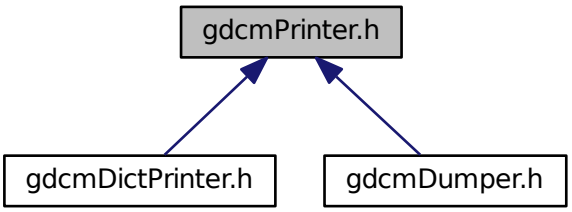
11.192 gdcmPrinter.h File Reference

```
#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.

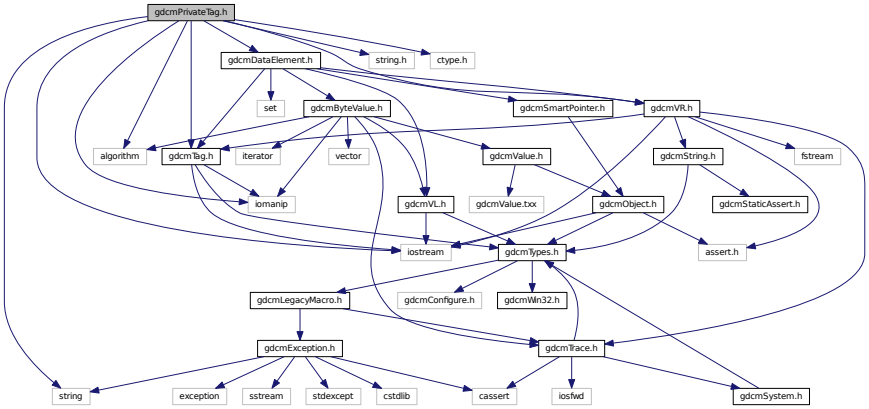
Namespaces

- gdcm

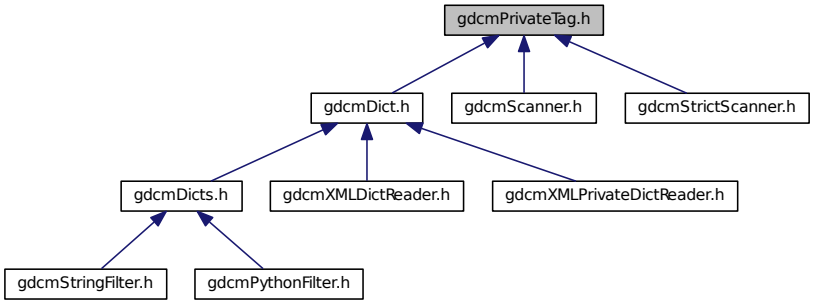
11.193 gdcmPrivateTag.h File Reference

```
#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

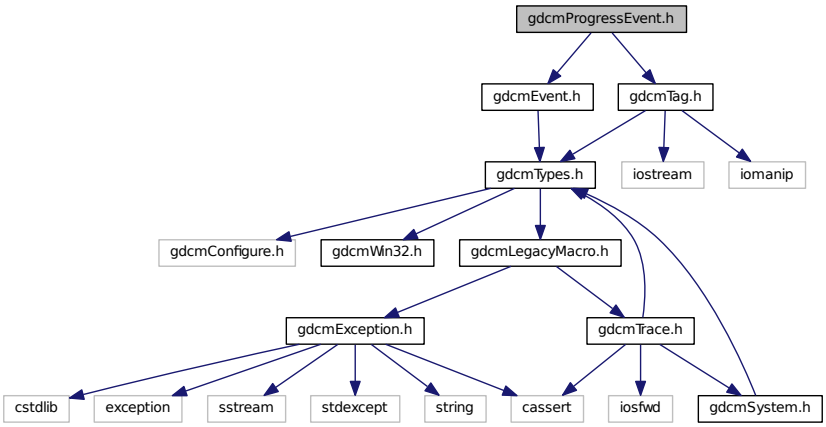
Functions

- 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 Special type of event triggered during.

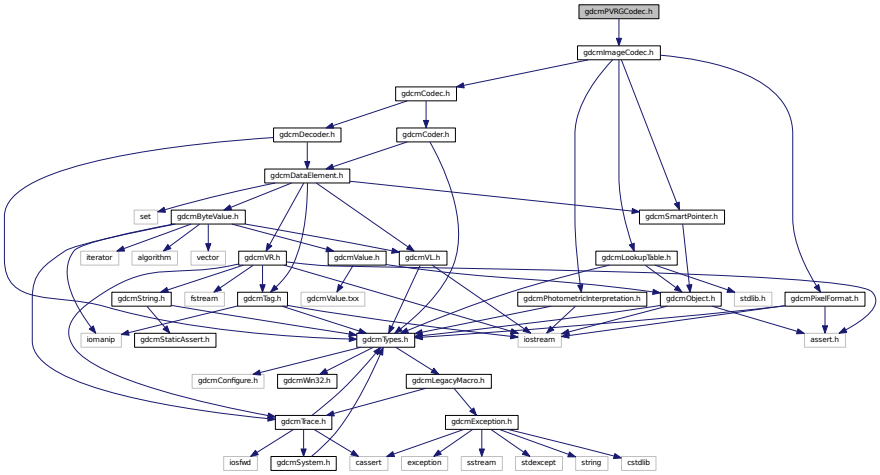
Namespaces

- gdcm

11.195 gdcmPVRGCodec.h File Reference

```
#include "gdcmImageCodec.h"
```

Include dependency graph for gdcmPVRGCodec.h:



Classes

- class gdcm::PVRGCodec
  PVRGCodec.

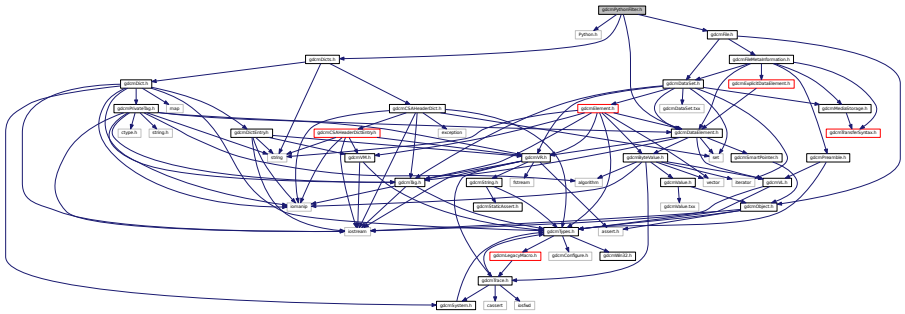
Namespaces

- 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.

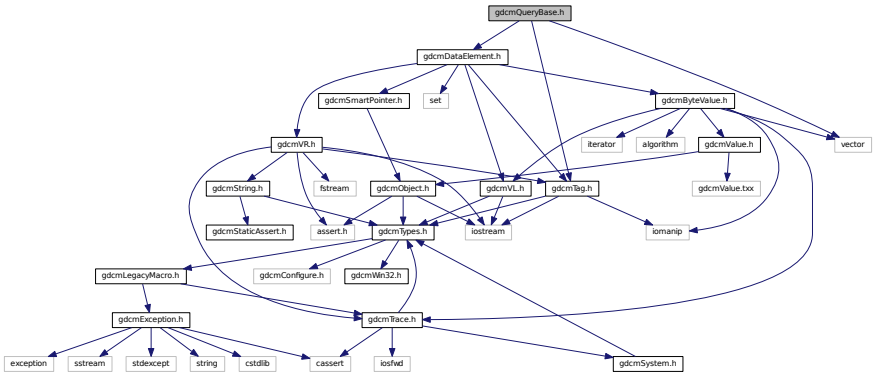
Namespaces

- gdcm

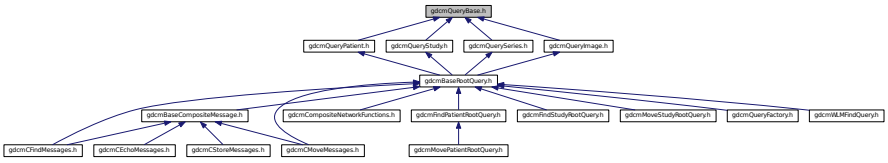
11.197 gdcmQueryBase.h File Reference

```
#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 contains: the base class for constructing a query dataset for a C-FIND and a C-MOVE.

Namespaces

- gdcm

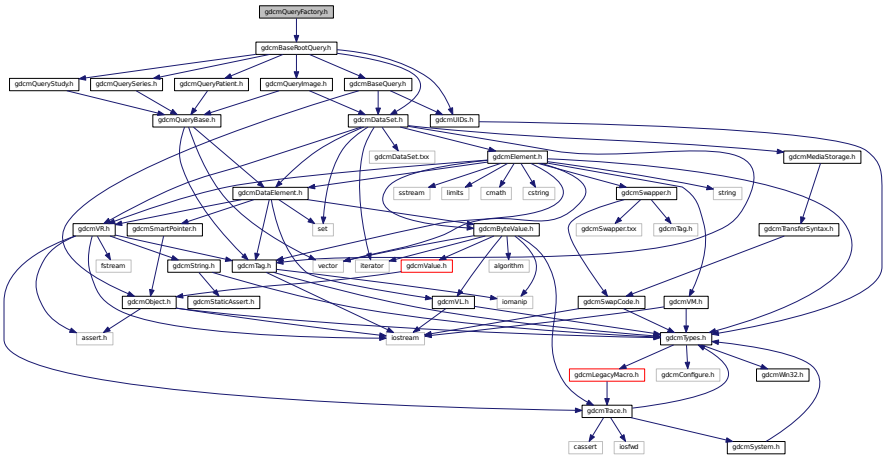
Enumerations

- 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

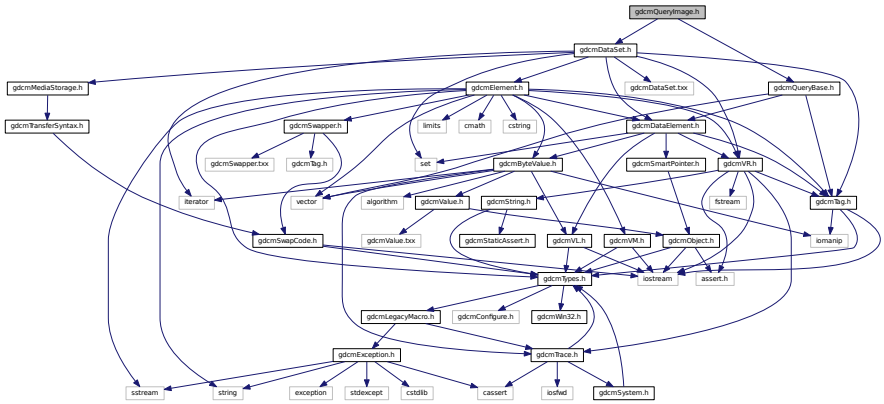
Enumerations

- 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 }

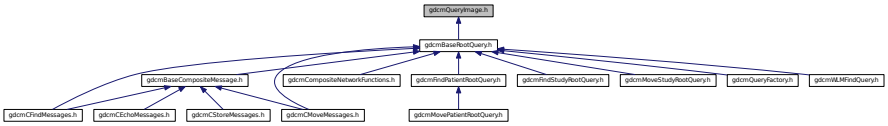
11.199 gdcmQueryImage.h File Reference

```
#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 contains: class to construct an image-based query for C-FIND and C-MOVE.

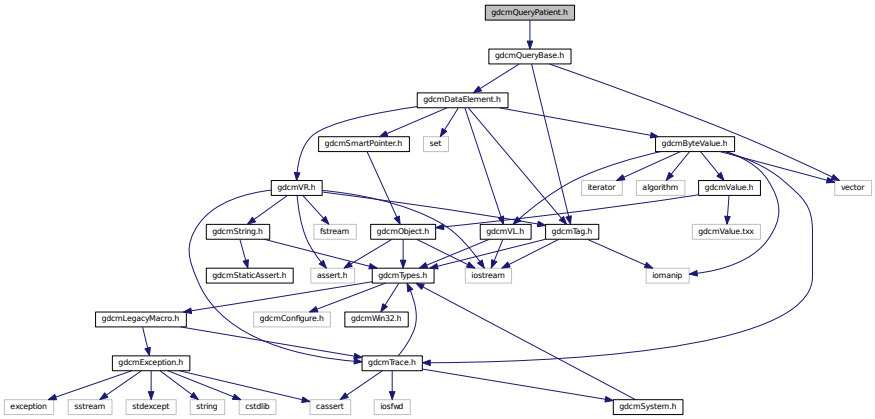
Namespaces

- gdcm

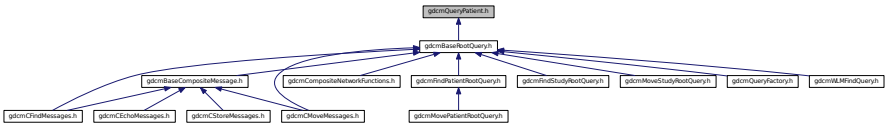
11.200 gdcmQueryPatient.h File Reference

```
#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 contains: class to construct a patient-based query for c-find and c-move.

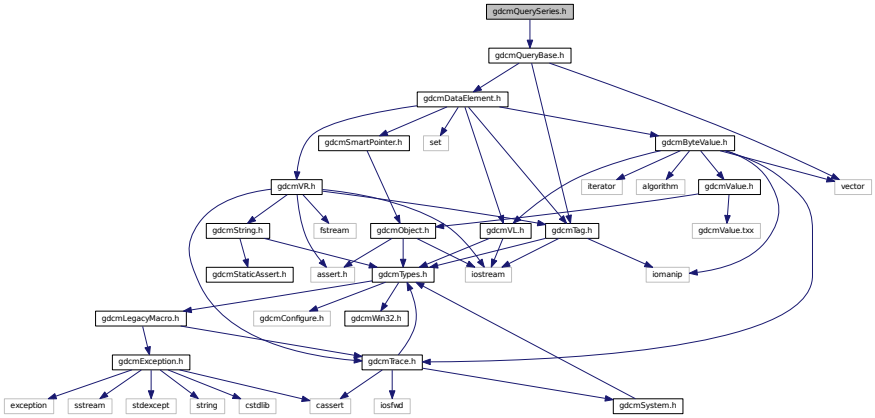
Namespaces

- gdcm

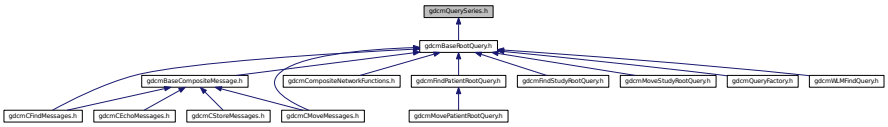
11.201 gdcmQuerySeries.h File Reference

```
#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 contains: class to construct a series-based query for c-find and c-move.

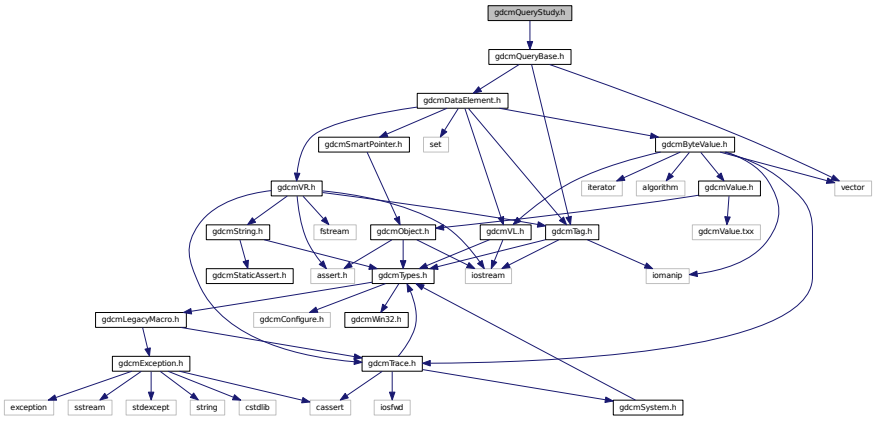
Namespaces

- gdcm

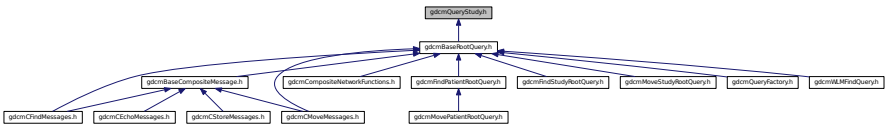
11.202 gdcmQueryStudy.h File Reference

```
#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 contains: class to construct a study-based query for C-FIND and C-MOVE.

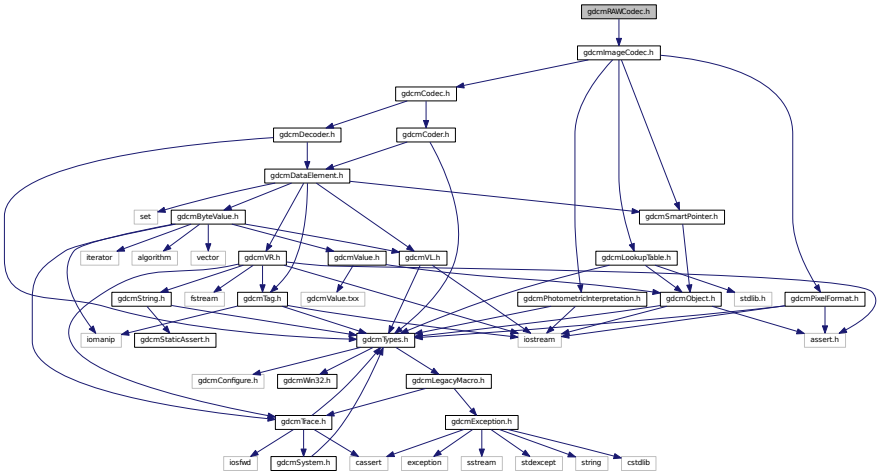
Namespaces

- gdcm

11.203 gdcmRAWCodec.h File Reference

```
#include "gdcmImageCodec.h"
```

Include dependency graph for gdcmRAWCodec.h:



Classes

- class gdcm::RAWCodec

  RAWCodec class.

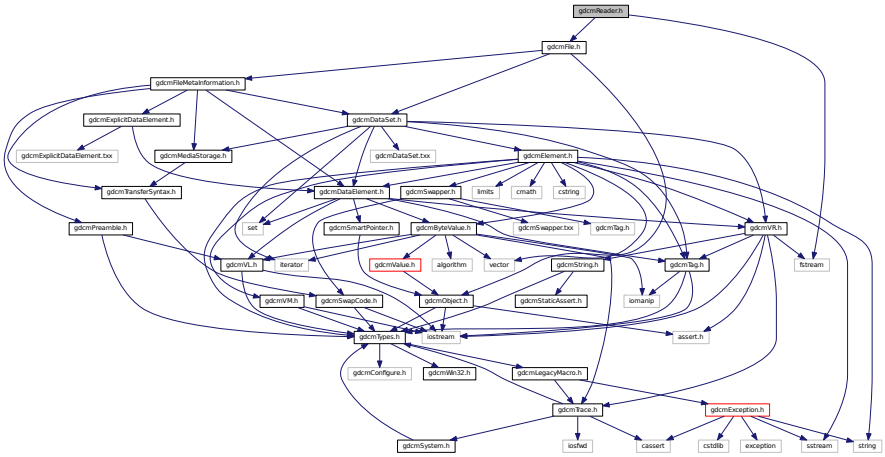
Namespaces

- gdcm

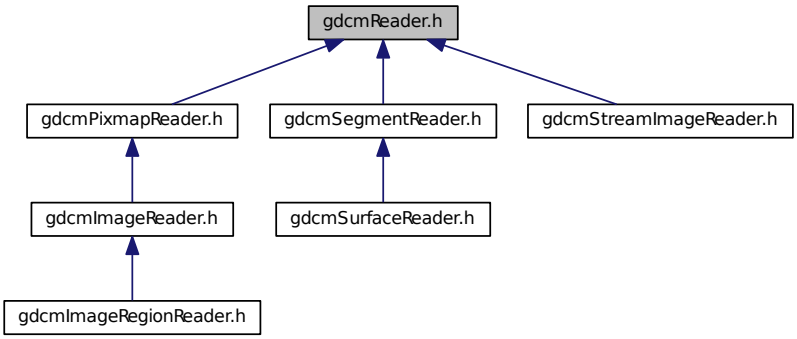
11.204 gdcmReader.h File Reference

```
#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)

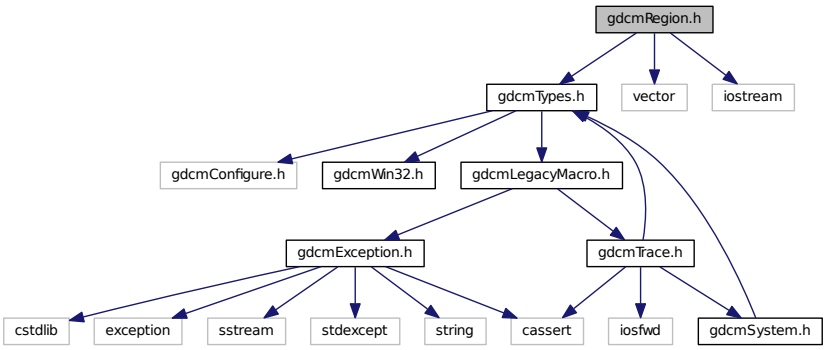
Namespaces

- gdcm

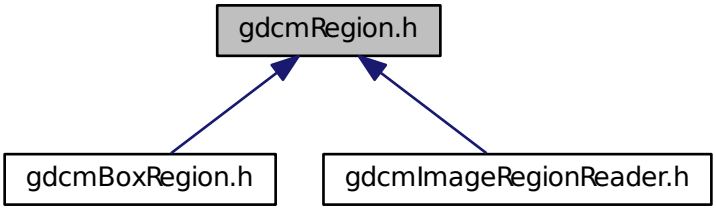
11.205 gdcmRegion.h File Reference

```
#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

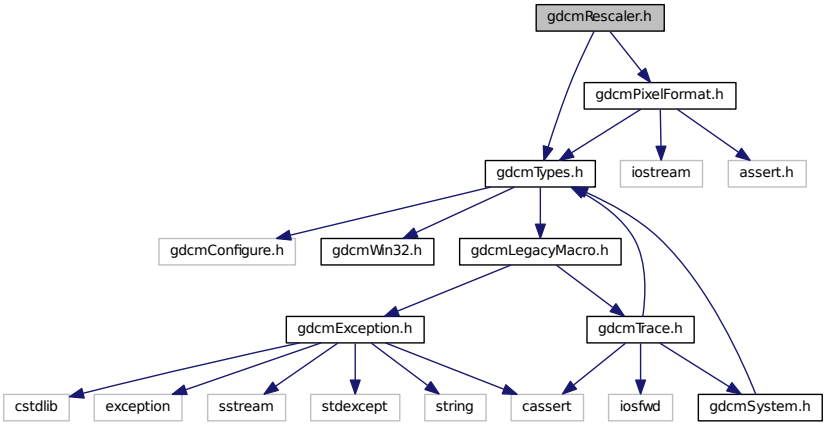
Functions

- 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 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:

  $$RWV = 1. * SV - 1024$$

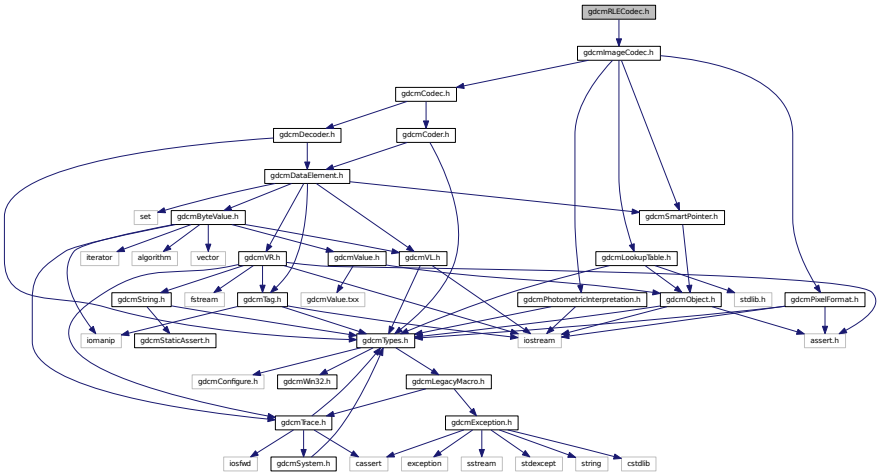
  So the best scalar to store the Real World Value will be 16 bits signed type.

Namespaces

- gdcm

11.207 gdcmRLECodec.h File Reference

`#include "gdcmImageCodec.h"`
Include dependency graph for gdcmRLECodec.h:



Classes

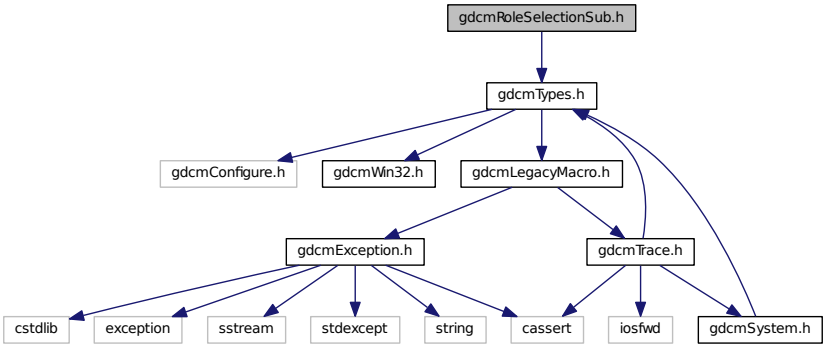
- class gdcm::RLECodec
  Class to do RLE.

Namespaces

- gdcm

11.208 gdcmRoleSelectionSub.h File Reference

`#include "gdcmTypes.h"`
Include dependency graph for gdcmRoleSelectionSub.h:



Classes

- class gdcm::network::RoleSelectionSub

  RoleSelectionSub PS 3.7 Table D.3-9 SCP/SCU ROLE SELECTION SUB-ITEM FIELDS (A-ASSOCIATE-RQ)

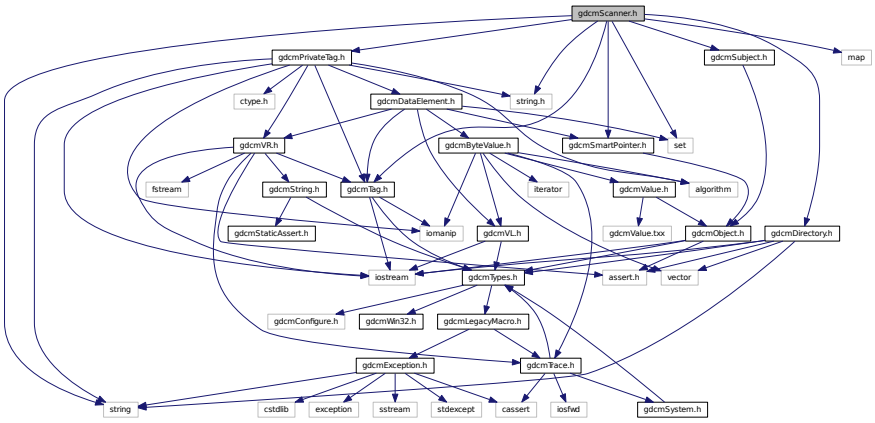
Namespaces

- 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 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.

Namespaces

- gdcm

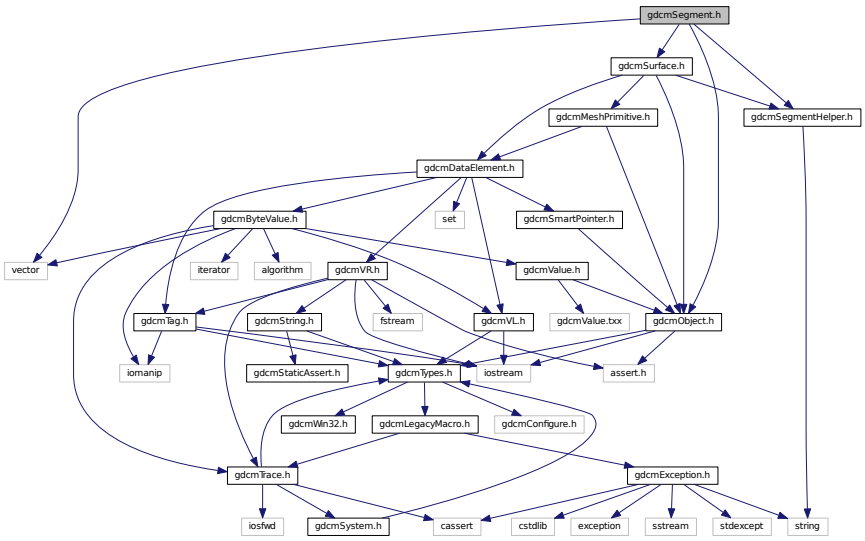
Functions

- std::ostream & gdcm::operator<< (std::ostream &os, const Scanner &s)

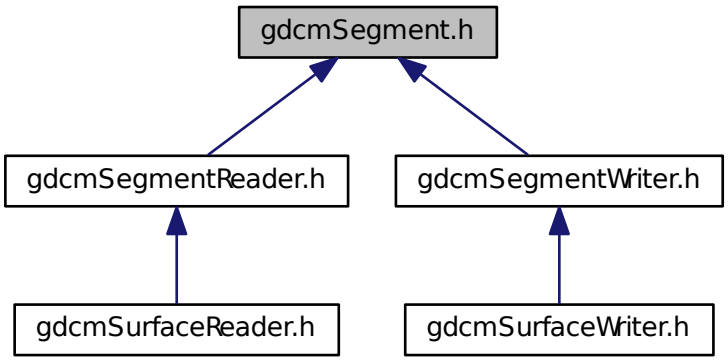
11.210 gdcmSegment.h File Reference

```
#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. It mainly contains attributes of group 0x0062. In addition, it can be associated with surface.

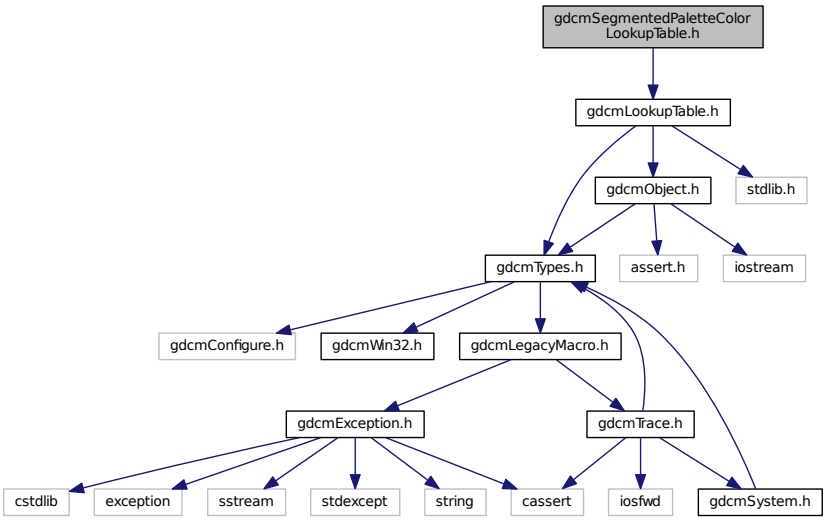
Namespaces

- gdcm

11.211 gdcmSegmentedPaletteColorLookupTable.h File Reference

```
#include "gdcmLookupTable.h"
```

Include dependency graph for gdcmSegmentedPaletteColorLookupTable.h:



Classes

- class gdcm::SegmentedPaletteColorLookupTable

  SegmentedPaletteColorLookupTable class.

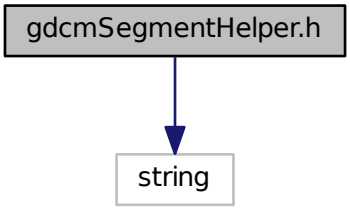
Namespaces

- gdcm

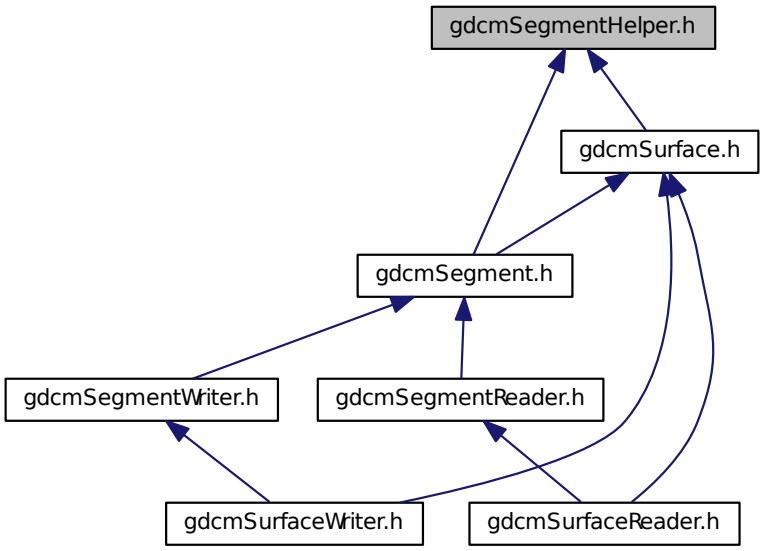
11.212 gdcmSegmentHelper.h File Reference

```
#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.

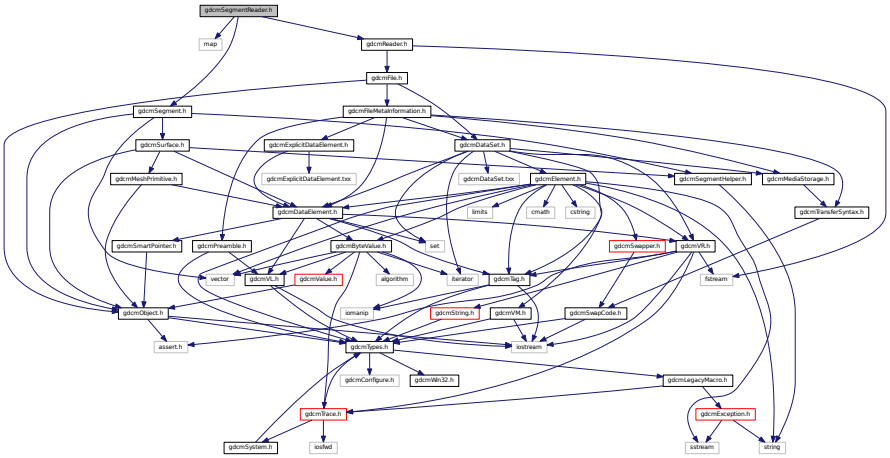
Namespaces

- gdcm
- gdcm::SegmentHelper

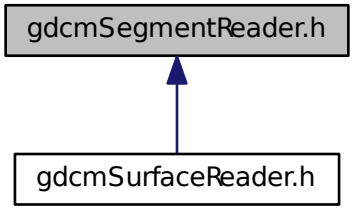
11.213 gdcmSegmentReader.h File Reference

```
#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. It reads attributes of group 0x0062.

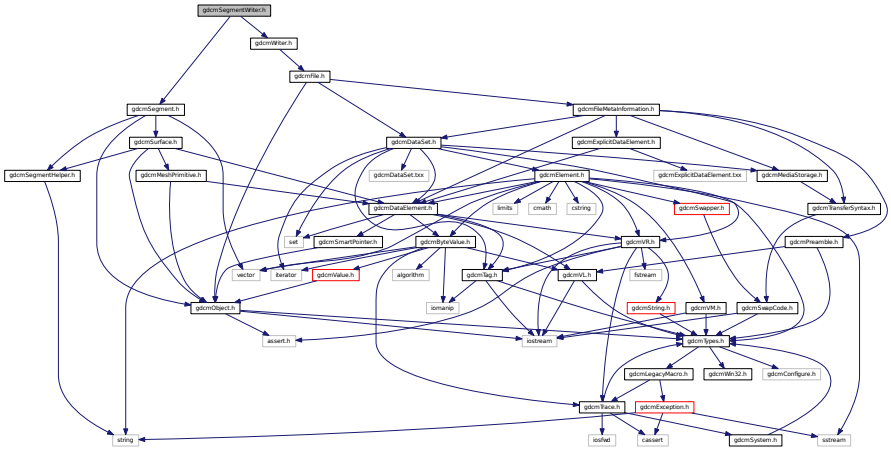
Namespaces

- gdcm

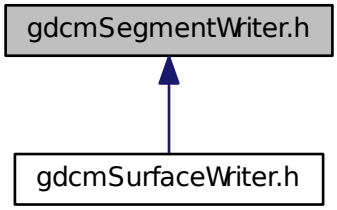
11.214 gdcmSegmentWriter.h File Reference

```
#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. It writes attributes of group 0x0062.

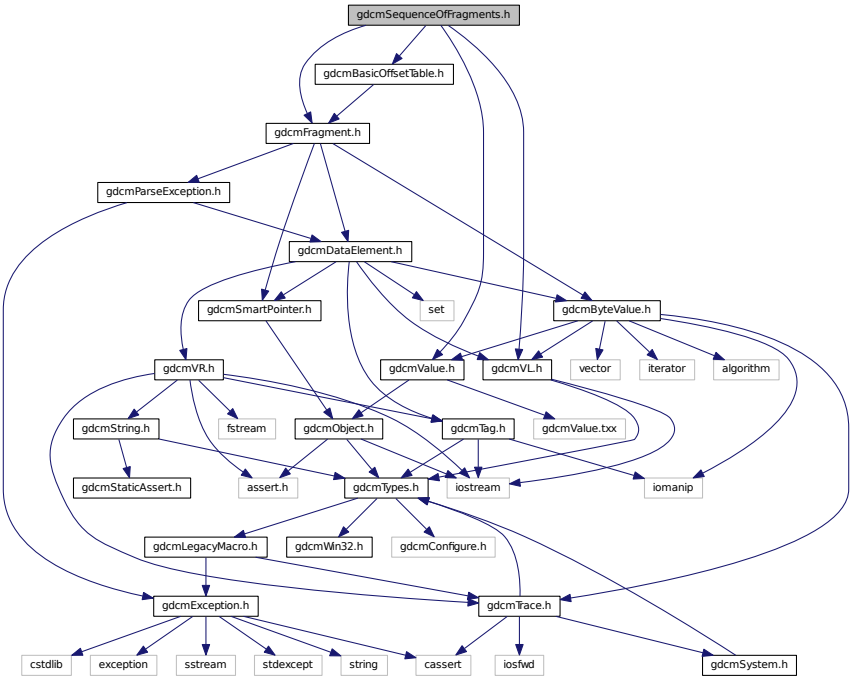
Namespaces

- 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

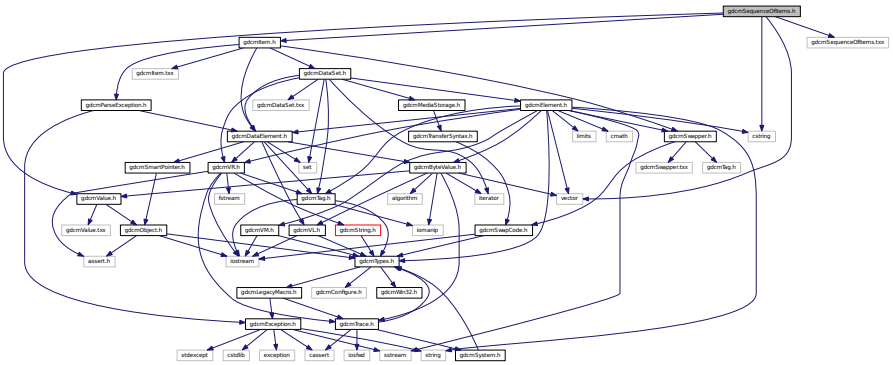
- gdcm

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 (value representation : SQ)

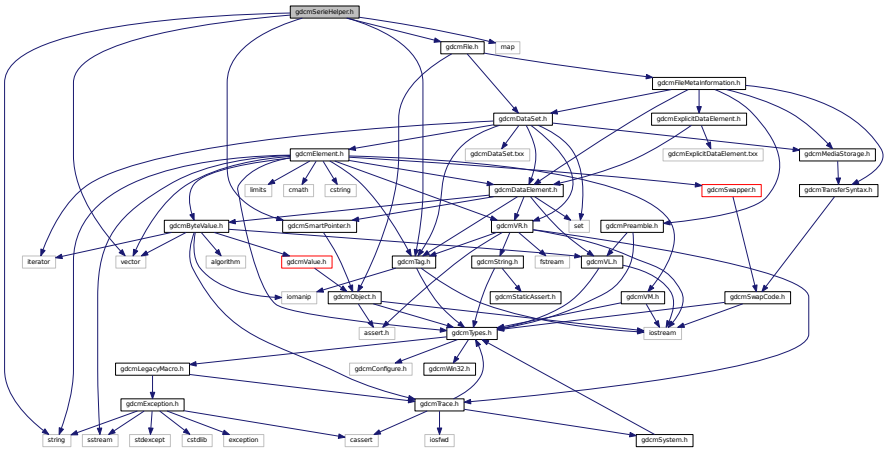
Namespaces

- 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

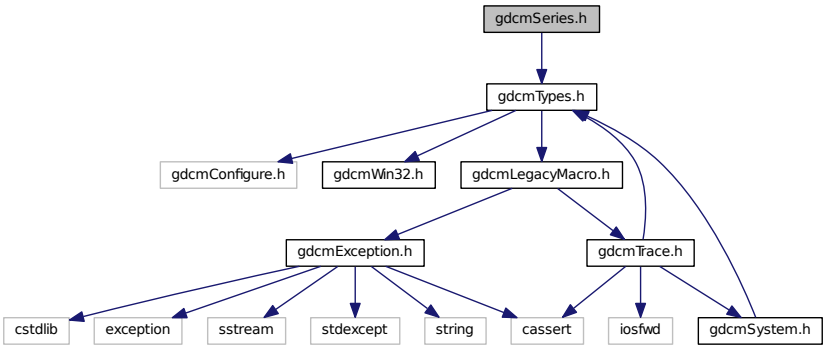
Enumerations

- 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 }

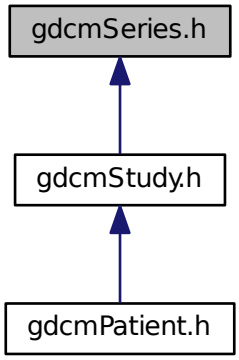
11.218 gdcmSeries.h File Reference

```
#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.

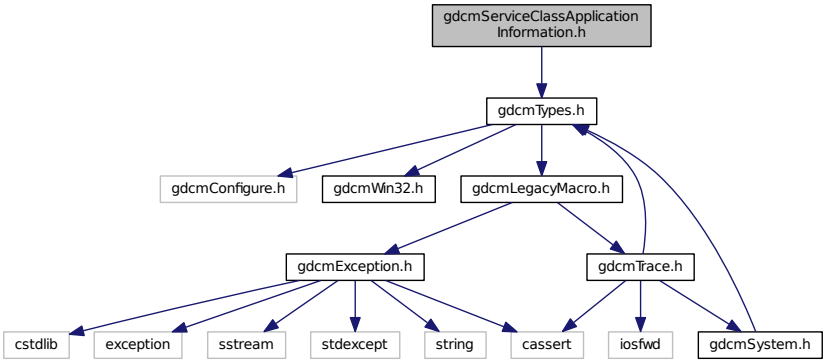
Namespaces

- gdcm

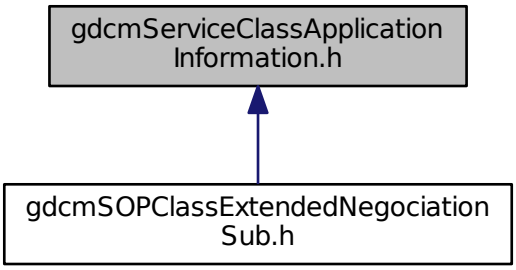
11.219 gdcmServiceClassApplicationInformation.h File Reference

```
#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

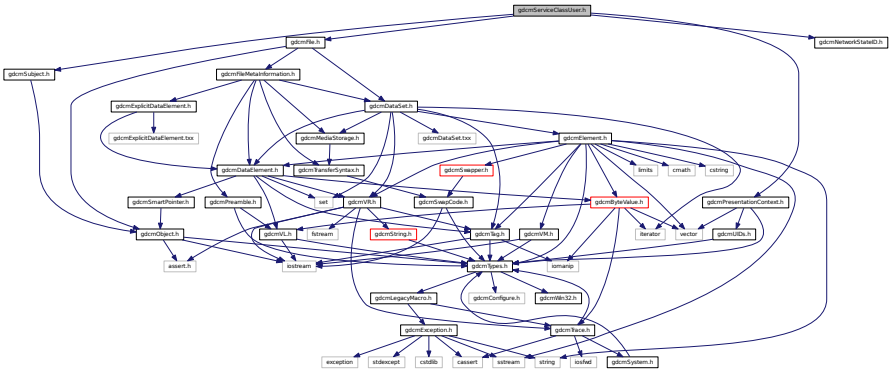
Namespaces

- 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.

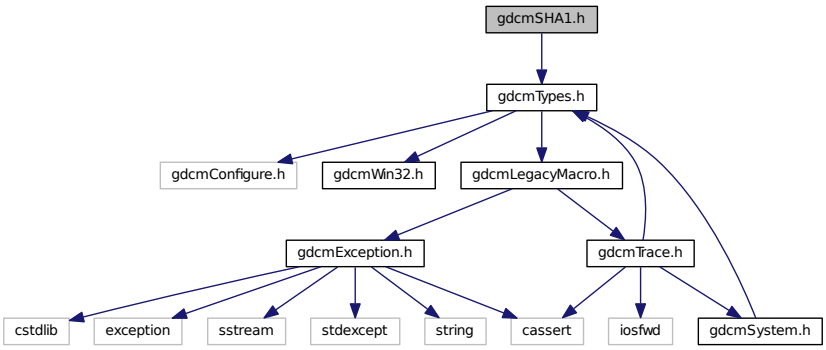
Namespaces

- 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.

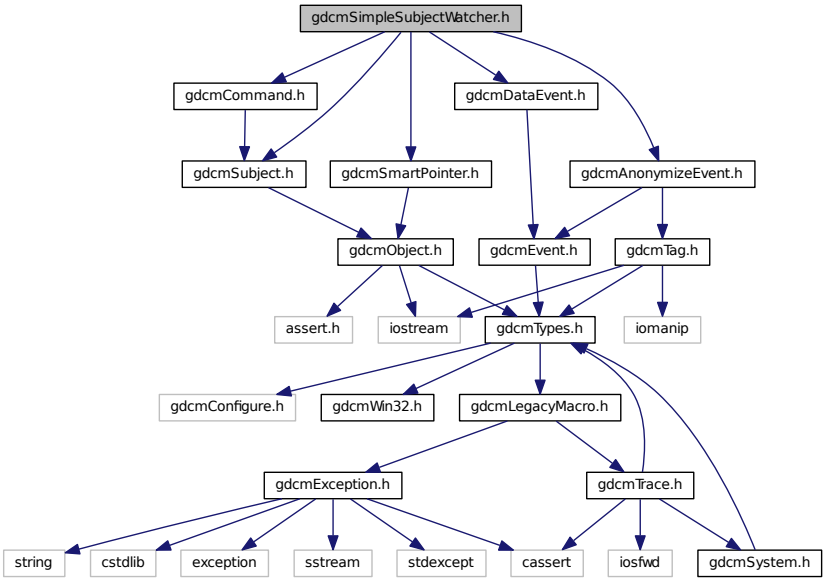
Namespaces

- 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 This is a typical Subject Watcher class. It will observe all events.

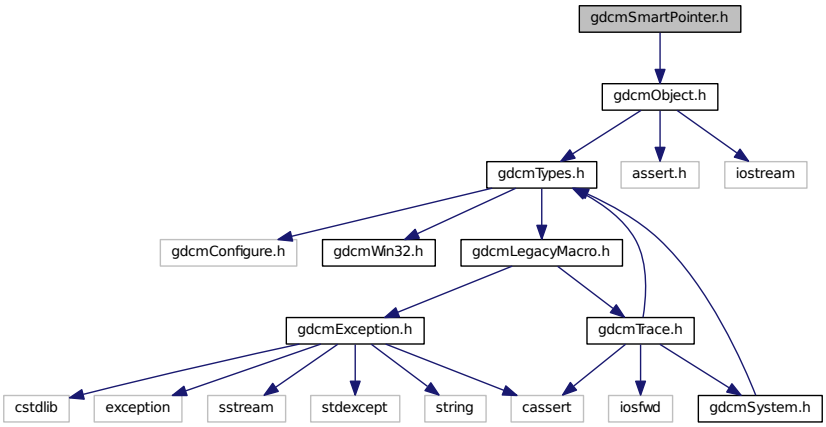
Namespaces

- gdcm

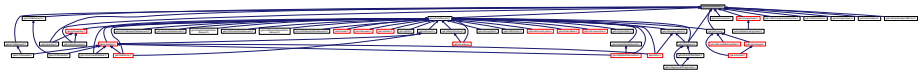
11.223 gdcmSmartPointer.h File Reference

```
#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.

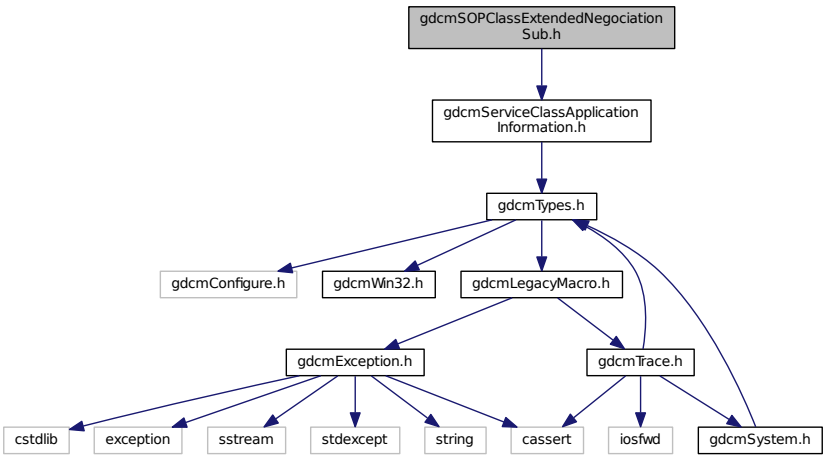
Namespaces

- gdcm

11.224 gdcmSOPClassExtendedNegociationSub.h File Reference

```
#include "gdcmServiceClassApplicationInformation.h"
```

Include dependency graph for gdcmSOPClassExtendedNegociationSub.h:



Classes

- class gdcm::network::SOPClassExtendedNegociationSub

  SOPClassExtendedNegociationSub PS 3.7 Table D.3-11 SOP CLASS EXTENDED NEGOTIATION SUB-ITEM FIELDS (A-ASSOCIATE-RQ and A-ASSOCIATE-AC)

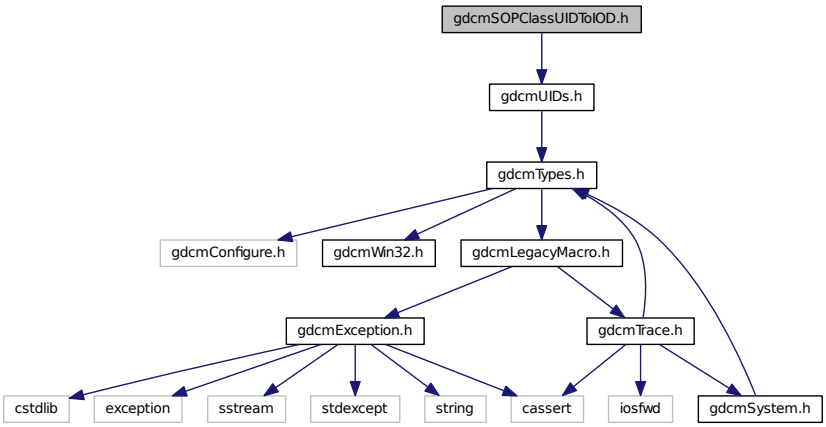
Namespaces

- 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.

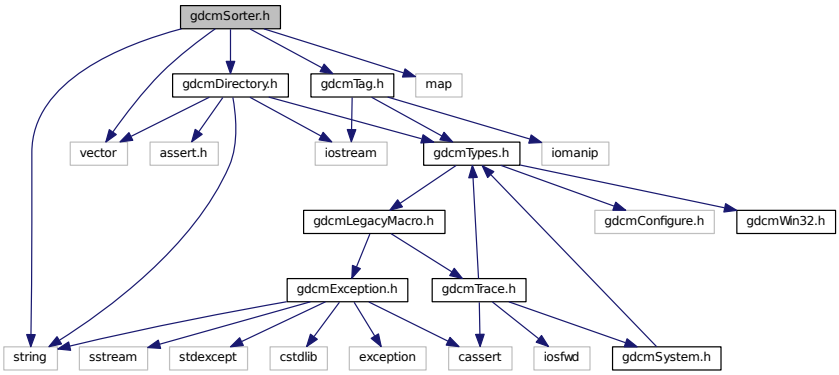
Namespaces

- gdcm

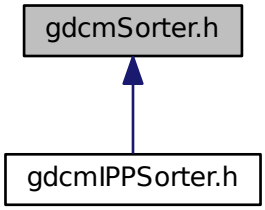
11.226 gdcmSorter.h File Reference

```
#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 General class to do sorting using a custom function You simply need to provide a function of type: Sorter::Sort↩ Function.

Namespaces

- gdcm

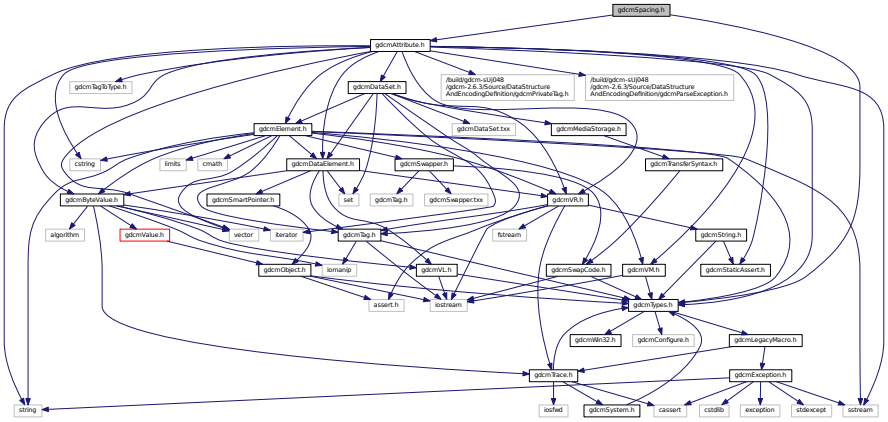
Functions

- 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.

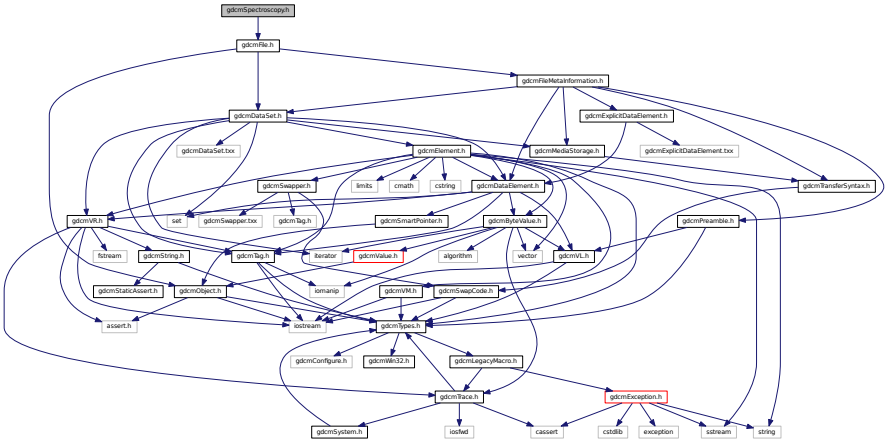
Namespaces

- gdcm

11.228 gdcmSpectroscopy.h File Reference

```
#include "gdcmFile.h"
```

Include dependency graph for gdcmSpectroscopy.h:



Classes

- class gdcm::Spectroscopy

  Spectroscopy class.

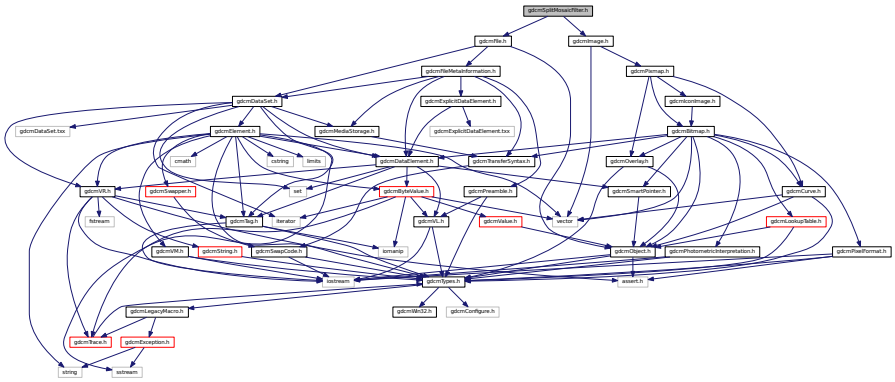
Namespaces

- gdcm

11.229 gdcmSplitMosaicFilter.h File Reference

```
#include "gdcmFile.h"
#include "gdcmImage.h"
```

Include dependency graph for gdcmSplitMosaicFilter.h:



Classes

- class gdcm::SplitMosaicFilter

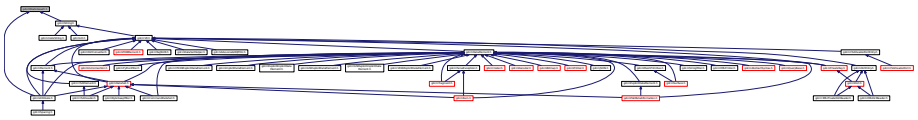
  SplitMosaicFilter class Class to reshuffle bytes for a SIEMENS Mosaic image Siemens CSA *Image* Header CSA:= Common Siemens Architecture, sometimes also known as Common syngo Architecture.

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 #define GDCM_DO_JOIN(*X, Y*) GDCM_DO_JOIN2(X,Y)

11.230.1.2 #define GDCM_DO_JOIN2(*X, Y*) X##Y

11.230.1.3 #define GDCM_JOIN(*X, Y*) GDCM_DO_JOIN(X, Y)

11.230.1.4 #define GDCM_STATIC_ASSERT(*B*)

Value:

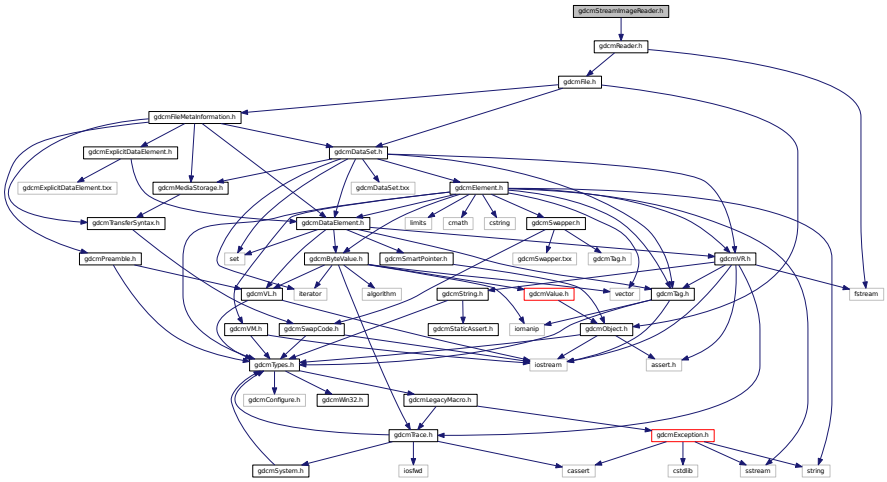
```
typedef ::gdcm::static_assert_test<\
    sizeof(::gdcm::STATIC_ASSERTION_FAILURE< (bool)( B ) >)>
      \
      GDCM_JOIN(gdcm_static_assert_typedef_, __LINE__)
```

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.

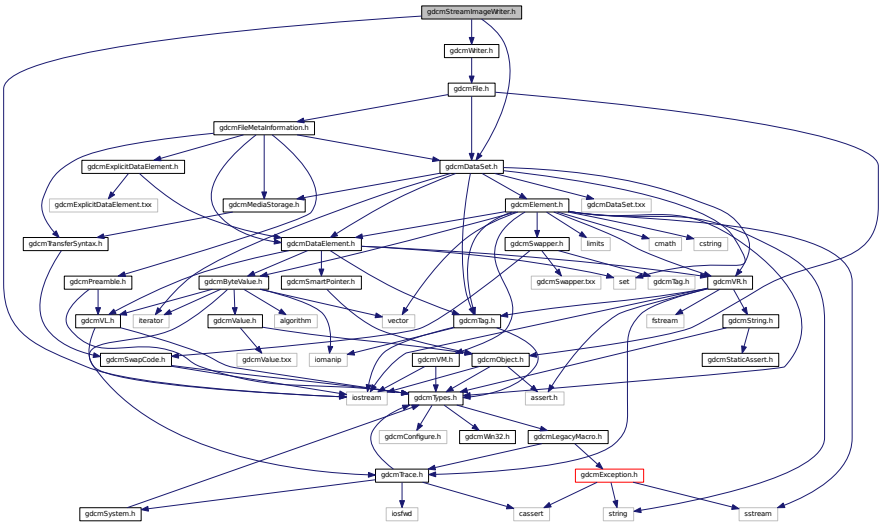
Namespaces

- 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.

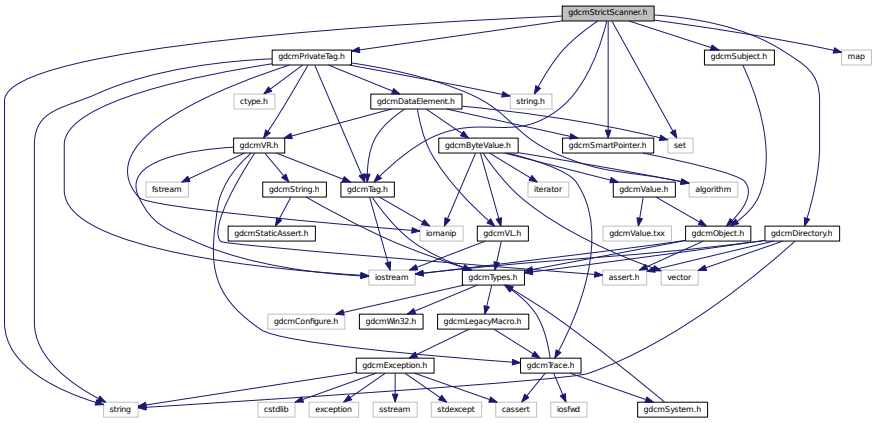
Namespaces

- 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 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.

Namespaces

- gdcm

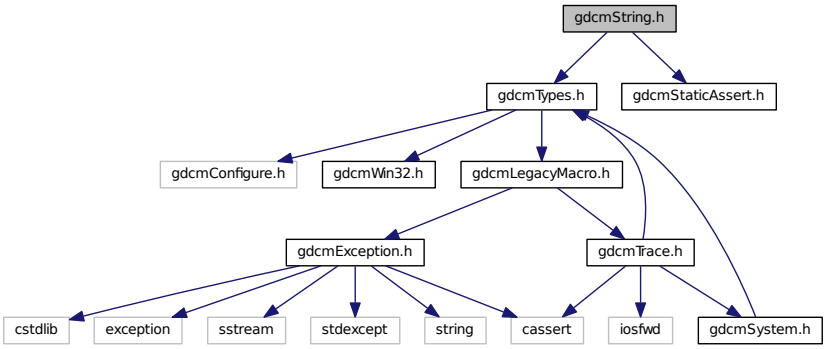
Functions

- std::ostream & gdcm::operator<< (std::ostream &os, const StrictScanner &s)

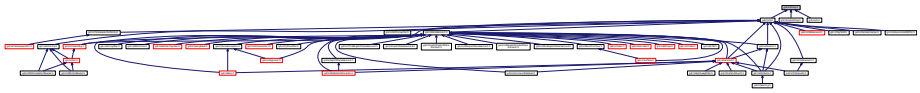
11.234 gdcmString.h File Reference

```
#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

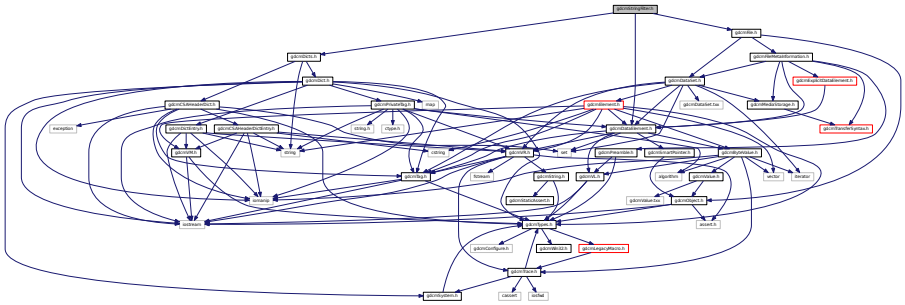
Functions

- 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 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.

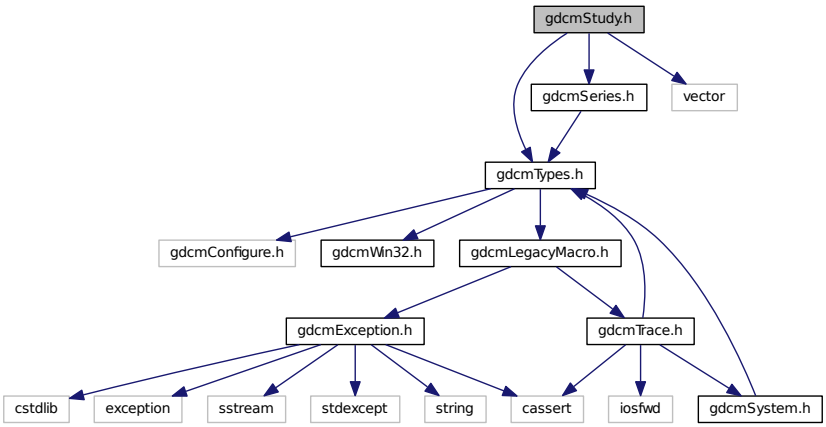
Namespaces

- gdcm

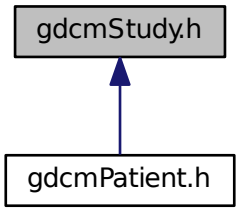
11.236 gdcmStudy.h File Reference

```
#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.

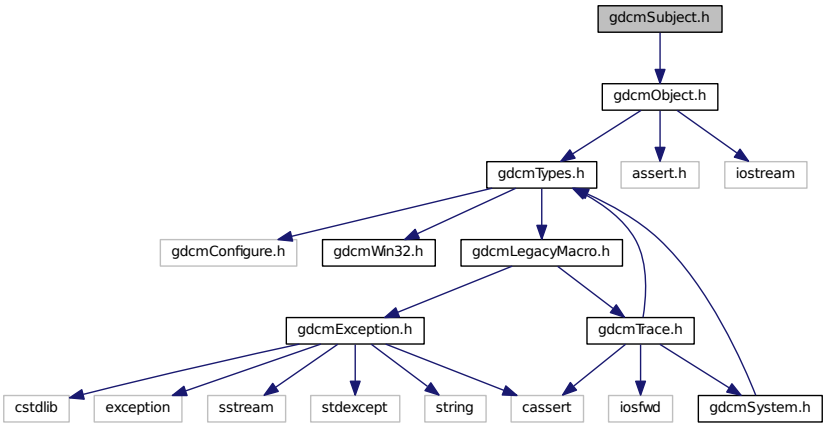
Namespaces

- 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.

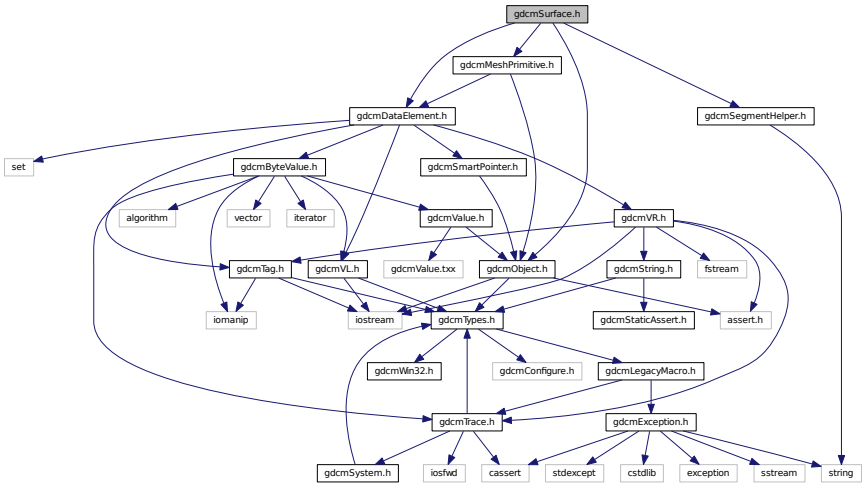
Namespaces

- gdcm

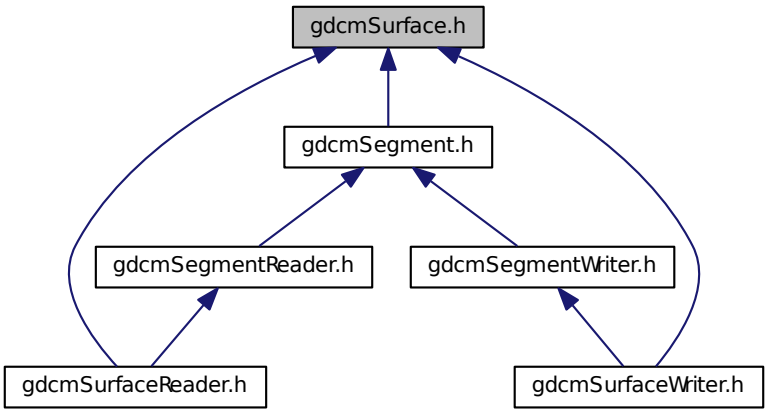
11.238 gdcmSurface.h File Reference

```
#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. This members are taken from required surface mesh module attributes.

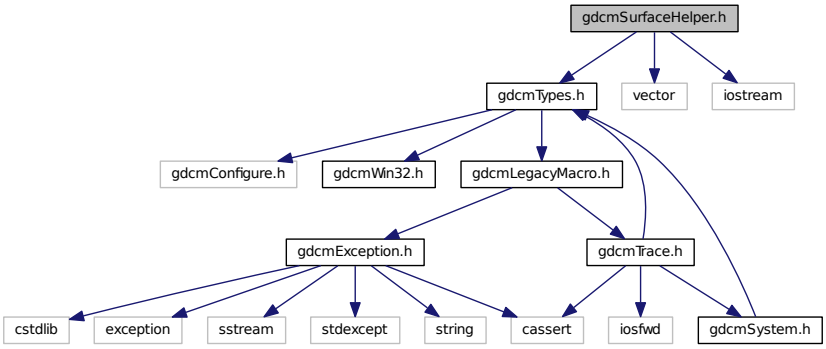
Namespaces

- 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 Helper class for Surface object.

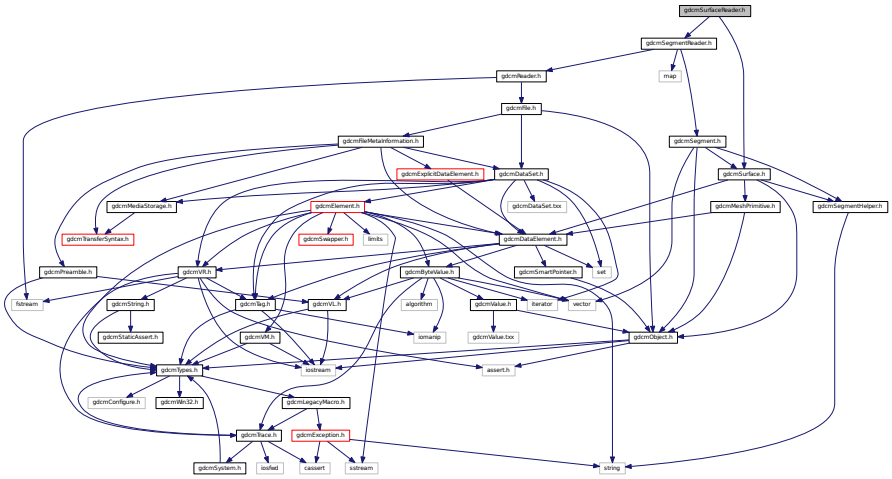
Namespaces

- 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. It reads surface mesh module attributes.

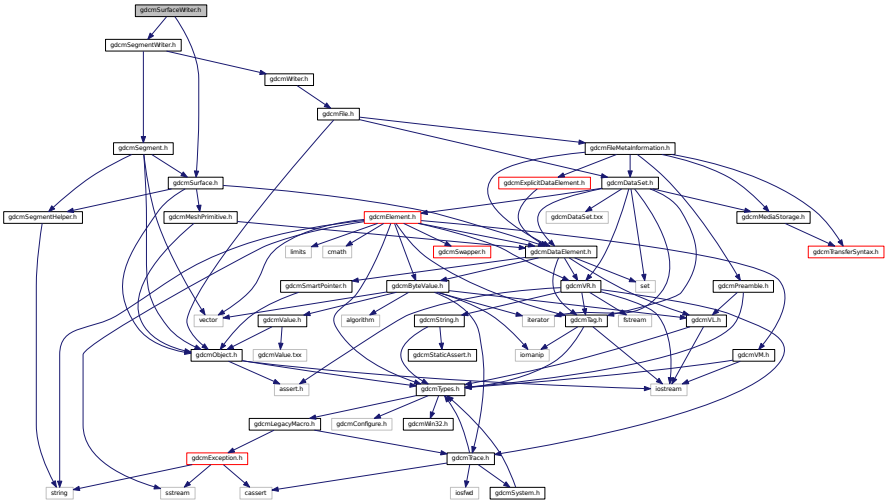
Namespaces

- 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. It writes surface mesh module attributes.

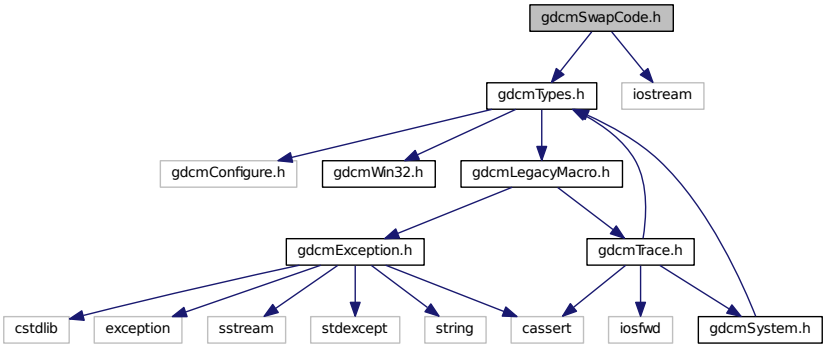
Namespaces

- gdcm

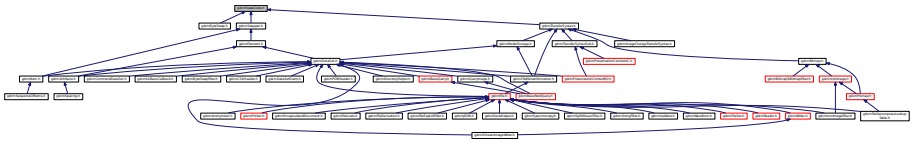
11.242 gdcmSwapCode.h File Reference

```
#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

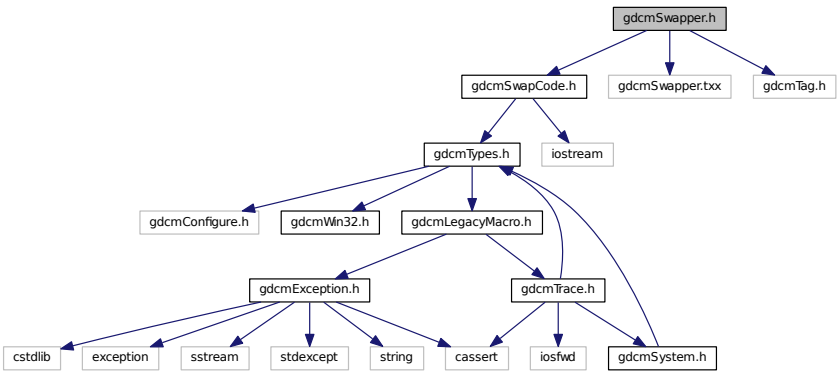
Functions

- 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

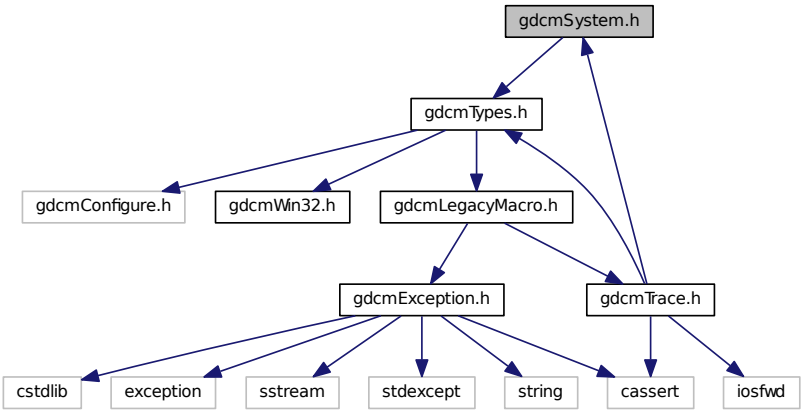
Namespaces

- gdcm

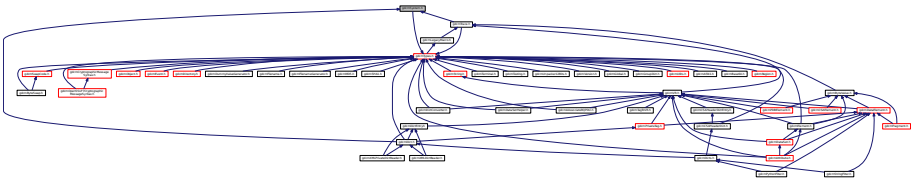
11.244 gdcmSystem.h File Reference

```
#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.

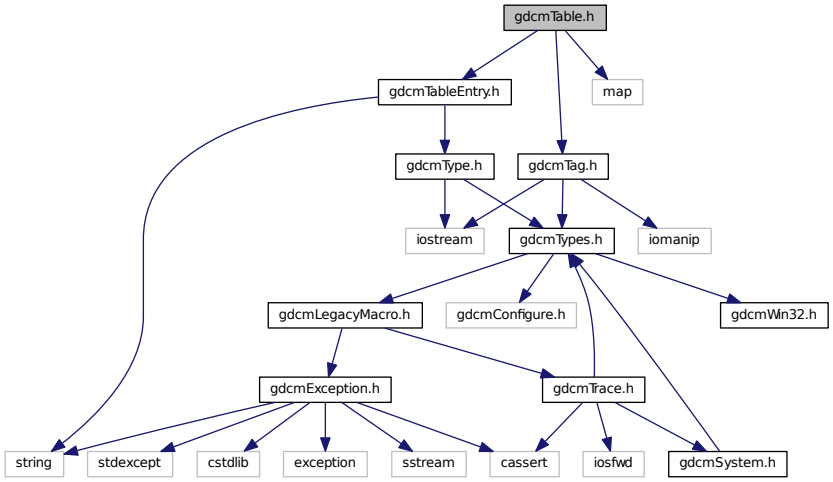
Namespaces

- 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.

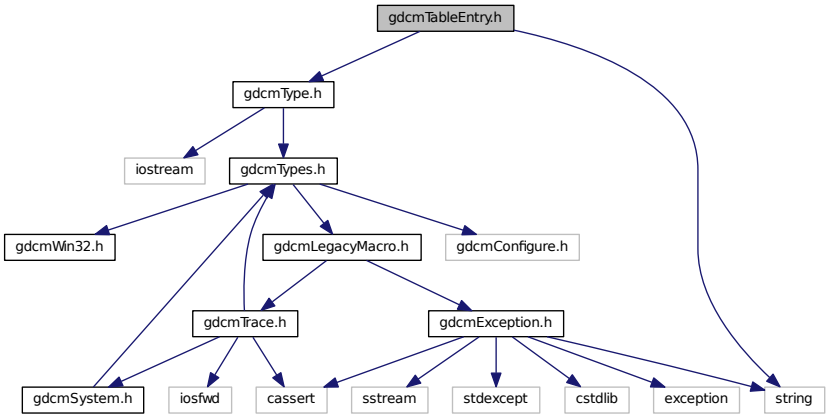
Namespaces

- gdcm

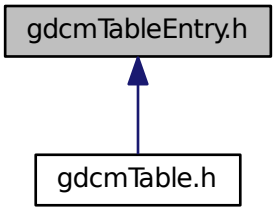
11.246 gdcmTableEntry.h File Reference

```
#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.

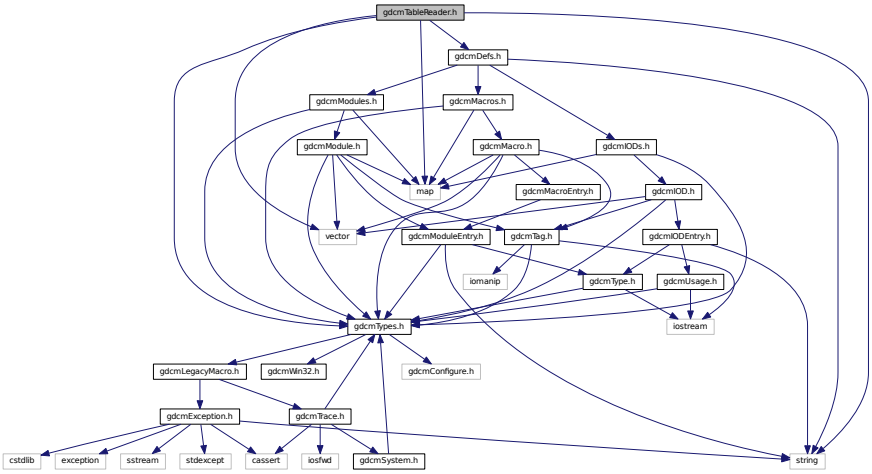
Namespaces

- gdcm

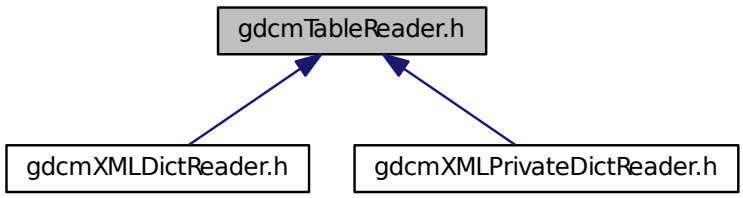
11.247 gdcmTableReader.h File Reference

```
#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.

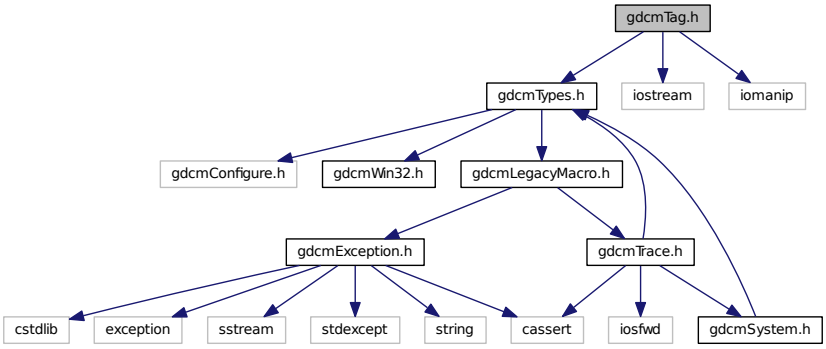
Namespaces

- gdcm

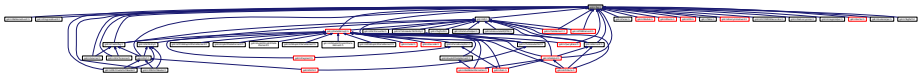
11.248 gdcmTag.h File Reference

```
#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). Basically an uint32_t which can also be expressed as two uint16_t (group and element)

Namespaces

- gdcm

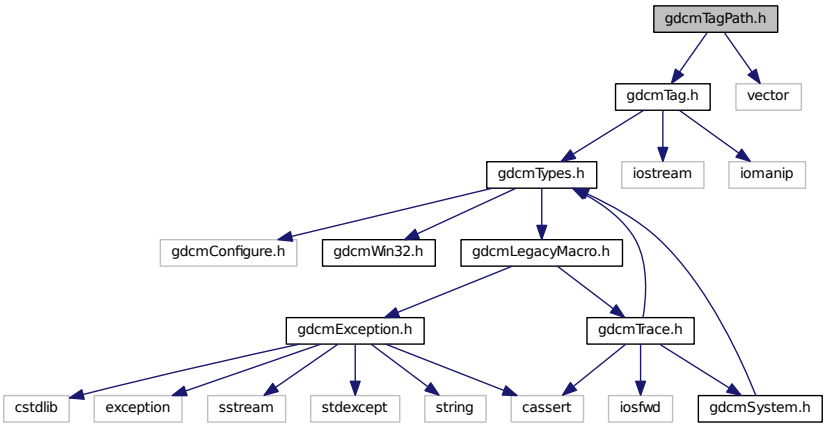
Functions

- 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.

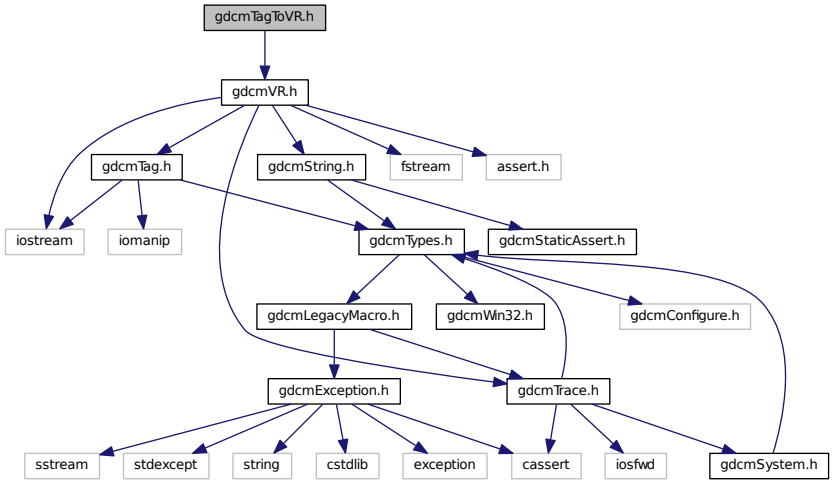
Namespaces

- gdcm

11.250 gdcmTagToVR.h File Reference

```
#include "gdcmVR.h"
```

Include dependency graph for gdcmTagToVR.h:



Namespaces

- gdcm

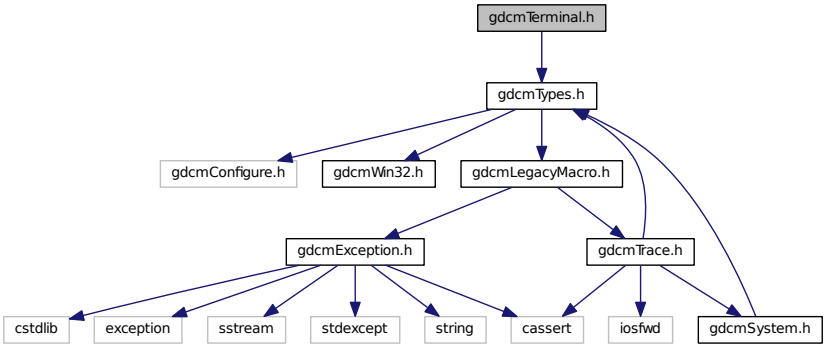
Functions

- 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 Allow one to print in color in a shell.

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 }

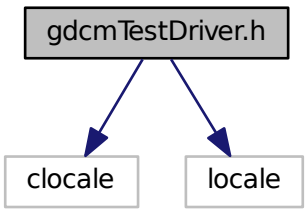
Functions

- 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>
```

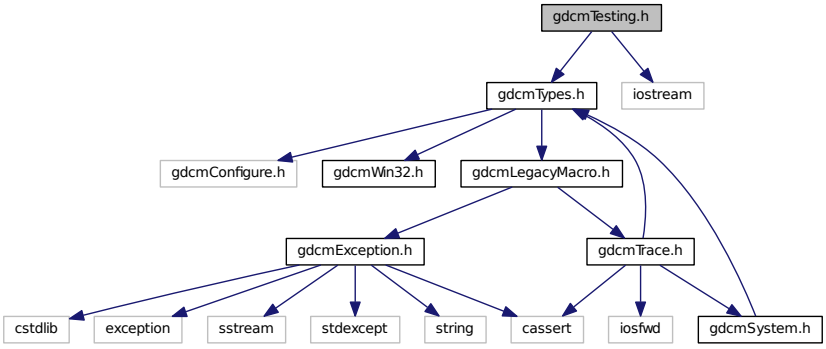
Include dependency graph for gdcmTestDriver.h:



11.253 gdcmTesting.h File Reference

```
#include "gdcmTypes.h"
#include <iostream>
```

Include dependency graph for gdcmTesting.h:



Classes

- class gdcm::Testing

  class for testing

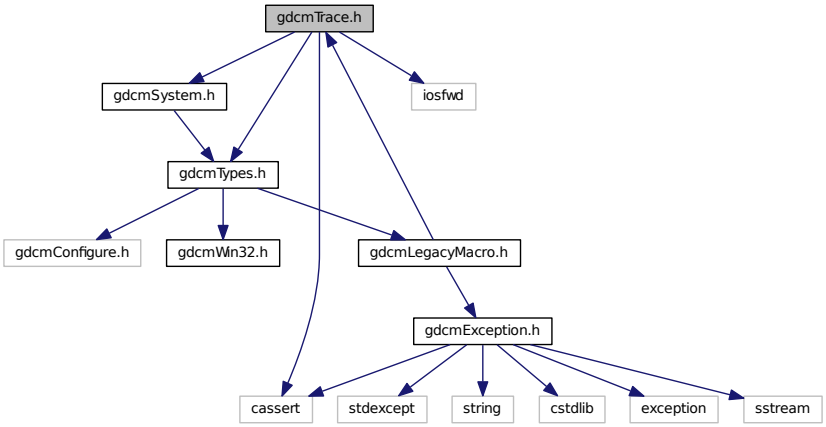
Namespaces

- gdcm

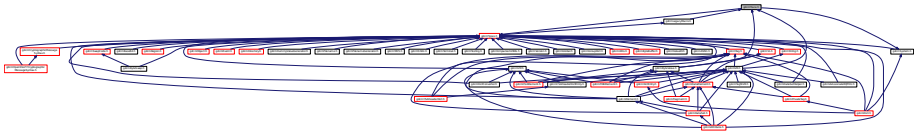
11.254 gdcmTrace.h File Reference

```
#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 #define GDCM_FUNCTION "<unknown>"

11.254.1.2 #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 #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 #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 #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 #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

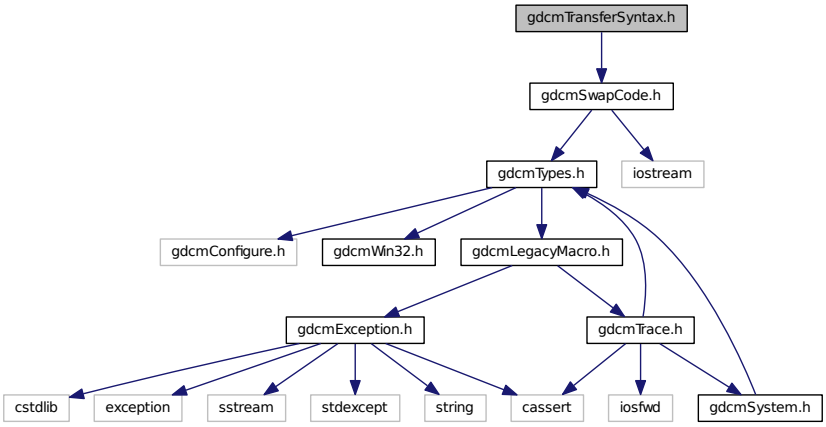
msg	message part

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().

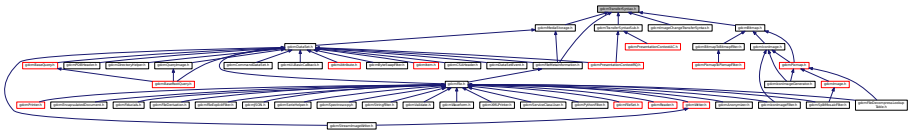
11.255 gdcmTransferSyntax.h File Reference

```
#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

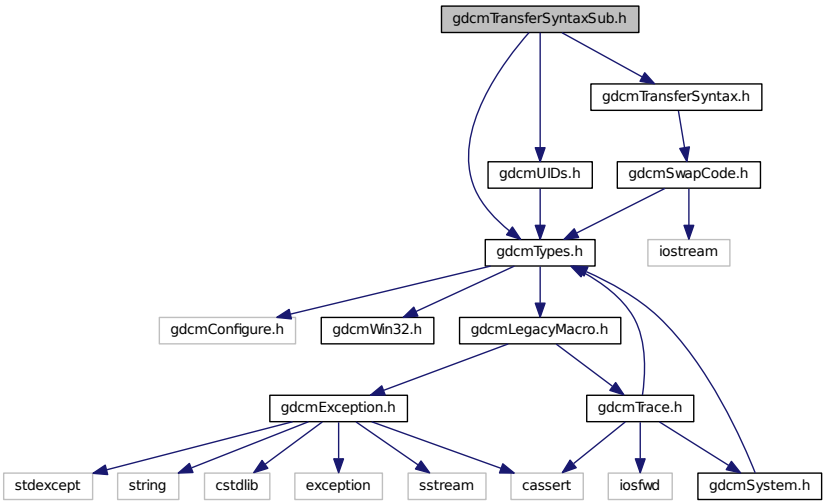
Functions

- std::ostream & gdcm::operator<< (std::ostream &_os, const TransferSyntax &ts)

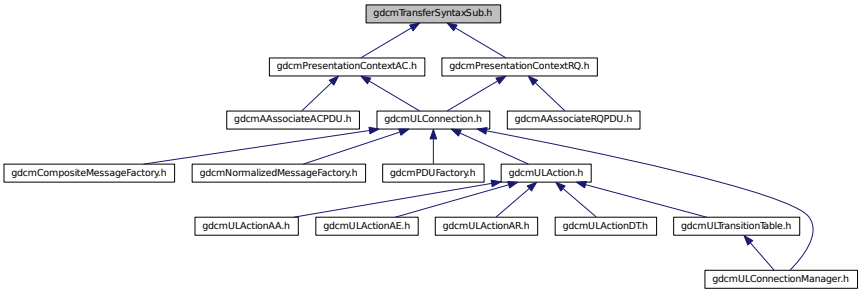
11.256 gdcmTransferSyntaxSub.h File Reference

```
#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 Table 9-15 TRANSFER SYNTAX SUB-ITEM FIELDS.

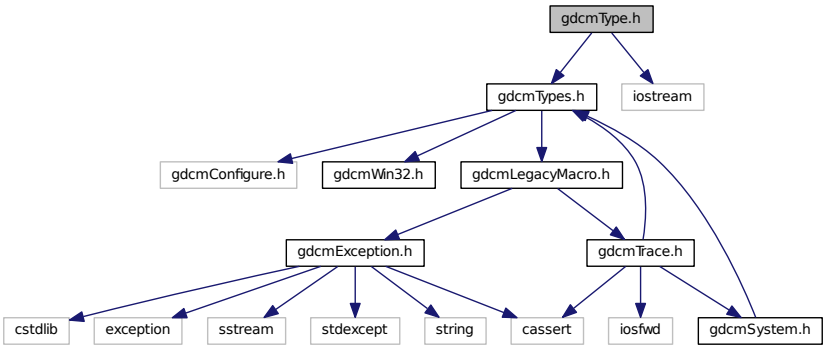
Namespaces

- gdcm
- gdcm::network

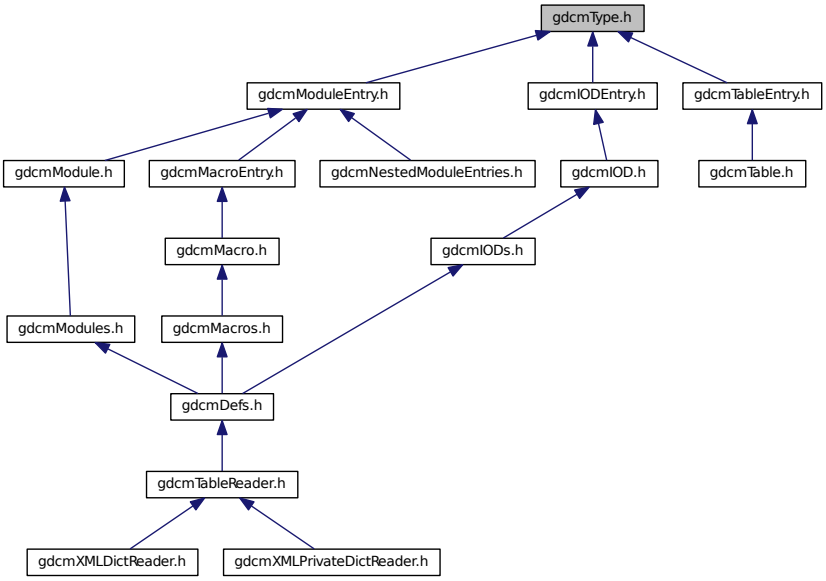
11.257 gdcmType.h File Reference

```
#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

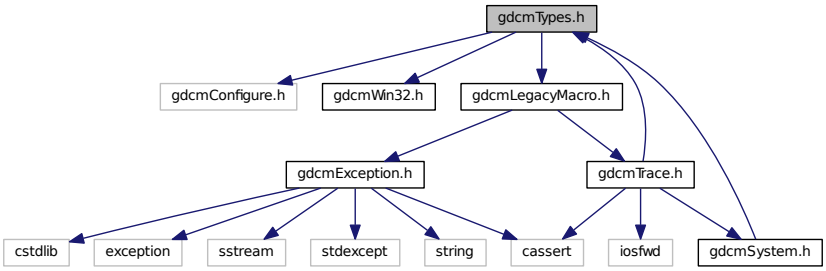
Functions

- 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"
```

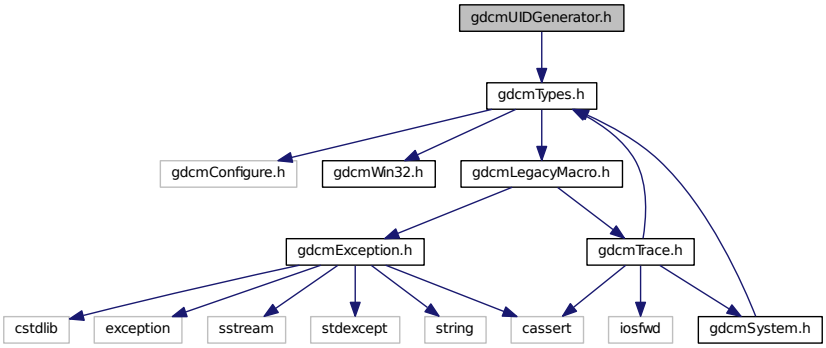
Include dependency graph for gdcmTypes.h:



11.259 gdcmUIDGenerator.h File Reference

```
#include "gdcmTypes.h"
```

Include dependency graph for gdcmUIDGenerator.h:



Classes

- class gdcm::UIDGenerator

  Class for generating unique UID.

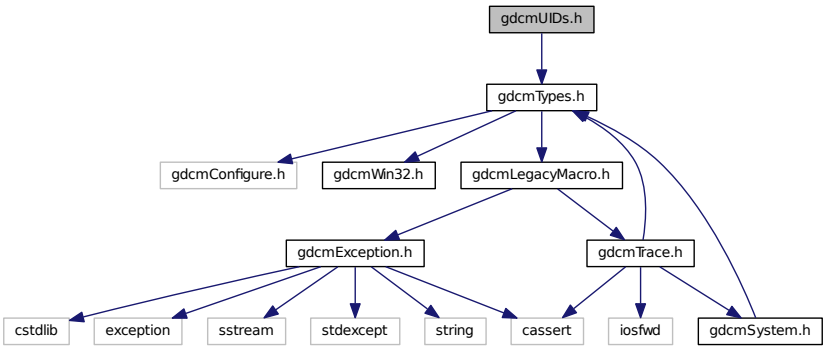
Namespaces

- 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

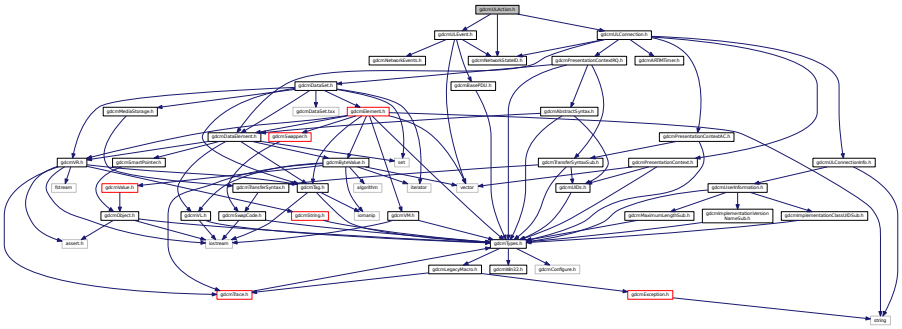
Functions

- std::ostream & gdcm::operator<< (std::ostream &_os, const UIDs &uid)

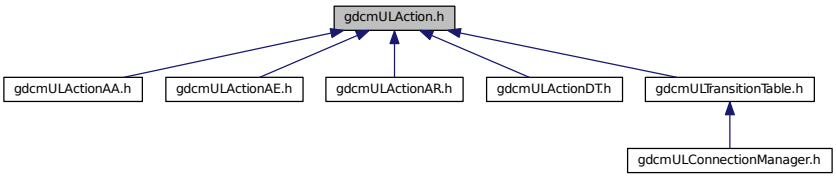
11.261 gdcmULAction.h File Reference

```
#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 A ULConnection in a given ULState can perform certain ULActions. This base class provides the interface for running those ULActions on a given ULConnection.

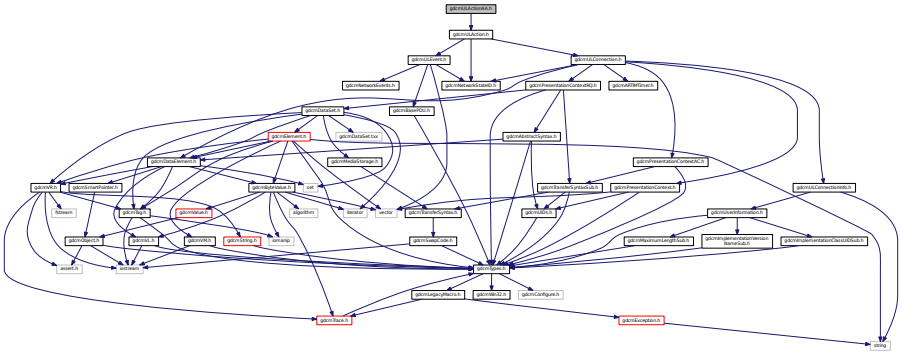
Namespaces

- 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

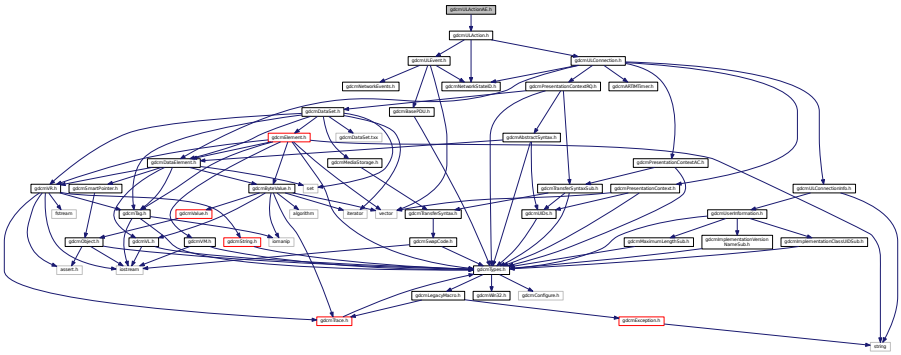
Namespaces

- 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

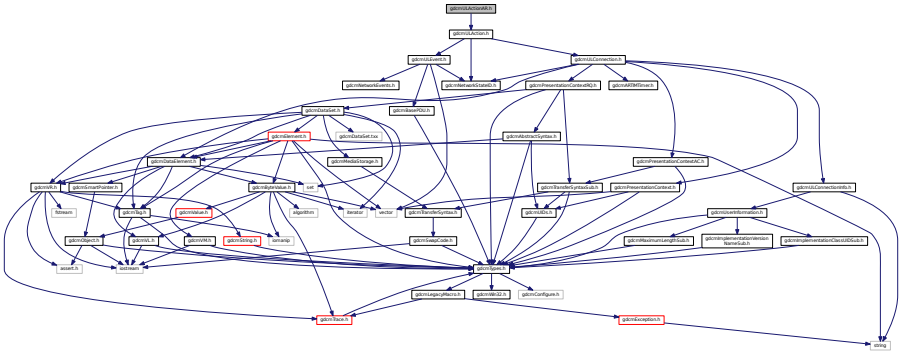
Namespaces

- 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

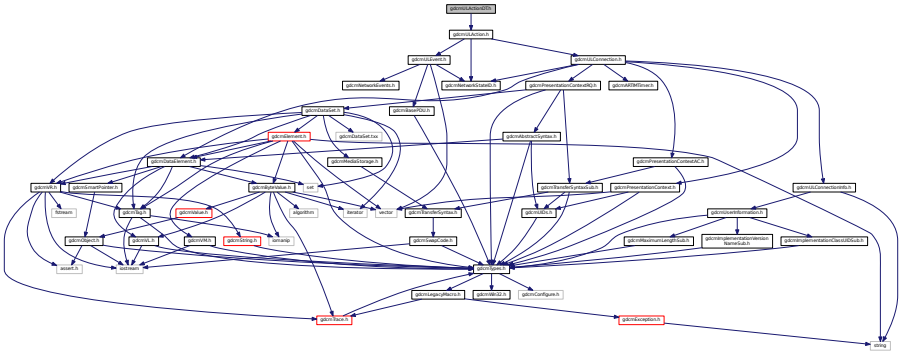
Namespaces

- 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

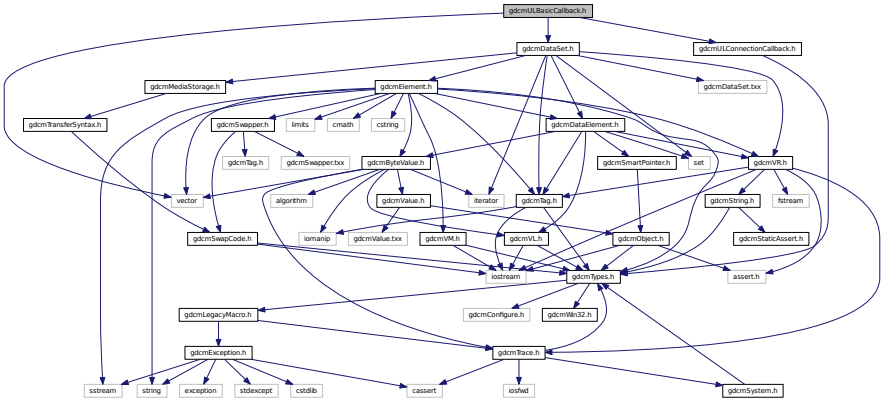
Namespaces

- 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 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.

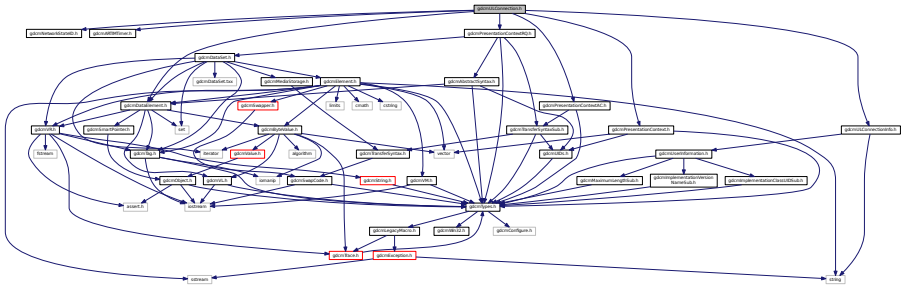
Namespaces

- gdcm
- gdcm::network

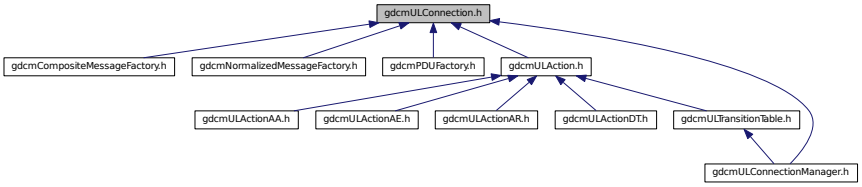
11.267 gdcmULConnection.h File Reference

```
#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 This is the class that contains the socket to another machine, and passes data through itself, as well as maintaining a sense of state.

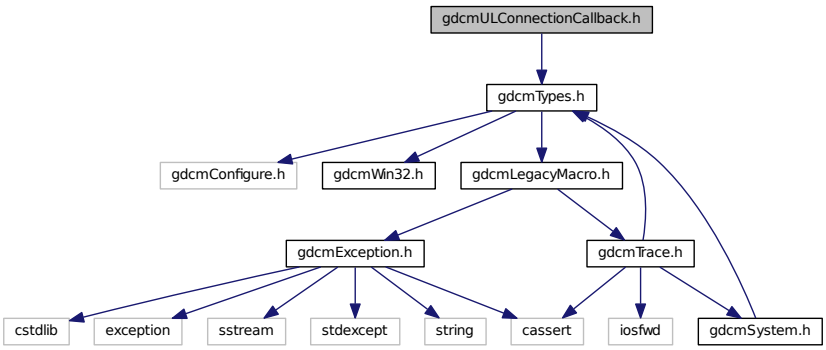
Namespaces

- gdcm
- gdcm::network

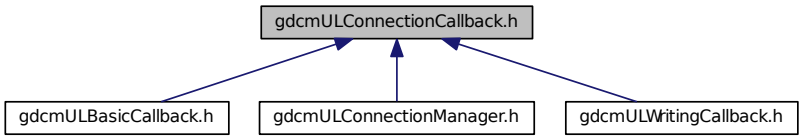
11.268 gdcmULConnectionCallback.h File Reference

```
#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

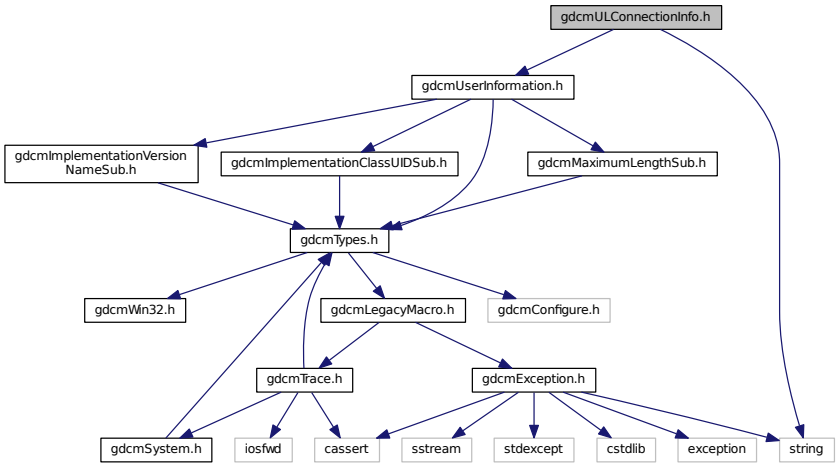
Namespaces

- gdcm
- gdcm::network

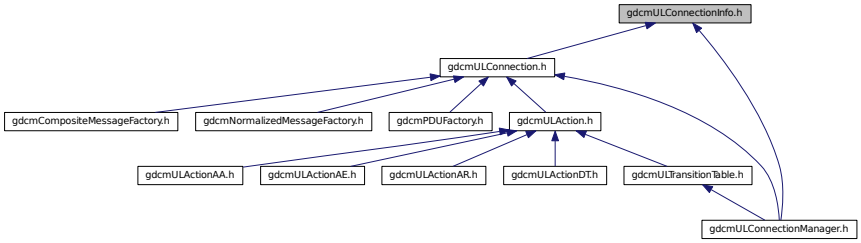
11.269 gdcmULConnectionInfo.h File Reference

```
#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 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.

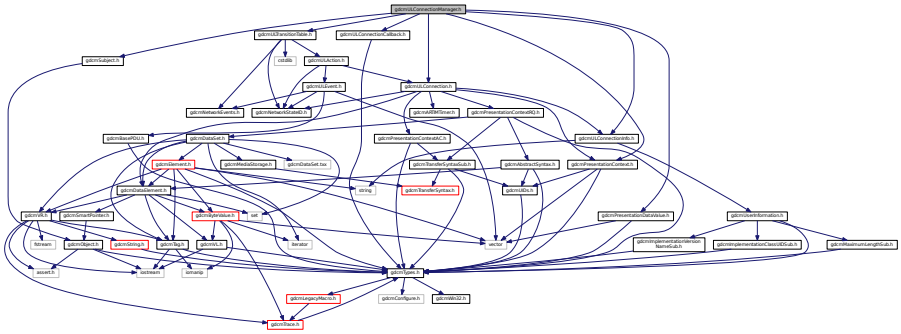
Namespaces

- 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 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).

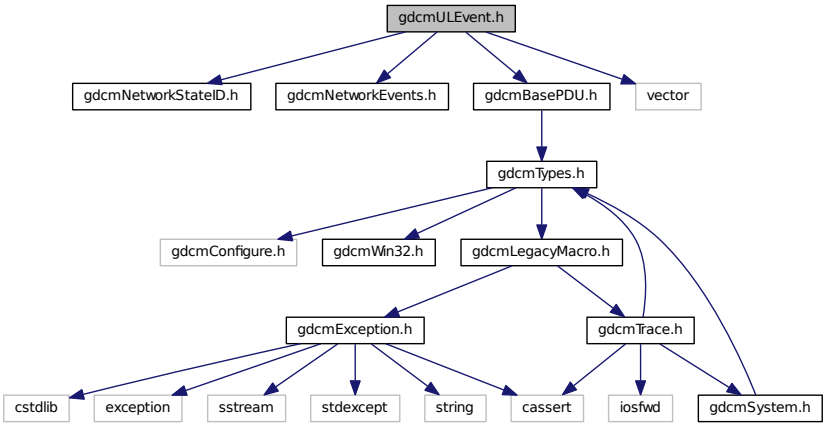
Namespaces

- gdcm
- gdcm::network

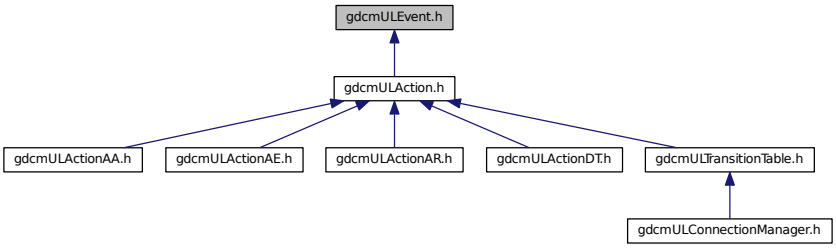
11.271 gdcmULEvent.h File Reference

```
#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 base class for network events.

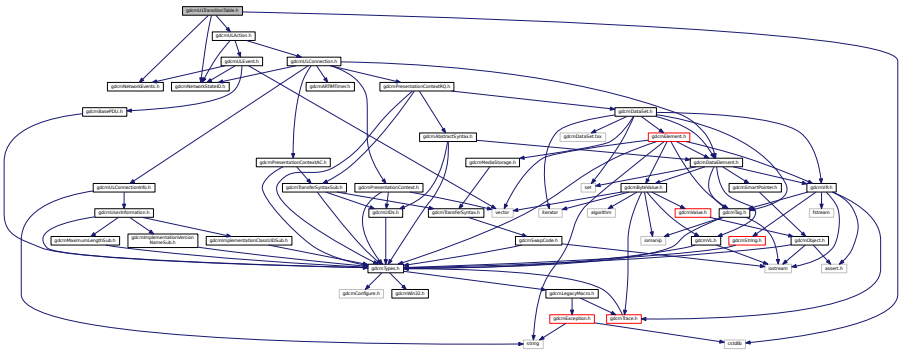
Namespaces

- gdcm
- gdcm::network

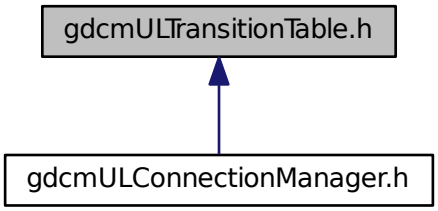
11.272 gdcmULTransitionTable.h File Reference

```
#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

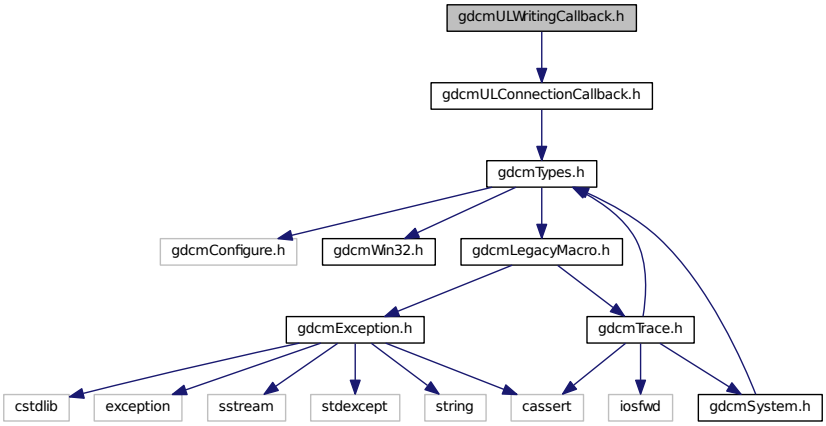
  ULTransitionTable The transition table of all the ULEvents, new ULActions, and ULStates.

Namespaces

- gdcm
- gdcm::network

11.273 gdcmULWritingCallback.h File Reference

```
#include "gdcmULConnectionCallback.h"
```
Include dependency graph for gdcmULWritingCallback.h:



Classes

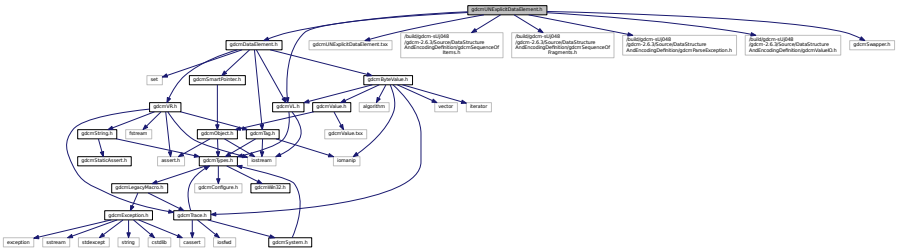
- class gdcm::network::ULWritingCallback

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.

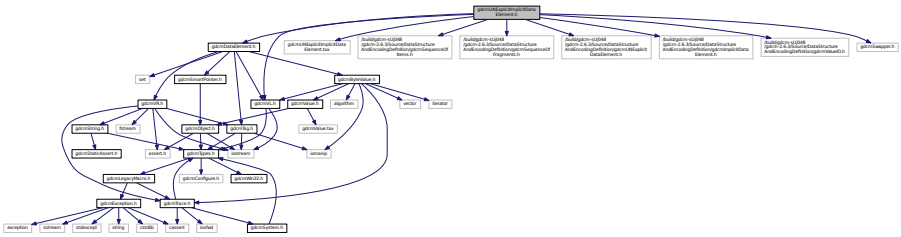
Namespaces

- 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 This class gather two known bugs:

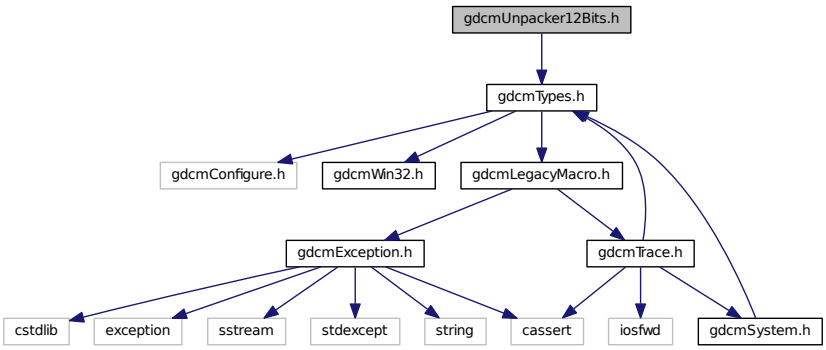
Namespaces

- 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.

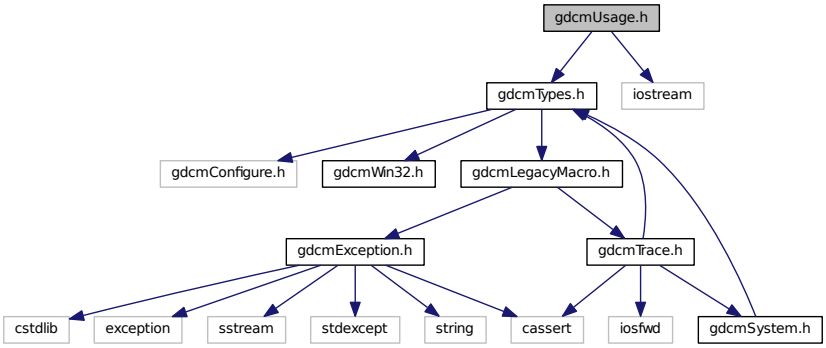
Namespaces

- gdcm

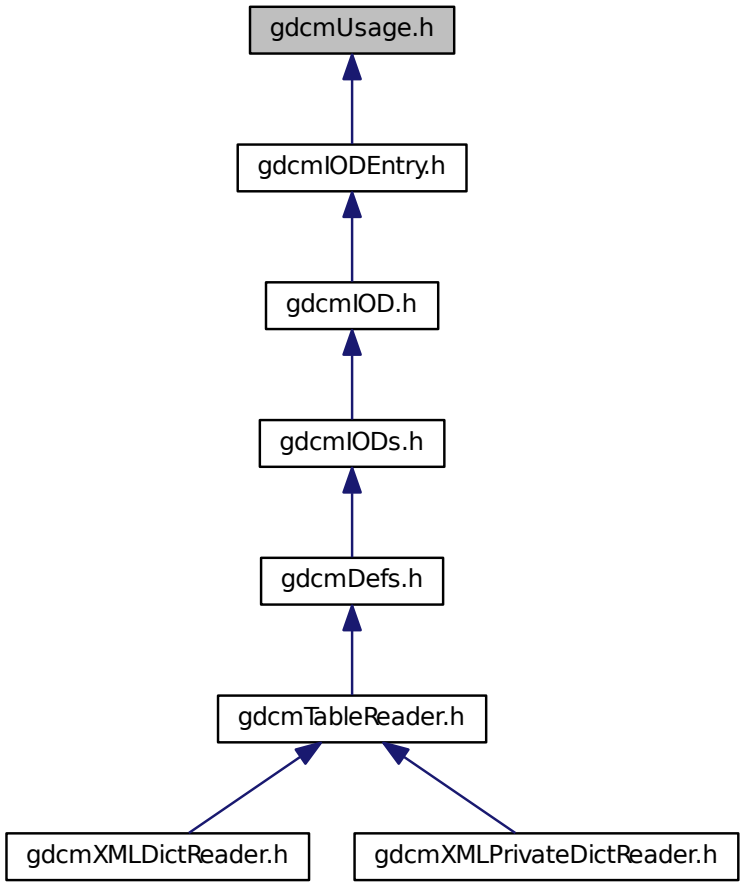
11.277 gdcmUsage.h File Reference

```
#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

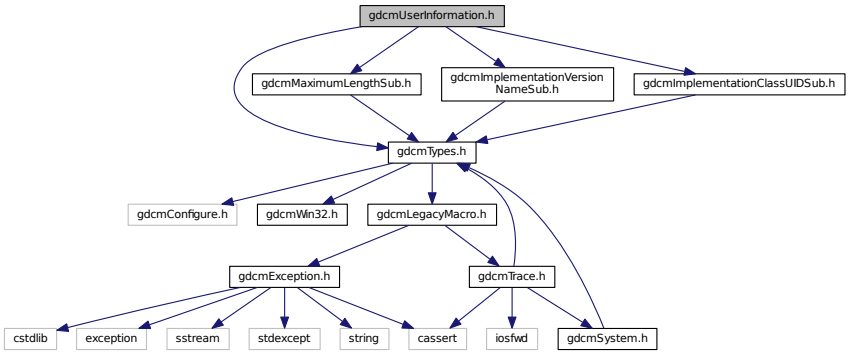
Functions

- std::ostream & gdcm::operator<< (std::ostream &_os, const Usage &val)

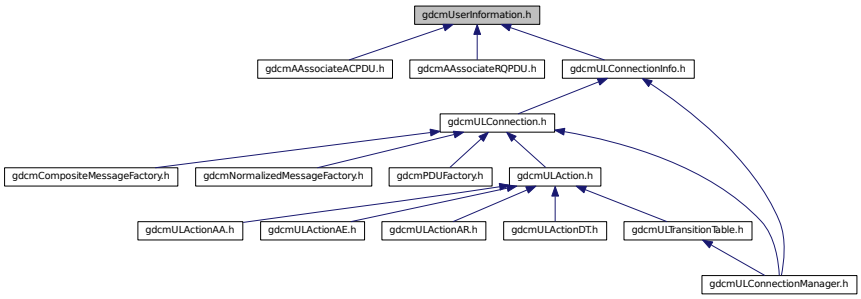
11.278 gdcmUserInformation.h File Reference

```
#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 Table 9-16 USER INFORMATION ITEM FIELDS.

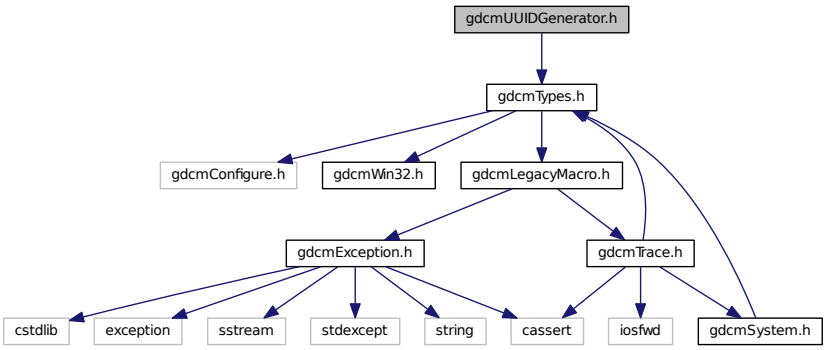
Namespaces

- 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 generate DCE 1.1 uid.

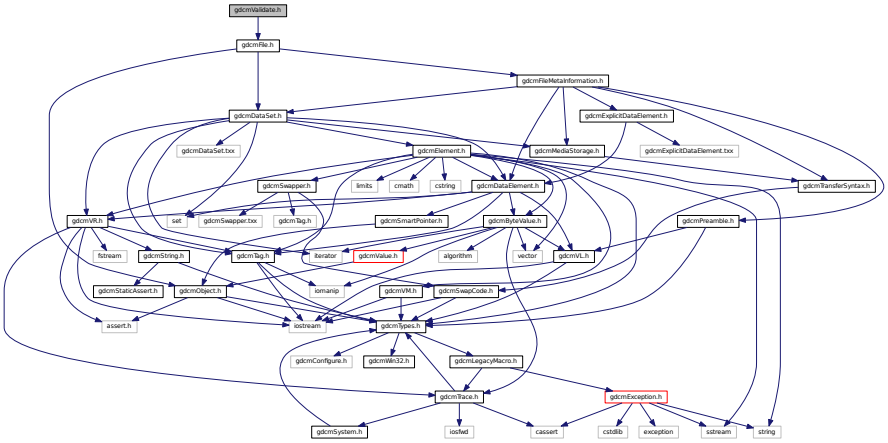
Namespaces

- gdcm

11.280 gdcmValidate.h File Reference

```
#include "gdcmFile.h"
```

Include dependency graph for gdcmValidate.h:



Classes

- class gdcm::Validate

  Validate class.

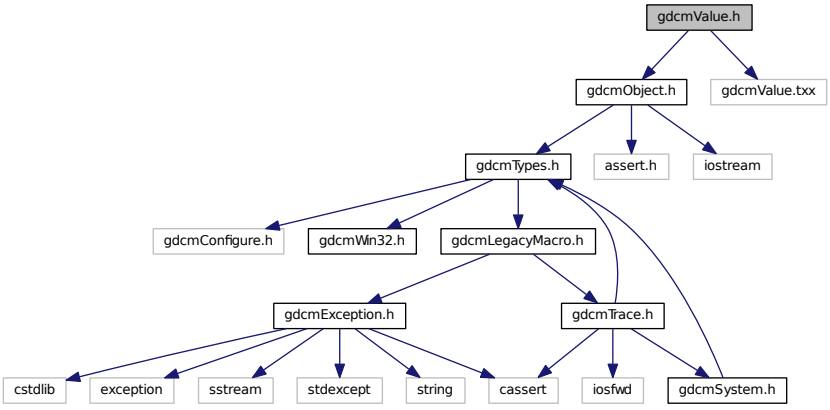
Namespaces

- gdcm

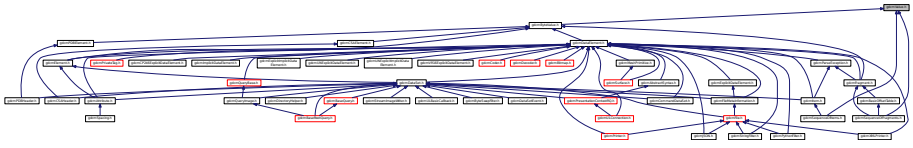
11.281 gdcmValue.h File Reference

```
#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.

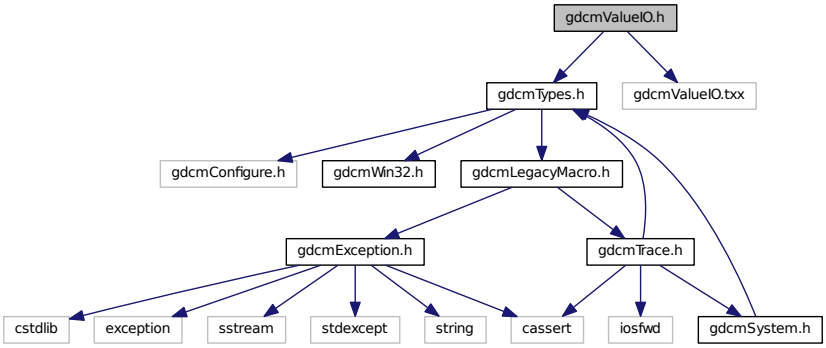
Namespaces

- 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.

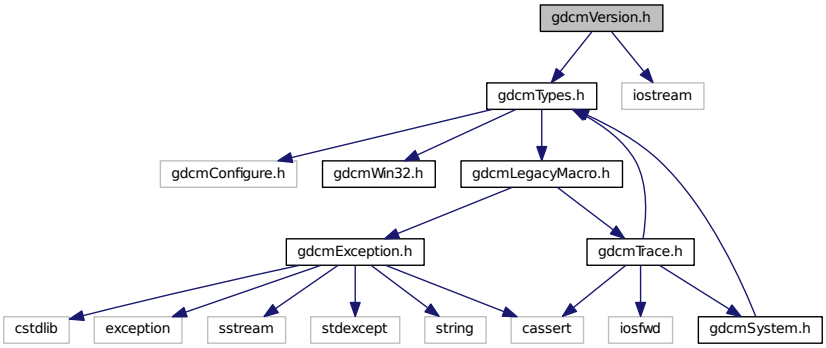
Namespaces

- 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

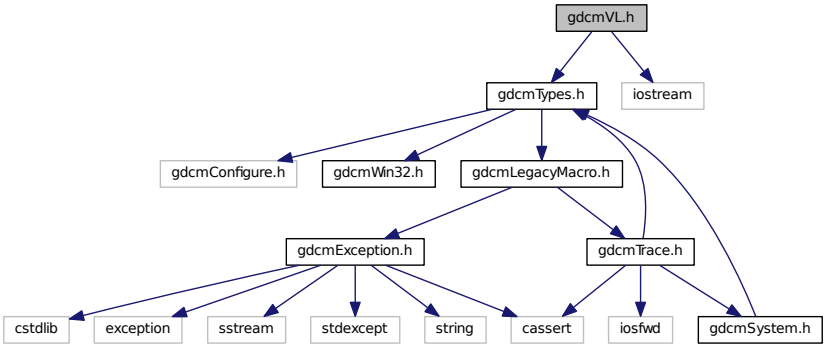
Functions

- std::ostream & gdcm::operator<< (std::ostream &os, const Version &v)

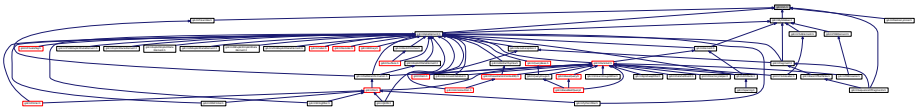
11.284 gdcmVL.h File Reference

```
#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

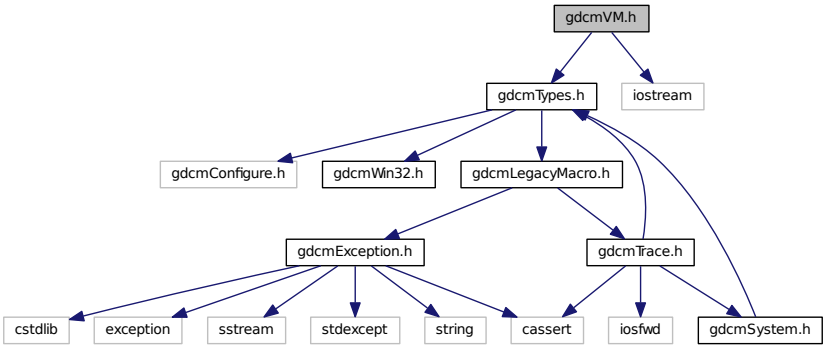
Functions

- 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 #define TYPETOLENGTH(*type, length*)

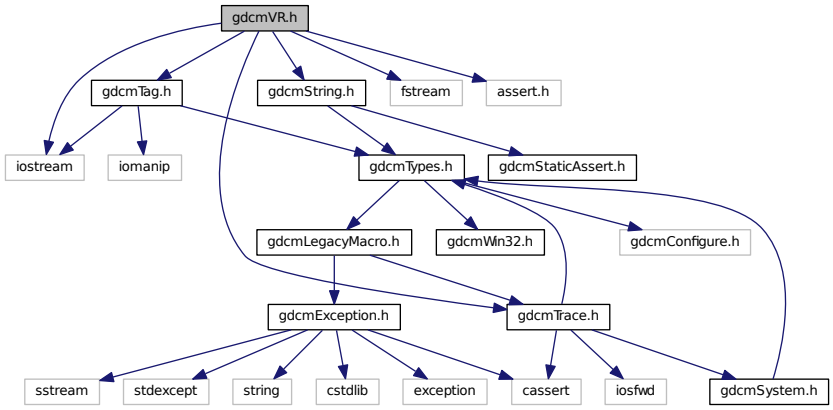
Value:

```
template<> struct VMToLength<VM::type> \
  { enum { Length = length }; };
```

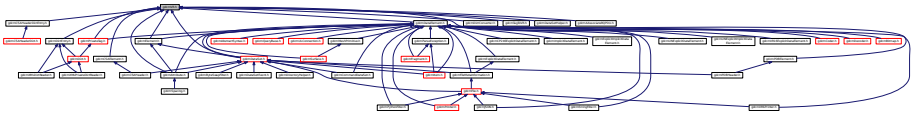
11.286 gdcmVR.h File Reference

```
#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 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.
- 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 #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 #define VRTypeTemplateCase(*type*)

Value:

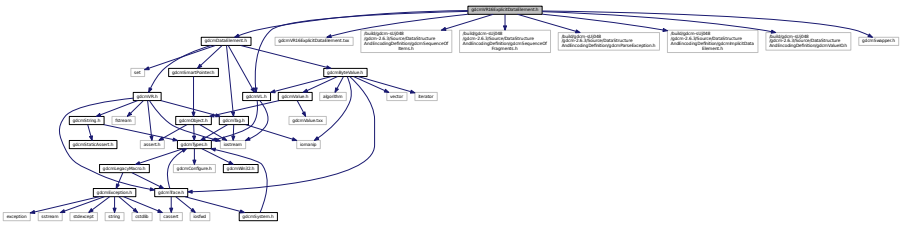
```
case VR::type: \
    return sizeof ( VRToType<VR::type>::Type );
```

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.

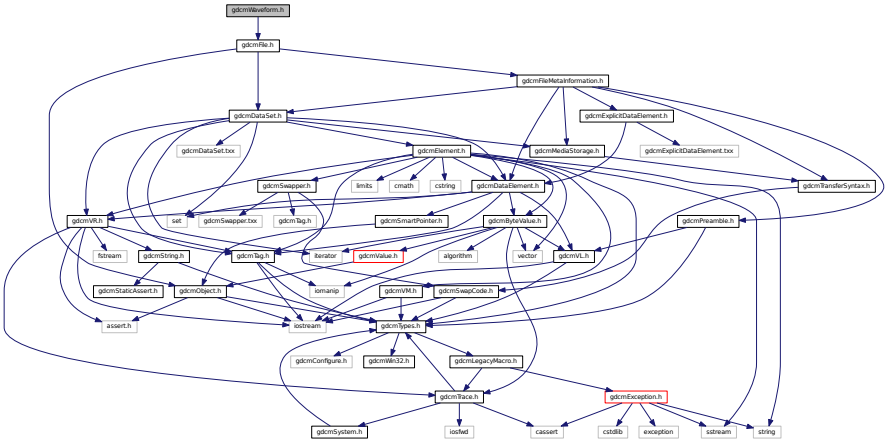
Namespaces

- 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

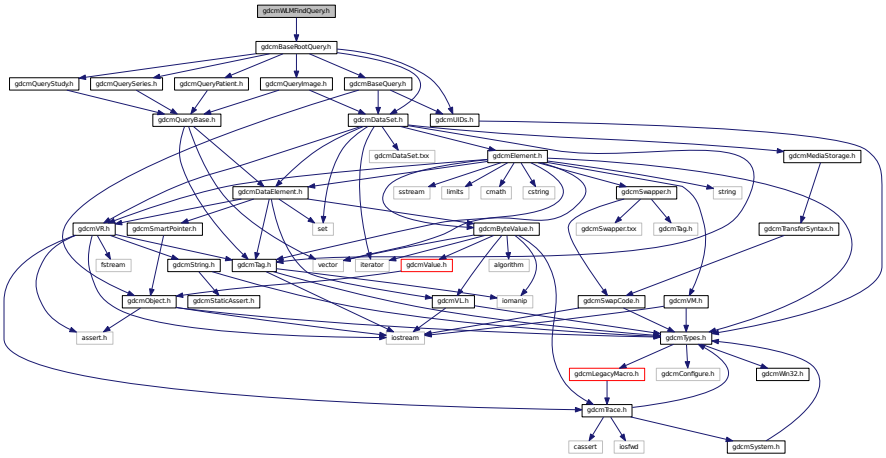
11.289.1 Macro Definition Documentation

11.289.1.1 #define GDCM_EXPORT

11.290 gdcmWLMFindQuery.h File Reference

```
#include "gdcmBaseRootQuery.h"
```

Include dependency graph for gdcmWLMFindQuery.h:



Classes

- class gdcm::WLMFindQuery

  PatientRootQuery contains: the class which will produce a dataset for c-find with patient root.

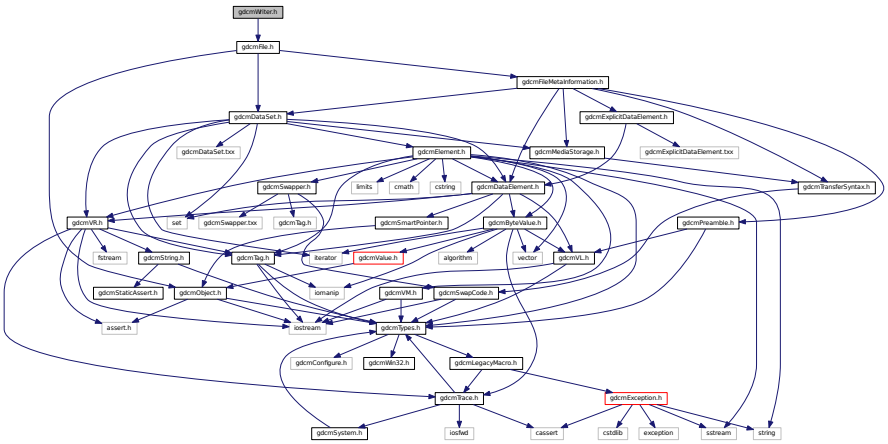
Namespaces

- gdcm

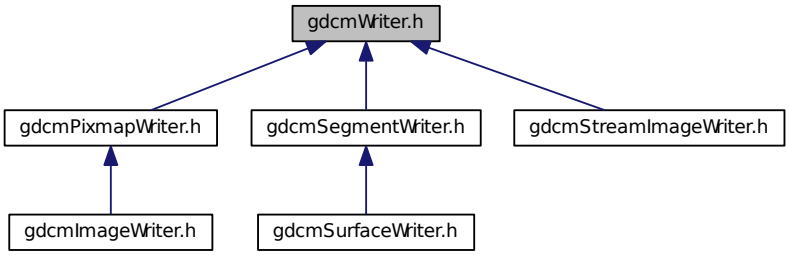
11.291 gdcmWriter.h File Reference

```
#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) This class is a non-validating writer, it will only performs well- formedness check only.

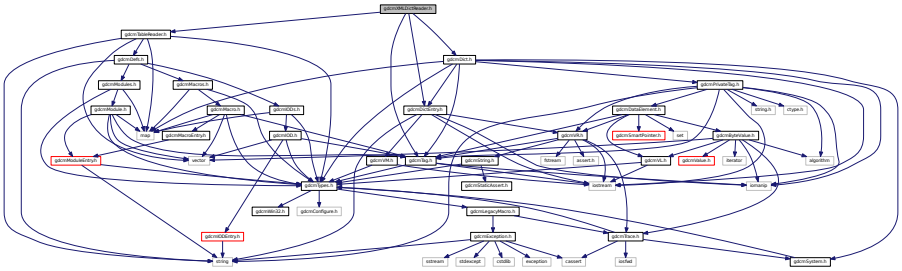
Namespaces

- 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.

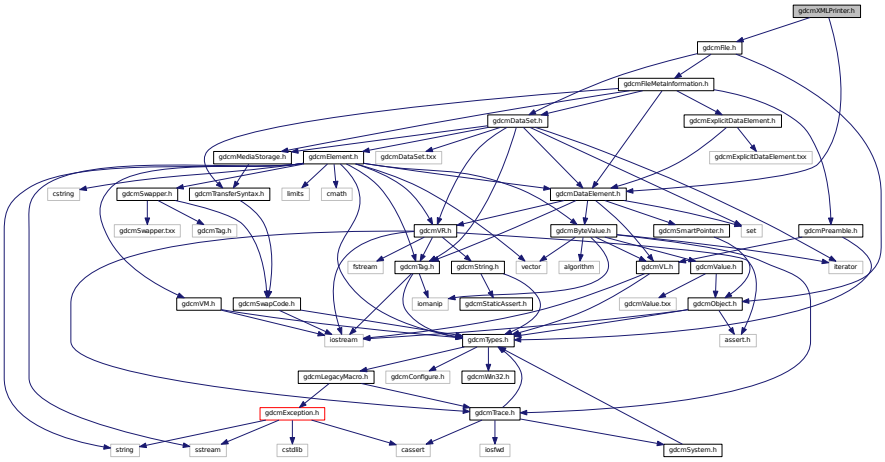
Namespaces

- gdcm

11.293 gdcmXMLPrinter.h File Reference

```
#include "gdcmFile.h"
#include "gdcmDataElement.h"
```

Include dependency graph for gdcmXMLPrinter.h:



Classes

- class gdcm::XMLPrinter

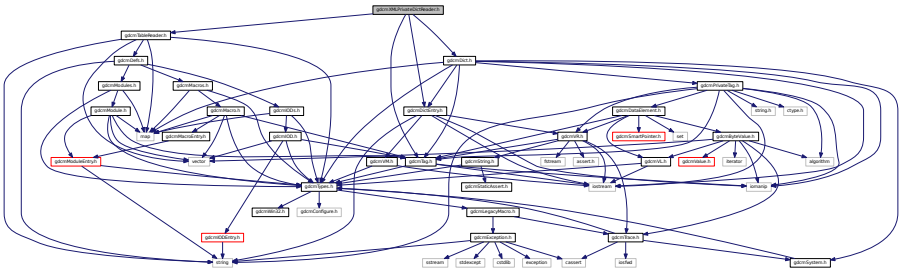
Namespaces

- 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

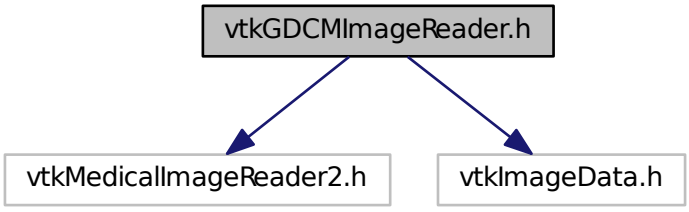
11.295 README.txt File Reference

11.296 TestsList.txt File Reference

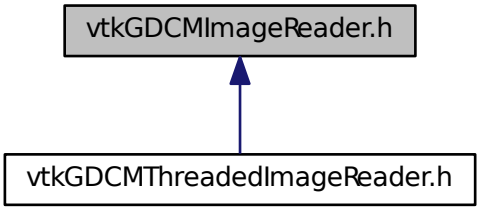
11.297 vtkGDCMImageReader.h File Reference

```
#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 #define VTK_CMYK 8

11.297.1.2 #define VTK_INVERSE_LUMINANCE 5

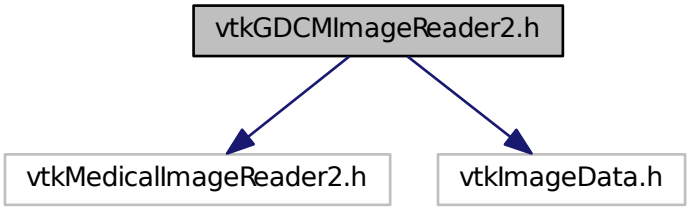
11.297.1.3 #define VTK_LOOKUP_TABLE 6

11.297.1.4 #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 #define VTK_CMYK 8

11.298.1.2 #define VTK_INVERSE_LUMINANCE 5

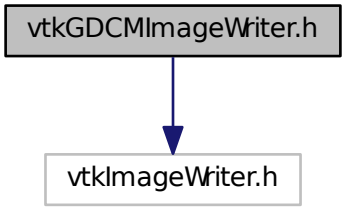
11.298.1.3 #define VTK_LOOKUP_TABLE 6

11.298.1.4 #define VTK_YBR 7

11.299 vtkGDCMImageWriter.h File Reference

```
#include "vtkImageWriter.h"
```

Include dependency graph for vtkGDCMImageWriter.h:

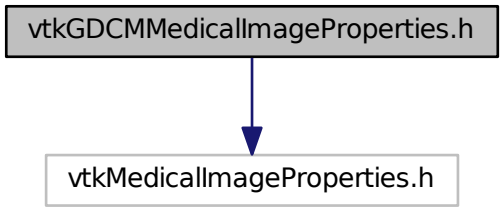


Classes

- class vtkGDCMImageWriter

11.300 vtkGDCMMedicalImageProperties.h File Reference

```
#include "vtkMedicalImageProperties.h"
```
Include dependency graph for vtkGDCMMedicalImageProperties.h:



Classes

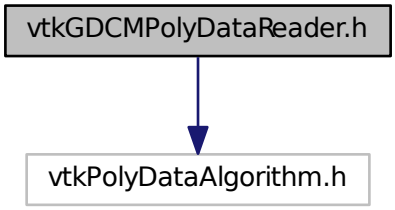
- class vtkGDCMMedicalImageProperties

Namespaces

- gdcm

11.301 vtkGDCMPolyDataReader.h File Reference

```
#include "vtkPolyDataAlgorithm.h"
```
Include dependency graph for vtkGDCMPolyDataReader.h:



Classes

- class vtkGDCMPolyDataReader

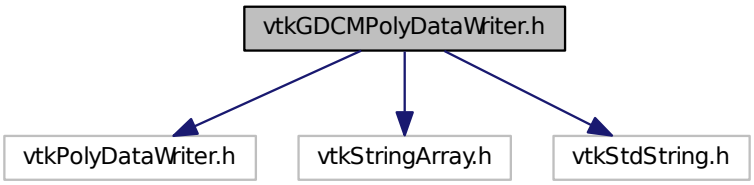
Namespaces

- 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

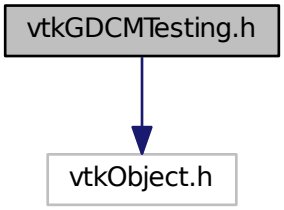
Namespaces

- gdcm

11.303 vtkGDCMTesting.h File Reference

```
#include "vtkObject.h"
```

Include dependency graph for vtkGDCMTesting.h:



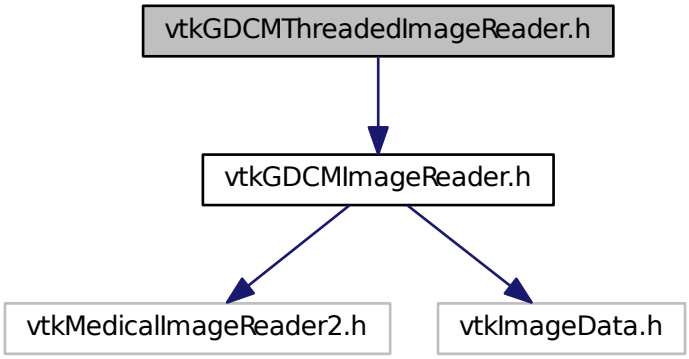
Classes

- class vtkGDCMTesting

11.304 vtkGDCMThreadedImageReader.h File Reference

```
#include "vtkGDCMImageReader.h"
```

Include dependency graph for vtkGDCMThreadedImageReader.h:

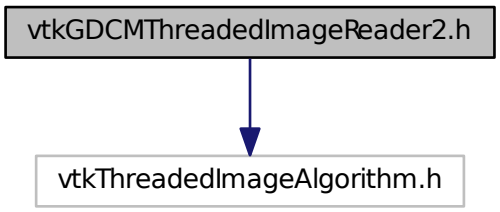


Classes

- class vtkGDCMThreadedImageReader

11.305 vtkGDCMThreadedImageReader2.h File Reference

```
#include "vtkThreadedImageAlgorithm.h"
```
Include dependency graph for vtkGDCMThreadedImageReader2.h:

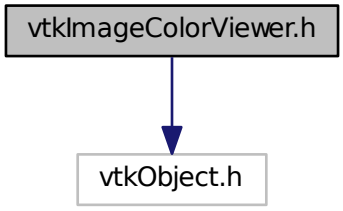


Classes

- class vtkGDCMThreadedImageReader2

11.306 vtkImageColorViewer.h File Reference

```
#include "vtkObject.h"
```
Include dependency graph for vtkImageColorViewer.h:

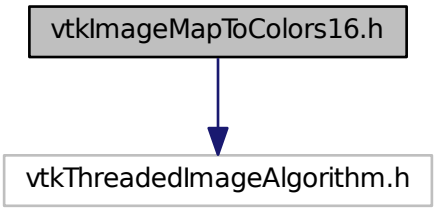


Classes

- class vtkImageColorViewer

11.307 vtkImageMapToColors16.h File Reference

```
#include "vtkThreadedImageAlgorithm.h"
```
Include dependency graph for vtkImageMapToColors16.h:

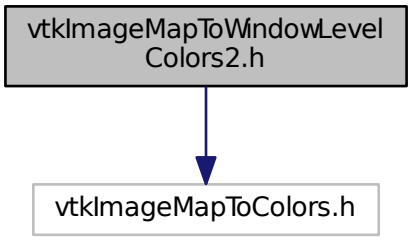


Classes

- class vtkImageMapToColors16

11.308 vtkImageMapToWindowLevelColors2.h File Reference

```
#include "vtkImageMapToColors.h"
```
Include dependency graph for vtkImageMapToWindowLevelColors2.h:



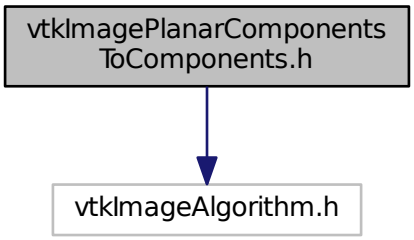
Classes

- class vtkImageMapToWindowLevelColors2

11.309 vtkImagePlanarComponentsToComponents.h File Reference

```
#include "vtkImageAlgorithm.h"
```

Include dependency graph for vtkImagePlanarComponentsToComponents.h:



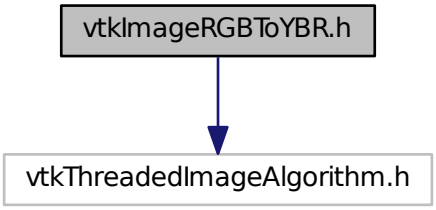
Classes

- class vtkImagePlanarComponentsToComponents

11.310 vtkImageRGBToYBR.h File Reference

```
#include "vtkThreadedImageAlgorithm.h"
```

Include dependency graph for vtkImageRGBToYBR.h:



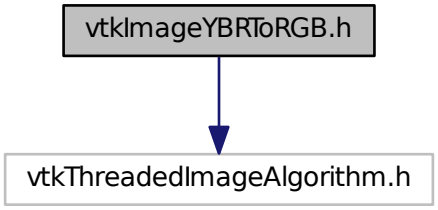
Classes

- class vtkImageRGBToYBR

11.311 vtkImageYBRToRGB.h File Reference

```
#include "vtkThreadedImageAlgorithm.h"
```

Include dependency graph for vtkImageYBRToRGB.h:



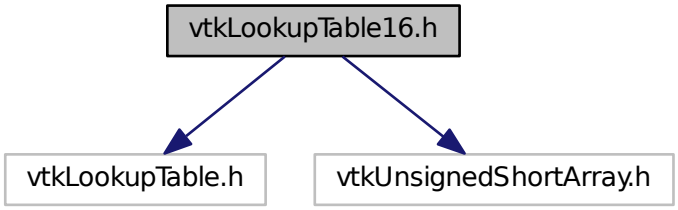
Classes

- class vtkImageYBRToRGB

11.312 vtkLookupTable16.h File Reference

```
#include "vtkLookupTable.h"
#include "vtkUnsignedShortArray.h"
```

Include dependency graph for vtkLookupTable16.h:



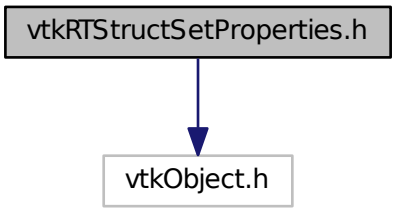
Classes

- 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 ############################################################################
2 #
3 #  Program: GDCM (Grassroots DICOM). A DICOM library
4 #
5 #  Copyright (c) 2006-2011 Mathieu Malaterre
6 #  All rights reserved.
7 #  See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.
8 #
9 #     This software is distributed WITHOUT ANY WARRANTY; without even
10 #     the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
11 #     PURPOSE.  See the above copyright notice for more information.
12 #
13 ############################################################################
14
15 """
16 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"

#ifndef vtkFloatingPointType
#define vtkFloatingPointType double
#endif

/*
 * Simple example to check computation of spacing within vtkGDCMImageReader2
 * This is a direct implementation of:
 *
 * http://gdcm.sourceforge.net/wiki/index.php/
     Using_GDCM_API#Automatic_ordering_of_slices_for_vtkGDCMImageReader.SetFileNames
 *
 * For more advanced information on how 3D spacing is being computed see:
 *
 *  - http://gdcm.sourceforge.net/html/classgdcm_1_1IPPSorter.html
 *
 * Usage:
 *
 * $ Compute3DSpacing SIEMENS_MAGNETOM-12-MONO2-FileSeq0.dcm \
 *     SIEMENS_MAGNETOM-12-MONO2-FileSeq1.dcm \
 *     SIEMENS_MAGNETOM-12-MONO2-FileSeq2.dcm \
 *     SIEMENS_MAGNETOM-12-MONO2-FileSeq3.dcm
 */

int main(int argc, char *argv[])
{
  if( argc < 2 ) return 1;

  std::vector<std::string> filenames;
  for( int  i = 1; i < argc; ++i )
    {
    filenames.push_back( argv[i] );
    }

  gdcm::IPPSorter s;
  s.SetComputeZSpacing( true );
  s.SetZSpacingTolerance( 1e-3 );
  bool b = s.Sort( filenames );
  if( !b )
    {
    std::cerr << "Failed to sort files" << std::endl;
    return 1;
    }
  std::cout << "Sorting succeeded:" << std::endl;
  //s.Print( std::cout );

  std::cout << "Found z-spacing:" << std::endl;
  std::cout << s.GetZSpacing() << std::endl;
  const double ippzspacing = s.GetZSpacing();

  const std::vector<std::string> & sorted = s.GetFilenames();
  vtkGDCMImageReader2 * reader = vtkGDCMImageReader2::New();
  vtkStringArray *files = vtkStringArray::New();
  std::vector< std::string >::const_iterator it = sorted.begin();
  for( ; it != sorted.end(); ++it)
    {
    const std::string &f = *it;
    files->InsertNextValue( f.c_str() );
    }
  reader->SetFileNames( files );
  reader->Update();

  const vtkFloatingPointType *spacing = reader->GetOutput()->GetSpacing();
  vtkImageChangeInformation *v16 = vtkImageChangeInformation::New();
#if (VTK_MAJOR_VERSION >= 6)
  v16->SetInputConnection( reader->GetOutputPort() );
#else
  v16->SetInput( reader->GetOutput() );
#endif
  v16->SetOutputSpacing( spacing[0], spacing[1], ippzspacing );
  v16->Update();

  v16->GetOutput()->Print( std::cout );

  return 0;
}
```

12.13 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 ##########################################################################
2 #
3 #  Program: GDCM (Grassroots DICOM). A DICOM library
4 #
5 #  Copyright (c) 2006-2011 Mathieu Malaterre
```

```
6 #  All rights reserved.
7 #  See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.
8 #
9 #     This software is distributed WITHOUT ANY WARRANTY; without even
10 #     the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
11 #     PURPOSE.  See the above copyright notice for more information.
12 #
13 ##########################################################################
14
15 """
16 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     ## use float for accurate scaling
63     result = numpy.frombuffer(gdcm_array, dtype=dtype).astype(float)
64     ## optional gamma scaling
65     #maxV = float(result[result.argmax()])
66     #result = result + .5*(maxV-result)
67     #result = numpy.log(result+50) ## apprx background level
68     result.shape = d
69     return result
70
71 if __name__ == "__main__":
72   import sys
73   r = gdcm.ImageReader()
74   filename = sys.argv[1]
75   r.SetFileName( filename )
76   if not r.Read():  sys.exit(1)
77   numpy_array = gdcm_to_numpy( r.GetImage() )
78
79   subplot(111)# one plot, on left
80   title(filename)
81   ## many colormaps are available
82   imshow(numpy_array, interpolation='bilinear', cmap=cm.jet)
83   ## set the plot sizes and placement
84   subplots_adjust(bottom=0.1, right=0.8, top=0.9)
85   cax = axes([0.85, 0.1, 0.075, 0.8])
86   colorbar(cax=cax)
```

```
87   title('values')
88   get_current_fig_manager().window.title('plot')
89   show()
```

12.15 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 ############################################################################
2 #
3 #  Program: GDCM (Grassroots DICOM). A DICOM library
4 #
5 #  Copyright (c) 2006-2011 Mathieu Malaterre
6 #  All rights reserved.
7 #  See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.
8 #
9 #     This software is distributed WITHOUT ANY WARRANTY; without even
10 #     the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
11 #     PURPOSE.  See the above copyright notice for more information.
12 #
13 ############################################################################
14
15 """
16 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 ##########################################################################
2 #
3 #  Program: GDCM (Grassroots DICOM). A DICOM library
4 #
5 #  Copyright (c) 2006-2011 Mathieu Malaterre
6 #  All rights reserved.
7 #  See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.
8 #
9 #     This software is distributed WITHOUT ANY WARRANTY; without even
10 #     the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
11 #     PURPOSE.  See the above copyright notice for more information.
12 #
13 ##########################################################################
14
15 """
16 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      ## linear gamma adjust
65      #result = result + .5*(maxV-result)
66      ## log gamma
67      result = numpy.log(result+50) ## 50 is apprx background level
68      maxV = float(result[result.argmax()])
69      result = result*(2.**8/maxV) ## histogram stretch
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    ## L is 8 bit grey
81    ## http://www.pythonware.com/library/pil/handbook/concepts.htm
82    pilImage = Image.frombuffer('L',
83                                numpy_array.shape,
84                                numpy_array.astype(numpy.uint8),
85                                'raw','L',0,1)
86    ## cutoff removes background noise and spikes
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 ##########################################################################
2 #
3 #  Program: GDCM (Grassroots DICOM). A DICOM library
4 #
5 #  Copyright (c) 2006-2011 Mathieu Malaterre
```

```
6 #  All rights reserved.
7 #  See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.
8 #
9 #     This software is distributed WITHOUT ANY WARRANTY; without even
10 #     the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
11 #     PURPOSE.  See the above copyright notice for more information.
12 #
13 ##########################################################################
14
15 """
16     <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 ############################################################################
2 #
3 #  Program: GDCM (Grassroots DICOM). A DICOM library
4 #
5 #  Copyright (c) 2006-2011 Mathieu Malaterre
6 #  All rights reserved.
7 #  See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.
8 #
9 #     This software is distributed WITHOUT ANY WARRANTY; without even
10 #     the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
11 #     PURPOSE.  See the above copyright notice for more information.
12 #
13 ############################################################################
14
15 """
16 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 ##########################################################################
2 #
3 #  Program: GDCM (Grassroots DICOM). A DICOM library
4 #
5 #  Copyright (c) 2006-2011 Mathieu Malaterre
6 #  All rights reserved.
7 #  See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.
8 #
9 #     This software is distributed WITHOUT ANY WARRANTY; without even
10 #     the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
11 #     PURPOSE.  See the above copyright notice for more information.
12 #
13 ##########################################################################
14
15 """
16 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 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
      assert( isvalidpdfstring( fn.c_str() ) );
      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;
}
```

12.40 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;
}
```

12.41 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 );

  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 = 0.004166666839752674; // 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;
}
```

12.42 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, 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 = totalsize - offsets[r] - 16;
    else
      sourceLen = offsets[r+1] - 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, 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;
}
```

12.43 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;
}
```

12.44 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;
}
```

12.45 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;
}
```

12.46 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;
}
```

12.47 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;
}
```

12.48 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;
    }
}
```

12.49 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;
}
```

12.50 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;
}
```

12.51 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;
    }
}
```

12.52 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");
        }
      }

    }
}
```

12.53 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;
    }
}
```

12.54 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;
}
```

12.55 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;
    }
}
```

12.56 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;
}
```

12.57 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;
    }
}
```

12.58 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" );
    }
}
```

12.59 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;
    }
}
```

12.60 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;
    }
}
```

12.61 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;
    }
}
```

12.62 FindAllPatientName.py

```
1 ############################################################################
2 #
3 #  Program: GDCM (Grassroots DICOM). A DICOM library
4 #
5 #  Copyright (c) 2006-2011 Mathieu Malaterre
6 #  All rights reserved.
7 #  See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.
8 #
9 #     This software is distributed WITHOUT ANY WARRANTY; without even
10 #     the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
11 #     PURPOSE.  See the above copyright notice for more information.
12 #
13 ############################################################################
14 """
15 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]
```

12.63 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;
}
```

12.64 FixCommaBug.py

```
1 ############################################################################
2 #
3 #  Program: GDCM (Grassroots DICOM). A DICOM library
4 #
5 #  Copyright (c) 2006-2011 Mathieu Malaterre
6 #  All rights reserved.
7 #  See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.
8 #
9 #     This software is distributed WITHOUT ANY WARRANTY; without even
10 #     the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
11 #     PURPOSE.  See the above copyright notice for more information.
12 #
13 ############################################################################
14
15 """
16 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
```

12.65 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;
}
```

12.66 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;
}
```

12.67 gdcmorthoplanes.cxx

```
/*=========================================================================

  Program: GDCM (Grassroots DICOM). A DICOM library

  Copyright (c) 2006-2011 Mathieu Malaterre
  All rights reserved.
  See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.

     This software is distributed WITHOUT ANY WARRANTY; without even
     the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
     PURPOSE.  See the above copyright notice for more information.

=========================================================================*/

#include "vtkActor.h"
#include "vtkCamera.h"
#include "vtkMatrix4x4.h"
#include "vtkTransform.h"
#include "vtkAssembly.h"
#include "vtkCellPicker.h"
#include "vtkCommand.h"
#include "vtkImageActor.h"
#include "vtkImageMapToColors.h"
#include "vtkImageOrthoPlanes.h"
#include "vtkImagePlaneWidget.h"
#include "vtkImageReader.h"
#include "vtkInteractorEventRecorder.h"
#include "vtkLookupTable.h"
#include "vtkOutlineFilter.h"
#include "vtkPolyDataMapper.h"
#include "vtkProperty.h"
#include "vtkRenderWindow.h"
#include "vtkRenderWindowInteractor.h"
#include "vtkRenderer.h"
#include "vtkVolume16Reader.h"
#include "vtkImageData.h"
#include "vtkImageChangeInformation.h"
#include "vtkOrientationMarkerWidget.h"
#include "vtkAnnotatedCubeActor.h"
#include "vtkAxesActor.h"
#include "vtkCaptionActor2D.h"
#include "vtkTextProperty.h"
#include "vtkPropAssembly.h"

#include "vtkGDCMImageReader.h"
#include "vtkGDCMImageWriter.h"
#include "vtkStringArray.h"
```

```
#include "gdcmSystem.h"
#include "gdcmDirectory.h"
#include "gdcmIPPSorter.h"

#ifndef vtkFloatingPointType
#define vtkFloatingPointType double
#endif

//----------------------------------------------------------------------------
class vtkOrthoPlanesCallback : public vtkCommand
{
public:
  static vtkOrthoPlanesCallback *New()
  { return new vtkOrthoPlanesCallback; }

  void Execute( vtkObject *caller, unsigned long vtkNotUsed( event ),
                void *callData )
  {
    vtkImagePlaneWidget* self =
      reinterpret_cast< vtkImagePlaneWidget* >( caller );
    if(!self) return;

    double* wl = static_cast<double*>( callData );

    if ( self == this->WidgetX )
      {
      this->WidgetY->SetWindowLevel(wl[0],wl[1],1);
      this->WidgetZ->SetWindowLevel(wl[0],wl[1],1);
      }
    else if( self == this->WidgetY )
      {
      this->WidgetX->SetWindowLevel(wl[0],wl[1],1);
      this->WidgetZ->SetWindowLevel(wl[0],wl[1],1);
      }
    else if (self == this->WidgetZ)
      {
      this->WidgetX->SetWindowLevel(wl[0],wl[1],1);
      this->WidgetY->SetWindowLevel(wl[0],wl[1],1);
      }
  }

  vtkOrthoPlanesCallback():WidgetX( 0 ), WidgetY( 0 ), WidgetZ ( 0 ) {}

  vtkImagePlaneWidget* WidgetX;
  vtkImagePlaneWidget* WidgetY;
  vtkImagePlaneWidget* WidgetZ;
};

int main( int argc, char *argv[] )
{
  //char* fname = vtkTestUtilities::ExpandDataFileName(argc, argv, "Data/headsq/quarter");

  //vtkVolume16Reader* v16 =  vtkVolume16Reader::New();
  //  v16->SetDataDimensions( 64, 64);
  //  v16->SetDataByteOrderToLittleEndian();
  //  v16->SetImageRange( 1, 93);
  //  v16->SetDataSpacing( 3.2, 3.2, 1.5);
  //  v16->SetFilePrefix( fname );
  //  v16->SetDataMask( 0x7fff);
  //  v16->Update();
  std::vector<std::string> filenames;
  if( argc < 2 )
    {
    std::cerr << argv[0] << " filename1.dcm [filename2.dcm ...]\n";
    return 1;
    }
  else
    {
    // Is it a single directory ? If so loop over all files contained in it:
    const char *filename = argv[1];
    if( argc == 2 && gdcm::System::FileIsDirectory( filename ) )
      {
      std::cout << "Loading directory: " << filename << std::endl;
      bool recursive = false;
      gdcm::Directory d;
      d.Load(filename, recursive);
      gdcm::Directory::FilenamesType const &files = d.
      GetFilenames();
      for( gdcm::Directory::FilenamesType::const_iterator it = files.begin(); it != files.end(); ++it )
        {
        filenames.push_back( it->c_str() );
```

```
      }
    }
  else // list of files passed directly on the cmd line:
    // discard non-existing or directory
    {
    for(int i=1; i < argc; ++i)
      {
      filename = argv[i];
      if( gdcm::System::FileExists( filename ) )
        {
        if( gdcm::System::FileIsDirectory( filename ) )
          {
          std::cerr << "Discarding directory: " << filename << std::endl;
          }
        else
          {
          filenames.push_back( filename );
          }
        }
      else
        {
        std::cerr << "Discarding non existing file: " << filename << std::endl;
        }
      }
    }
  //names->Print( std::cout );
  }

vtkGDCMImageReader * reader = vtkGDCMImageReader::New();
double ippzspacing;
if( filenames.size() > 1 )
  {
  //gdcm::Trace::DebugOn();
  //gdcm::Trace::WarningOn();
  gdcm::IPPSorter s;
  s.SetComputeZSpacing( true );
  s.SetZSpacingTolerance( 1e-3 );
  bool b = s.Sort( filenames );
  if( !b )
    {
    std::cerr << "Failed to sort files" << std::endl;
    return 1;
    }
  std::cout << "Sorting succeeded:" << std::endl;
  s.Print( std::cout );

  std::cout << "Found z-spacing:" << std::endl;
  std::cout << s.GetZSpacing() << std::endl;
  ippzspacing = s.GetZSpacing();

  const std::vector<std::string> & sorted = s.GetFilenames();
  vtkStringArray *files = vtkStringArray::New();
  std::vector< std::string >::const_iterator it = sorted.begin();
  for( ; it != sorted.end(); ++it)
    {
    const std::string &f = *it;
    files->InsertNextValue( f.c_str() );
    }
  reader->SetFileNames( files );
  //reader->SetFileLowerLeft( 1 );
  reader->Update(); // important
  files->Delete();
  }
else
  {
  reader->SetFileName( argv[1] );
  reader->Update(); // important
  ippzspacing = reader->GetOutput()->GetSpacing()[2];
  ippzspacing = 4;
  }

//reader->GetOutput()->Print( std::cout );
//vtkFloatingPointType range[2];
//reader->GetOutput()->GetScalarRange(range);
//std::cout << "Range: " << range[0] << " " << range[1] << std::endl;

const vtkFloatingPointType *spacing = reader->GetOutput()->GetSpacing();

vtkImageChangeInformation *v16 = vtkImageChangeInformation::New();
#if (VTK_MAJOR_VERSION >= 6)
v16->SetInputConnection( reader->GetOutputPort() );
```

```
#else
  v16->SetInput( reader->GetOutput() );
#endif
  v16->SetOutputSpacing( spacing[0], spacing[1], ippzspacing );
  v16->Update();

#if 0
    vtkGDCMImageWriter *writer = vtkGDCMImageWriter::New();
    writer->SetInput( v16->GetOutput() );
    writer->SetFileLowerLeft( reader->GetFileLowerLeft() );
    writer->SetDirectionCosines( reader->GetDirectionCosines() );
    writer->SetImageFormat( reader->GetImageFormat() );
    writer->SetFileDimensionality( 3); //reader->GetFileDimensionality() );
    writer->SetMedicalImageProperties( reader->GetMedicalImageProperties() );
    writer->SetShift( reader->GetShift() );
    writer->SetScale( reader->GetScale() );
    writer->SetFileName( "out.dcm" );
    writer->Write();
#endif


  vtkOutlineFilter* outline = vtkOutlineFilter::New();
    outline->SetInputConnection(v16->GetOutputPort());

  vtkPolyDataMapper* outlineMapper = vtkPolyDataMapper::New();
    outlineMapper->SetInputConnection(outline->GetOutputPort());

  vtkActor* outlineActor =  vtkActor::New();
    outlineActor->SetMapper( outlineMapper);

  vtkRenderer* ren1 = vtkRenderer::New();
  vtkRenderer* ren2 = vtkRenderer::New();

  vtkRenderWindow* renWin = vtkRenderWindow::New();
    renWin->AddRenderer(ren2);
    renWin->AddRenderer(ren1);

  vtkRenderWindowInteractor* iren = vtkRenderWindowInteractor::New();
    iren->SetRenderWindow(renWin);

  vtkCellPicker* picker = vtkCellPicker::New();
    picker->SetTolerance(0.005);

  vtkProperty* ipwProp = vtkProperty::New();
   //assign default props to the ipw's texture plane actor

  vtkImagePlaneWidget* planeWidgetX = vtkImagePlaneWidget::New();
    planeWidgetX->SetInteractor( iren);
    planeWidgetX->SetKeyPressActivationValue('x');
    planeWidgetX->SetPicker(picker);
    planeWidgetX->RestrictPlaneToVolumeOn();
    planeWidgetX->GetPlaneProperty()->SetColor(1,0,0);
    planeWidgetX->SetTexturePlaneProperty(ipwProp);
    planeWidgetX->TextureInterpolateOff();
    planeWidgetX->SetResliceInterpolateToNearestNeighbour();
#if (VTK_MAJOR_VERSION >= 6)
    planeWidgetX->SetInputConnection(v16->GetOutputPort());
#else
    planeWidgetX->SetInput(v16->GetOutput());
#endif
    planeWidgetX->SetPlaneOrientationToXAxes();
    //planeWidgetX->SetSliceIndex(32);
    planeWidgetX->DisplayTextOn();
    planeWidgetX->On();
    planeWidgetX->InteractionOff();
    planeWidgetX->InteractionOn();

  vtkImagePlaneWidget* planeWidgetY = vtkImagePlaneWidget::New();
    planeWidgetY->SetInteractor( iren);
    planeWidgetY->SetKeyPressActivationValue('y');
    planeWidgetY->SetPicker(picker);
    planeWidgetY->GetPlaneProperty()->SetColor(1,1,0);
    planeWidgetY->SetTexturePlaneProperty(ipwProp);
    planeWidgetY->TextureInterpolateOn();
    planeWidgetY->SetResliceInterpolateToLinear();
#if (VTK_MAJOR_VERSION >= 6)
    planeWidgetY->SetInputConnection(v16->GetOutputPort());
#else
    planeWidgetY->SetInput(v16->GetOutput());
#endif
    planeWidgetY->SetPlaneOrientationToYAxes();
```

```
    //planeWidgetY->SetSlicePosition(102.4);
    planeWidgetY->SetLookupTable( planeWidgetX->GetLookupTable());
    planeWidgetY->DisplayTextOn();
    planeWidgetY->UpdatePlacement();
    planeWidgetY->On();

  vtkImagePlaneWidget* planeWidgetZ = vtkImagePlaneWidget::New();
    planeWidgetZ->SetInteractor( iren);
    planeWidgetZ->SetKeyPressActivationValue('z');
    planeWidgetZ->SetPicker(picker);
    planeWidgetZ->GetPlaneProperty()->SetColor(0,0,1);
    planeWidgetZ->SetTexturePlaneProperty(ipwProp);
    planeWidgetZ->TextureInterpolateOn();
    planeWidgetZ->SetResliceInterpolateToCubic();
#if (VTK_MAJOR_VERSION >= 6)
    planeWidgetZ->SetInputConnection(v16->GetOutputPort());
#else
    planeWidgetZ->SetInput(v16->GetOutput());
#endif
    planeWidgetZ->SetPlaneOrientationToZAxes();
    //planeWidgetZ->SetSliceIndex(25);
    planeWidgetZ->SetLookupTable( planeWidgetX->GetLookupTable());
    planeWidgetZ->DisplayTextOn();
    planeWidgetZ->On();

  vtkImageOrthoPlanes *orthoPlanes = vtkImageOrthoPlanes::New();
    orthoPlanes->SetPlane(0, planeWidgetX);
    orthoPlanes->SetPlane(1, planeWidgetY);
    orthoPlanes->SetPlane(2, planeWidgetZ);
    orthoPlanes->ResetPlanes();

   vtkOrthoPlanesCallback* cbk = vtkOrthoPlanesCallback::New();
   cbk->WidgetX = planeWidgetX;
   cbk->WidgetY = planeWidgetY;
   cbk->WidgetZ = planeWidgetZ;
   planeWidgetX->AddObserver( vtkCommand::EndWindowLevelEvent, cbk );
   planeWidgetY->AddObserver( vtkCommand::EndWindowLevelEvent, cbk );
   planeWidgetZ->AddObserver( vtkCommand::EndWindowLevelEvent, cbk );
   cbk->Delete();

  double wl[2];
  planeWidgetZ->GetWindowLevel(wl);

  // Add a 2D image to test the GetReslice method
  //
  vtkImageMapToColors* colorMap = vtkImageMapToColors::New();
    colorMap->PassAlphaToOutputOff();
    colorMap->SetActiveComponent(0);
    colorMap->SetOutputFormatToLuminance();
#if (VTK_MAJOR_VERSION >= 6)
    colorMap->SetInputData(planeWidgetZ->GetResliceOutput());
#else
    colorMap->SetInput(planeWidgetZ->GetResliceOutput());
#endif
    colorMap->SetLookupTable(planeWidgetX->GetLookupTable());

  vtkImageActor* imageActor = vtkImageActor::New();
    imageActor->PickableOff();
#if (VTK_MAJOR_VERSION >= 6)
    imageActor->SetInputData(colorMap->GetOutput());
#else
    imageActor->SetInput(colorMap->GetOutput());
#endif

  // Add the actors
  //
  ren1->AddActor( outlineActor);
  ren2->AddActor( imageActor);

  ren1->SetBackground( 0.1, 0.1, 0.2);
  ren2->SetBackground( 0.2, 0.1, 0.2);

  renWin->SetSize( 600, 350);

  ren1->SetViewport(0,0,0.58333,1);
  ren2->SetViewport(0.58333,0,1,1);

  // Set the actors' postions
  //
  renWin->Render();
  //iren->SetEventPosition( 175,175);
```

```
//iren->SetKeyCode('r');
//iren->InvokeEvent(vtkCommand::CharEvent,NULL);
//iren->SetEventPosition( 475,175);
//iren->SetKeyCode('r');
//iren->InvokeEvent(vtkCommand::CharEvent,NULL);
//renWin->Render();

//ren1->GetActiveCamera()->Elevation(110);
//ren1->GetActiveCamera()->SetViewUp(0, 0, -1);
//ren1->GetActiveCamera()->Azimuth(45);
//ren1->GetActiveCamera()->Dolly(1.15);
ren1->ResetCameraClippingRange();

vtkAnnotatedCubeActor* cube = vtkAnnotatedCubeActor::New();
cube->SetXPlusFaceText ( "R" );
cube->SetXMinusFaceText( "L" );
cube->SetYPlusFaceText ( "A" );
cube->SetYMinusFaceText( "P" );
cube->SetZPlusFaceText ( "H" );
cube->SetZMinusFaceText( "F" );
cube->SetFaceTextScale( 0.666667 );

vtkAxesActor* axes2 = vtkAxesActor::New();

vtkMatrix4x4 *invert = vtkMatrix4x4::New();
invert->DeepCopy( reader->GetDirectionCosines() );
invert->Invert();

// simulate a left-handed coordinate system
//
vtkTransform *transform = vtkTransform::New();
transform->Identity();
//transform->RotateY(90);
transform->Concatenate(invert);
axes2->SetShaftTypeToCylinder();
axes2->SetUserTransform( transform );
cube->GetAssembly()->SetUserTransform( transform );

axes2->SetTotalLength( 1.5, 1.5, 1.5 );
axes2->SetCylinderRadius( 0.500 * axes2->GetCylinderRadius() );
axes2->SetConeRadius    ( 1.025 * axes2->GetConeRadius() );
axes2->SetSphereRadius  ( 1.500 * axes2->GetSphereRadius() );

vtkTextProperty* tprop = axes2->GetXAxisCaptionActor2D()->
  GetCaptionTextProperty();
tprop->ItalicOn();
tprop->ShadowOn();
tprop->SetFontFamilyToTimes();

axes2->GetYAxisCaptionActor2D()->GetCaptionTextProperty()->ShallowCopy( tprop );
axes2->GetZAxisCaptionActor2D()->GetCaptionTextProperty()->ShallowCopy( tprop );

vtkPropAssembly* assembly = vtkPropAssembly::New();
assembly->AddPart( axes2 );
assembly->AddPart( cube );

vtkOrientationMarkerWidget* widget = vtkOrientationMarkerWidget::New();
widget->SetOutlineColor( 0.9300, 0.5700, 0.1300 );
widget->SetOrientationMarker( assembly );
widget->SetInteractor( iren );
widget->SetViewport( 0.0, 0.0, 0.4, 0.4 );
widget->SetEnabled( 1 );
widget->InteractiveOff();
widget->InteractiveOn();

// Playback recorded events
//
//vtkInteractorEventRecorder *recorder = vtkInteractorEventRecorder::New();
//recorder->SetInteractor(iren);
//recorder->ReadFromInputStringOn();
//recorder->SetInputString(IOPeventLog);

// Interact with data
// Render the image
//
iren->Initialize();
renWin->Render();

// Test SetKeyPressActivationValue for one of the widgets
//
//iren->SetKeyCode('z');
```

```
  //iren->InvokeEvent(vtkCommand::CharEvent,NULL);
  //iren->SetKeyCode('z');
  //iren->InvokeEvent(vtkCommand::CharEvent,NULL);

  //int retVal = vtkRegressionTestImage( renWin );
  //
  //if ( retVal == vtkRegressionTester::DO_INTERACTOR)
    {
    iren->Start();
    }

  // Clean up
  //
  //recorder->Off();
  //recorder->Delete();

  ipwProp->Delete();
  orthoPlanes->Delete();
  planeWidgetX->Delete();
  planeWidgetY->Delete();
  planeWidgetZ->Delete();
  colorMap->Delete();
  imageActor->Delete();
  picker->Delete();
  outlineActor->Delete();
  outlineMapper->Delete();
  outline->Delete();
  iren->Delete();
  renWin->Delete();
  ren1->Delete();
  ren2->Delete();
  v16->Delete();
  reader->Delete();

  return 0;
}
```

12.68 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;
}
```

12.69 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;
}
```

12.70 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;
}
```

12.71 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;
}
```

12.72 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;
}
```

12.73 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 "vtkGDCMImageReader.h"
#include "vtkPiecewiseFunction.h"
#include "vtkColorTransferFunction.h"
#include "vtkVolume.h"
#include "vtkVolumeProperty.h"
#include "vtkVolumeTextureMapper3D.h"
#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;
}
```

12.74 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;
}
```

12.75 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;
    }
}
```

12.76 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;
}
```

12.77 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;
}
```

12.78 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;
}
```

12.79 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;
}
```

12.80 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;
}
```

12.81 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;
}
```

12.82 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;
    }
}
```

12.83 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;
}
```

12.84 GetPortionCSAHeader.py

```
1 ##########################################################################
2 #
3 #  Program: GDCM (Grassroots DICOM). A DICOM library
4 #
5 #  Copyright (c) 2006-2011 Mathieu Malaterre
6 #  All rights reserved.
7 #  See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.
8 #
9 #     This software is distributed WITHOUT ANY WARRANTY; without even
10 #     the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
11 #     PURPOSE.  See the above copyright notice for more information.
12 #
13 ##########################################################################
14
15 """
16 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("\\")
```

12.85 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;
}
```

12.86 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;
}
```

12.87 headsq2dcm.py

```
1 ##########################################################################
2 #
3 #  Program: GDCM (Grassroots DICOM). A DICOM library
4 #
5 #  Copyright (c) 2006-2011 Mathieu Malaterre
6 #  All rights reserved.
7 #  See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.
8 #
9 #     This software is distributed WITHOUT ANY WARRANTY; without even
10 #     the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
11 #     PURPOSE.  See the above copyright notice for more information.
12 #
13 ##########################################################################
14
15 """
16 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
37 writer = vtkgdcm.vtkGDCMImageWriter()
38 writer.SetFileName( "headsq.dcm" )
39 writer.SetInput( reader.GetOutput() )
40 # cast -> Multiframe Grayscale Byte Secondary Capture Image Storage
41 #writer.SetInput( cast.GetOutput() )
42 writer.SetFileDimensionality( 3 )
43 writer.Write()
```

12.88 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;
    }
}
```

12.89 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;
    }
}
```

12.90 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;
    }
}
```

12.91 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;
    }
}
```

12.92 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;
    }
}
```

12.93 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 );
    }
}
```

12.94 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;
}
```

12.95 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() ); // new
    writer.SetInput( reader.GetOutput() ); // old
    writer.Write();

    return 0;
    }
}
```

12.96 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 );
    }
}
```

12.97 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.SetInput( reader.GetOutput() );
    cast.SetOutputScalarTypeToUnsignedChar();

    // By default this is creating a Multiframe Grayscale Word Secondary Capture Image Storage
    vtkGDCMImageWriter writer = vtkGDCMImageWriter.
      New();
    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();

    return 0;
    }
}
```

12.98 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"
```

```cpp
#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.
 */
```

12.99 HelloWorld.py

```
1 ##########################################################################
2 #
3 #  Program: GDCM (Grassroots DICOM). A DICOM library
4 #
5 #  Copyright (c) 2006-2011 Mathieu Malaterre
6 #  All rights reserved.
7 #  See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.
8 #
9 #     This software is distributed WITHOUT ANY WARRANTY; without even
10 #     the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
11 #     PURPOSE.  See the above copyright notice for more information.
12 #
13 ##########################################################################
14
```

```
15 """
16 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))
```

12.100 iU22tomultisc.cxx

```
/*=========================================================================

  Program: GDCM (Grassroots DICOM). A DICOM library

  Copyright (c) 2006-2011 Mathieu Malaterre
  All rights reserved.
  See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.

     This software is distributed WITHOUT ANY WARRANTY; without even
     the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
     PURPOSE.  See the above copyright notice for more information.

=========================================================================*/
/*
 * iU22 Raw Data extractor
 */
#include "gdcmReader.h"
#include "gdcmImageWriter.h"
#include "gdcmAttribute.h"
#include "gdcmPrivateTag.h"

#include <math.h>

int main(int argc, char *argv [])
{
  if( argc < 2 ) return 1;
  // IM_001
```

```
  const char *filename = argv[1];

  gdcm::Reader reader; // Do not use ImageReader
  reader.SetFileName( filename );
  if( !reader.Read() )
    {
    std::cerr << "Failed to read: " << filename << std::endl;
    return 1;
    }

 // * The data is simply 8-bit unsigned in the obvious x/y/z order
 // * 200D,300B contains the data
 // * 200D,3001 contains the no. of voxels (416,412,256 in this case)
 // * 200D,3003 contains the voxel sizes (0.156184527398215 /
 // 0.1223749613981957 / 0.328479990704639 in this case)

  const gdcm::File &file = reader.GetFile();
  const gdcm::DataSet &ds = file.GetDataSet();
  const gdcm::PrivateTag trawdataus( 0x200d, 0x0b, "Philips US Imaging DD 033" );
  const gdcm::DataElement &rawdataus = ds.GetDataElement( trawdataus );

  const gdcm::PrivateTag tcolsrowsframes( 0x200d, 0x01, "Philips US Imaging DD 036" );
  const gdcm::DataElement &colsrowsframes = ds.GetDataElement(
      tcolsrowsframes );
  // const gdcm::PrivateTag tcolsrowsframes( 0x200d, 0x02, "Philips US Imaging DD 036" );
  // this is just a duplicate previous tag.
  const gdcm::PrivateTag tvoxelspacing( 0x200d, 0x03, "Philips US Imaging DD 036" );
  const gdcm::DataElement &voxelspacing = ds.GetDataElement( tvoxelspacing )
      ;

  gdcm::Element<gdcm::VR::DS,gdcm::VM::VM3> dims; // Use DS to
       interpret value stored in LO
  dims.SetFromDataElement( colsrowsframes );

  gdcm::Element<gdcm::VR::DS,gdcm::VM::VM3> spacing;
  spacing.SetFromDataElement( voxelspacing );

  gdcm::ImageWriter writer;

  gdcm::Image &image = writer.GetImage();
  image.SetNumberOfDimensions( 3 ); // good default
  image.SetDimension(0, (unsigned int)dims[0] );
  image.SetDimension(1, (unsigned int)dims[1] );
  image.SetDimension(2, (unsigned int)dims[2] );
  image.SetSpacing(0, spacing[0] );
  image.SetSpacing(1, spacing[1] );
  image.SetSpacing(2, spacing[2] );
  gdcm::PixelFormat pixeltype = gdcm::PixelFormat::UINT8;

  gdcm::PhotometricInterpretation pi;
  pi = gdcm::PhotometricInterpretation::MONOCHROME2;
  image.SetPhotometricInterpretation( pi );
  image.SetPixelFormat( pixeltype );

  image.SetDataElement( rawdataus );

  std::string outfilename = "outiu22.dcm";

  gdcm::DataElement de( gdcm::Tag(0x8,0x16) ); // SOP Class UID
  de.SetVR( gdcm::VR::UI );
  gdcm::MediaStorage ms(
    gdcm::MediaStorage::UltrasoundMultiFrameImageStorage
       );
//    gdcm::MediaStorage::MultiframeGrayscaleByteSecondaryCaptureImageStorage );
  de.SetByteValue( ms.GetString(), (uint32_t)strlen(ms.
      GetString()));
  writer.GetFile().GetDataSet().Replace( de );

  writer.SetFileName( outfilename.c_str() );
  if( !writer.Write() )
    {
    std::cerr << "could not write: " << outfilename << std::endl;
    return 1;
    }


  return 0;
}
```

12.101 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;
}
```

12.102 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;
}
```

12.103 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;
}
```

12.104 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;
    }
}
```

12.105 ManipulateFile.py

```
1 ##########################################################################
2 #
3 #  Program: GDCM (Grassroots DICOM). A DICOM library
4 #
5 #  Copyright (c) 2006-2011 Mathieu Malaterre
6 #  All rights reserved.
7 #  See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.
8 #
9 #     This software is distributed WITHOUT ANY WARRANTY; without even
10 #     the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
11 #     PURPOSE.  See the above copyright notice for more information.
12 #
13 ##########################################################################
14
15 """
16 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)
```

12.106 ManipulateSequence.py

```
1 ######################################################################
2 #
3 #  Program: GDCM (Grassroots DICOM). A DICOM library
4 #
5 #  Copyright (c) 2006-2011 Mathieu Malaterre
6 #  All rights reserved.
7 #  See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.
8 #
9 #     This software is distributed WITHOUT ANY WARRANTY; without even
10 #     the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
11 #     PURPOSE.  See the above copyright notice for more information.
12 #
13 ######################################################################
14
15 """
16 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
```

```
WARNING:
Do not use as-is in production, this is just an example
This example works in an undefined length Item only (you need to explicitly recompute the length otherwise)
"""

import sys
import gdcm

if __name__ == "__main__":

  file1 = sys.argv[1]
  file2 = sys.argv[2]

  r = gdcm.Reader()
  r.SetFileName( file1 )
  if not r.Read():
    sys.exit(1)

  f = r.GetFile()
  ds = f.GetDataSet()
  tsis = gdcm.Tag(0x0008,0x2112) # SourceImageSequence
  if ds.FindDataElement( tsis ):
    sis = ds.GetDataElement( tsis )
    #sqsis = sis.GetSequenceOfItems()
    # GetValueAsSQ handle more cases
    sqsis = sis.GetValueAsSQ()
    if sqsis.GetNumberOfItems():
      item1 = sqsis.GetItem(1)
      nestedds = item1.GetNestedDataSet()
      tprcs = gdcm.Tag(0x0040,0xa170) # PurposeOfReferenceCodeSequence
      if nestedds.FindDataElement( tprcs ):
        prcs = nestedds.GetDataElement( tprcs )
        sqprcs = prcs.GetSequenceOfItems()
        if sqprcs.GetNumberOfItems():
          item2 = sqprcs.GetItem(1)
          nestedds2 = item2.GetNestedDataSet()
          # (0008,0104) LO [Uncompressed predecessor]               #  24, 1 CodeMeaning
          tcm = gdcm.Tag(0x0008,0x0104)
          if nestedds2.FindDataElement( tcm ):
            cm = nestedds2.GetDataElement( tcm )
            mystr = "GDCM was here"
            cm.SetByteValue( mystr, gdcm.VL( len(mystr) ) )

  w = gdcm.Writer()
  w.SetFile( f )
  w.SetFileName( file2 )
  if not w.Write():
    sys.exit(1)
```

12.107 MergeFile.py

```
##########################################################################
#
#  Program: GDCM (Grassroots DICOM). A DICOM library
#
#  Copyright (c) 2006-2011 Mathieu Malaterre
#  All rights reserved.
#  See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.
#
#     This software is distributed 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:

 python MergeFile.py input1.dcm input2.dcm

  It will produce a 'merge.dcm' output file, which contains all meta information from input1.dcm
  and copy the Stored Pixel values from input2.dcm
  This script even works when input2.dcm is a Secondary Capture and does not contains information
  such as IOP and IPP...
"""
```

```
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)
```

12.108 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;
}
```

12.109 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;
    }
}
```

12.110 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.SetInput(change.GetOutput());

    //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();
    }
}
```

12.111 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;
```

```
    }
}
```

12.112 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();
    }
}
```

12.113 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 wextent[] = image.GetWholeExtent();
    int xMin = wextent[0];
    int xMax = wextent[1];
    int yMin = wextent[2];
    int yMax = wextent[3];
    int zMin = wextent[4];
    int zMax = wextent[5];

    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.SetInput(change.GetOutput());
    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.SetInput(change.GetOutput());
    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.SetInput(change.GetOutput());
    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 );
    }
}
```

12.114 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;
}
```

12.115 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;
    }
}
```

12.116 NewSequence.py

```
1 ############################################################################
2 #
3 #  Program: GDCM (Grassroots DICOM). A DICOM library
4 #
5 #  Copyright (c) 2006-2011 Mathieu Malaterre
6 #  All rights reserved.
7 #  See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.
8 #
9 #     This software is distributed WITHOUT ANY WARRANTY; without even
10 #     the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
11 #     PURPOSE.  See the above copyright notice for more information.
12 #
13 ############################################################################
14
15 """
16 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)
```

12.117 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;
}
```

12.118 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;
}
```

12.119 PhilipsPrivateRescaleInterceptSlope.py

```
1 ##########################################################################
2 #
3 #  Program: GDCM (Grassroots DICOM). A DICOM library
4 #
5 #  Copyright (c) 2006-2011 Mathieu Malaterre
6 #  All rights reserved.
7 #  See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.
8 #
9 #     This software is distributed WITHOUT ANY WARRANTY; without even
10 #     the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
11 #     PURPOSE.  See the above copyright notice for more information.
12 #
13 ##########################################################################
14
15 """
16 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"
```

12.120 PlaySound.py

```
1 ##########################################################################
2 #
3 #  Program: GDCM (Grassroots DICOM). A DICOM library
4 #
5 #  Copyright (c) 2006-2011 Mathieu Malaterre
6 #  All rights reserved.
7 #  See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.
8 #
9 #     This software is distributed WITHOUT ANY WARRANTY; without even
10 #     the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
11 #     PURPOSE.  See the above copyright notice for more information.
12 #
13 ##########################################################################
14
15 """
16 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()
```

12.121 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;
}
```

12.122 PrivateDict.py

```
1 ##########################################################################
2 #
3 #  Program: GDCM (Grassroots DICOM). A DICOM library
4 #
5 #  Copyright (c) 2006-2011 Mathieu Malaterre
6 #  All rights reserved.
7 #  See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.
8 #
9 #     This software is distributed WITHOUT ANY WARRANTY; without even
10 #     the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
11 #     PURPOSE.  See the above copyright notice for more information.
12 #
13 ##########################################################################
14
```

```
15 """
16 """
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") )
```

12.123 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;
}
```

12.124 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;
}
```

12.125 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;
}
```

12.126 ReadAndDumpDICOMDIR.py

```
1 ############################################################################
2 #
3 #  Program: GDCM (Grassroots DICOM). A DICOM library
4 #
5 #  Copyright (c) 2006-2011 Mathieu Malaterre
6 #  All rights reserved.
7 #  See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.
8 #
9 #     This software is distributed WITHOUT ANY WARRANTY; without even
10 #     the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
11 #     PURPOSE.  See the above copyright notice for more information.
12 #
13 #  File: ReadAndDumpDICOMDIR.py
14 #
15 #  Author: Lukas Batteau (lbatteau gmail)
16 #
17 #  This example shows how to read and dump a DICOMDIR File.
18 #  Based on Tom Marynowski's (lordglub gmail) example.
19 #
20 #  Usage:
21 #  python ReadAndDumpDICOMDIR.py [DICOMDIR file]
22 ############################################################################
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
```

12.127 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;
}
```

12.128 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;
}
```

12.129 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);
      }
    }
}
```

12.130 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;
}
```

12.131 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;
}
```

12.132 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.SetInput( reader.GetOutput() );
    writer.SetFileName( "ReadSeriesIntoVTK.mhd" );
    writer.Write();

    System.out.println("Success writing: " + writer.GetFileName() );
    }
}
```

12.133 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;
}
```

12.134 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;
    }
}
```

12.135 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;
    }
}
```

12.136 RemovePrivateTags.py

```
1 ############################################################################
2 #
3 #  Program: GDCM (Grassroots DICOM). A DICOM library
4 #
5 #  Copyright (c) 2006-2011 Mathieu Malaterre
6 #  All rights reserved.
7 #  See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.
8 #
9 #     This software is distributed WITHOUT ANY WARRANTY; without even
10 #     the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
11 #     PURPOSE.  See the above copyright notice for more information.
12 #
13 ############################################################################
14
15 """
16 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)
```

12.137 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;
    }
}
```

12.138 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;
}
```

12.139 ReWriteSCAsMR.py

```
1 ############################################################################
2 #
3 #  Program: GDCM (Grassroots DICOM). A DICOM library
4 #
5 #  Copyright (c) 2006-2011 Mathieu Malaterre
6 #  All rights reserved.
7 #  See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.
8 #
9 #     This software is distributed WITHOUT ANY WARRANTY; without even
10 #     the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
11 #     PURPOSE.  See the above copyright notice for more information.
12 #
13 ############################################################################
14
15 """
16 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)
```

12.140 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;
}
```

12.141 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;
}
```

12.142 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;
    }
}
```

12.143 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" );
    }
}
```

12.144 ScanDirectory.py

```
1 ##########################################################################
2 #
3 #  Program: GDCM (Grassroots DICOM). A DICOM library
4 #
5 #  Copyright (c) 2006-2011 Mathieu Malaterre
6 #  All rights reserved.
7 #  See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.
8 #
9 #     This software is distributed WITHOUT ANY WARRANTY; without even
10 #     the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
11 #     PURPOSE.  See the above copyright notice for more information.
12 #
13 ##########################################################################
14
15 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)
```

12.145 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;
    }
}
```

12.146 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;
    }
}
```

12.147 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;
      }
    }
}
```

12.148 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;
}
```

12.149 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;
}
```

12.150 SortImage.py

```
1 ############################################################################
2 #
3 #  Program: GDCM (Grassroots DICOM). A DICOM library
4 #
5 #  Copyright (c) 2006-2011 Mathieu Malaterre
6 #  All rights reserved.
```

```
7 #  See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.
8 #
9 #     This software is distributed WITHOUT ANY WARRANTY; without even
10 #     the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
11 #     PURPOSE.  See the above copyright notice for more information.
12 #
13 ##########################################################################
14
15 """
16 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
```

12.151 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;
    }
}
```

12.152 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;
    }
}
```

12.153 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;
}
```

12.154 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;
}
```

12.155 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;
}
```

12.156 TestReader.py

This is a C++ example on how to use gdcm::Reader

```
1 ##########################################################################
2 #
3 #  Program: GDCM (Grassroots DICOM). A DICOM library
4 #
5 #  Copyright (c) 2006-2011 Mathieu Malaterre
6 #  All rights reserved.
7 #  See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.
8 #
9 #     This software is distributed WITHOUT ANY WARRANTY; without even
10 #     the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
11 #     PURPOSE.  See the above copyright notice for more information.
12 #
13 ##########################################################################
14
15 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)
```

12.157 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;
}
```

12.158 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;
}
```

12.159 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 );
    }
}
```

12.160 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;
}
```

12.161 WriteBuffer.py

```
1 ##########################################################################
2 #
3 #  Program: GDCM (Grassroots DICOM). A DICOM library
4 #
5 #  Copyright (c) 2006-2011 Mathieu Malaterre
6 #  All rights reserved.
7 #  See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.
8 #
9 #     This software is distributed WITHOUT ANY WARRANTY; without even
10 #     the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
11 #     PURPOSE.  See the above copyright notice for more information.
12 #
13 ##########################################################################
14
15 """
16 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, 111
~AnonymizeEvent
gdcm::AnonymizeEvent, 94
~Anonymizer
gdcm::Anonymizer, 98
~Attribute
gdcm::Attribute< Group, Element, TVR, VM::VM1_n >, 126
~AudioCodec
gdcm::AudioCodec, 136
~BaseCompositeMessage
gdcm::network::BaseCompositeMessage, 140
~BaseNormalizedMessage
gdcm::network::BaseNormalizedMessage, 142
~BasePDU
gdcm::network::BasePDU, 144
~BaseQuery
gdcm::BaseQuery, 147
~BaseRootQuery
gdcm::BaseRootQuery, 150
~Bitmap
gdcm::Bitmap, 159
~BitmapToBitmapFilter
gdcm::BitmapToBitmapFilter, 166
~BoxRegion
gdcm::BoxRegion, 168
~ByteSwapFilter
gdcm::ByteSwapFilter, 173
~ByteValue
gdcm::ByteValue, 176
~CAPICryptographicMessageSyntax
gdcm::CAPICryptographicMessageSyntax, 181
~CSAHeader
gdcm::CSAHeader, 220
~Coder
gdcm::Coder, 195
~Command
gdcm::Command, 199
~CommandDataSet
gdcm::CommandDataSet, 201
~CryptoFactory
gdcm::CryptoFactory, 211
~CryptographicMessageSyntax
gdcm::CryptographicMessageSyntax, 212
~Curve
gdcm::Curve, 232
~DICOMDIRGenerator
gdcm::DICOMDIRGenerator, 265
~DataEvent
gdcm::DataEvent, 245
~DataSetEvent
gdcm::DataSetEvent, 255
~Decoder
gdcm::Decoder, 258
~Defs
gdcm::Defs, 260
~DeltaEncodingCodec
gdcm::DeltaEncodingCodec, 263
~DictConverter
gdcm::DictConverter, 271
~DictPrinter
gdcm::DictPrinter, 276
~Dicts
gdcm::Dicts, 278
~DirectionCosines
gdcm::DirectionCosines, 282
~Directory
gdcm::Directory, 285
~Dumper
gdcm::Dumper, 290
~Element
gdcm::Element< TVR, VM::VM1_n >, 296
~Event
gdcm::Event, 316
~Exception
gdcm::Exception, 318
~File
gdcm::File, 327
~FileAnonymizer
gdcm::FileAnonymizer, 330
~FileChangeTransferSyntax
gdcm::FileChangeTransferSyntax, 333
~FileDecompressLookupTable
gdcm::FileDecompressLookupTable, 336
~FileDerivation
gdcm::FileDerivation, 338
~FileExplicitFilter
gdcm::FileExplicitFilter, 341
~FileMetaInformation

gdcm::FileMetaInformation, 344
~FileNameEvent
gdcm::FileNameEvent, 352
~FileStreamer
gdcm::FileStreamer, 359
~FilenameGenerator
gdcm::FilenameGenerator, 354
~Global
gdcm::Global, 371
~GroupDict
gdcm::GroupDict, 374
~IconImageFilter
gdcm::IconImageFilter, 376
~IconImageGenerator
gdcm::IconImageGenerator, 378
~Image
gdcm::Image, 383
~ImageApplyLookupTable
gdcm::ImageApplyLookupTable, 387
~ImageChangePhotometricInterpretation
gdcm::ImageChangePhotometricInterpretation, 389
~ImageChangePlanarConfiguration
gdcm::ImageChangePlanarConfiguration, 392
~ImageChangeTransferSyntax
gdcm::ImageChangeTransferSyntax, 395
~ImageCodec
gdcm::ImageCodec, 399
~ImageConverter
gdcm::ImageConverter, 405
~ImageFragmentSplitter
gdcm::ImageFragmentSplitter, 407
~ImageReader
gdcm::ImageReader, 414
~ImageRegionReader
gdcm::ImageRegionReader, 418
~ImageToImageFilter
gdcm::ImageToImageFilter, 421
~ImageWriter
gdcm::ImageWriter, 423
~JPEG12Codec
gdcm::JPEG12Codec, 445
~JPEG16Codec
gdcm::JPEG16Codec, 447
~JPEG2000Codec
gdcm::JPEG2000Codec, 450
~JPEG8Codec
gdcm::JPEG8Codec, 454
~JPEGCodec
gdcm::JPEGCodec, 457
~JPEGLSCodec
gdcm::JPEGLSCodec, 463
~JSON
gdcm::JSON, 465
~KAKADUCodec
gdcm::KAKADUCodec, 467
~LookupTable
gdcm::LookupTable, 473
~MD5
gdcm::MD5, 482
~MemberCommand
gdcm::MemberCommand, 493
~MeshPrimitive
gdcm::MeshPrimitive, 497
~ModuleEntry
gdcm::ModuleEntry, 509
~Object
gdcm::Object, 540
~OpenSSLCryptographicMessageSyntax
gdcm::OpenSSLCryptographicMessageSyntax, 544
~OpenSSLP7CryptographicMessageSyntax
gdcm::OpenSSLP7CryptographicMessageSyntax, 548
~Orientation
gdcm::Orientation, 550
~Overlay
gdcm::Overlay, 554
~PDBHeader
gdcm::PDBHeader, 568
~PDFCodec
gdcm::PDFCodec, 571
~PGXCodec
gdcm::PGXCodec, 576
~PNMCodec
gdcm::PNMCodec, 599
~PVRGCodec
gdcm::PVRGCodec, 624
~ParseException
gdcm::ParseException, 559
~Parser
gdcm::Parser, 562
~Pixmap
gdcm::Pixmap, 588
~PixmapReader
gdcm::PixmapReader, 591
~PixmapToPixmapFilter
gdcm::PixmapToPixmapFilter, 594
~PixmapWriter
gdcm::PixmapWriter, 596
~Preamble
gdcm::Preamble, 601
~Printer
gdcm::Printer, 614
~PrivateDict
gdcm::PrivateDict, 617
~ProgressEvent
gdcm::ProgressEvent, 622
~PythonFilter
gdcm::PythonFilter, 626

~QueryBase
gdcm::QueryBase, 627
~RAWCodec
gdcm::RAWCodec, 639
~RLECodec
gdcm::RLECodec, 655
~Reader
gdcm::Reader, 644
~Region
gdcm::Region, 649
~Rescaler
gdcm::Rescaler, 651
~SHA1
gdcm::SHA1, 701
~Scanner
gdcm::Scanner, 662
~Segment
gdcm::Segment, 668
~SegmentReader
gdcm::SegmentReader, 675
~SegmentWriter
gdcm::SegmentWriter, 677
~SegmentedPaletteColorLookupTable
gdcm::SegmentedPaletteColorLookupTable, 672
~SerieHelper
gdcm::SerieHelper, 691
~ServiceClassUser
gdcm::ServiceClassUser, 696
~SimpleMemberCommand
gdcm::SimpleMemberCommand, 703
~SimpleSubjectWatcher
gdcm::SimpleSubjectWatcher, 705
~SmartPointer
gdcm::SmartPointer, 708
~Sorter
gdcm::Sorter, 714
~Spacing
gdcm::Spacing, 717
~SplitMosaicFilter
gdcm::SplitMosaicFilter, 718
~StreamImageReader
gdcm::StreamImageReader, 722
~StreamImageWriter
gdcm::StreamImageWriter, 727
~StrictScanner
gdcm::StrictScanner, 733
~StringFilter
gdcm::StringFilter, 741
~Subject
gdcm::Subject, 745
~Surface
gdcm::Surface, 749
~SurfaceReader
gdcm::SurfaceReader, 758
~SurfaceWriter
gdcm::SurfaceWriter, 760
~Table
gdcm::Table, 769
~TableEntry
gdcm::TableEntry, 770
~TableReader
gdcm::TableReader, 772
~TableRow
gdcm::network::TableRow, 773
~TagPath
gdcm::TagPath, 782
~Testing
gdcm::Testing, 785
~Trace
gdcm::Trace, 790
~Transition
gdcm::network::Transition, 799
~ULAction
gdcm::network::ULAction, 827
~ULBasicCallback
gdcm::network::ULBasicCallback, 858
~ULConnection
gdcm::network::ULConnection, 860
~ULConnectionCallback
gdcm::network::ULConnectionCallback, 862
~ULConnectionManager
gdcm::network::ULConnectionManager, 867
~ULEvent
gdcm::network::ULEvent, 869
~ULWritingCallback
gdcm::network::ULWritingCallback, 872
~UserInformation
gdcm::network::UserInformation, 881
~Validate
gdcm::Validate, 884
~Value
gdcm::Value, 885
~Version
gdcm::Version, 888
~Writer
gdcm::Writer, 972
~XMLDictReader
gdcm::XMLDictReader, 975
~XMLPrinter
gdcm::XMLPrinter, 977
~XMLPrivateDictReader
gdcm::XMLPrivateDictReader, 979
~vtkGDCMImageReader
vtkGDCMImageReader, 907
~vtkGDCMImageReader2
vtkGDCMImageReader2, 913
~vtkGDCMImageWriter
vtkGDCMImageWriter, 920

~vtkGDCMMedicalImageProperties
vtkGDCMMedicalImageProperties, 924
~vtkGDCMPolyDataReader
vtkGDCMPolyDataReader, 926
~vtkGDCMPolyDataWriter
vtkGDCMPolyDataWriter, 929
~vtkGDCMTesting
vtkGDCMTesting, 932
~vtkGDCMThreadedImageReader
vtkGDCMThreadedImageReader, 935
~vtkGDCMThreadedImageReader2
vtkGDCMThreadedImageReader2, 937
~vtkImageColorViewer
vtkImageColorViewer, 943
~vtkImageMapToColors16
vtkImageMapToColors16, 948
~vtkImageMapToWindowLevelColors2
vtkImageMapToWindowLevelColors2, 951
~vtkImagePlanarComponentsToComponents
vtkImagePlanarComponentsToComponents, 953
~vtkImageRGBToYBR
vtkImageRGBToYBR, 955
~vtkImageYBRToRGB
vtkImageYBRToRGB, 957
~vtkLookupTable16
vtkLookupTable16, 959
~vtkRTStructSetProperties
vtkRTStructSetProperties, 962

AAbortPDU
gdcm::network::AAbortPDU, 80
AAssociateACPDU
gdcm::network::AAssociateACPDU, 82
gdcm::network::AAssociateRQPDU, 90
AAssociateRJPDU
gdcm::network::AAssociateRJPDU, 85
AAssociateRQPDU
gdcm::network::AAssociateACPDU, 83
gdcm::network::AAssociateRQPDU, 88
AEComp
gdcm, 61
AES128_CIPHER
gdcm::CryptographicMessageSyntax, 212
AES192_CIPHER
gdcm::CryptographicMessageSyntax, 212
AES256_CIPHER
gdcm::CryptographicMessageSyntax, 212
ALGOType
gdcm::Segment, 668
ALGOType_END
gdcm::Segment, 668
ARGB
gdcm::PhotometricInterpretation, 578
ARTIMTimer
gdcm::network::ARTIMTimer, 110
AReleaseRPPDU
gdcm::network::AReleaseRPPDU, 107
AReleaseRQPDU
gdcm::network::AReleaseRQPDU, 109
ASComp
gdcm, 61
ASN1
gdcm::ASN1, 111
AUTOMATIC
gdcm::Segment, 668
AXIAL
gdcm::Orientation, 550
AbstractSyntax
gdcm::PresentationContext, 604
gdcm::network::AbstractSyntax, 92
ActiveComponent
vtkImageMapToColors16, 949
Add
gdcm::GroupDict, 374
AddAcceptedPresentationContext
gdcm::network::ULConnection, 860
AddCSAHeaderDictEntry
gdcm::CSAHeaderDict, 224
AddContourReferencedFrameOfReference
vtkRTStructSetProperties, 962
AddDerivationDescription
gdcm::FileDerivation, 338
AddDictEntry
gdcm::Dict, 268
gdcm::PrivateDict, 617
AddFile
gdcm::FileSet, 356
gdcm::SerieHelper, 691
AddFileName
gdcm::SerieHelper, 691
AddFragment
gdcm::SequenceOfFragments, 681
AddFromFile
gdcm::PresentationContextGenerator, 607
AddGroupLength
gdcm::DictConverter, 271
AddIODEntry
gdcm::IOD, 430
AddIOD
gdcm::IODs, 434
AddImageDirectoryRecord
gdcm::DICOMDIRGenerator, 265
AddInput
vtkImageColorViewer, 943
AddInputConnection
vtkImageColorViewer, 943
AddItem
gdcm::SequenceOfItems, 686

AddMacro
gdcm::Macros, 480
gdcm::Module, 506
AddMacroEntry
gdcm::Macro, 478
AddModule
gdcm::Modules, 511
AddModuleEntry
gdcm::Module, 506
gdcm::NestedModuleEntries, 527
AddNewUndefinedLengthItem
gdcm::SequenceOfItems, 686
AddObserver
gdcm::Subject, 745
AddPatientDirectoryRecord
gdcm::DICOMDIRGenerator, 266
AddPresentationContext
gdcm::PresentationContextGenerator, 607
gdcm::network::AAssociateRQPDU, 88
AddPresentationContextAC
gdcm::network::AAssociateACPDU, 82
AddPresentationDataValue
gdcm::network::PDataTFPDU, 564
AddPrimitiveData
gdcm::MeshPrimitive, 497
AddPrivateTag
gdcm::Scanner, 662
gdcm::StrictScanner, 733
AddPurposeOfReferenceCodeSequence
gdcm::FileDerivation, 338
AddQueryDataSet
gdcm::BaseQuery, 147
AddReference
gdcm::FileDerivation, 338
AddReferencedFrameOfReference
vtkRTStructSetProperties, 962
AddRestriction
gdcm::SerieHelper, 692
AddRoleSelectionSub
gdcm::network::UserInformation, 881
AddSOPClassExtendedNegociationSub
gdcm::network::UserInformation, 881
AddSegment
gdcm::SegmentWriter, 677
AddSelect
gdcm::Sorter, 714
AddSeriesDirectoryRecord
gdcm::DICOMDIRGenerator, 266
AddSkipTag
gdcm::Scanner, 662
gdcm::StrictScanner, 733
AddSourceImageSequence
gdcm::FileDerivation, 338
AddStructureSetROIObservation
vtkRTStructSetProperties, 962
AddStructureSetROI
vtkRTStructSetProperties, 962
AddStudyDirectoryRecord
gdcm::DICOMDIRGenerator, 266
AddSurface
gdcm::Segment, 668
AddTag
gdcm::Scanner, 662
gdcm::StrictScanner, 733
AddTransferSyntax
gdcm::PresentationContext, 603
gdcm::network::PresentationContextRQ, 609
AE
gdcm::VR, 898
AffectedSOPClassUID
gdcm::network::CEchoRQ, 183
Allocate
gdcm::LookupTable, 473
AmbulatoryECGWaveformStorage
gdcm::MediaStorage, 487
gdcm::UIDs, 813
AnatomicRegion
gdcm::Segment, 670
AnonymizeEvent
gdcm::AnonymizeEvent, 94
Anonymizer
gdcm::Anonymizer, 98
Append
gdcm::ByteValue, 176
gdcm::Global, 371
AppendFrameEncode
gdcm::ImageCodec, 399
gdcm::JPEG2000Codec, 450
gdcm::JPEGCodec, 457
gdcm::JPEGLSCodec, 463
gdcm::RLECodec, 655
AppendImplementationClassUID
gdcm::FileMetaInformation, 345
AppendRowEncode
gdcm::ImageCodec, 399
gdcm::JPEG2000Codec, 450
gdcm::JPEGCodec, 457
gdcm::JPEGLSCodec, 463
gdcm::RLECodec, 655
AppendToDataElement
gdcm::FileStreamer, 359
AppendToGroupDataElement
gdcm::FileStreamer, 359
ApplicationContext
gdcm::network::ApplicationContext, 103
Apply
gdcm::ImageApplyLookupTable, 387
ApplyInverseVideo

vtkGDCMImageReader, 910
vtkGDCMImageReader2, 916
ApplyLookupTable
vtkGDCMImageReader, 910
vtkGDCMImageReader2, 916
ApplyPlanarConfiguration
vtkGDCMImageReader, 910
vtkGDCMImageReader2, 916
ApplyShiftScale
vtkGDCMImageReader, 910
vtkGDCMImageReader2, 916
ApplyYBRToRGB
vtkGDCMImageReader, 910
vtkGDCMImageReader2, 916
AreOverlaysInPixelData
gdcm::Bitmap, 159
gdcm::Pixmap, 588
Area
gdcm::BoxRegion, 169
gdcm::Region, 649
ArrayIncludeMacrosType
gdcm::Macro, 477
gdcm::Module, 506
ArrayType
gdcm::Attribute, 115
gdcm::Attribute< Group, Element, TVR, VM::VM1 >, 120
gdcm::Attribute< Group, Element, TVR, VM::VM1_n >, 126
AS
gdcm::VR, 898
AsynchronousOperationsWindowSub
gdcm::network::AsynchronousOperationsWindowSub, 112
AT
gdcm::VR, 898
Attribute
gdcm::Attribute< Group, Element, TVR, VM::VM1_n >, 126
gdcm::terminal, 77
Audio
gdcm::MediaStorage, 488
AudioCodec
gdcm::AudioCodec, 136
AudioSRStorageTrialRetired
gdcm::UIDs, 814
AutoPixelMinMax
gdcm::IconImageGenerator, 378

BALCPProtect
gdcm::Anonymizer, 98
BLUE
gdcm::LookupTable, 473
BOOL_FUNCTION_PFILE_PFILE_POINTER
gdcm, 61
backslash
gdcm, 63
BadBigEndian
gdcm::SwapCode, 761
BadLittleEndian
gdcm::SwapCode, 761
BaseQuery
gdcm::BaseQuery, 147
BaseRootQuery
gdcm::BaseRootQuery, 150
BasicAnnotationBoxSOPClass
gdcm::UIDs, 812
BasicApplicationLevelConfidentialityProfile
gdcm::Anonymizer, 98
BasicCodedEntry
gdcm::SegmentHelper::BasicCodedEntry, 153
BasicColorImageBoxSOPClass
gdcm::UIDs, 812
BasicColorPrintManagementMetaSOPClass
gdcm::UIDs, 813
BasicFilmBoxSOPClass
gdcm::UIDs, 812
BasicFilmSessionSOPClass
gdcm::UIDs, 812
BasicGrayscaleImageBoxSOPClass
gdcm::UIDs, 812
BasicGrayscalePrintManagementMetaSOPClass
gdcm::UIDs, 812
BasicOffsetTable
gdcm::BasicOffsetTable, 155
BasicPrintImageOverlayBoxSOPClassRetired
gdcm::UIDs, 813
BasicStudyContentNotificationSOPClassRetired
gdcm::UIDs, 812
BasicTextSRStorage
gdcm::UIDs, 814
BasicTextSR
gdcm::MediaStorage, 487
BasicVoiceAudioWaveformStorage
gdcm::MediaStorage, 487
gdcm::UIDs, 814
Begin
gdcm::CSAHeaderDict, 224
gdcm::DataSet, 249
gdcm::Dict, 268
gdcm::IODs, 434
gdcm::Scanner, 663
gdcm::SequenceOfFragments, 681
gdcm::SequenceOfItems, 686, 687
gdcm::StrictScanner, 733
BigEndian
gdcm::SwapCode, 761
BitSample

gdcm::JPEGCodec, 460
gdcm::LookupTable, 475
Bitmap
gdcm::Bitmap, 159
gdcm::JPEG2000Codec, 452
gdcm::PixelFormat, 585
BitmapToBitmapFilter
gdcm::BitmapToBitmapFilter, 166
black
gdcm::terminal, 77
BlendingSoftcopyPresentationStateStorageSOPClass
gdcm::UIDs, 814
blink
gdcm::terminal, 77
blue
gdcm::terminal, 77
BoundingBox
gdcm::BoxRegion, 169
BoxRegion
gdcm::BoxRegion, 168
BreakConnection
gdcm::network::ULConnectionManager, 867
BreakConnectionNow
gdcm::network::ULConnectionManager, 867
BreastImagingRelevantPatientInformationQuery
gdcm::UIDs, 816
BreastTomosynthesisImageStorage
gdcm::MediaStorage, 488
gdcm::UIDs, 817
bright
gdcm::terminal, 77
Build
vtkLookupTable16, 959
ByteBuffer
gdcm::ByteBuffer, 171
ByteSwap
gdcm::ByteSwapFilter, 173
ByteSwapFilter
gdcm::ByteSwapFilter, 173
ByteValue
gdcm::ByteValue, 176
bytes
gdcm::Tag, 781

C_CANCEL_RQ
gdcm::network::DIMSE, 281
C_ECHO_RSP
gdcm::network::DIMSE, 280
C_ECHO_RQ
gdcm::network::DIMSE, 280
C_FIND_RSP
gdcm::network::DIMSE, 280
C_FIND_RQ
gdcm::network::DIMSE, 280
C_GET_RSP
gdcm::network::DIMSE, 280
C_GET_RQ
gdcm::network::DIMSE, 280
C_MOVE_RSP
gdcm::network::DIMSE, 280
C_MOVE_RQ
gdcm::network::DIMSE, 280
C_STORE_RSP
gdcm::network::DIMSE, 280
C_STORE_RQ
gdcm::network::DIMSE, 280
CALIBRATED
gdcm::Spacing, 717
CAPICryptoFactory
gdcm::CAPICryptoFactory, 179
CAPICryptographicMessageSyntax
gdcm::CAPICryptographicMessageSyntax, 181
CAPI
gdcm::CryptoFactory, 210
CEcho
gdcm::CompositeNetworkFunctions, 204
CFind
gdcm::CompositeNetworkFunctions, 204
CMYK
gdcm::PhotometricInterpretation, 578
cMaxEventID
gdcm::network, 75
cMaxStateID
gdcm::network, 75
CMove
gdcm::CompositeNetworkFunctions, 205
CONDENSED_STYLE
gdcm::Printer, 614
CONSOLE
gdcm::terminal, 77
CORONAL
gdcm::Orientation, 550
CSAElement
gdcm::CSAElement, 215
CSAHeader
gdcm::CSAHeader, 220
gdcm::DataSet, 253
CSAHeaderDict
gdcm::CSAHeaderDict, 224
CSAHeaderDictEntry
gdcm::CSAHeaderDictEntry, 226
CSAHeaderType
gdcm::CSAHeader, 220
CSANonImageStorage
gdcm::MediaStorage, 487
CSComp
gdcm, 61
CSD

gdcm::SegmentHelper::BasicCodedEntry, 154
CStore
gdcm::CompositeNetworkFunctions, 206
CSV
gdcm::SegmentHelper::BasicCodedEntry, 154
CT_private_ELE
gdcm::TransferSyntax, 795
CTImageStorage
gdcm::MediaStorage, 486
gdcm::UIDs, 813
CanCode
gdcm::AudioCodec, 136
gdcm::Coder, 195
gdcm::ImageCodec, 400
gdcm::JPEG2000Codec, 450
gdcm::JPEGCodec, 458
gdcm::JPEGLSCodec, 463
gdcm::KAKADUCodec, 467
gdcm::PDFCodec, 571
gdcm::PGXCodec, 576
gdcm::PNMCodec, 599
gdcm::PVRGCodec, 624
gdcm::RAWCodec, 639
gdcm::RLECodec, 655
CanDecode
gdcm::AudioCodec, 136
gdcm::Decoder, 258
gdcm::DeltaEncodingCodec, 263
gdcm::ImageCodec, 400
gdcm::JPEG2000Codec, 450
gdcm::JPEGCodec, 458
gdcm::JPEGLSCodec, 463
gdcm::KAKADUCodec, 467
gdcm::PDFCodec, 571
gdcm::PGXCodec, 576
gdcm::PNMCodec, 599
gdcm::PVRGCodec, 624
gdcm::RAWCodec, 639
gdcm::RLECodec, 655
CanDisplay
gdcm::VR, 899
CanEmptyTag
gdcm::Anonymizer, 98
CanRead
gdcm::Reader, 644
CanReadFile
vtkGDCMImageReader, 907
vtkGDCMImageReader2, 913
CanReadImage
gdcm::StreamImageReader, 722
CanStoreLossy
gdcm::TransferSyntax, 795
CanWriteFile
gdcm::StreamImageWriter, 727
CardiacElectrophysiologyWaveformStorage
gdcm::MediaStorage, 487
gdcm::UIDs, 814
CardiacRelevantPatientInformationQuery
gdcm::UIDs, 816
Change
gdcm::FileChangeTransferSyntax, 333
gdcm::FileDecompressLookupTable, 336
gdcm::FileExplicitFilter, 341
gdcm::ImageChangePhotometricInterpretation, 389
gdcm::ImageChangePlanarConfiguration, 392
gdcm::ImageChangeTransferSyntax, 395
ChangeFMI
gdcm::FileExplicitFilter, 341
ChangeMonochrome
gdcm::ImageChangePhotometricInterpretation, 389
CharacterDataHandler
gdcm::TableReader, 772
gdcm::XMLDictReader, 975
gdcm::XMLPrivateDictReader, 979
CheckDataElement
gdcm::FileStreamer, 359
CheckEvent
gdcm::AnonymizeEvent, 94
gdcm::DataEvent, 245
gdcm::DataSetEvent, 255
gdcm::Event, 316
gdcm::FileNameEvent, 352
gdcm::ProgressEvent, 622
CheckFileMetaInformationOff
gdcm::Writer, 972
CheckFileMetaInformationOn
gdcm::Writer, 972
CheckTemplateFileName
gdcm::FileStreamer, 359
ChestCADSRStorage
gdcm::UIDs, 815
CipherTypes
gdcm::CryptographicMessageSyntax, 212
Clear
gdcm::Bitmap, 159
gdcm::ByteValue, 176
gdcm::DataElement, 237
gdcm::DataSet, 249
gdcm::IODs, 434
gdcm::IOD, 430
gdcm::Item, 441
gdcm::LookupTable, 473
gdcm::Macro, 478
gdcm::Macros, 480
gdcm::Module, 506
gdcm::Modules, 511
gdcm::Preamble, 601
gdcm::SequenceOfFragments, 681

gdcm::SequenceOfItems, 687
gdcm::SerieHelper, 692
gdcm::Value, 886
vtkGDCMMedicalImageProperties, 924
vtkRTStructSetProperties, 962
ClearInternalUIDs
gdcm::Anonymizer, 98
ClearSkipTags
gdcm::Scanner, 663
gdcm::StrictScanner, 733
ClearTags
gdcm::Scanner, 663
gdcm::StrictScanner, 733
Clone
gdcm::BoxRegion, 169
gdcm::ImageCodec, 400
gdcm::JPEG2000Codec, 451
gdcm::JPEGCodec, 458
gdcm::JPEGLSCodec, 463
gdcm::KAKADUCodec, 468
gdcm::PGXCodec, 577
gdcm::PNMCodec, 599
gdcm::PVRGCodec, 624
gdcm::RAWCodec, 640
gdcm::RLECodec, 655
gdcm::Region, 649
CM
gdcm::SegmentHelper::BasicCodedEntry, 153
Code
gdcm::Coder, 195
gdcm::JPEG2000Codec, 451
gdcm::JPEGCodec, 458
gdcm::JPEGLSCodec, 463
gdcm::JSON, 465
gdcm::KAKADUCodec, 468
gdcm::PVRGCodec, 625
gdcm::RAWCodec, 640
gdcm::RLECodec, 656
CodeMeaning
gdcm::RealWorldValueMappingContent, 647
CodeString
gdcm::CodeString, 197
CodeValue
gdcm::RealWorldValueMappingContent, 647
Color
gdcm::terminal, 77
ColorArray
gdcm::SurfaceHelper, 754
ColorSoftcopyPresentationStateStorageSOPClass
gdcm::UIDs, 814
Command
gdcm::Command, 199
CommandDataSet
gdcm::CommandDataSet, 201
CommandTypes
gdcm::network::DIMSE, 280
CompOperators
gdcm, 61
Compatible
gdcm::VM, 894
gdcm::VR, 899
Component
gdcm::PersonName, 575
ComprehensiveSRStorage
gdcm::UIDs, 815
ComprehensiveSRStorageTrialRetired
gdcm::UIDs, 814
ComprehensiveSR
gdcm::MediaStorage, 487
CompressionTypes
vtkGDCMImageWriter, 919
Compute
gdcm::MD5, 482
gdcm::SHA1, 701
ComputeBoundingBox
gdcm::BoxRegion, 169
gdcm::Region, 649
ComputeBufferLength
gdcm::ImageRegionReader, 418
ComputeByteLength
gdcm::SequenceOfFragments, 681
ComputeDataElement
gdcm::DataSet, 249
ComputeDataSetMediaStorageSOPClass
gdcm::FileMetaInformation, 345
ComputeDataSetTransferSyntax
gdcm::FileMetaInformation, 345
ComputeDistAlongNormal
gdcm::DirectionCosines, 282
ComputeFile
gdcm::MD5, 482
gdcm::SHA1, 701
ComputeFileMD5
gdcm::Testing, 785
ComputeGroupLength
gdcm::DataSet, 249
ComputeInterceptSlopePixelType
gdcm::Rescaler, 651
ComputeLength
gdcm::ByteValue, 176
gdcm::Fragment, 369
gdcm::SequenceOfFragments, 681
gdcm::SequenceOfItems, 687
ComputeLossyFlag
gdcm::Bitmap, 159
ComputeMD5
gdcm::Testing, 785
ComputeMOSAICDimensions

gdcm::SplitMosaicFilter, 718
ComputeMediaStorageFromModality
gdcm::ImageHelper, 409
ComputeNumberOfSurfaces
gdcm::SurfaceWriter, 760
ComputeOffsetTable
gdcm::JPEGCodec, 458
ComputePixelAspectRatioFromPixelSpacing
gdcm::Spacing, 717
ComputePixelTypeFromMinMax
gdcm::Rescaler, 651
ComputeSpacingFromImagePositionPatient
gdcm::ImageHelper, 409
ComputeTargetMediaStorage
gdcm::ImageWriter, 423
ComputeVR
gdcm::DataSetHelper, 256
ComputeZSpacing
gdcm::IPPSorter, 439
ComputedRadiographyImageStorage
gdcm::MediaStorage, 486
gdcm::UIDs, 813
ConcatenatePDVBlobs
gdcm::network::PresentationDataValue, 611
ConcatenatePDVBlobsAsExplicit
gdcm::network::PresentationDataValue, 611
Conditional
gdcm::Usage, 878
const
gdcm::SOPClassUIDToIOD, 711
const_iterator
gdcm::CodeString, 197
gdcm::LO, 470
gdcm::String, 738
const_reference
gdcm::CodeString, 197
gdcm::LO, 470
gdcm::String, 738
const_reverse_iterator
gdcm::CodeString, 197
gdcm::LO, 470
gdcm::String, 738
ConstCharWrapper
gdcm::ConstCharWrapper, 207
ConstIterator
gdcm::CSAHeaderDict, 224
gdcm::DataSet, 249
gdcm::Dict, 268
gdcm::Scanner, 662
gdcm::SequenceOfFragments, 680
gdcm::SequenceOfItems, 686
gdcm::StrictScanner, 732
Construct
gdcm::BaseRootQuery, 150
ConstructAbortPDU
gdcm::network::PDUFactory, 573
ConstructCEchoRQ
gdcm::network::CompositeMessageFactory, 203
ConstructCFindRQ
gdcm::network::CompositeMessageFactory, 203
ConstructCMoveRQ
gdcm::network::CompositeMessageFactory, 203
ConstructCStoreRSP
gdcm::network::CompositeMessageFactory, 203
ConstructCStoreRQ
gdcm::network::CompositeMessageFactory, 203
ConstructFromString
gdcm::TagPath, 782
ConstructFromTagList
gdcm::TagPath, 782
ConstructNAction
gdcm::network::NormalizedMessageFactory, 534
ConstructNCreate
gdcm::network::NormalizedMessageFactory, 534
ConstructNDelete
gdcm::network::NormalizedMessageFactory, 534
ConstructNEventReport
gdcm::network::NormalizedMessageFactory, 534
ConstructNGet
gdcm::network::NormalizedMessageFactory, 534
ConstructNSet
gdcm::network::NormalizedMessageFactory, 534
ConstructPDVByDataSet
gdcm::network::CEchoRSP, 185
gdcm::network::CFindCancelRQ, 186
gdcm::network::CFindRSP, 189
gdcm::network::CMoveCancelRq, 190
gdcm::network::CMoveRSP, 193
gdcm::network::NActionRSP, 519
gdcm::network::NCreateRSP, 522
gdcm::network::NDeleteRSP, 525
gdcm::network::NEventReportRSP, 530
gdcm::network::NGetRSP, 533
gdcm::network::NSetRSP, 539
ConstructPDU
gdcm::network::PDUFactory, 573
ConstructPDV
gdcm::network::BaseCompositeMessage, 140
gdcm::network::BaseNormalizedMessage, 142
gdcm::network::CEchoRQ, 183
gdcm::network::CFindRQ, 188
gdcm::network::CMoveRQ, 191
gdcm::network::CStoreRSP, 230
gdcm::network::CStoreRQ, 228
gdcm::network::NActionRQ, 518
gdcm::network::NCreateRQ, 521
gdcm::network::NDeleteRQ, 523
gdcm::network::NEventReportRQ, 529

gdcm::network::NGetRQ, 531
gdcm::network::NSetRQ, 537
ConstructQuery
gdcm::CompositeNetworkFunctions, 205, 206
gdcm::NormalizedNetworkFunctions, 536
ConstructReleasePDU
gdcm::network::PDUFactory, 573
ConstructorType
gdcm::Dicts, 278
Convert
gdcm::DictConverter, 271
gdcm::ImageConverter, 405
ConvertRGBToPaletteColor
gdcm::IconImageGenerator, 378
ConvertToCXX
gdcm::DictConverter, 271
ConvertToXML
gdcm::DictConverter, 271
Create
gdcm::Preamble, 601
CreateCEchoPDU
gdcm::network::PDUFactory, 573
CreateCFindPDU
gdcm::network::PDUFactory, 573
CreateCMSProvider
gdcm::CAPICryptoFactory, 180
gdcm::CryptoFactory, 211
gdcm::OpenSSLCryptoFactory, 542
gdcm::OpenSSLP7CryptoFactory, 546
CreateCMovePDU
gdcm::network::PDUFactory, 573
CreateCStoreRQPDU
gdcm::network::PDUFactory, 573
CreateCStoreRSPPDU
gdcm::network::PDUFactory, 573
CreateDefaultUniqueSeriesIdentifier
gdcm::SerieHelper, 692
CreateNActionPDU
gdcm::network::PDUFactory, 573
CreateNCreatePDU
gdcm::network::PDUFactory, 573
CreateNDeletePDU
gdcm::network::PDUFactory, 573
CreateNEventReportPDU
gdcm::network::PDUFactory, 573
CreateNGetPDU
gdcm::network::PDUFactory, 573
CreateNSetPDU
gdcm::network::PDUFactory, 573
CreateUniqueSeriesIdentifier
gdcm::SerieHelper, 692
Cross
gdcm::DirectionCosines, 282
CrossDot
gdcm::DirectionCosines, 282
CryptoFactory
gdcm::CryptoFactory, 210
CryptoLib
gdcm::CryptoFactory, 210
CryptographicMessageSyntax
gdcm::CryptographicMessageSyntax, 212
CS
gdcm::VR, 898
Curve
gdcm::Curve, 232
vtkGDCMImageReader, 910
vtkGDCMImageReader2, 916
Curves
gdcm::Pixmap, 589
CV
gdcm::SegmentHelper::BasicCodedEntry, 154
cyan
gdcm::terminal, 77

DAComp
gdcm, 61
DATASET_FORMAT
gdcm::CSAHeader, 220
DEFAULT
gdcm::CryptoFactory, 210
DES3_CIPHER
gdcm::CryptographicMessageSyntax, 212
DETECTOR
gdcm::Spacing, 717
DICOMApplicationContextName
gdcm::UIDs, 812
DICOMControlledTerminology
gdcm::UIDs, 812
DICOMDIRGenerator
gdcm::DICOMDIRGenerator, 265
DICOMDIR
gdcm::DICOMDIR, 264
DICOMUIDRegistry
gdcm::UIDs, 812
DICT_DEBUG
gdcm::DictConverter, 270
DICT_DEFAULT
gdcm::DictConverter, 270
DICT_XML
gdcm::DictConverter, 270
DTComp
gdcm, 61
DA
gdcm::VR, 898
DataElement
gdcm::DataElement, 237
gdcm::Value, 886
DataElementSet

gdcm::DataSet, 249
DataElementType
gdcm::ModuleEntry, 510
DataEvent
gdcm::DataEvent, 245
DataField
gdcm::CSAElement, 217
DataPtr
gdcm::CSAElement, 215
DataSetEvent
gdcm::DataSetEvent, 255
DataSetHandled
gdcm::network::ULConnectionCallback, 862
DataSetHandles
gdcm::network::ULConnectionCallback, 862
DataSetMS
gdcm::FileMetaInformation, 347
DataSetTS
gdcm::FileMetaInformation, 347
DataWasPassed
vtkImageMapToColors16, 949
DebugOff
gdcm::Trace, 790
DebugOn
gdcm::Trace, 790
Decode
gdcm::AudioCodec, 137
gdcm::Base64, 137
gdcm::Curve, 232
gdcm::Decoder, 258
gdcm::DeltaEncodingCodec, 263
gdcm::ImageCodec, 400
gdcm::JPEG2000Codec, 451
gdcm::JPEGCodec, 458
gdcm::JPEGLSCodec, 463, 464
gdcm::JSON, 465
gdcm::KAKADUCodec, 468
gdcm::LookupTable, 473
gdcm::PDFCodec, 571
gdcm::PVRGCodec, 625
gdcm::RAWCodec, 640
gdcm::RLECodec, 656
DecodeByStreams
gdcm::Decoder, 258
gdcm::ImageCodec, 400
gdcm::JPEG12Codec, 445
gdcm::JPEG16Codec, 447
gdcm::JPEG2000Codec, 451
gdcm::JPEG8Codec, 454
gdcm::JPEGCodec, 458
gdcm::RAWCodec, 640
gdcm::RLECodec, 656
DecodeBytes
gdcm::RAWCodec, 640
DecodeExtent
gdcm::JPEG2000Codec, 451
gdcm::JPEGCodec, 458
gdcm::JPEGLSCodec, 464
gdcm::RLECodec, 656
Decompress
gdcm::Overlay, 554
Decrypt
gdcm::CAPICryptographicMessageSyntax, 181
gdcm::CryptographicMessageSyntax, 212
gdcm::OpenSSLCryptographicMessageSyntax, 544
gdcm::OpenSSLP7CryptographicMessageSyntax, 548
DeepCopy
vtkRTStructSetProperties, 962
Default
gdcm::FileMetaInformation, 345
DefinePixelExtent
gdcm::StreamImageReader, 722
gdcm::StreamImageWriter, 727
DefineProperBufferLength
gdcm::StreamImageReader, 722
gdcm::StreamImageWriter, 727
DefinedTerms
gdcm::DefinedTerms, 259
DeflatedExplicitVRLittleEndian
gdcm::TransferSyntax, 795
gdcm::UIDs, 810
DeformableSpatialRegistrationStorage
gdcm::UIDs, 814
Defs
gdcm::Defs, 260
DeleteDirectory
gdcm::System, 765
DeltaEncodingCodec
gdcm::DeltaEncodingCodec, 263
Derive
gdcm::FileDerivation, 338
Description
gdcm::ModuleEntry, 509
DescriptionField
gdcm::ModuleEntry, 510
DetachedInterpretationManagementSOPClassRetired
gdcm::UIDs, 812
DetachedPatientManagementMetaSOPClassRetired
gdcm::UIDs, 812
DetachedPatientManagementSOPClass
gdcm::MediaStorage, 487
DetachedPatientManagementSOPClassRetired
gdcm::UIDs, 812
DetachedResultsManagementMetaSOPClassRetired
gdcm::UIDs, 812
DetachedResultsManagementSOPClassRetired
gdcm::UIDs, 812

DetachedStudyManagementMetaSOPClassRetired
gdcm::UIDs, 812
DetachedStudyManagementSOPClass
gdcm::MediaStorage, 487
DetachedStudyManagementSOPClassRetired
gdcm::UIDs, 812
DetachedVisitManagementSOPClass
gdcm::MediaStorage, 487
DetachedVisitManagementSOPClassRetired
gdcm::UIDs, 812
DetailSRStorageTrialRetired
gdcm::UIDs, 814
DetermineEventByPDU
gdcm::network::PDUFactory, 573
dicomAETitle
gdcm::UIDs, 816
dicomApplicationCluster
gdcm::UIDs, 816
dicomAssociationAcceptor
gdcm::UIDs, 816
dicomAssociationInitiator
gdcm::UIDs, 816
dicomAuthorizedNodeCertificateReference
gdcm::UIDs, 816
dicomConfigurationRoot
gdcm::UIDs, 816
dicomDescription
gdcm::UIDs, 816
dicomDevice
gdcm::UIDs, 817
dicomDeviceName
gdcm::UIDs, 816
dicomDeviceSerialNumber
gdcm::UIDs, 816
dicomDevicesRoot
gdcm::UIDs, 816
dicomHostname
gdcm::UIDs, 816
dicomInstalled
gdcm::UIDs, 816
dicomInstitutionAddress
gdcm::UIDs, 816
dicomInstitutionDepartmentName
gdcm::UIDs, 816
dicomInstitutionName
gdcm::UIDs, 816
dicomIssuerOfPatientID
gdcm::UIDs, 816
dicomManufacturer
gdcm::UIDs, 816
dicomManufacturerModelName
gdcm::UIDs, 816
dicomNetworkAE
gdcm::UIDs, 817
dicomNetworkConnection
gdcm::UIDs, 817
dicomNetworkConnectionReference
gdcm::UIDs, 816
dicomPort
gdcm::UIDs, 816
dicomPreferredCalledAETitle
gdcm::UIDs, 816
dicomPreferredCallingAETitle
gdcm::UIDs, 816
dicomPrimaryDeviceType
gdcm::UIDs, 816
dicomRelatedDeviceReference
gdcm::UIDs, 816
dicomSOPClass
gdcm::UIDs, 816
dicomSoftwareVersion
gdcm::UIDs, 816
dicomStationName
gdcm::UIDs, 816
dicomSupportedCharacterSet
gdcm::UIDs, 816
dicomTLSCyphersuite
gdcm::UIDs, 816
dicomThisNodeCertificateReference
gdcm::UIDs, 816
dicomTransferCapability
gdcm::UIDs, 817
dicomTransferRole
gdcm::UIDs, 816
dicomTransferSyntax
gdcm::UIDs, 816
dicomUniqueAETitle
gdcm::UIDs, 817
dicomUniqueAETitlesRegistryRoot
gdcm::UIDs, 817
dicomVendorData
gdcm::UIDs, 816
Dict
gdcm::Dict, 268
gdcm::DictEntry, 274
DictConverter
gdcm::DictConverter, 271
DictEntry
gdcm::DictEntry, 273
DictPrinter
gdcm::DictPrinter, 276
Dicts
gdcm::CSAHeaderDict, 224
gdcm::Dict, 269
gdcm::Dicts, 278
gdcm::PrivateDict, 617
difference_type
gdcm::CodeString, 197

gdcm::LO, 470
gdcm::String, 738
DigitalIntraoralXRayImageStorageForPresentation
gdcm::UIDs, 813
DigitalIntraoralXRayImageStorageForProcessing
gdcm::MediaStorage, 486
gdcm::UIDs, 813
DigitalIntraoralXrayImageStorageForPresentation
gdcm::MediaStorage, 486
DigitalMammographyImageStorageForPresentation
gdcm::MediaStorage, 486
DigitalMammographyImageStorageForProcessing
gdcm::MediaStorage, 486
DigitalMammographyXRayImageStorageForPresentation
gdcm::UIDs, 813
DigitalMammographyXRayImageStorageForProcessing
gdcm::UIDs, 813
DigitalXRayImageStorageForPresentation
gdcm::MediaStorage, 486
gdcm::UIDs, 813
DigitalXRayImageStorageForProcessing
gdcm::MediaStorage, 486
gdcm::UIDs, 813
dim
gdcm::terminal, 77
Dimensions
gdcm::Bitmap, 164
gdcm::ImageCodec, 403
DirCosTolerance
gdcm::IPPSorter, 439
DirectionCosines
gdcm::DirectionCosines, 282
vtkGDCMImageReader, 910
vtkGDCMImageReader2, 916
Directory
gdcm::Directory, 285
DoByteSwap
gdcm::ImageCodec, 400
DoIconImage
gdcm::PixmapWriter, 596
DoInvertMonochrome
gdcm::ImageCodec, 401
DoOverlayCleanup
gdcm::ImageCodec, 401
DoPaddedCompositePixelCode
gdcm::ImageCodec, 401
DoPlanarConfiguration
gdcm::ImageCodec, 401
DoSimpleCopy
gdcm::ImageCodec, 401
DoYBR
gdcm::ImageCodec, 401
Dot
gdcm::DirectionCosines, 282
DropDuplicatePositions
gdcm::IPPSorter, 439
DS
gdcm::VR, 898
DT
gdcm::VR, 898
Dumper
gdcm::Dumper, 290
DuplicateAttributeError
gdcm::Parser, 561

eAABORTPDUReceivedOpen
gdcm::network, 74
eAABORTRequest
gdcm::network, 74
eAASSOCIATE_RQPDUreceived
gdcm::network, 74
eAASSOCIATERequestLocalUser
gdcm::network, 74
eAASSOCIATEresponseAccept
gdcm::network, 74
eAASSOCIATEresponseReject
gdcm::network, 74
eARELEASE_RPPDUReceived
gdcm::network, 74
eARELEASE_RQPDUReceivedOpen
gdcm::network, 74
eARELEASERequest
gdcm::network, 74
eARELEASEResponse
gdcm::network, 74
eARTIMTimerExpired
gdcm::network, 74
eASSOCIATE_ACPDUreceived
gdcm::network, 74
eASSOCIATE_RJPDUreceived
gdcm::network, 74
eArabic
gdcm, 62
ECharSet
gdcm, 61
eCreateMMPS
gdcm, 62
eCyrillic
gdcm, 62
EDGE
gdcm::MeshPrimitive, 497
eEventDoesNotExist
gdcm::network, 74
EEventID
gdcm::network, 74
eFind
gdcm, 63
eGB18030

gdcm, 62
eGreek
gdcm, 62
eHebrew
gdcm, 62
eImage
gdcm, 62
eJapanese
gdcm, 62
eJapaneseKanjiMultibyte
gdcm, 62
eJapaneseSupplementaryKanjiMultibyte
gdcm, 62
eKoreanHangulHanjaMultibyte
gdcm, 62
eLatin1
gdcm, 62
eLatin2
gdcm, 62
eLatin3
gdcm, 62
eLatin4
gdcm, 62
eLatin5
gdcm, 62
eMove
gdcm, 63
ENQueryType
gdcm, 62
ePDATATFPDU
gdcm::network, 74
ePDATArequest
gdcm::network, 74
ePatient
gdcm, 62
ePatientRootType
gdcm, 63
EQueryLevel
gdcm, 62
EQueryType
gdcm, 62
ERootType
gdcm, 63
eSeries
gdcm, 62
eSetMMPS
gdcm, 62
eSta10ReleaseCollisionAc
gdcm::network, 75
eSta11ReleaseCollisionRq
gdcm::network, 75
eSta12ReleaseCollisionAcLocal
gdcm::network, 75
eSta13AwaitingClose
gdcm::network, 75
eSta1Idle
gdcm::network, 75
eSta2Open
gdcm::network, 75
eSta3WaitLocalAssoc
gdcm::network, 75
eSta4LocalAssocDone
gdcm::network, 75
eSta5WaitRemoteAssoc
gdcm::network, 75
eSta6TransferReady
gdcm::network, 75
eSta7WaitRelease
gdcm::network, 75
eSta8WaitLocalRelease
gdcm::network, 75
eSta9ReleaseCollisionRqLocal
gdcm::network, 75
eStaDoesNotExist
gdcm::network, 75
EStateID
gdcm::network, 74
eStudy
gdcm, 62
eStudyRootType
gdcm, 63
eThai
gdcm, 62
eTransportConnConfirmLocal
gdcm::network, 74
eTransportConnIndicLocal
gdcm::network, 74
eTransportConnectionClosed
gdcm::network, 74
eUTF8
gdcm, 62
eUnrecognizedPDUReceived
gdcm::network, 74
eWLMFind
gdcm, 63
elem
gdcm::SerieHelper::Rule, 659
Element
gdcm::Element< TVR, VM::VM1_n >, 296
Empty
gdcm::Anonymizer, 98
gdcm::BoxRegion, 169
gdcm::DataElement, 237
gdcm::FileAnonymizer, 330
gdcm::Region, 649
EncapsulatedCDAStorage
gdcm::MediaStorage, 487
gdcm::UIDs, 815

EncapsulatedDocument
gdcm::EncapsulatedDocument, 310
EncapsulatedPDFStorage
gdcm::MediaStorage, 487
gdcm::UIDs, 815
Encode
gdcm::Base64, 138
EncodeBuffer
gdcm::JPEG12Codec, 445
gdcm::JPEG16Codec, 447
gdcm::JPEG8Codec, 454
gdcm::JPEGCodec, 459
EncodeBytes
gdcm::System, 765
Encrypt
gdcm::CAPICryptographicMessageSyntax, 181
gdcm::CryptographicMessageSyntax, 212
gdcm::OpenSSLCryptographicMessageSyntax, 544
gdcm::OpenSSLP7CryptographicMessageSyntax, 548
End
gdcm::CSAHeaderDict, 224
gdcm::DataSet, 249
gdcm::Dict, 268
gdcm::IODs, 435
gdcm::Scanner, 663
gdcm::SequenceOfFragments, 681
gdcm::SequenceOfItems, 687
gdcm::StrictScanner, 733
EndElement
gdcm::TableReader, 772
gdcm::XMLDictReader, 975
gdcm::XMLPrivateDictReader, 979
EndElementHandler
gdcm::Parser, 561
EndFilter
gdcm::SimpleSubjectWatcher, 705
EndWith
gdcm::Filename, 348
EnhancedCTImageStorage
gdcm::MediaStorage, 486
gdcm::UIDs, 813
EnhancedMRImageStorage
gdcm::MediaStorage, 486
gdcm::UIDs, 813
EnhancedPETImageStorage
gdcm::MediaStorage, 488
EnhancedSRStorage
gdcm::UIDs, 814
EnhancedSR
gdcm::MediaStorage, 487
EnhancedUSVolumeStorage
gdcm::MediaStorage, 488
gdcm::UIDs, 817
EnhancedXAImageStorage
gdcm::MediaStorage, 488
gdcm::UIDs, 814
EnhancedXRFImageStorage
gdcm::UIDs, 814
EnumeratedValues
gdcm::EnumeratedValues, 314
ErrorOff
gdcm::Trace, 790
ErrorOn
gdcm::Trace, 790
ErrorType
gdcm::Parser, 561
EstablishConnection
gdcm::network::ULConnectionManager, 867
EstablishConnectionMove
gdcm::network::ULConnectionManager, 867
Event
gdcm::Event, 316
Exception
gdcm::Exception, 318
Execute
gdcm::Command, 199
gdcm::MemberCommand, 493
gdcm::SimpleMemberCommand, 703
ExecuteData
vtkGDCMImageReader, 907
vtkGDCMThreadedImageReader, 935
ExecuteInformation
vtkGDCMImageReader, 907
vtkGDCMThreadedImageReader, 935
ExecuteQuery
gdcm::StringFilter, 741
Explicit
gdcm::TransferSyntax, 794
ExplicitVRBigEndian
gdcm::TransferSyntax, 795
gdcm::UIDs, 810
ExplicitVRLittleEndian
gdcm::TransferSyntax, 795
gdcm::UIDs, 810
Explore
gdcm::Directory, 285
Extract
gdcm::IconImageFilter, 376
ExtractIconImages
gdcm::IconImageFilter, 376
ExtractVeproIconImages
gdcm::IconImageFilter, 376

F
gdcm::Printer, 616
gdcm::Reader, 646
gdcm::Validate, 884

gdcm::XMLPrinter, 977
FACET
gdcm::MeshPrimitive, 497
FLOAT16
gdcm::PixelFormat, 582
FLOAT32
gdcm::PixelFormat, 582
FLOAT64
gdcm::PixelFormat, 582
FD
gdcm::VR, 898
Fiducials
gdcm::Fiducials, 324
File
gdcm::File, 327
FileAnonymizer
gdcm::FileAnonymizer, 330
FileChangeTransferSyntax
gdcm::FileChangeTransferSyntax, 333
gdcm::ImageCodec, 403
FileDecompressLookupTable
gdcm::FileDecompressLookupTable, 336
FileDerivation
gdcm::FileDerivation, 338
FileExists
gdcm::System, 765
FileExplicitFilter
gdcm::FileExplicitFilter, 341
FileIsDirectory
gdcm::System, 765
FileIsSymlink
gdcm::System, 765
FileList
gdcm, 61
FileMetaInformation
gdcm::FileMetaInformation, 344
FileName
vtkGDCMPolyDataReader, 927
FileNameEvent
gdcm::FileNameEvent, 352
FileNameOrdering
gdcm::SerieHelper, 692
FileNames
vtkGDCMImageReader, 910
FileSet
gdcm::FileSet, 356
FileSize
gdcm::System, 765
FileStreamer
gdcm::FileStreamer, 359
FileTime
gdcm::System, 766
FileType
gdcm::FileSet, 356
FileWithName
gdcm::FileWithName, 362
Filename
gdcm::Filename, 348
filename
gdcm::FileWithName, 362
FilenameGenerator
gdcm::FilenameGenerator, 354
FilenameType
gdcm::DICOMDIRGenerator, 265
gdcm::Directory, 284
gdcm::FilenameGenerator, 354
Filenames
gdcm::Sorter, 715
FilenamesType
gdcm::DICOMDIRGenerator, 265
gdcm::Directory, 284
gdcm::FilenameGenerator, 354
FilesType
gdcm::FileSet, 356
Fill
gdcm::ByteValue, 176
FillFromDataSet
gdcm::FileMetaInformation, 345
FillMedicalImageInformation
vtkGDCMImageReader, 907
vtkGDCMImageReader2, 913
vtkGDCMPolyDataReader, 926
FindCSAElementByName
gdcm::CSAHeader, 220
FindContext
gdcm::network::ULConnection, 860
FindDataElement
gdcm::DataSet, 250
gdcm::Item, 441
gdcm::SequenceOfItems, 687
FindDictEntry
gdcm::PrivateDict, 617
FindMacroEntry
gdcm::Macro, 478
FindModuleEntryInMacros
gdcm::Module, 506
FindNextDataElement
gdcm::DataSet, 250
FindPDBElementByName
gdcm::PDBHeader, 568
FindPatientRootQuery
gdcm::FindPatientRootQuery, 364
FindStudyRootQuery
gdcm::FindStudyRootQuery, 366
FirstRender
vtkImageColorViewer, 946
FL
gdcm::VR, 898

ForceRescale
vtkGDCMImageReader, 910
vtkGDCMImageReader2, 916
FormatDateTime
gdcm::System, 766
Fragment
gdcm::Fragment, 369
FragmentVector
gdcm::SequenceOfFragments, 680
FromString
gdcm::StringFilter, 741
FujiPrivateCRImageStorage
gdcm::MediaStorage, 488

GDCM_DIFFERENT
gdcm, 61
GDCM_DO_JOIN2
gdcmStaticAssert.h, 1194
GDCM_DO_JOIN
gdcmStaticAssert.h, 1194
GDCM_EQUAL
gdcm, 61
GDCM_EXPORT
gdcmWin32.h, 1250
GDCM_FUNCTION
gdcmTrace.h, 1217
GDCM_GREATEROREQUAL
gdcm, 61
GDCM_GREATER
gdcm, 61
GDCM_JOIN
gdcmStaticAssert.h, 1194
GDCM_LEGACY_BODY
gdcmLegacyMacro.h, 1102
GDCM_LEGACY_REPLACED_BODY
gdcmLegacyMacro.h, 1102
GDCM_LEGACY
gdcmLegacyMacro.h, 1102
GDCM_LESSOREQUAL
gdcm, 61
GDCM_LESS
gdcm, 61
GDCM_STATIC_ASSERT
gdcm::Attribute, 115
gdcm::Attribute< Group, Element, TVR, VM::VM1 >, 120
gdcm::Attribute< Group, Element, TVR, VM::VM1_n >, 127
gdcmStaticAssert.h, 1194
GDCMMACROENTRY_H
gdcmMacroEntry.h, 1107
GEMS
gdcm::Dicts, 278
GEPrivate3DModelStorage
gdcm::MediaStorage, 487
GRAY
gdcm::LookupTable, 473
GREEN
gdcm::LookupTable, 473
gdcm, 45
AEComp, 61
ASComp, 61
BOOL_FUNCTION_PFILE_PFILE_POINTER, 61
backslash, 63
CSComp, 61
CompOperators, 61
DAComp, 61
DTComp, 61
eArabic, 62
ECharSet, 61
eCreateMMPS, 62
eCyrillic, 62
eFind, 63
eGB18030, 62
eGreek, 62
eHebrew, 62
eImage, 62
eJapanese, 62
eJapaneseKanjiMultibyte, 62
eJapaneseSupplementaryKanjiMultibyte, 62
eKoreanHangulHanjaMultibyte, 62
eLatin1, 62
eLatin2, 62
eLatin3, 62
eLatin4, 62
eLatin5, 62
eMove, 63
ENQueryType, 62
ePatient, 62
ePatientRootType, 63
EQueryLevel, 62
EQueryType, 62
ERootType, 63
eSeries, 62
eSetMMPS, 62
eStudy, 62
eStudyRootType, 63
eThai, 62
eUTF8, 62
eWLMFind, 63
FileList, 61
GDCM_DIFFERENT, 61
GDCM_EQUAL, 61
GDCM_GREATEROREQUAL, 61
GDCM_GREATER, 61
GDCM_LESSOREQUAL, 61
GDCM_LESS, 61
GetVRFromTag, 63

GlobalInstance, 69
IconImage, 61
LD_ALL, 63
LD_NOSEQ, 63
LD_NOSHADOWSEQ, 63
LD_NOSHADOW, 63
LOComp, 61
LTComp, 61
LodModeType, 63
MacroEntry, 61
NestedMacroEntries, 61
operator!=, 63
operator<<, 63–68
operator>>, 68, 69
operator==, 68
PNComp, 61
SHComp, 61
STComp, 61
TMComp, 61
TYPETOENCODING, 69
to_string, 69
UIComp, 61
UTComp, 61
VRBINARY, 69
gdcm::ASN1, 111
~ASN1, 111
ASN1, 111
ParseDump, 111
ParseDumpFile, 111
TestPBKDF2, 111
gdcm::AbortEvent, 90
gdcm::AnonymizeEvent, 93
~AnonymizeEvent, 94
AnonymizeEvent, 94
CheckEvent, 94
GetEventName, 94
GetTag, 94
MakeObject, 95
Self, 94
SetTag, 95
Superclass, 94
gdcm::Anonymizer, 95
~Anonymizer, 98
Anonymizer, 98
BALCPProtect, 98
BasicApplicationLevelConfidentialityProfile, 98
CanEmptyTag, 98
ClearInternalUIDs, 98
Empty, 98
GetBasicApplicationLevelConfidentialityProfile↩
Attributes, 98
GetCryptographicMessageSyntax, 99
GetFile, 99
New, 99
RecurseDataSet, 99
Remove, 99
RemoveGroupLength, 99
RemovePrivateTags, 99
RemoveRetired, 99
Replace, 99, 100
SetCryptographicMessageSyntax, 100
SetFile, 100
gdcm::AnyEvent, 100
gdcm::ApplicationEntity, 104
Internal, 105
IsValid, 105
MaxLength, 105
MaxNumberOfComponents, 105
Padding, 105
Print, 105
Separator, 105
SetBlob, 105
Squeeze, 105
gdcm::Attribute
ArrayType, 115
GDCM_STATIC_ASSERT, 115
GetAsDataElement, 115
GetDictVM, 115
GetDictVR, 115
GetNumberOfValues, 115
GetTag, 115
GetValue, 116
GetValues, 116
GetVM, 116
GetVR, 116
Internal, 118
operator!=, 116
operator<, 116
operator==, 116
operator[], 116, 117
Print, 117
Set, 117
SetByteValue, 117
SetByteValueNoSwap, 117
SetFromDataElement, 117
SetFromDataSet, 117
SetValue, 117
SetValues, 118
VMType, 115
gdcm::Attribute< Group, Element, TVR, TVM >, 113
gdcm::Attribute< Group, Element, TVR, VM::VM1 >, 118
ArrayType, 120
GDCM_STATIC_ASSERT, 120
GetAsDataElement, 120
GetDictVM, 120
GetDictVR, 120
GetNumberOfValues, 120
GetTag, 120

GetValue, 121
GetValues, 121
GetVM, 121
GetVR, 121
Internal, 122
operator!=, 121
operator<, 121
operator==, 121
Print, 121
Set, 121
SetByteValue, 121
SetByteValueNoSwap, 122
SetFromDataElement, 122
SetFromDataSet, 122
SetValue, 122
VMType, 120
gdcm::Attribute< Group, Element, TVR, VM::VM1_3 >, 123
GetVM, 124
gdcm::Attribute< Group, Element, TVR, VM::VM1_8 >, 124
GetVM, 125
gdcm::Attribute< Group, Element, TVR, VM::VM1_n >, 125
~Attribute, 126
ArrayType, 126
Attribute, 126
GDCM_STATIC_ASSERT, 127
GetAsDataElement, 127
GetDictVM, 127
GetDictVR, 127
GetNumberOfValues, 127
GetTag, 127
GetValue, 127
GetValues, 127
GetVM, 127
GetVR, 127
operator[], 128
Print, 128
Set, 128
SetByteValue, 128
SetFromDataElement, 128
SetFromDataSet, 128
SetNumberOfValues, 128
SetValue, 128
SetValues, 128
gdcm::Attribute< Group, Element, TVR, VM::VM2_2n >, 129
GetVM, 130
gdcm::Attribute< Group, Element, TVR, VM::VM2_n >, 130
GetVM, 132
gdcm::Attribute< Group, Element, TVR, VM::VM3_3n >, 132
GetVM, 133
gdcm::Attribute< Group, Element, TVR, VM::VM3_n >, 134
GetVM, 135
gdcm::AudioCodec, 135
~AudioCodec, 136
AudioCodec, 136
CanCode, 136
CanDecode, 136
Decode, 137
gdcm::Base64, 137
Decode, 137
Encode, 138
GetDecodeLength, 138
GetEncodeLength, 138
gdcm::BaseQuery, 145
~BaseQuery, 147
AddQueryDataSet, 147
BaseQuery, 147
GetAbstractSyntaxUID, 147
GetQueryDataSet, 147
GetSOPInstanceUID, 147
mDataSet, 148
mHelpDescription, 148
mSopInstanceUID, 148
Print, 147
QueryFactory, 148
SetSOPInstanceUID, 148
SetSearchParameter, 147, 148
ValidDataSet, 148
ValidateQuery, 148
WriteHelpFile, 148
WriteQuery, 148
gdcm::BaseRootQuery, 149
~BaseRootQuery, 150
BaseRootQuery, 150
Construct, 150
GetQueryLevelFromQueryRoot, 150
GetQueryLevelFromString, 150
GetQueryLevelString, 150
GetTagListByLevel, 150
InitializeDataSet, 150
mHelpDescription, 151
mImage, 151
mPatient, 151
mRootType, 151
mSeries, 151
mStudy, 151
QueryFactory, 151
ValidateQuery, 151
gdcm::BasicOffsetTable, 154
BasicOffsetTable, 155
operator<<, 156
Read, 156

gdcm::Bitmap, 156
~Bitmap, 159
AreOverlaysInPixelData, 159
Bitmap, 159
Clear, 159
ComputeLossyFlag, 159
Dimensions, 164
GetBuffer, 159
GetBuffer2, 159
GetBufferLength, 160
GetColumns, 160
GetDataElement, 160
GetDimension, 160
GetDimensions, 160
GetLUT, 160
GetNeedByteSwap, 161
GetNumberOfDimensions, 161
GetPhotometricInterpretation, 161
GetPixelFormat, 161
GetPlanarConfiguration, 161
GetRows, 161
GetTransferSyntax, 161
ImageChangeTransferSyntax, 164
IsEmpty, 161
IsLossy, 162
IsTransferSyntaxCompatible, 162
LUTPtr, 159
LUT, 164
LossyFlag, 164
NeedByteSwap, 164
NumberOfDimensions, 164
PF, 164
PI, 164
PixelData, 164
PixmapReader, 164
PlanarConfiguration, 164
Print, 162
SetColumns, 162
SetDataElement, 162
SetDimension, 162
SetDimensions, 162
SetLUT, 162
SetLossyFlag, 162
SetNeedByteSwap, 163
SetNumberOfDimensions, 163
SetPhotometricInterpretation, 163
SetPixelFormat, 163
SetPlanarConfiguration, 163
SetRows, 163
SetTransferSyntax, 163
TryJPEG2000Codec, 163
TryJPEG2000Codec2, 164
TryJPEGCodec, 164
TryJPEGCodec2, 164
TryJPEGLSCodec, 164
TryKAKADUCodec, 164
TryPVRGCodec, 164
TryRAWCodec, 164
TryRLECodec, 164
TS, 164
gdcm::BitmapToBitmapFilter, 165
~BitmapToBitmapFilter, 166
BitmapToBitmapFilter, 166
GetOutput, 166
GetOutputAsBitmap, 166
Input, 166
Output, 166
SetInput, 166
gdcm::BoxRegion, 167
~BoxRegion, 168
Area, 169
BoundingBox, 169
BoxRegion, 168
Clone, 169
ComputeBoundingBox, 169
Empty, 169
GetXMax, 169
GetXMin, 169
GetYMax, 169
GetYMin, 170
GetZMax, 170
GetZMin, 170
IsValid, 170
operator=, 170
Print, 170
SetDomain, 170
gdcm::ByteBuffer, 170
ByteBuffer, 171
Get, 171
GetStart, 171
ShiftEnd, 171
UpdatePosition, 171
gdcm::ByteSwap
Swap, 172
SwapFromSwapCodeIntoSystem, 172
SwapRange, 172
SwapRangeFromSwapCodeIntoSystem, 172
SystemIsBigEndian, 172
SystemIsLittleEndian, 172
gdcm::ByteSwap< T >, 171
gdcm::ByteSwapFilter, 173
~ByteSwapFilter, 173
ByteSwap, 173
ByteSwapFilter, 173
SetByteSwapTag, 173
gdcm::ByteValue, 174
~ByteValue, 176
Append, 176

ByteValue, 176
Clear, 176
ComputeLength, 176
Fill, 176
GetBuffer, 176
GetLength, 176
GetPointer, 177
IsEmpty, 177
IsPrintable, 177
operator const std::vector< char > &, 177
operator=, 177
operator==, 177
Print, 177
PrintASCIIXML, 178
PrintASCII, 178
PrintGroupLength, 178
PrintHex, 178
PrintHexXML, 178
PrintPNXML, 178
Read, 178
SetLength, 178
SetLengthOnly, 178
Write, 178
WriteBuffer, 178
gdcm::CAPICryptoFactory, 179
CAPICryptoFactory, 179
CreateCMSProvider, 180
gdcm::CAPICryptographicMessageSyntax, 180
~CAPICryptographicMessageSyntax, 181
CAPICryptographicMessageSyntax, 181
Decrypt, 181
Encrypt, 181
GetCipherType, 181
GetInitialized, 181
ParseCertificateFile, 182
ParseKeyFile, 182
SetCipherType, 182
SetPassword, 182
gdcm::CP246ExplicitDataElement, 207
GetLength, 209
Read, 209
ReadPreValue, 209
ReadValue, 209
ReadWithLength, 209
gdcm::CSAElement, 213
CSAElement, 215
DataField, 217
DataPtr, 215
GetByteValue, 215
GetKey, 215
GetName, 216
GetNoOfItems, 216
GetSyngoDT, 216
GetValue, 216
GetVM, 216
GetVR, 216
IsEmpty, 216
KeyField, 217
NameField, 218
NoOfItemsField, 218
operator<, 217
operator<<, 217
operator=, 217
operator==, 217
SetByteValue, 217
SetKey, 217
SetName, 217
SetNoOfItems, 217
SetSyngoDT, 217
SetValue, 217
SetVM, 217
SetVR, 217
SyngoDTField, 218
VRField, 218
ValueMultiplicityField, 218
gdcm::CSAHeader, 218
~CSAHeader, 220
CSAHeader, 220
CSAHeaderType, 220
DATASET_FORMAT, 220
FindCSAElementByName, 220
GetCSADataInfo, 220
GetCSAEEnd, 221
GetCSAElementByName, 221
GetCSAImageHeaderInfoTag, 221
GetCSASeriesHeaderInfoTag, 221
GetDataSet, 221
GetFormat, 221
GetInterfile, 222
INTERFILE, 220
LoadFromDataElement, 222
NOMAGIC, 220
operator<<, 222
Print, 222
Read, 222
SV10, 220
UNKNOWN, 220
Write, 222
ZEROED_OUT, 220
gdcm::CSAHeaderDict, 223
AddCSAHeaderDictEntry, 224
Begin, 224
CSAHeaderDict, 224
ConstIterator, 224
Dicts, 224
End, 224
GetCSAHeaderDictEntry, 224
IsEmpty, 224

Iterator, 224
LoadDefault, 224
MapCSAHeaderDictEntry, 224
operator<<, 224
gdcm::CSAHeaderDictEntry, 225
CSAHeaderDictEntry, 226
GetDescription, 226
GetName, 226
GetVM, 226
GetVR, 226
operator<, 226
operator<<, 226
SetDescription, 226
SetName, 226
SetVM, 226
SetVR, 226
gdcm::CSAHeaderDictException, 227
gdcm::CodeString, 195
CodeString, 197
const_iterator, 197
const_reference, 197
const_reverse_iterator, 197
difference_type, 197
GetAsString, 197
IsValid, 197
iterator, 197
operator!=, 198
operator<<, 198
operator==, 198
pointer, 197
reference, 197
reverse_iterator, 197
Size, 198
size_type, 197
TrimInternal, 198
value_type, 197
gdcm::Codec, 193
gdcm::Coder, 194
~Coder, 195
CanCode, 195
Code, 195
InternalCode, 195
gdcm::Command, 198
~Command, 199
Command, 199
Execute, 199
gdcm::CommandDataSet, 200
~CommandDataSet, 201
CommandDataSet, 201
Insert, 201
operator<<, 202
Read, 201
Replace, 201
Write, 202
gdcm::CompositeNetworkFunctions, 203
CEcho, 204
CFind, 204
CMove, 205
CStore, 206
ConstructQuery, 205, 206
KeyValuePairArrayType, 204
KeyValuePairType, 204
gdcm::ConstCharWrapper, 206
ConstCharWrapper, 207
operator const char *, 207
gdcm::CryptoFactory, 209
~CryptoFactory, 211
CAPI, 210
CreateCMSProvider, 211
CryptoFactory, 210
CryptoLib, 210
DEFAULT, 210
GetFactoryInstance, 211
OPENSSLP7, 210
OPENSSL, 210
gdcm::CryptographicMessageSyntax, 211
~CryptographicMessageSyntax, 212
AES128_CIPHER, 212
AES192_CIPHER, 212
AES256_CIPHER, 212
CipherTypes, 212
CryptographicMessageSyntax, 212
DES3_CIPHER, 212
Decrypt, 212
Encrypt, 212
GetCipherType, 212
ParseCertificateFile, 213
ParseKeyFile, 213
SetCipherType, 213
SetPassword, 213
gdcm::Curve, 230
~Curve, 232
Curve, 232
Decode, 232
GetAsPoints, 232
GetCurveDataDescriptor, 232
GetDataValueRepresentation, 232
GetDimensions, 232
GetGroup, 232
GetNumberOfCurves, 232
GetNumberOfPoints, 232
GetTypeOfData, 232
GetTypeOfDataDescription, 232
IsEmpty, 232
Print, 232
SetCoordinateStartValue, 232
SetCoordinateStepValue, 233
SetCurve, 233

SetCurveDataDescriptor, 233
SetCurveDescription, 233
SetDataValueRepresentation, 233
SetDimensions, 233
SetGroup, 233
SetNumberOfPoints, 233
SetTypeOfData, 233
Update, 233
gdcm::DICOMDIRGenerator, 264
~DICOMDIRGenerator, 265
AddImageDirectoryRecord, 265
AddPatientDirectoryRecord, 266
AddSeriesDirectoryRecord, 266
AddStudyDirectoryRecord, 266
DICOMDIRGenerator, 265
FilenameType, 265
FilenamesType, 265
Generate, 266
GetFile, 266
GetScanner, 266
SetDescriptor, 266
SetFile, 266
SetFilenames, 266
SetRootDirectory, 266
gdcm::DICOMDIR, 263
DICOMDIR, 264
gdcm::DataElement, 233
Clear, 237
DataElement, 237
Empty, 237
GetByteValue, 237
GetLength, 237
GetSequenceOfFragments, 238
GetTag, 238
GetValue, 238
GetValueAsSQ, 239
GetVL, 239
GetVR, 239
IsEmpty, 239
IsUndefinedLength, 240
operator<, 240
operator<<, 242
operator=, 240
operator==, 240
Read, 240
ReadOrSkip, 240
ReadPreValue, 240
ReadValue, 240
ReadValueWithLength, 240
ReadWithLength, 240
SetByteValue, 240
SetTag, 241
SetVLToUndefined, 241
SetValue, 241
SetValueFieldLength, 241
SetVL, 241
SetVR, 242
TagField, 242
VRField, 243
ValueField, 242
ValueLengthField, 242
ValuePtr, 237
Write, 242
gdcm::DataElementException, 243
gdcm::DataEvent, 244
~DataEvent, 245
CheckEvent, 245
DataEvent, 245
GetData, 245
GetDataLength, 245
GetEventName, 245
MakeObject, 245
Self, 245
SetData, 246
Superclass, 245
gdcm::DataSet, 246
Begin, 249
CSAHeader, 253
Clear, 249
ComputeDataElement, 249
ComputeGroupLength, 249
ConstIterator, 249
DataElementSet, 249
End, 249
FindDataElement, 250
FindNextDataElement, 250
GetDEEnd, 251
GetDES, 251
GetDataElement, 250
GetLength, 251
GetMediaStorage, 251
GetPrivateCreator, 251
Insert, 251
InsertDataElement, 251
IsEmpty, 252
Iterator, 249
operator<<, 253
operator(), 252
operator=, 252
operator[], 252
Print, 252
Read, 252
ReadNested, 252
ReadSelectedPrivateTags, 252
ReadSelectedPrivateTagsWithLength, 252
ReadSelectedTags, 252
ReadSelectedTagsWithLength, 252
ReadUpToTag, 252

ReadUpToTagWithLength, 252
ReadWithLength, 252
Remove, 252
Replace, 252
ReplaceEmpty, 253
Size, 253
SizeType, 249
Write, 253
gdcm::DataSetEvent, 254
~DataSetEvent, 255
CheckEvent, 255
DataSetEvent, 255
GetDataSet, 255
GetEventName, 255
MakeObject, 255
Self, 255
Superclass, 255
gdcm::DataSetHelper, 256
ComputeVR, 256
gdcm::Decoder, 257
~Decoder, 258
CanDecode, 258
Decode, 258
DecodeByStreams, 258
gdcm::DefinedTerms, 258
DefinedTerms, 259
gdcm::Defs, 259
~Defs, 260
Defs, 260
GetIODFromFile, 260
GetIODNameFromMediaStorage, 260
GetIODs, 260, 261
GetMacros, 261
GetModules, 261
GetTypeFromTag, 261
Global, 261
IsEmpty, 261
LoadDefaults, 261
LoadFromFile, 261
Verify, 261
gdcm::DeltaEncodingCodec, 262
~DeltaEncodingCodec, 263
CanDecode, 263
Decode, 263
DeltaEncodingCodec, 263
gdcm::Dict, 267
AddDictEntry, 268
Begin, 268
ConstIterator, 268
Dict, 268
Dicts, 269
End, 268
GetDictEntry, 268
GetDictEntryByKeyword, 268
GetDictEntryByName, 268
GetKeywordFromTag, 269
IsEmpty, 269
Iterator, 268
LoadDefault, 269
MapDictEntry, 268
operator<<, 269
gdcm::DictConverter, 269
~DictConverter, 271
AddGroupLength, 271
Convert, 271
ConvertToCXX, 271
ConvertToXML, 271
DICT_DEBUG, 270
DICT_DEFAULT, 270
DICT_XML, 270
DictConverter, 271
GetDictName, 271
GetInputFilename, 271
GetOutputFilename, 271
GetOutputType, 271
OutputTypes, 270
ReadVM, 271
ReadVR, 271
Readuint16, 271
SetDictName, 271
SetInputFileName, 271
SetOutputFileName, 271
SetOutputType, 271
WriteFooter, 271
WriteHeader, 271
gdcm::DictEntry, 272
Dict, 274
DictEntry, 273
GetKeyword, 273
GetName, 273
GetRetired, 273
GetVM, 273
GetVR, 273
IsUnique, 273
operator<<, 274
SetElementXX, 274
SetGroupXX, 274
SetKeyword, 274
SetName, 274
SetRetired, 274
SetVM, 274
SetVR, 274
gdcm::DictPrinter, 275
~DictPrinter, 276
DictPrinter, 276
Print, 276
PrintDataElement2, 276
PrintDataSet2, 276

gdcm::Dicts, 276
~Dicts, 278
ConstructorType, 278
Dicts, 278
GEMS, 278
GetCSAHeaderDict, 278
GetConstructorString, 278
GetDictEntry, 278
GetPrivateDict, 279
GetPublicDict, 279
Global, 279
IsEmpty, 279
LoadDefaults, 279
operator<<, 279
PHILIPS, 278
SIEMENS, 278
gdcm::DirectionCosines, 281
~DirectionCosines, 282
ComputeDistAlongNormal, 282
Cross, 282
CrossDot, 282
DirectionCosines, 282
Dot, 282
IsValid, 282
Normalize, 282
operator const double *, 283
Print, 283
SetFromString, 283
gdcm::Directory, 283
~Directory, 285
Directory, 285
Explore, 285
FilenameType, 284
FilenamesType, 284
GetDirectories, 285
GetFilenames, 285
GetToplevel, 285
Load, 285
operator<<, 286
Print, 285
gdcm::DirectoryHelper, 286
GetCTImageSeriesUIDs, 287
GetFilenamesFromSeriesUIDs, 287
GetFrameOfReference, 287
GetMRImageSeriesUIDs, 287
GetRTStructSeriesUIDs, 287
GetSOPClassUID, 287
GetSeriesUIDsBySOPClassUID, 287
GetStringValueFromTag, 287
LoadImageFromFiles, 287
RetrieveSOPInstanceUIDFromIndex, 287
RetrieveSOPInstanceUIDFromZPosition, 287
gdcm::DummyValueGenerator, 288
Generate, 288
gdcm::Dumper, 288
~Dumper, 290
Dumper, 290
gdcm::Element
GetAsDataElement, 292
GetLength, 292
GetValue, 292
GetValues, 292
GetVM, 292
GetVR, 293
Internal, 293
operator[], 293
Print, 293
Read, 293
Set, 293
SetFromDataElement, 293
SetNoSwap, 293
SetValue, 293
Type, 292
Write, 293
gdcm::Element< TVR, TVM >, 290
gdcm::Element< TVR, VM::VM1_2 >, 294
Parent, 295
SetLength, 295
gdcm::Element< TVR, VM::VM1_n >, 295
~Element, 296
Element, 296
GetAsDataElement, 296
GetLength, 296
GetValue, 297
GetVM, 297
GetVR, 297
operator=, 297
operator[], 297
Print, 297
Read, 297
Set, 297
SetArray, 297
SetFromDataElement, 297
SetLength, 297
SetNoSwap, 297
SetValue, 297
Type, 296
Write, 298
WriteASCII, 298
gdcm::Element< TVR, VM::VM2_2n >, 298
Parent, 299
SetLength, 299
gdcm::Element< TVR, VM::VM2_n >, 300
Parent, 301
SetLength, 301
gdcm::Element< TVR, VM::VM3_3n >, 301
Parent, 302
SetLength, 302

gdcm::Element< TVR, VM::VM3_n >, 303
Parent, 304
SetLength, 304
gdcm::Element< VR::AS, VM::VM5 >, 304
GetLength, 304
Internal, 304
Print, 304
gdcm::Element< VR::OB, VM::VM1 >, 305
gdcm::Element< VR::OW, VM::VM1 >, 306
gdcm::ElementDisableCombinations< TVR, TVM >, 308
gdcm::ElementDisableCombinations< VR::OB, VM::VM1_n >, 309
gdcm::ElementDisableCombinations< VR::OW, VM::VM1_n >, 309
gdcm::EncapsulatedDocument, 309
EncapsulatedDocument, 310
gdcm::EncodingImplementation< T >, 310
gdcm::EncodingImplementation< VR::VRASCII >, 310
Read, 311
ReadComputeLength, 311
ReadNoSwap, 311
Write, 311
gdcm::EncodingImplementation< VR::VRBINARY >, 312
Read, 312
ReadComputeLength, 312
ReadNoSwap, 312
Write, 312
gdcm::EndEvent, 312
gdcm::EnumeratedValues, 314
EnumeratedValues, 314
gdcm::Event, 314
~Event, 316
CheckEvent, 316
Event, 316
GetEventName, 316
MakeObject, 316
Print, 316
gdcm::Exception, 317
~Exception, 318
Exception, 318
GetDescription, 318
what, 318
gdcm::ExitEvent, 319
gdcm::ExplicitDataElement, 320
GetLength, 322
Read, 322
ReadPreValue, 322
ReadValue, 322
ReadWithLength, 322
Write, 322
gdcm::ExplicitImplicitDataElement, 322
GetLength, 324
Read, 324
ReadPreValue, 324
ReadValue, 324
ReadWithLength, 324
gdcm::Fiducials, 324
Fiducials, 324
gdcm::File, 325
~File, 327
File, 327
GetDataSet, 327
GetHeader, 327
operator<<, 328
Read, 327
SetDataSet, 328
SetHeader, 328
Write, 328
gdcm::FileAnonymizer, 328
~FileAnonymizer, 330
Empty, 330
FileAnonymizer, 330
Remove, 330
Replace, 330
SetInputFileName, 331
SetOutputFileName, 331
Write, 331
gdcm::FileChangeTransferSyntax, 332
~FileChangeTransferSyntax, 333
Change, 333
FileChangeTransferSyntax, 333
GetCodec, 333
New, 334
SetInputFileName, 334
SetOutputFileName, 334
SetTransferSyntax, 334
gdcm::FileDecompressLookupTable, 334
~FileDecompressLookupTable, 336
Change, 336
FileDecompressLookupTable, 336
GetFile, 336
GetPixmap, 336
SetFile, 336
SetPixmap, 336
gdcm::FileDerivation, 337
~FileDerivation, 338
AddDerivationDescription, 338
AddPurposeOfReferenceCodeSequence, 338
AddReference, 338
AddSourceImageSequence, 338
Derive, 338
FileDerivation, 338
GetFile, 338
SetDerivationCodeSequenceCodeValue, 339
SetDerivationDescription, 339
SetFile, 339
SetPurposeOfReferenceCodeSequenceCodeValue, 339

gdcm::FileExplicitFilter, 339
~FileExplicitFilter, 341
Change, 341
ChangeFMI, 341
FileExplicitFilter, 341
GetFile, 341
ProcessDataSet, 341
SetChangePrivateTags, 341
SetFile, 341
SetRecomputeItemLength, 341
SetRecomputeSequenceLength, 341
SetUseVRUN, 342
gdcm::FileMetaInformation, 342
~FileMetaInformation, 344
AppendImplementationClassUID, 345
ComputeDataSetMediaStorageSOPClass, 345
ComputeDataSetTransferSyntax, 345
DataSetMS, 347
DataSetTS, 347
Default, 345
FileMetaInformation, 344
FillFromDataSet, 345
GetDataSetTransferSyntax, 345
GetFileMetaInformationVersion, 345
GetFullLength, 345
GetGDCMImplementationClassUID, 345
GetGDCMImplementationVersionName, 345
GetGDCMSourceApplicationEntityTitle, 345
GetImplementationClassUID, 345
GetImplementationVersionName, 345
GetMediaStorage, 345
GetMediaStorageAsString, 345
GetMetaInformationTS, 345
GetPreamble, 345
GetSourceApplicationEntityTitle, 346
Insert, 346
IsValid, 346
MetaInformationTS, 347
operator<<, 347
Read, 346
ReadCompat, 346
ReadCompatInternal, 346
Replace, 346
SetDataSetTransferSyntax, 346
SetImplementationClassUID, 346
SetImplementationVersionName, 346
SetPreamble, 347
SetSourceApplicationEntityTitle, 347
Write, 347
gdcm::FileNameEvent, 350
~FileNameEvent, 352
CheckEvent, 352
FileNameEvent, 352
GetEventName, 352
GetFileName, 352
MakeObject, 352
Self, 352
SetFileName, 352
Superclass, 352
gdcm::FileSet, 355
AddFile, 356
FileSet, 356
FileType, 356
FilesType, 356
GetFiles, 356
operator<<, 357
SetFiles, 357
gdcm::FileStreamer, 357
~FileStreamer, 359
AppendToDataElement, 359
AppendToGroupDataElement, 359
CheckDataElement, 359
CheckTemplateFileName, 359
FileStreamer, 359
New, 360
ReserveDataElement, 360
ReserveGroupDataElement, 360
SetOutputFileName, 360
SetTemplateFileName, 360
StartDataElement, 360
StartGroupDataElement, 360
StopDataElement, 361
StopGroupDataElement, 361
gdcm::FileWithName, 361
FileWithName, 362
filename, 362
gdcm::Filename, 347
EndWith, 348
Filename, 348
GetExtension, 348
GetFileName, 349
GetName, 349
GetPath, 349
IsEmpty, 349
IsIdentical, 349
Join, 349
operator const char *, 349
ToUnixSlashes, 349
ToWindowsSlashes, 350
gdcm::FilenameGenerator, 353
~FilenameGenerator, 354
FilenameGenerator, 354
FilenameType, 354
FilenamesType, 354
Generate, 354
GetFilename, 354
GetFilenames, 354
GetNumberOfFilenames, 354

GetPattern, 354
GetPrefix, 355
SetNumberOfFilenames, 355
SetPattern, 355
SetPrefix, 355
SizeType, 354
gdcm::FindPatientRootQuery, 363
FindPatientRootQuery, 364
GetAbstractSyntaxUID, 364
GetTagListByLevel, 364
InitializeDataSet, 364
QueryFactory, 365
ValidateQuery, 364
gdcm::FindStudyRootQuery, 365
FindStudyRootQuery, 366
GetAbstractSyntaxUID, 366
GetTagListByLevel, 366
InitializeDataSet, 367
QueryFactory, 367
ValidateQuery, 367
gdcm::Fragment, 367
ComputeLength, 369
Fragment, 369
GetLength, 369
operator<<, 370
Read, 369
ReadBacktrack, 369
ReadPreValue, 369
ReadValue, 370
Write, 370
gdcm::Global, 370
~Global, 371
Append, 371
GetDefs, 371
GetDicts, 371, 372
GetInstance, 372
Global, 371
LoadResourcesFiles, 372
Locate, 372
operator<<, 373
Prepend, 372
gdcm::GroupDict, 373
~GroupDict, 374
Add, 374
GetAbbreviation, 374
GetName, 374
GroupDict, 374
GroupStringVector, 374
Insert, 374
operator<<, 374
Size, 374
gdcm::IODEntry, 431
GetIE, 432
GetName, 432
GetRef, 432
GetUsage, 432
GetUsageType, 433
IODEntry, 432
operator<<, 433
SetIE, 433
SetName, 433
SetRef, 433
SetUsage, 433
gdcm::IODs, 433
AddIOD, 434
Begin, 434
Clear, 434
End, 435
GetIOD, 435
IODMapType, 434
IODMapTypeConstIterator, 434
IODName, 434
IODs, 434
operator<<, 435
gdcm::IOD, 429
AddIODEntry, 430
Clear, 430
GetIODEntry, 430
GetNumberOfIODs, 430
GetTypeFromTag, 431
IOD, 430
MapIODEntry, 430
operator<<, 431
SizeType, 430
gdcm::IPPSorter, 435
ComputeZSpacing, 439
DirCosTolerance, 439
DropDuplicatePositions, 439
GetDirectionCosinesTolerance, 437
GetZSpacing, 437
GetZSpacingTolerance, 437
IPPSorter, 437
SetComputeZSpacing, 438
SetDirectionCosinesTolerance, 438
SetDropDuplicatePositions, 438
SetZSpacingTolerance, 438
Sort, 438
ZSpacing, 439
ZTolerance, 439
gdcm::IconImageFilter, 375
~IconImageFilter, 376
Extract, 376
ExtractIconImages, 376
ExtractVeproIconImages, 376
GetFile, 376
GetIconImage, 376
GetNumberOfIconImages, 376
IconImageFilter, 376

SetFile, 377
gdcm::IconImageGenerator, 377
~IconImageGenerator, 378
AutoPixelMinMax, 378
ConvertRGBToPaletteColor, 378
Generate, 378
GetIconImage, 379
GetPixmap, 379
IconImageGenerator, 378
SetOutputDimensions, 379
SetOutsideValuePixel, 379
SetPixelMinMax, 379
SetPixmap, 379
gdcm::Image, 381
~Image, 383
GetDirectionCosines, 383
GetIntercept, 383
GetOrigin, 383
GetSlope, 383
GetSpacing, 383
Image, 383
Print, 384
SetDirectionCosines, 384
SetIntercept, 384
SetOrigin, 384
SetSlope, 384
SetSpacing, 384
gdcm::ImageApplyLookupTable, 385
~ImageApplyLookupTable, 387
Apply, 387
ImageApplyLookupTable, 387
gdcm::ImageChangePhotometricInterpretation, 387
~ImageChangePhotometricInterpretation, 389
Change, 389
ChangeMonochrome, 389
GetPhotometricInterpretation, 389
ImageChangePhotometricInterpretation, 389
RGB2YBR, 389
SetPhotometricInterpretation, 389
YBR2RGB, 390
gdcm::ImageChangePlanarConfiguration, 390
~ImageChangePlanarConfiguration, 392
Change, 392
GetPlanarConfiguration, 392
ImageChangePlanarConfiguration, 392
RGBPixelsToRGBPlanes, 392
RGBPlanesToRGBPixels, 392
SetPlanarConfiguration, 392
gdcm::ImageChangeTransferSyntax, 393
~ImageChangeTransferSyntax, 395
Change, 395
GetTransferSyntax, 395
ImageChangeTransferSyntax, 395
SetCompressIconImage, 396
SetForce, 396
SetTransferSyntax, 396
SetUserCodec, 396
TryJPEG2000Codec, 396
TryJPEGCodec, 396
TryJPEGLSCodec, 396
TryRAWCodec, 396
TryRLECodec, 396
gdcm::ImageCodec, 397
~ImageCodec, 399
AppendFrameEncode, 399
AppendRowEncode, 399
CanCode, 400
CanDecode, 400
Clone, 400
Decode, 400
DecodeByStreams, 400
Dimensions, 403
DoByteSwap, 400
DoInvertMonochrome, 401
DoOverlayCleanup, 401
DoPaddedCompositePixelCode, 401
DoPlanarConfiguration, 401
DoSimpleCopy, 401
DoYBR, 401
FileChangeTransferSyntax, 403
GetDimensions, 401
GetHeaderInfo, 401
GetLUT, 401
GetLossyFlag, 401
GetNeedByteSwap, 401
GetNumberOfDimensions, 401
GetPhotometricInterpretation, 401
GetPixelFormat, 401
GetPlanarConfiguration, 402
ImageChangePhotometricInterpretation, 403
ImageCodec, 399
IsFrameEncoder, 402
IsLossy, 402
IsRowEncoder, 402
IsValid, 402
LUTPtr, 399
LUT, 403
LossyFlag, 403
NeedByteSwap, 403
NeedOverlayCleanup, 403
NumberOfDimensions, 404
PF, 404
PI, 404
PlanarConfiguration, 404
RequestPaddedCompositePixelCode, 404
RequestPlanarConfiguration, 404
SetDimensions, 402
SetLUT, 402

SetLossyFlag, 402
SetNeedByteSwap, 402
SetNeedOverlayCleanup, 402
SetNumberOfDimensions, 402
SetPhotometricInterpretation, 402
SetPixelFormat, 402
SetPlanarConfiguration, 403
StartEncode, 403
StopEncode, 403
gdcm::ImageConverter, 404
~ImageConverter, 405
Convert, 405
GetOuput, 405
ImageConverter, 405
SetInput, 405
gdcm::ImageFragmentSplitter, 405
~ImageFragmentSplitter, 407
GetFragmentSizeMax, 407
ImageFragmentSplitter, 407
SetForce, 407
SetFragmentSizeMax, 407
Split, 407
gdcm::ImageHelper, 407
ComputeMediaStorageFromModality, 409
ComputeSpacingFromImagePositionPatient, 409
GetDimensionsValue, 409
GetDirectionCosinesFromDataSet, 409
GetDirectionCosinesValue, 409
GetForcePixelSpacing, 409
GetForceRescaleInterceptSlope, 409
GetLUT, 409
GetOriginValue, 409
GetPMSRescaleInterceptSlope, 410
GetPhotometricInterpretationValue, 409
GetPixelFormatValue, 410
GetPlanarConfigurationValue, 410
GetPointerFromElement, 410
GetRealWorldValueMappingContent, 410
GetRescaleInterceptSlopeValue, 410
GetSpacingTagFromMediaStorage, 410
GetSpacingValue, 410
GetZSpacingTagFromMediaStorage, 410
SetDimensionsValue, 410
SetDirectionCosinesValue, 410
SetForcePixelSpacing, 410
SetForceRescaleInterceptSlope, 411
SetOriginValue, 411
SetPMSRescaleInterceptSlope, 411
SetRescaleInterceptSlopeValue, 411
SetSpacingValue, 411
gdcm::ImageReader, 411
~ImageReader, 414
GetImage, 414
ImageReader, 414
Read, 415
ReadACRNEMAImage, 415
ReadImage, 415
gdcm::ImageRegionReader, 416
~ImageRegionReader, 418
ComputeBufferLength, 418
GetRegion, 418
ImageRegionReader, 418
Read, 418
ReadInformation, 418
ReadIntoBuffer, 419
SetRegion, 419
gdcm::ImageToImageFilter, 419
~ImageToImageFilter, 421
GetInput, 421
GetOutput, 421
ImageToImageFilter, 421
gdcm::ImageWriter, 421
~ImageWriter, 423
ComputeTargetMediaStorage, 423
GetImage, 423
ImageWriter, 423
Write, 423
gdcm::ImplicitDataElement, 426
GetLength, 428
Read, 428
ReadPreValue, 428
ReadValue, 428
ReadValueWithLength, 428
ReadWithLength, 428
Write, 428
gdcm::InitializeEvent, 428
gdcm::Item, 439
Clear, 441
FindDataElement, 441
GetDataElement, 441
GetLength, 442
GetNestedDataSet, 442
InsertDataElement, 442
Item, 441
operator<<, 442
Read, 442
SetNestedDataSet, 442
Write, 442
gdcm::IterationEvent, 443
gdcm::JPEG12Codec, 444
~JPEG12Codec, 445
DecodeByStreams, 445
EncodeBuffer, 445
GetHeaderInfo, 445
InternalCode, 445
IsStateSuspension, 446
JPEG12Codec, 445
gdcm::JPEG16Codec, 446

~JPEG16Codec, 447
DecodeByStreams, 447
EncodeBuffer, 447
GetHeaderInfo, 448
InternalCode, 448
IsStateSuspension, 448
JPEG16Codec, 447
gdcm::JPEG2000Codec, 448
~JPEG2000Codec, 450
AppendFrameEncode, 450
AppendRowEncode, 450
Bitmap, 452
CanCode, 450
CanDecode, 450
Clone, 451
Code, 451
Decode, 451
DecodeByStreams, 451
DecodeExtent, 451
GetHeaderInfo, 451
GetQuality, 451
GetRate, 451
ImageRegionReader, 452
IsFrameEncoder, 451
IsRowEncoder, 451
JPEG2000Codec, 450
SetNumberOfResolutions, 452
SetQuality, 452
SetRate, 452
SetReversible, 452
SetTileSize, 452
StartEncode, 452
StopEncode, 452
gdcm::JPEG8Codec, 453
~JPEG8Codec, 454
DecodeByStreams, 454
EncodeBuffer, 454
GetHeaderInfo, 454
InternalCode, 454
IsStateSuspension, 455
JPEG8Codec, 454
gdcm::JPEGCodec, 455
~JPEGCodec, 457
AppendFrameEncode, 457
AppendRowEncode, 457
BitSample, 460
CanCode, 458
CanDecode, 458
Clone, 458
Code, 458
ComputeOffsetTable, 458
Decode, 458
DecodeByStreams, 458
DecodeExtent, 458
EncodeBuffer, 459
GetHeaderInfo, 459
GetLossless, 459
GetQuality, 459
ImageRegionReader, 460
IsFrameEncoder, 459
IsRowEncoder, 459
IsStateSuspension, 459
IsValid, 459
JPEGCodec, 457
Quality, 460
SetBitSample, 459
SetLossless, 460
SetPixelFormat, 460
SetQuality, 460
StartEncode, 460
StopEncode, 460
gdcm::JPEGLSCodec, 461
~JPEGLSCodec, 463
AppendFrameEncode, 463
AppendRowEncode, 463
CanCode, 463
CanDecode, 463
Clone, 463
Code, 463
Decode, 463, 464
DecodeExtent, 464
GetBufferLength, 464
GetHeaderInfo, 464
GetLossless, 464
ImageRegionReader, 465
IsFrameEncoder, 464
IsRowEncoder, 464
JPEGLSCodec, 463
SetBufferLength, 464
SetLossless, 464
SetLossyError, 464
StartEncode, 464
StopEncode, 464
gdcm::JSON, 465
~JSON, 465
Code, 465
Decode, 465
GetPrettyPrint, 466
JSON, 465
PrettyPrintOff, 466
PrettyPrintOn, 466
SetPrettyPrint, 466
gdcm::KAKADUCodec, 466
~KAKADUCodec, 467
CanCode, 467
CanDecode, 467
Clone, 468
Code, 468

Decode, 468
KAKADUCodec, 467
gdcm::LO, 468
const_iterator, 470
const_reference, 470
const_reverse_iterator, 470
difference_type, 470
IsValid, 470
iterator, 470
LO, 470
pointer, 470
reference, 470
reverse_iterator, 470
size_type, 470
Superclass, 470
value_type, 470
gdcm::LookupTable, 471
∼LookupTable, 473
Allocate, 473
BLUE, 473
BitSample, 475
Clear, 473
Decode, 473
GRAY, 473
GREEN, 473
GetBitSample, 473
GetBufferAsRGBA, 473
GetLUTDescriptor, 474
GetLUTLength, 474
GetLUT, 474
GetPointer, 474
IncompleteLUT, 475
InitializeBlueLUT, 474
InitializeGreenLUT, 474
InitializeLUT, 474
InitializeRedLUT, 474
Initialized, 474
Internal, 475
LookupTable, 473
LookupTableType, 473
Print, 474
RED, 473
SetBlueLUT, 475
SetGreenLUT, 475
SetLUT, 475
SetRedLUT, 475
UNKNOWN, 473
WriteBufferAsRGBA, 475
gdcm::MD5, 481
∼MD5, 482
Compute, 482
ComputeFile, 482
MD5, 482
gdcm::Macro, 476
AddMacroEntry, 478
ArrayIncludeMacrosType, 477
Clear, 478
FindMacroEntry, 478
GetMacroEntry, 478
GetName, 478
Macro, 477
MapModuleEntry, 477
operator<<, 478
SetName, 478
Verify, 478
gdcm::Macros, 478
AddMacro, 480
Clear, 480
GetMacro, 480
IsEmpty, 480
Macros, 479
ModuleMapType, 479
operator<<, 480
gdcm::MediaStorage, 482
AmbulatoryECGWaveformStorage, 487
Audio, 488
BasicTextSR, 487
BasicVoiceAudioWaveformStorage, 487
BreastTomosynthesisImageStorage, 488
CSANonImageStorage, 487
CTImageStorage, 486
CardiacElectrophysiologyWaveformStorage, 487
ComprehensiveSR, 487
ComputedRadiographyImageStorage, 486
DetachedPatientManagementSOPClass, 487
DetachedStudyManagementSOPClass, 487
DetachedVisitManagementSOPClass, 487
DigitalIntraoralXRayImageStorageForProcessing, 486
DigitalIntraoralXrayImageStorageForPresentation, 486
DigitalMammographyImageStorageForPresentation, 486
DigitalMammographyImageStorageForProcessing, 486
DigitalXRayImageStorageForPresentation, 486
DigitalXRayImageStorageForProcessing, 486
EncapsulatedCDAStorage, 487
EncapsulatedPDFStorage, 487
EnhancedCTImageStorage, 486
EnhancedMRImageStorage, 486
EnhancedPETImageStorage, 488
EnhancedSR, 487
EnhancedUSVolumeStorage, 488
EnhancedXAImageStorage, 488
FujiPrivateCRImageStorage, 488
GEPrivate3DModelStorage, 487
GeneralECGWaveformStorage, 487

GeneralElectricMagneticResonanceImageStorage, 487
GetMSString, 489
GetMSType, 489
GetModality, 489
GetModalityDimension, 489
GetNumberOfMSString, 489
GetNumberOfMSType, 489
GetNumberOfModality, 489
GetString, 489
GrayscaleSoftcopyPresentationStateStorageSOP↩ Class, 487
GuessFromModality, 489
HangingProtocolStorage, 488
HardcopyGrayscaleImageStorage, 487
HemodynamicWaveformStorage, 487
IsImage, 490
IsUndefined, 490
KeyObjectSelectionDocument, 488
LeadECGWaveformStorage, 487
MRImageStorage, 486
MRSpectroscopyStorage, 486
MS_END, 488
MSType, 486
MammographyCADSR, 488
MediaStorage, 489
MediaStorageDirectoryStorage, 486
ModalityPerformedProcedureStepSOPClass, 488
MultiframeGrayscaleByteSecondaryCaptureImage↩ Storage, 487
MultiframeGrayscaleWordSecondaryCapture↩ ImageStorage, 487
MultiframeSingleBitSecondaryCaptureImage↩ Storage, 487
MultiframeTrueColorSecondaryCaptureImage↩ Storage, 487
NoObject, 488
NuclearMedicineImageStorage, 487
NuclearMedicineImageStorageRetired, 486
ObjectEnd, 488
ObjectType, 488
operator MSType, 490
operator<<, 491
OphthalmicPhotography8BitImageStorage, 488
OphthalmicTomographyImageStorage, 488
PDF, 488
PETImageStorage, 487
Philips3D, 487
PhilipsPrivateMRSyntheticImageStorage, 488
RTDoseStorage, 487
RTImageStorage, 487
RTIonBeamsTreatmentRecordStorage, 488
RTIonPlanStorage, 488
RTPlanStorage, 487
RTStructureSetStorage, 487
RTTreatmentSummaryRecordStorage, 488
RawDataStorage, 487
SecondaryCaptureImageStorage, 486
Segmentation, 488
SegmentationStorage, 488
SetFromDataSet, 490
SetFromFile, 490
SetFromHeader, 490
SetFromModality, 491
SetFromSourceImageSequence, 491
SpacialFiducialsStorage, 487
SpacialRegistrationStorage, 487
StandaloneCurveStorage, 487
StandaloneModalityLUTStorage, 487
StandaloneOverlayStorage, 487
StandaloneVOILUTStorage, 487
StudyComponentManagementSOPClass, 487
SurfaceSegmentationStorage, 488
ToshibaPrivateDataStorage, 488
URI, 488
UltrasoundImageStorage, 486
UltrasoundImageStorageRetired, 486
UltrasoundMultiFrameImageStorage, 486
UltrasoundMultiFrameImageStorageRetired, 486
VLEndoscopicImageStorage, 488
VLMicroscopicImageStorage, 488
VLPhotographicImageStorage, 488
VLWholeSlideMicroscopyImageStorage, 488
Video, 488
VideoEndoscopicImageStorage, 487
VideoPhotographicImageStorage, 488
Waveform, 488
XRay3DAngiographicImageStorage, 488
XRayAngiographicBiPlaneImageStorageRetired, 487
XRayAngiographicImageStorage, 487
XRayRadiationDoseSR, 488
XRayRadiofluoroscopingImageStorage, 487
gdcm::MemberCommand
~MemberCommand, 493
Execute, 493
m_ConstMemberFunction, 494
m_MemberFunction, 494
m_This, 494
MemberCommand, 493
New, 494
Self, 493
SetCallbackFunction, 494
TConstMemberFunctionPointer, 493
TMemberFunctionPointer, 493
gdcm::MemberCommand< T >, 491
gdcm::MeshPrimitive, 495
~MeshPrimitive, 497
AddPrimitiveData, 497

EDGE, 497
FACET, 497
GetMPType, 497
GetMPTypeString, 497
GetNumberOfPrimitivesData, 497
GetPrimitiveData, 497, 498
GetPrimitiveType, 498
GetPrimitivesData, 498
LINE, 497
MPType, 497
MPType_END, 497
MeshPrimitive, 497
PrimitiveData, 498
PrimitiveType, 498
PrimitivesData, 497
SetPrimitiveData, 498
SetPrimitiveType, 498
SetPrimitivesData, 498
TRIANGLE_FAN, 497
TRIANGLE_STRIP, 497
TRIANGLE, 497
VERTEX, 497
gdcm::ModalityPerformedProcedureStepCreateQuery, 499
GetAbstractSyntaxUID, 501
GetRequiredDataSet, 501
ModalityPerformedProcedureStepCreateQuery, 501
QueryFactory, 501
ValidateQuery, 501
gdcm::ModalityPerformedProcedureStepSetQuery, 501
GetAbstractSyntaxUID, 503
GetRequiredDataSet, 503
ModalityPerformedProcedureStepSetQuery, 503
QueryFactory, 503
ValidateQuery, 503
gdcm::ModifiedEvent, 504
gdcm::Module, 505
AddMacro, 506
AddModuleEntry, 506
ArrayIncludeMacrosType, 506
Clear, 506
FindModuleEntryInMacros, 506
GetModuleEntryInMacros, 506
GetName, 506
MapModuleEntry, 506
Module, 506
operator<<, 507
SetName, 507
Verify, 507
gdcm::ModuleEntry, 507
~ModuleEntry, 509
DataElementType, 510
Description, 509
DescriptionField, 510
GetDescription, 509
GetName, 509
GetType, 509
ModuleEntry, 509
Name, 510
operator<<, 510
SetDescription, 509
SetName, 510
SetType, 510
gdcm::Modules, 510
AddModule, 511
Clear, 511
GetModule, 511
IsEmpty, 511
ModuleMapType, 511
Modules, 511
operator<<, 512
gdcm::MovePatientRootQuery, 512
GetAbstractSyntaxUID, 513
GetTagListByLevel, 513
InitializeDataSet, 514
MovePatientRootQuery, 513
QueryFactory, 514
ValidateQuery, 514
gdcm::MoveStudyRootQuery, 515
GetAbstractSyntaxUID, 516
GetTagListByLevel, 516
InitializeDataSet, 516
MoveStudyRootQuery, 516
QueryFactory, 517
ValidateQuery, 516
gdcm::NestedModuleEntries, 525
AddModuleEntry, 527
GetModuleEntry, 527
GetNumberOfModuleEntries, 527
NestedModuleEntries, 527
operator<<, 527
SizeType, 527
gdcm::NoEvent, 533
gdcm::NormalizedNetworkFunctions, 535
ConstructQuery, 536
NAction, 536
NCreate, 536
NDelete, 536
NEventReport, 536
NGet, 536
NSet, 536
gdcm::Object, 539
~Object, 540
Object, 540
operator<<, 541
operator=, 541
Print, 541
Register, 541

SmartPointer, 541
UnRegister, 541
gdcm::OpenSSLCryptoFactory, 541
CreateCMSProvider, 542
InitOpenSSL, 542
OpenSSLCryptoFactory, 542
gdcm::OpenSSLCryptographicMessageSyntax, 543
~OpenSSLCryptographicMessageSyntax, 544
Decrypt, 544
Encrypt, 544
GetCipherType, 544
OpenSSLCryptographicMessageSyntax, 544
ParseCertificateFile, 544
ParseKeyFile, 544
SetCipherType, 544
SetPassword, 545
gdcm::OpenSSLP7CryptoFactory, 545
CreateCMSProvider, 546
OpenSSLP7CryptoFactory, 546
gdcm::OpenSSLP7CryptographicMessageSyntax, 546
~OpenSSLP7CryptographicMessageSyntax, 548
Decrypt, 548
Encrypt, 548
GetCipherType, 548
OpenSSLP7CryptographicMessageSyntax, 548
ParseCertificateFile, 548
ParseKeyFile, 548
SetCipherType, 548
SetPassword, 548
gdcm::Orientation, 549
~Orientation, 550
AXIAL, 550
CORONAL, 550
GetLabel, 550
GetMajorAxisFromPatientRelativeDirectionCosine, 550
GetObliquityThresholdCosineValue, 550
GetType, 550
OBLIQUE, 550
operator<<, 551
Orientation, 550
OrientationType, 550
Print, 550
SAGITTAL, 550
SetObliquityThresholdCosineValue, 551
UNKNOWN, 550
gdcm::Overlay, 551
~Overlay, 554
Decompress, 554
GetBitPosition, 554
GetBitsAllocated, 554
GetColumns, 555
GetDescription, 555
GetGroup, 555
GetOrigin, 555
GetOverlayData, 555
GetOverlayTypeAsString, 555
GetOverlayTypeFromString, 555
GetRows, 555
GetType, 555
GetTypeAsEnum, 555
GetUnpackBuffer, 556
GetUnpackBufferLength, 556
GrabOverlayFromPixelData, 556
Graphics, 554
Invalid, 554
IsEmpty, 556
IsInPixelData, 556
IsZero, 556
operator=, 556
Overlay, 554
OverlayType, 554
Print, 556
ROI, 554
SetBitPosition, 556
SetBitsAllocated, 556
SetColumns, 557
SetDescription, 557
SetFrameOrigin, 557
SetGroup, 557
SetNumberOfFrames, 557
SetOrigin, 557
SetOverlay, 557
SetRows, 557
SetType, 557
Update, 557
gdcm::PDBElement, 565
GetName, 566
GetValue, 566
NameField, 567
operator<<, 567
operator==, 566
PDBElement, 566
SetName, 567
SetValue, 567
ValueField, 567
gdcm::PDBHeader, 567
~PDBHeader, 568
FindPDBElementByName, 568
GetPDBEEnd, 568
GetPDBElementByName, 569
GetPDBInfoTag, 569
LoadFromDataElement, 569
operator<<, 569
PDBHeader, 568
Print, 569
gdcm::PDFCodec, 570
~PDFCodec, 571

CanCode, 571
CanDecode, 571
Decode, 571
PDFCodec, 571
gdcm::PGXCodec, 575
~PGXCodec, 576
CanCode, 576
CanDecode, 576
Clone, 577
GetHeaderInfo, 577
PGXCodec, 576
Read, 577
Write, 577
gdcm::PNMCodec, 598
~PNMCodec, 599
CanCode, 599
CanDecode, 599
Clone, 599
GetBufferLength, 600
GetHeaderInfo, 600
PNMCodec, 599
Read, 600
SetBufferLength, 600
Write, 600
gdcm::PVRGCodec, 623
~PVRGCodec, 624
CanCode, 624
CanDecode, 624
Clone, 624
Code, 625
Decode, 625
PVRGCodec, 624
SetLossyFlag, 625
gdcm::ParseException, 558
~ParseException, 559
GetLastElement, 559
operator=, 559
ParseException, 559
SetLastElement, 559
gdcm::Parser, 560
~Parser, 562
DuplicateAttributeError, 561
EndElementHandler, 561
ErrorType, 561
GetBuffer, 562
GetCurrentByteIndex, 562
GetErrorCode, 562
GetErrorString, 562
GetUserData, 562
JunkAfterDocElementError, 561
NoElementsError, 561
NoError, 561
NoMemoryError, 561
Parse, 562
ParseBuffer, 562
Parser, 562
Process, 562
SetElementHandler, 562
SetUserData, 562
StartElementHandler, 561
SyntaxError, 561
TagMismatchError, 561
UndefinedEntityError, 561
UnexpectedStateError, 561
gdcm::Patient, 562
Patient, 563
gdcm::PersonName, 574
Component, 575
GetMaxLength, 574
GetNumberOfComponents, 574
MaxLength, 575
MaxNumberOfComponents, 575
Padding, 575
Print, 574
Separator, 575
SetBlob, 574
SetComponents, 574, 575
gdcm::PhotometricInterpretation, 577
ARGB, 578
CMYK, 578
GetPIString, 579
GetPIType, 579
GetSamplesPerPixel, 579
GetString, 579
GetType, 579
HSV, 578
IsLossless, 579
IsLossy, 579
IsRetired, 579
IsSameColorSpace, 579
MONOCHROME1, 578
MONOCHROME2, 578
operator PIType, 579
operator<<, 579
PALETTE_COLOR, 578
PI_END, 578
PIType, 578
PhotometricInterpretation, 579
RGB, 578
UNKNOWN, 578
YBR_FULL_422, 578
YBR_FULL, 578
YBR_ICT, 578
YBR_PARTIAL_420, 578
YBR_PARTIAL_422, 578
YBR_RCT, 578
gdcm::PixelFormat, 580
Bitmap, 585

FLOAT16, 582
FLOAT32, 582
FLOAT64, 582
GetBitsAllocated, 582
GetBitsStored, 582
GetHighBit, 583
GetMax, 583
GetMin, 583
GetPixelRepresentation, 583
GetPixelSize, 583
GetSamplesPerPixel, 583
GetScalarType, 584
GetScalarTypeAsString, 584
INT12, 582
INT16, 582
INT32, 582
INT64, 582
INT8, 582
IsCompatible, 584
IsValid, 584
operator ScalarType, 584
operator!=, 584
operator<<, 585
operator==, 584
PixelFormat, 582
Print, 584
SINGLEBIT, 582
ScalarType, 582
SetBitsAllocated, 584
SetBitsStored, 585
SetHighBit, 585
SetPixelRepresentation, 585
SetSamplesPerPixel, 585
SetScalarType, 585
UINT12, 582
UINT16, 582
UINT32, 582
UINT64, 582
UINT8, 582
UNKNOWN, 582
Validate, 585
gdcm::Pixmap, 586
~Pixmap, 588
AreOverlaysInPixelData, 588
Curves, 589
GetCurve, 588
GetIconImage, 588
GetNumberOfCurves, 588
GetNumberOfOverlays, 588
GetOverlay, 588
Icon, 589
Overlays, 589
Pixmap, 588
Print, 588
RemoveOverlay, 588
SetIconImage, 589
SetNumberOfCurves, 589
SetNumberOfOverlays, 589
gdcm::PixmapReader, 589
~PixmapReader, 591
GetPixmap, 591
PixelData, 592
PixmapReader, 591
Read, 591
ReadACRNEMAImage, 591
ReadImage, 592
ReadImageInternal, 592
gdcm::PixmapToPixmapFilter, 592
~PixmapToPixmapFilter, 594
GetInput, 594
GetOutput, 594
GetOutputAsPixmap, 594
PixmapToPixmapFilter, 594
gdcm::PixmapWriter, 594
~PixmapWriter, 596
DoIconImage, 596
GetImage, 596
GetPixmap, 596, 597
PixelData, 597
PixmapWriter, 596
PrepareWrite, 597
SetImage, 597
SetPixmap, 597
Write, 597
gdcm::Preamble, 600
~Preamble, 601
Clear, 601
Create, 601
GetInternal, 601
GetLength, 601
IsEmpty, 601
IsValid, 601
operator<<, 601
operator=, 601
Preamble, 601
Print, 601
Read, 601
Remove, 601
Valid, 601
Write, 601
gdcm::PresentationContext, 602
AbstractSyntax, 604
AddTransferSyntax, 603
GetAbstractSyntax, 603
GetNumberOfTransferSyntaxes, 603
GetPresentationContextID, 603
GetTransferSyntax, 603
ID, 604

operator==, 603
PresentationContext, 603
Print, 603
SetAbstractSyntax, 604
SetPresentationContextID, 604
SizeType, 603
TransferSyntaxArrayType, 603
TransferSyntaxes, 604
gdcm::PresentationContextGenerator, 605
AddFromFile, 607
AddPresentationContext, 607
GenerateFromFilenames, 607
GenerateFromUID, 607
GetDefaultTransferSyntax, 607
GetPresentationContexts, 607
PresentationContextArrayType, 607
PresentationContextGenerator, 607
SetDefaultTransferSyntax, 607
SetMergeModeToAbstractSyntax, 608
SetMergeModeToTransferSyntax, 608
SizeType, 607
gdcm::Printer, 612
∼Printer, 614
CONDENSED_STYLE, 614
F, 616
GetPrintStyle, 614
MaxPrintLength, 616
Print, 614
PrintDataElement, 615
PrintDataSet, 615
PrintSQ, 615
PrintStyle, 616
PrintStyles, 614
Printer, 614
SetColor, 615
SetFile, 615
SetStyle, 615
VERBOSE_STYLE, 614
XML, 614
gdcm::PrivateDict, 616
∼PrivateDict, 617
AddDictEntry, 617
Dicts, 617
FindDictEntry, 617
GetDictEntry, 617
IsEmpty, 617
LoadDefault, 617
operator<<, 617
PrintXML, 617
PrivateDict, 617
RemoveDictEntry, 617
gdcm::PrivateTag, 618
GetAsDataElement, 619
GetOwner, 619
operator<, 619
operator<<, 620
PrivateTag, 619
ReadFromCommaSeparatedString, 619
SetOwner, 619
gdcm::ProgressEvent, 620
∼ProgressEvent, 622
CheckEvent, 622
GetEventName, 622
GetProgress, 622
MakeObject, 622
ProgressEvent, 622
Self, 622
SetProgress, 622
Superclass, 622
gdcm::PythonFilter, 625
∼PythonFilter, 626
GetFile, 626
PythonFilter, 626
SetDicts, 626
SetFile, 626
ToPyObject, 626
UseDictAlways, 626
gdcm::QueryBase, 626
∼QueryBase, 627
GetAllRequiredTags, 627
GetAllTags, 627
GetHierachicalSearchTags, 628
GetName, 628
GetOptionalTags, 628
GetQueryLevel, 628
GetRequiredTags, 628
GetUniqueTags, 628
gdcm::QueryFactory, 629
GetCharacterFromCurrentLocale, 629
ListCharSets, 629
ProduceCharacterSetDataElement, 629
ProduceQuery, 630
gdcm::QueryImage, 630
GetHierachicalSearchTags, 631
GetName, 631
GetOptionalTags, 631
GetQueryLevel, 632
GetRequiredTags, 632
GetUniqueTags, 632
gdcm::QueryPatient, 632
GetHierachicalSearchTags, 633
GetName, 633
GetOptionalTags, 633
GetQueryLevel, 634
GetRequiredTags, 634
GetUniqueTags, 634
gdcm::QuerySeries, 634
GetHierachicalSearchTags, 635

GetName, 635
GetOptionalTags, 635
GetQueryLevel, 636
GetRequiredTags, 636
GetUniqueTags, 636
gdcm::QueryStudy, 636
GetHierachicalSearchTags, 637
GetName, 637
GetOptionalTags, 637
GetQueryLevel, 638
GetRequiredTags, 638
GetUniqueTags, 638
gdcm::RAWCodec, 638
~RAWCodec, 639
CanCode, 639
CanDecode, 639
Clone, 640
Code, 640
Decode, 640
DecodeByStreams, 640
DecodeBytes, 640
GetHeaderInfo, 640
RAWCodec, 639
gdcm::RLECodec, 653
~RLECodec, 655
AppendFrameEncode, 655
AppendRowEncode, 655
CanCode, 655
CanDecode, 655
Clone, 655
Code, 656
Decode, 656
DecodeByStreams, 656
DecodeExtent, 656
GetBufferLength, 656
GetHeaderInfo, 656
ImageRegionReader, 657
IsFrameEncoder, 656
IsRowEncoder, 656
RLECodec, 655
SetBufferLength, 656
SetLength, 657
StartEncode, 657
StopEncode, 657
gdcm::Reader, 641
~Reader, 644
CanRead, 644
F, 646
GetFile, 644
GetStreamCurrentPosition, 644
GetStreamPtr, 644
Read, 645
ReadDataSet, 645
ReadMetaInformation, 645
ReadPreamble, 645
ReadSelectedPrivateTags, 645
ReadSelectedTags, 645
ReadUpToTag, 645
Reader, 644
SetFile, 646
SetFileName, 646
SetStream, 646
StreamImageReader, 646
gdcm::RealWorldValueMappingContent, 647
CodeMeaning, 647
CodeValue, 647
RealWorldValueIntercept, 647
RealWorldValueSlope, 647
gdcm::Region, 648
~Region, 649
Area, 649
Clone, 649
ComputeBoundingBox, 649
Empty, 649
IsValid, 649
Print, 649
Region, 649
gdcm::Rescaler, 650
~Rescaler, 651
ComputeInterceptSlopePixelType, 651
ComputePixelTypeFromMinMax, 651
GetIntercept, 652
GetSlope, 652
InverseRescale, 652
InverseRescaleFunctionIntoBestFit, 652
Rescale, 652
RescaleFunctionIntoBestFit, 652
Rescaler, 651
SetIntercept, 652
SetMinMaxForPixelType, 652
SetPixelFormat, 652
SetSlope, 652
SetTargetPixelType, 652
SetUseTargetPixelType, 653
gdcm::SHA1, 700
~SHA1, 701
Compute, 701
ComputeFile, 701
SHA1, 701
gdcm::SOPClassUIDToIOD, 710
const, 711
GetIODFromSOPClassUID, 711
GetIOD, 711
GetNumberOfSOPClassToIOD, 711
GetSOPClassUIDFromIOD, 711
GetSOPClassUIDToIODs, 712
GetSOPClassUIDToIOD, 712
gdcm::STATIC_ASSERTION_FAILURE< true >, 721

value, 721
gdcm::STATIC_ASSERTION_FAILURE< x >, 720
gdcm::Scanner, 659
~Scanner, 662
AddPrivateTag, 662
AddSkipTag, 662
AddTag, 662
Begin, 663
ClearSkipTags, 663
ClearTags, 663
ConstIterator, 662
End, 663
GetAllFilenamesFromTagToValue, 663
GetFilenameFromTagToValue, 663
GetFilenames, 663
GetKeys, 663
GetMapping, 663
GetMappingFromTagToValue, 663
GetMappings, 664
GetOrderedValues, 664
GetValue, 664
GetValues, 664
IsKey, 664
MappingType, 662
New, 664
operator<<, 665
Print, 665
ProcessPublicTag, 665
Scan, 665
Scanner, 662
TagToValue, 662
TagToValueValueType, 662
ValuesType, 662
gdcm::Scanner::ltstr, 475
operator(), 476
gdcm::Segment, 666
~Segment, 668
ALGOType, 668
ALGOType_END, 668
AUTOMATIC, 668
AddSurface, 668
AnatomicRegion, 670
GetALGOType, 668
GetALGOTypeString, 668
GetAnatomicRegion, 668
GetPropertyCategory, 668, 669
GetPropertyType, 669
GetSegmentAlgorithmName, 669
GetSegmentAlgorithmType, 669
GetSegmentDescription, 669
GetSegmentLabel, 669
GetSegmentNumber, 669
GetSurface, 669
GetSurfaceCount, 669
GetSurfaces, 669
MANUAL, 668
PropertyCategory, 670
PropertyType, 670
Segment, 668
SegmentAlgorithmName, 670
SegmentAlgorithmType, 670
SegmentDescription, 670
SegmentLabel, 670
SegmentNumber, 670
SetAnatomicRegion, 669
SetPropertyCategory, 669
SetPropertyType, 669
SetSegmentAlgorithmName, 669
SetSegmentAlgorithmType, 669
SetSegmentDescription, 669
SetSegmentLabel, 669
SetSegmentNumber, 669
SetSurfaceCount, 670
SurfaceCount, 670
SurfaceVector, 668
Surfaces, 670
gdcm::SegmentHelper, 76
gdcm::SegmentHelper::BasicCodedEntry, 152
BasicCodedEntry, 153
CSD, 154
CSV, 154
CM, 153
CV, 154
IsEmpty, 153
gdcm::SegmentReader, 673
~SegmentReader, 675
GetSegments, 675
Read, 675
ReadSegment, 675
ReadSegments, 675
SegmentMap, 675
SegmentReader, 675
SegmentVector, 675
Segments, 675
gdcm::SegmentWriter, 676
~SegmentWriter, 677
AddSegment, 677
GetNumberOfSegments, 677
GetSegment, 677
GetSegments, 677
PrepareWrite, 677
SegmentVector, 677
SegmentWriter, 677
Segments, 678
SetNumberOfSegments, 677
SetSegments, 677
Write, 677
gdcm::SegmentedPaletteColorLookupTable, 671

~SegmentedPaletteColorLookupTable, 672
Print, 672
SegmentedPaletteColorLookupTable, 672
SetLUT, 672
gdcm::SequenceOfFragments, 678
AddFragment, 681
Begin, 681
Clear, 681
ComputeByteLength, 681
ComputeLength, 681
ConstIterator, 680
End, 681
FragmentVector, 680
GetBuffer, 681
GetFragBuffer, 681
GetFragment, 681
GetLength, 681
GetNumberOfFragments, 681
GetTable, 682
Iterator, 680
New, 682
operator==, 682
Print, 682
Read, 682
ReadPreValue, 682
ReadValue, 682
SequenceOfFragments, 680
SetLength, 682
SizeType, 680
Write, 682
WriteBuffer, 683
gdcm::SequenceOfItems, 683
AddItem, 686
AddNewUndefinedLengthItem, 686
Begin, 686, 687
Clear, 687
ComputeLength, 687
ConstIterator, 686
End, 687
FindDataElement, 687
GetItem, 687
GetLength, 687
GetNumberOfItems, 687
IsUndefinedLength, 687
ItemVector, 686
Items, 689
Iterator, 686
New, 688
operator=, 688
operator==, 688
Print, 688
Read, 688
RemoveItemByIndex, 688
SequenceLengthField, 689
SequenceOfItems, 686
SetLength, 688
SetLengthToUndefined, 689
SetNumberOfItems, 689
SizeType, 686
Write, 689
gdcm::SerieHelper, 690
~SerieHelper, 691
AddFile, 691
AddFileName, 691
AddRestriction, 692
Clear, 692
CreateDefaultUniqueSeriesIdentifier, 692
CreateUniqueSeriesIdentifier, 692
FileNameOrdering, 692
GetFirstSingleSerieUIDFileSet, 692
GetNextSingleSerieUIDFileSet, 692
ImagePositionPatientOrdering, 692
ItFileSetHt, 692
OrderFileList, 692
SerieHelper, 691
SerieRestrictions, 691
SetDirectory, 692
SetLoadMode, 692
SetUseSeriesDetails, 692
SingleSerieUIDFileSetHT, 692
SingleSerieUIDFileSetmap, 691
UserOrdering, 692
gdcm::SerieHelper::Rule, 658
elem, 659
group, 659
op, 659
value, 659
gdcm::Series, 693
Series, 693
gdcm::ServiceClassUser, 694
~ServiceClassUser, 696
GetAETitle, 697
GetCalledAETitle, 697
GetTimeout, 697
InitializeConnection, 697
IsPresentationContextAccepted, 697
New, 697
SendEcho, 697
SendFind, 697
SendMove, 697, 698
SendStore, 698
ServiceClassUser, 696
SetAETitle, 698
SetCalledAETitle, 698
SetHostname, 698
SetPort, 699
SetPortSCP, 699
SetPresentationContexts, 699

SetTimeout, 699
StartAssociation, 699
StopAssociation, 699
gdcm::SimpleMemberCommand
~SimpleMemberCommand, 703
Execute, 703
m_MemberFunction, 704
m_This, 704
New, 704
Self, 703
SetCallbackFunction, 704
SimpleMemberCommand, 703
TMemberFunctionPointer, 703
gdcm::SimpleMemberCommand< T >, 701
gdcm::SimpleSubjectWatcher, 704
~SimpleSubjectWatcher, 705
EndFilter, 705
ShowAbort, 705
ShowAnonymization, 705
ShowData, 705
ShowDataSet, 705
ShowFileName, 705
ShowIteration, 705
ShowProgress, 706
SimpleSubjectWatcher, 705
StartFilter, 706
TestAbortOff, 706
TestAbortOn, 706
gdcm::SmartPointer
~SmartPointer, 708
GetPointer, 708
operator ObjectType *, 708
operator*, 709
operator->, 709
operator=, 709
SmartPointer, 708
gdcm::SmartPointer< ObjectType >, 706
gdcm::Sorter, 712
~Sorter, 714
AddSelect, 714
Filenames, 715
GetFilenames, 714
operator<<, 715
Print, 714
Selection, 715
SelectionMap, 714
SetSortFunction, 714
Sort, 714
SortFunc, 715
SortFunction, 714
Sorter, 714
StableSort, 715
gdcm::Spacing, 715
~Spacing, 717
CALIBRATED, 717
ComputePixelAspectRatioFromPixelSpacing, 717
DETECTOR, 717
MAGNIFIED, 717
Spacing, 717
SpacingType, 717
UNKNOWN, 717
gdcm::Spectroscopy, 717
Spectroscopy, 718
gdcm::SplitMosaicFilter, 718
~SplitMosaicFilter, 718
ComputeMOSAICDimensions, 718
GetFile, 718, 719
GetImage, 719
SetFile, 719
SetImage, 719
Split, 719
SplitMosaicFilter, 718
gdcm::StartEvent, 719
gdcm::StreamImageReader, 721
~StreamImageReader, 722
CanReadImage, 722
DefinePixelExtent, 722
DefineProperBufferLength, 722
GetDimensionsValueForResolution, 723
GetFile, 723
Read, 723
ReadImageInformation, 723
SetFileName, 723
SetStream, 724
StreamImageReader, 722
gdcm::StreamImageWriter, 724
~StreamImageWriter, 727
CanWriteFile, 727
DefinePixelExtent, 727
DefineProperBufferLength, 727
mElementOffsets, 729
mElementOffsets1, 729
mWriter, 729
mXMax, 729
mXMin, 729
mYMax, 729
mYMin, 729
mZMax, 729
mZMin, 729
mspFile, 729
SetFile, 727
SetFileName, 727
SetStream, 728
StreamImageWriter, 727
Write, 728
WriteImageInformation, 728
WriteImageSubregionRAW, 728
WriteRawHeader, 728

gdcm::StrictScanner, 730
~StrictScanner, 733
AddPrivateTag, 733
AddSkipTag, 733
AddTag, 733
Begin, 733
ClearSkipTags, 733
ClearTags, 733
ConstIterator, 732
End, 733
GetAllFilenamesFromTagToValue, 733
GetFilenameFromTagToValue, 733
GetFilenames, 733
GetKeys, 734
GetMapping, 734
GetMappingFromTagToValue, 734
GetMappings, 734
GetOrderedValues, 734
GetValue, 734
GetValues, 734
IsKey, 735
MappingType, 732
New, 735
operator<<, 735
Print, 735
ProcessPublicTag, 735
Scan, 735
StrictScanner, 733
TagToValue, 732
TagToValueValueType, 732
ValuesType, 733
gdcm::StrictScanner::ltstr, 476
operator(), 476
gdcm::String
const_iterator, 738
const_reference, 738
const_reverse_iterator, 738
difference_type, 738
IsValid, 739
iterator, 738
operator const char *, 739
pointer, 738
reference, 738
reverse_iterator, 738
size_type, 738
String, 738, 739
Trim, 739
Truncate, 739
value_type, 738
gdcm::String< TDelimiter, TMaxLength, TPadChar >, 736
gdcm::StringFilter, 740
~StringFilter, 741
ExecuteQuery, 741
FromString, 741
GetFile, 741
SetDicts, 741
SetFile, 741
StringFilter, 741
ToString, 741, 742
ToStringPair, 742
UseDictAlways, 742
gdcm::Study, 742
Study, 743
gdcm::Subject, 743
~Subject, 745
AddObserver, 745
GetCommand, 745
HasObserver, 745
InvokeEvent, 745
RemoveAllObservers, 745
RemoveObserver, 745
Subject, 745
gdcm::Surface, 746
~Surface, 749
GetAlgorithmFamily, 749
GetAlgorithmName, 749
GetAlgorithmVersion, 749
GetAxisOfRotation, 749
GetCenterOfRotation, 749
GetFiniteVolume, 750
GetManifold, 750
GetMaximumPointDistance, 750
GetMeanPointDistance, 750
GetMeshPrimitive, 750
GetNumberOfSurfacePoints, 750
GetNumberOfVectors, 750
GetPointCoordinatesData, 750
GetPointPositionAccuracy, 750
GetPointsBoundingBoxCoordinates, 750
GetProcessingAlgorithm, 750, 751
GetRecommendedDisplayCIELabValue, 751
GetRecommendedDisplayGrayscaleValue, 751
GetRecommendedPresentationOpacity, 751
GetRecommendedPresentationType, 751
GetSTATESString, 751
GetSTATES, 751
GetSurfaceComments, 751
GetSurfaceNumber, 751
GetSurfaceProcessing, 751
GetSurfaceProcessingDescription, 751
GetSurfaceProcessingRatio, 751
GetVIEWType, 751
GetVIEWTypeString, 751
GetVectorAccuracy, 751
GetVectorCoordinateData, 751
GetVectorDimensionality, 751
NO, 749
POINTS, 749

STATES_END, 749
STATES, 749
SURFACE, 749
SetAlgorithmFamily, 751
SetAlgorithmName, 751
SetAlgorithmVersion, 752
SetAxisOfRotation, 752
SetCenterOfRotation, 752
SetFiniteVolume, 752
SetManifold, 752
SetMaximumPointDistance, 752
SetMeanPointDistance, 752
SetMeshPrimitive, 752
SetNumberOfSurfacePoints, 752
SetNumberOfVectors, 752
SetPointCoordinatesData, 752
SetPointPositionAccuracy, 752
SetPointsBoundingBoxCoordinates, 752
SetProcessingAlgorithm, 752
SetRecommendedDisplayCIELabValue, 752
SetRecommendedDisplayGrayscaleValue, 752
SetRecommendedPresentationOpacity, 752
SetRecommendedPresentationType, 752
SetSurfaceComments, 752
SetSurfaceNumber, 753
SetSurfaceProcessing, 753
SetSurfaceProcessingDescription, 753
SetSurfaceProcessingRatio, 753
SetVectorAccuracy, 753
SetVectorCoordinateData, 753
SetVectorDimensionality, 753
Surface, 749
UNKNOWN, 749
VIEWType, 749
VIEWType_END, 749
WIREFRAME, 749
YES, 749
gdcm::SurfaceHelper, 753
ColorArray, 754
RGBToRecommendedDisplayCIELab, 755
RGBToRecommendedDisplayGrayscale, 755
RecommendedDisplayCIELabToRGB, 754
gdcm::SurfaceReader, 756
~SurfaceReader, 758
GetNumberOfSurfaces, 758
Read, 758
ReadPointMacro, 758
ReadSurface, 758
ReadSurfaces, 758
SurfaceReader, 758
gdcm::SurfaceWriter, 758
~SurfaceWriter, 760
ComputeNumberOfSurfaces, 760
GetNumberOfSurfaces, 760
NumberOfSurfaces, 760
PrepareWrite, 760
PrepareWritePointMacro, 760
SetNumberOfSurfaces, 760
SurfaceWriter, 760
Write, 760
gdcm::SwapCode, 760
BadBigEndian, 761
BadLittleEndian, 761
BigEndian, 761
GetIndex, 762
GetSwapCodeString, 762
LittleEndian, 761
operator SwapCode::SwapCodeType, 762
operator<<, 762
SwapCode, 762
SwapCodeType, 761
Unknown, 761
gdcm::SwapperDoOp, 762
Swap, 763
SwapArray, 763
gdcm::SwapperNoOp, 763
Swap, 763
SwapArray, 763
gdcm::System, 763
DeleteDirectory, 765
EncodeBytes, 765
FileExists, 765
FileIsDirectory, 765
FileIsSymlink, 765
FileSize, 765
FileTime, 766
FormatDateTime, 766
GetCWD, 766
GetCurrentDateTime, 766
GetCurrentModuleFileName, 766
GetCurrentProcessFileName, 766
GetCurrentResourcesDirectory, 766
GetHostName, 767
GetLastSystemError, 767
GetLocaleCharset, 767
GetPermissions, 767
GetTimezoneOffsetFromUTC, 767
MakeDirectory, 767
ParseDateTime, 767
RemoveFile, 768
SetPermissions, 768
StrCaseCmp, 768
StrNCaseCmp, 768
StrSep, 768
StrTokR, 768
gdcm::Table, 768
~Table, 769
GetTableEntry, 769

InsertEntry, 769
MapTableEntry, 769
operator<<, 770
Table, 769
gdcm::TableEntry, 770
~TableEntry, 770
TableEntry, 770
gdcm::TableReader, 771
~TableReader, 772
CharacterDataHandler, 772
EndElement, 772
GetDefs, 772
GetFilename, 772
HandleIODEntry, 772
HandleIOD, 772
HandleMacro, 772
HandleMacroEntry, 772
HandleMacroEntryDescription, 772
HandleModule, 772
HandleModuleEntry, 772
HandleModuleEntryDescription, 772
HandleModuleInclude, 772
Read, 772
SetFilename, 772
StartElement, 772
TableReader, 772
gdcm::Tag, 774
bytes, 781
GetElement, 776
GetElementTag, 776
GetGroup, 777
GetLength, 777
GetPrivateCreator, 777
IsGroupLength, 777
IsGroupXX, 777
IsIllegal, 777
IsPrivate, 777
IsPrivateCreator, 778
IsPublic, 778
operator!=, 778
operator<, 778
operator<<, 781
operator<=, 778
operator>>, 781
operator=, 778
operator==, 778
operator[], 779
PrintAsContinuousString, 779
PrintAsContinuousUpperCaseString, 779
PrintAsPipeSeparatedString, 779
Read, 779
ReadFromCommaSeparatedString, 779
ReadFromContinuousString, 779
ReadFromPipeSeparatedString, 780
SetElement, 780
SetElementTag, 780
SetGroup, 780
SetPrivateCreator, 780
Tag, 776
tag, 781
tags, 781
Write, 781
gdcm::TagPath, 781
~TagPath, 782
ConstructFromString, 782
ConstructFromTagList, 782
IsValid, 782
Print, 782
Push, 783
TagPath, 782
gdcm::Testing, 783
~Testing, 785
ComputeFileMD5, 785
ComputeMD5, 785
GetDataExtraRoot, 785
GetDataRoot, 785
GetFileName, 785
GetFileNames, 785
GetLossyFlagFromFile, 786
GetMD5DataImage, 786
GetMD5DataImages, 786
GetMD5FromBrokenFile, 786
GetMD5FromFile, 786
GetMediaStorageDataFile, 786
GetMediaStorageDataFiles, 786
GetMediaStorageFromFile, 786
GetNumberOfFileNames, 786
GetNumberOfMD5DataImages, 786
GetNumberOfMediaStorageDataFiles, 787
GetPixelSpacingDataRoot, 787
GetSelectedPrivateGroupOffsetFromFile, 787
GetSelectedTagsOffsetFromFile, 787
GetSourceDirectory, 787
GetStreamOffsetFromFile, 787
GetTempDirectory, 787
GetTempDirectoryW, 787
GetTempFilename, 787
GetTempFilenameW, 788
MD5DataImagesType, 784
MediaStorageDataFilesType, 784
Print, 788
Testing, 785
gdcm::Trace, 788
~Trace, 790
DebugOff, 790
DebugOn, 790
ErrorOff, 790
ErrorOn, 790

GetDebugFlag, 790
GetDebugStream, 790
GetErrorFlag, 790
GetErrorStream, 790
GetStream, 790
GetWarningFlag, 790
GetWarningStream, 790
SetDebug, 790
SetDebugStream, 790
SetError, 791
SetErrorStream, 791
SetStream, 791
SetStreamToFile, 791
SetWarning, 791
SetWarningStream, 791
Trace, 790
WarningOff, 791
WarningOn, 792
gdcm::TransferSyntax, 792
CT_private_ELE, 795
CanStoreLossy, 795
DeflatedExplicitVRLittleEndian, 795
Explicit, 794
ExplicitVRBigEndian, 795
ExplicitVRLittleEndian, 795
GetNegociatedType, 795
GetString, 796
GetSwapCode, 796
GetTSString, 796
GetTSType, 796
Implicit, 794
ImplicitVRBigEndianACRNEMA, 795
ImplicitVRBigEndianPrivateGE, 795
ImplicitVRLittleEndian, 795
IsEncapsulated, 796
IsEncoded, 796
IsExplicit, 796
IsImplicit, 796
IsLossless, 796
IsLossy, 796
IsValid, 796
JPEG2000, 795
JPEG2000Lossless, 795
JPEG2000Part2, 795
JPEG2000Part2Lossless, 795
JPEGBaselineProcess1, 795
JPEGExtendedProcess2_4, 795
JPEGExtendedProcess3_5, 795
JPEGFullProgressionProcess10_12, 795
JPEGLSLossless, 795
JPEGLSNearLossless, 795
JPEGLosslessProcess14, 795
JPEGLosslessProcess14_1, 795
JPEGSpectralSelectionProcess6_8, 795
JPIPReferenced, 795
MPEG2MainProfile, 795
MPEG2MainProfileHighLevel, 795
MPEG4AVCH264BDcompatibleHighProfileLevel4_1, 795
MPEG4AVCH264HighProfileLevel4_1, 795
NegociatedType, 794
operator TSType, 797
operator<<, 797
RLELossless, 795
TS_END, 795
TSType, 794
TransferSyntax, 795
Unknown, 794
gdcm::Type, 800
GetTypeString, 802
GetTypeType, 802
operator TypeType, 802
operator<<, 802
T1, 801
T1C, 801
T2, 801
T2C, 801
T3, 801
Type, 802
TypeType, 801
UNKNOWN, 801
gdcm::UIDGenerator, 803
Generate, 804
GenerateUUID, 804
GetGDCMUID, 804
GetRoot, 804
IsValid, 804
SetRoot, 805
UIDGenerator, 804
gdcm::UIDs, 805
AmbulatoryECGWaveformStorage, 813
AudioSRStorageTrialRetired, 814
BasicAnnotationBoxSOPClass, 812
BasicColorImageBoxSOPClass, 812
BasicColorPrintManagementMetaSOPClass, 813
BasicFilmBoxSOPClass, 812
BasicFilmSessionSOPClass, 812
BasicGrayscaleImageBoxSOPClass, 812
BasicGrayscalePrintManagementMetaSOPClass, 812
BasicPrintImageOverlayBoxSOPClassRetired, 813
BasicStudyContentNotificationSOPClassRetired, 812
BasicTextSRStorage, 814
BasicVoiceAudioWaveformStorage, 814
BlendingSoftcopyPresentationStateStorageSOPClass, 814
BreastImagingRelevantPatientInformationQuery, 816

BreastTomosynthesisImageStorage, 817
CTImageStorage, 813
CardiacElectrophysiologyWaveformStorage, 814
CardiacRelevantPatientInformationQuery, 816
ChestCADSRStorage, 815
ColorSoftcopyPresentationStateStorageSOPClass, 814
ComprehensiveSRStorage, 815
ComprehensiveSRStorageTrialRetired, 814
ComputedRadiographyImageStorage, 813
DICOMApplicationContextName, 812
DICOMControlledTerminology, 812
DICOMUIDRegistry, 812
DeflatedExplicitVRLittleEndian, 810
DeformableSpatialRegistrationStorage, 814
DetachedInterpretationManagementSOPClass↩ Retired, 812
DetachedPatientManagementMetaSOPClass↩ Retired, 812
DetachedPatientManagementSOPClassRetired, 812
DetachedResultsManagementMetaSOPClass↩ Retired, 812
DetachedResultsManagementSOPClassRetired, 812
DetachedStudyManagementMetaSOPClassRetired, 812
DetachedStudyManagementSOPClassRetired, 812
DetachedVisitManagementSOPClassRetired, 812
DetailSRStorageTrialRetired, 814
dicomAETitle, 816
dicomApplicationCluster, 816
dicomAssociationAcceptor, 816
dicomAssociationInitiator, 816
dicomAuthorizedNodeCertificateReference, 816
dicomConfigurationRoot, 816
dicomDescription, 816
dicomDevice, 817
dicomDeviceName, 816
dicomDeviceSerialNumber, 816
dicomDevicesRoot, 816
dicomHostname, 816
dicomInstalled, 816
dicomInstitutionAddress, 816
dicomInstitutionDepartmentName, 816
dicomInstitutionName, 816
dicomIssuerOfPatientID, 816
dicomManufacturer, 816
dicomManufacturerModelName, 816
dicomNetworkAE, 817
dicomNetworkConnection, 817
dicomNetworkConnectionReference, 816
dicomPort, 816
dicomPreferredCalledAETitle, 816
dicomPreferredCallingAETitle, 816
dicomPrimaryDeviceType, 816
dicomRelatedDeviceReference, 816
dicomSOPClass, 816
dicomSoftwareVersion, 816
dicomStationName, 816
dicomSupportedCharacterSet, 816
dicomTLSCyphersuite, 816
dicomThisNodeCertificateReference, 816
dicomTransferCapability, 817
dicomTransferRole, 816
dicomTransferSyntax, 816
dicomUniqueAETitle, 817
dicomUniqueAETitlesRegistryRoot, 817
dicomVendorData, 816
DigitalIntraoralXRayImageStorageForPresentation, 813
DigitalIntraoralXRayImageStorageForProcessing, 813
DigitalMammographyXRayImageStorageFor↩ Presentation, 813
DigitalMammographyXRayImageStorageFor↩ Processing, 813
DigitalXRayImageStorageForPresentation, 813
DigitalXRayImageStorageForProcessing, 813
EncapsulatedCDAStorage, 815
EncapsulatedPDFStorage, 815
EnhancedCTImageStorage, 813
EnhancedMRImageStorage, 813
EnhancedSRStorage, 814
EnhancedUSVolumeStorage, 817
EnhancedXAImageStorage, 814
EnhancedXRFImageStorage, 814
ExplicitVRBigEndian, 810
ExplicitVRLittleEndian, 810
GeneralECGWaveformStorage, 813
GeneralPurposePerformedProcedureStepSOP↩ Class, 815
GeneralPurposeScheduledProcedureStepSOP↩ Class, 815
GeneralPurposeWorklistInformationModelFIND, 815
GeneralPurposeWorklistManagementMetaSOP↩ Class, 815
GeneralRelevantPatientInformationQuery, 816
GetName, 824
GetNumberOfTransferSyntaxStrings, 824
GetString, 824
GetTransferSyntaxString, 824
GetTransferSyntaxStrings, 824
GetUIDName, 824
GetUIDString, 824
GrayscaleSoftcopyPresentationStateStorageSOP↩ Class, 814
HangingProtocolInformationModelFIND, 816
HangingProtocolInformationModelMOVE, 816

HangingProtocolStorage, 816
HardcopyColorImageStorageSOPClassRetired, 813
HardcopyGrayscaleImageStorageSOPClassRetired, 813
HemodynamicWaveformStorage, 814
ICBM452T1FrameofReference, 812
ICBMSingleSubjectMRIFrameofReference, 812
ImageOverlayBoxSOPClassRetired, 813
ImplicitVRLittleEndianDefaultTransferSyntaxforDICOM, 810
InstanceAvailabilityNotificationSOPClass, 815
JPEG2000ImageCompression, 811
JPEG2000ImageCompressionLosslessOnly, 811
JPEG2000Part2MulticomponentImageCompression, 811
JPEG2000Part2MulticomponentImageCompressionLosslessOnly, 811
JPEGBaselineProcess1DefaultTransferSyntaxforLossyJPEG8BitImageCompression, 810
JPEGExtendedHierarchicalProcess1618Retired, 811
JPEGExtendedHierarchicalProcess1719Retired, 811
JPEGExtendedProcess24DefaultTransferSyntaxforLossyJPEG12BitImageCompressionProcess4only, 810
JPEGExtendedProcess35Retired, 810
JPEGFullProgressionHierarchicalProcess2426Retired, 811
JPEGFullProgressionHierarchicalProcess2527Retired, 811
JPEGFullProgressionNonHierarchicalProcess1012Retired, 811
JPEGFullProgressionNonHierarchicalProcess1113Retired, 811
JPEGLSLosslessImageCompression, 811
JPEGLSLossyNearLosslessImageCompression, 811
JPEGLosslessHierarchicalProcess28Retired, 811
JPEGLosslessHierarchicalProcess29Retired, 811
JPEGLosslessNonHierarchicalFirstOrderPredictionProcess14SelectionValue1DefaultTransferSyntaxforLosslessJPEGImageCompression, 811
JPEGLosslessNonHierarchicalProcess14, 811
JPEGLosslessNonHierarchicalProcess15Retired, 811
JPEGSpectralSelectionHierarchicalProcess2022Retired, 811
JPEGSpectralSelectionHierarchicalProcess2123Retired, 811
JPEGSpectralSelectionNonHierarchicalProcess68Retired, 810
JPEGSpectralSelectionNonHierarchicalProcess79Retired, 810
JPIPReferenced, 811
JPIPReferencedDeflate, 811
KeyObjectSelectionDocumentStorage, 815
MPEG2MainProfileMainLevel, 811
MRImageStorage, 813
MRSpectroscopyStorage, 813
MammographyCADSRStorage, 815
MediaCreationManagementSOPClassUID, 813
MediaStorageDirectoryStorage, 811
ModalityPerformedProcedureStepNotificationSOPClass, 812
ModalityPerformedProcedureStepRetrieveSOPClass, 812
ModalityPerformedProcedureStepSOPClass, 812
ModalityWorklistInformationModelFIND, 815
MultiframeGrayscaleByteSecondaryCaptureImageStorage, 813
MultiframeGrayscaleWordSecondaryCaptureImageStorage, 813
MultiframeSingleBitSecondaryCaptureImageStorage, 813
MultiframeTrueColorSecondaryCaptureImageStorage, 813
NuclearMedicineImageStorage, 814
NuclearMedicineImageStorageRetired, 813
operator TSType, 824
OphthalmicPhotography16BitImageStorage, 814
OphthalmicPhotography8BitImageStorage, 814
OphthalmicTomographyImageStorage, 814
PatientRootQueryRetrieveInformationModelFIND, 815
PatientRootQueryRetrieveInformationModelGET, 815
PatientRootQueryRetrieveInformationModelMOVE, 815
PatientStudyOnlyQueryRetrieveInformationModelFINDRetired, 815
PatientStudyOnlyQueryRetrieveInformationModelGETRetired, 815
PatientStudyOnlyQueryRetrieveInformationModelMOVERetired, 815
PositronEmissionTomographyImageStorage, 815
PresentationLUTSOPClass, 813
PrintJobSOPClass, 812
PrintQueueManagementSOPClassRetired, 813
PrintQueueSOPInstanceRetired, 813
PrinterConfigurationRetrievalSOPClass, 812
PrinterConfigurationRetrievalSOPInstance, 813
PrinterSOPClass, 812
PrinterSOPInstance, 812
ProceduralEventLoggingSOPClass, 812
ProceduralEventLoggingSOPInstance, 812
ProcedureLogStorage, 815
ProductCharacteristicsQuerySOPClass, 816
PseudoColorSoftcopyPresentationStateStorageSOPClass, 814

PullPrintRequestSOPClassRetired, 813
PullStoredPrintManagementMetaSOPClassRetired, 813
RFC2557MIMEencapsulation, 811
RLELossless, 811
RTBeamsDeliveryInstructionStorageSupplement74↩ FrozenDraft, 815
RTBeamsTreatmentRecordStorage, 815
RTBrachyTreatmentRecordStorage, 815
RTConventionalMachineVerificationSupplement74↩ FrozenDraft, 815
RTDoseStorage, 815
RTImageStorage, 815
RTIonBeamsTreatmentRecordStorage, 815
RTIonMachineVerificationSupplement74FrozenDraft, 815
RTIonPlanStorage, 815
RTPlanStorage, 815
RTStructureSetStorage, 815
RTTreatmentSummaryRecordStorage, 815
RawDataStorage, 814
RealWorldValueMappingStorage, 814
ReferencedColorPrintManagementMetaSOPClass↩ Retired, 813
ReferencedGrayscalePrintManagementMetaSOP↩ ClassRetired, 812
ReferencedImageBoxSOPClassRetired, 812
SPM2AVG152PDFrameofReference, 812
SPM2AVG152T1FrameofReference, 811
SPM2AVG152T2FrameofReference, 812
SPM2AVG305T1FrameofReference, 811
SPM2BRAINMASKFrameofReference, 811
SPM2CSFFrameofReference, 811
SPM2EPIFrameofReference, 811
SPM2FILT1FrameofReference, 811
SPM2GRAYFrameofReference, 811
SPM2PDFrameofReference, 811
SPM2PETFrameofReference, 811
SPM2SINGLESUBJT1FrameofReference, 812
SPM2SPECTFrameofReference, 811
SPM2T1FrameofReference, 811
SPM2T2FrameofReference, 811
SPM2TRANSMFrameofReference, 811
SPM2WHITEFrameofReference, 811
SecondaryCaptureImageStorage, 813
SegmentationStorage, 814
SetFromUID, 824
SpatialFiducialsStorage, 814
SpatialRegistrationStorage, 814
StandaloneCurveStorageRetired, 813
StandaloneModalityLUTStorageRetired, 814
StandaloneOverlayStorageRetired, 813
StandalonePETCurveStorageRetired, 815
StandaloneVOILUTStorageRetired, 814
StereometricRelationshipStorage, 814
StorageCommitmentPullModelSOPClassRetired, 812
StorageCommitmentPullModelSOPInstanceRetired, 812
StorageCommitmentPushModelSOPClass, 812
StorageCommitmentPushModelSOPInstance, 812
StorageServiceClass, 812
StoredPrintStorageSOPClassRetired, 813
StudyComponentManagementSOPClassRetired, 812
StudyRootQueryRetrieveInformationModelFIND, 815
StudyRootQueryRetrieveInformationModelGET, 815
StudyRootQueryRetrieveInformationModelMOVE, 815
SubstanceAdministrationLoggingSOPClass, 812
SubstanceAdministrationLoggingSOPInstance, 812
SubstanceApprovalQuerySOPClass, 816
SurfaceSegmentationStorage, 817
TSName, 810
TSType, 817
TalairachBrainAtlasFrameofReference, 811
TextSRStorageTrialRetired, 814
TransferSyntaxStringsType, 810
uid_1_2_840_10008_15_0_3_1, 822
uid_1_2_840_10008_15_0_3_10, 823
uid_1_2_840_10008_15_0_3_11, 823
uid_1_2_840_10008_15_0_3_12, 823
uid_1_2_840_10008_15_0_3_13, 823
uid_1_2_840_10008_15_0_3_14, 823
uid_1_2_840_10008_15_0_3_15, 823
uid_1_2_840_10008_15_0_3_16, 823
uid_1_2_840_10008_15_0_3_17, 823
uid_1_2_840_10008_15_0_3_18, 823
uid_1_2_840_10008_15_0_3_19, 823
uid_1_2_840_10008_15_0_3_2, 822
uid_1_2_840_10008_15_0_3_20, 823
uid_1_2_840_10008_15_0_3_21, 823
uid_1_2_840_10008_15_0_3_22, 823
uid_1_2_840_10008_15_0_3_23, 823
uid_1_2_840_10008_15_0_3_24, 823
uid_1_2_840_10008_15_0_3_25, 823
uid_1_2_840_10008_15_0_3_26, 823
uid_1_2_840_10008_15_0_3_27, 823
uid_1_2_840_10008_15_0_3_28, 823
uid_1_2_840_10008_15_0_3_29, 823
uid_1_2_840_10008_15_0_3_3, 822
uid_1_2_840_10008_15_0_3_30, 823
uid_1_2_840_10008_15_0_3_31, 823
uid_1_2_840_10008_15_0_3_4, 822
uid_1_2_840_10008_15_0_3_5, 822
uid_1_2_840_10008_15_0_3_6, 822
uid_1_2_840_10008_15_0_3_7, 822
uid_1_2_840_10008_15_0_3_8, 822

uid_1_2_840_10008_15_0_3_9, 823
uid_1_2_840_10008_15_0_4_1, 823
uid_1_2_840_10008_15_0_4_2, 823
uid_1_2_840_10008_15_0_4_3, 823
uid_1_2_840_10008_15_0_4_4, 823
uid_1_2_840_10008_15_0_4_5, 823
uid_1_2_840_10008_15_0_4_6, 823
uid_1_2_840_10008_15_0_4_7, 823
uid_1_2_840_10008_15_0_4_8, 823
uid_1_2_840_10008_1_1, 817
uid_1_2_840_10008_1_2, 817
uid_1_2_840_10008_1_20_1, 818
uid_1_2_840_10008_1_20_1_1, 818
uid_1_2_840_10008_1_20_2, 818
uid_1_2_840_10008_1_20_2_1, 818
uid_1_2_840_10008_1_2_1, 817
uid_1_2_840_10008_1_2_1_99, 817
uid_1_2_840_10008_1_2_2, 817
uid_1_2_840_10008_1_2_4_100, 818
uid_1_2_840_10008_1_2_4_101, 823
uid_1_2_840_10008_1_2_4_102, 823
uid_1_2_840_10008_1_2_4_103, 823
uid_1_2_840_10008_1_2_4_50, 817
uid_1_2_840_10008_1_2_4_51, 817
uid_1_2_840_10008_1_2_4_52, 817
uid_1_2_840_10008_1_2_4_53, 817
uid_1_2_840_10008_1_2_4_54, 817
uid_1_2_840_10008_1_2_4_55, 817
uid_1_2_840_10008_1_2_4_56, 817
uid_1_2_840_10008_1_2_4_57, 817
uid_1_2_840_10008_1_2_4_58, 817
uid_1_2_840_10008_1_2_4_59, 817
uid_1_2_840_10008_1_2_4_60, 817
uid_1_2_840_10008_1_2_4_61, 817
uid_1_2_840_10008_1_2_4_62, 817
uid_1_2_840_10008_1_2_4_63, 817
uid_1_2_840_10008_1_2_4_64, 817
uid_1_2_840_10008_1_2_4_65, 817
uid_1_2_840_10008_1_2_4_66, 817
uid_1_2_840_10008_1_2_4_70, 817
uid_1_2_840_10008_1_2_4_80, 817
uid_1_2_840_10008_1_2_4_81, 817
uid_1_2_840_10008_1_2_4_90, 817
uid_1_2_840_10008_1_2_4_91, 817
uid_1_2_840_10008_1_2_4_92, 818
uid_1_2_840_10008_1_2_4_93, 818
uid_1_2_840_10008_1_2_4_94, 818
uid_1_2_840_10008_1_2_4_95, 818
uid_1_2_840_10008_1_2_5, 818
uid_1_2_840_10008_1_2_6_1, 818
uid_1_2_840_10008_1_2_6_2, 818
uid_1_2_840_10008_1_3_10, 818
uid_1_2_840_10008_1_40, 818
uid_1_2_840_10008_1_40_1, 818
uid_1_2_840_10008_1_42, 818
uid_1_2_840_10008_1_42_1, 818
uid_1_2_840_10008_1_4_1_1, 818
uid_1_2_840_10008_1_4_1_10, 818
uid_1_2_840_10008_1_4_1_11, 818
uid_1_2_840_10008_1_4_1_12, 818
uid_1_2_840_10008_1_4_1_13, 818
uid_1_2_840_10008_1_4_1_14, 818
uid_1_2_840_10008_1_4_1_15, 818
uid_1_2_840_10008_1_4_1_16, 818
uid_1_2_840_10008_1_4_1_17, 818
uid_1_2_840_10008_1_4_1_18, 818
uid_1_2_840_10008_1_4_1_2, 818
uid_1_2_840_10008_1_4_1_3, 818
uid_1_2_840_10008_1_4_1_4, 818
uid_1_2_840_10008_1_4_1_5, 818
uid_1_2_840_10008_1_4_1_6, 818
uid_1_2_840_10008_1_4_1_7, 818
uid_1_2_840_10008_1_4_1_8, 818
uid_1_2_840_10008_1_4_1_9, 818
uid_1_2_840_10008_1_4_2_1, 818
uid_1_2_840_10008_1_4_2_2, 818
uid_1_2_840_10008_1_9, 818
uid_1_2_840_10008_2_16_4, 818
uid_1_2_840_10008_2_6_1, 818
uid_1_2_840_10008_3_1_1_1, 818
uid_1_2_840_10008_3_1_2_1_1, 818
uid_1_2_840_10008_3_1_2_1_4, 819
uid_1_2_840_10008_3_1_2_2_1, 819
uid_1_2_840_10008_3_1_2_3_1, 819
uid_1_2_840_10008_3_1_2_3_2, 819
uid_1_2_840_10008_3_1_2_3_3, 819
uid_1_2_840_10008_3_1_2_3_4, 819
uid_1_2_840_10008_3_1_2_3_5, 819
uid_1_2_840_10008_3_1_2_5_1, 819
uid_1_2_840_10008_3_1_2_5_4, 819
uid_1_2_840_10008_3_1_2_5_5, 819
uid_1_2_840_10008_3_1_2_6_1, 819
uid_1_2_840_10008_4_2, 819
uid_1_2_840_10008_5_1_1_1, 819
uid_1_2_840_10008_5_1_1_14, 819
uid_1_2_840_10008_5_1_1_15, 819
uid_1_2_840_10008_5_1_1_16, 819
uid_1_2_840_10008_5_1_1_16_376, 819
uid_1_2_840_10008_5_1_1_17, 819
uid_1_2_840_10008_5_1_1_17_376, 819
uid_1_2_840_10008_5_1_1_18, 819
uid_1_2_840_10008_5_1_1_18_1, 819
uid_1_2_840_10008_5_1_1_2, 819
uid_1_2_840_10008_5_1_1_22, 819
uid_1_2_840_10008_5_1_1_23, 819
uid_1_2_840_10008_5_1_1_24, 819
uid_1_2_840_10008_5_1_1_24_1, 819
uid_1_2_840_10008_5_1_1_25, 819

uid_1_2_840_10008_5_1_1_26, 819
uid_1_2_840_10008_5_1_1_27, 819
uid_1_2_840_10008_5_1_1_29, 819
uid_1_2_840_10008_5_1_1_30, 819
uid_1_2_840_10008_5_1_1_31, 819
uid_1_2_840_10008_5_1_1_32, 819
uid_1_2_840_10008_5_1_1_33, 819
uid_1_2_840_10008_5_1_1_4, 819
uid_1_2_840_10008_5_1_1_4_1, 819
uid_1_2_840_10008_5_1_1_4_2, 819
uid_1_2_840_10008_5_1_1_9, 819
uid_1_2_840_10008_5_1_1_9_1, 819
uid_1_2_840_10008_5_1_4_1_1_1, 819
uid_1_2_840_10008_5_1_4_1_1_10, 820
uid_1_2_840_10008_5_1_4_1_1_104_1, 821
uid_1_2_840_10008_5_1_4_1_1_104_2, 821
uid_1_2_840_10008_5_1_4_1_1_11, 820
uid_1_2_840_10008_5_1_4_1_1_11_1, 820
uid_1_2_840_10008_5_1_4_1_1_11_2, 820
uid_1_2_840_10008_5_1_4_1_1_11_3, 820
uid_1_2_840_10008_5_1_4_1_1_11_4, 820
uid_1_2_840_10008_5_1_4_1_1_128, 821
uid_1_2_840_10008_5_1_4_1_1_129, 821
uid_1_2_840_10008_5_1_4_1_1_12_1, 820
uid_1_2_840_10008_5_1_4_1_1_12_1_1, 820
uid_1_2_840_10008_5_1_4_1_1_12_2, 820
uid_1_2_840_10008_5_1_4_1_1_12_2_1, 820
uid_1_2_840_10008_5_1_4_1_1_12_3, 820
uid_1_2_840_10008_5_1_4_1_1_13_1_1, 820
uid_1_2_840_10008_5_1_4_1_1_13_1_2, 820
uid_1_2_840_10008_5_1_4_1_1_13_1_3, 823
uid_1_2_840_10008_5_1_4_1_1_1_1, 819
uid_1_2_840_10008_5_1_4_1_1_1_1_1, 819
uid_1_2_840_10008_5_1_4_1_1_1_2, 820
uid_1_2_840_10008_5_1_4_1_1_1_2_1, 820
uid_1_2_840_10008_5_1_4_1_1_1_3, 820
uid_1_2_840_10008_5_1_4_1_1_1_3_1, 820
uid_1_2_840_10008_5_1_4_1_1_2, 820
uid_1_2_840_10008_5_1_4_1_1_20, 820
uid_1_2_840_10008_5_1_4_1_1_2_1, 820
uid_1_2_840_10008_5_1_4_1_1_3, 820
uid_1_2_840_10008_5_1_4_1_1_3_1, 820
uid_1_2_840_10008_5_1_4_1_1_4, 820
uid_1_2_840_10008_5_1_4_1_1_481_1, 821
uid_1_2_840_10008_5_1_4_1_1_481_2, 821
uid_1_2_840_10008_5_1_4_1_1_481_3, 821
uid_1_2_840_10008_5_1_4_1_1_481_4, 821
uid_1_2_840_10008_5_1_4_1_1_481_5, 821
uid_1_2_840_10008_5_1_4_1_1_481_6, 821
uid_1_2_840_10008_5_1_4_1_1_481_7, 821
uid_1_2_840_10008_5_1_4_1_1_481_8, 822
uid_1_2_840_10008_5_1_4_1_1_481_9, 822
uid_1_2_840_10008_5_1_4_1_1_4_1, 820
uid_1_2_840_10008_5_1_4_1_1_4_2, 820
uid_1_2_840_10008_5_1_4_1_1_5, 820
uid_1_2_840_10008_5_1_4_1_1_6, 820
uid_1_2_840_10008_5_1_4_1_1_66, 821
uid_1_2_840_10008_5_1_4_1_1_66_1, 821
uid_1_2_840_10008_5_1_4_1_1_66_2, 821
uid_1_2_840_10008_5_1_4_1_1_66_3, 821
uid_1_2_840_10008_5_1_4_1_1_66_4, 821
uid_1_2_840_10008_5_1_4_1_1_66_5, 823
uid_1_2_840_10008_5_1_4_1_1_67, 821
uid_1_2_840_10008_5_1_4_1_1_6_1, 820
uid_1_2_840_10008_5_1_4_1_1_6_2, 823
uid_1_2_840_10008_5_1_4_1_1_7, 820
uid_1_2_840_10008_5_1_4_1_1_77_1, 821
uid_1_2_840_10008_5_1_4_1_1_77_1_1, 821
uid_1_2_840_10008_5_1_4_1_1_77_1_1_1, 821
uid_1_2_840_10008_5_1_4_1_1_77_1_2, 821
uid_1_2_840_10008_5_1_4_1_1_77_1_2_1, 821
uid_1_2_840_10008_5_1_4_1_1_77_1_3, 821
uid_1_2_840_10008_5_1_4_1_1_77_1_4, 821
uid_1_2_840_10008_5_1_4_1_1_77_1_4_1, 821
uid_1_2_840_10008_5_1_4_1_1_77_1_5_1, 821
uid_1_2_840_10008_5_1_4_1_1_77_1_5_2, 821
uid_1_2_840_10008_5_1_4_1_1_77_1_5_3, 821
uid_1_2_840_10008_5_1_4_1_1_77_1_5_4, 821
uid_1_2_840_10008_5_1_4_1_1_77_1_6, 823
uid_1_2_840_10008_5_1_4_1_1_77_2, 821
uid_1_2_840_10008_5_1_4_1_1_7_1, 820
uid_1_2_840_10008_5_1_4_1_1_7_2, 820
uid_1_2_840_10008_5_1_4_1_1_7_3, 820
uid_1_2_840_10008_5_1_4_1_1_7_4, 820
uid_1_2_840_10008_5_1_4_1_1_8, 820
uid_1_2_840_10008_5_1_4_1_1_88_1, 821
uid_1_2_840_10008_5_1_4_1_1_88_11, 821
uid_1_2_840_10008_5_1_4_1_1_88_2, 821
uid_1_2_840_10008_5_1_4_1_1_88_22, 821
uid_1_2_840_10008_5_1_4_1_1_88_3, 821
uid_1_2_840_10008_5_1_4_1_1_88_33, 821
uid_1_2_840_10008_5_1_4_1_1_88_4, 821
uid_1_2_840_10008_5_1_4_1_1_88_40, 821
uid_1_2_840_10008_5_1_4_1_1_88_50, 821
uid_1_2_840_10008_5_1_4_1_1_88_59, 821
uid_1_2_840_10008_5_1_4_1_1_88_65, 821
uid_1_2_840_10008_5_1_4_1_1_88_67, 821
uid_1_2_840_10008_5_1_4_1_1_9, 820
uid_1_2_840_10008_5_1_4_1_1_9_1, 820
uid_1_2_840_10008_5_1_4_1_1_9_1_1, 820
uid_1_2_840_10008_5_1_4_1_1_9_1_2, 820
uid_1_2_840_10008_5_1_4_1_1_9_1_3, 820
uid_1_2_840_10008_5_1_4_1_1_9_2_1, 820
uid_1_2_840_10008_5_1_4_1_1_9_3_1, 820
uid_1_2_840_10008_5_1_4_1_1_9_4_1, 820
uid_1_2_840_10008_5_1_4_1_2_1_1, 822
uid_1_2_840_10008_5_1_4_1_2_1_2, 822
uid_1_2_840_10008_5_1_4_1_2_1_3, 822

uid_1_2_840_10008_5_1_4_1_2_2_1, 822
uid_1_2_840_10008_5_1_4_1_2_2_2, 822
uid_1_2_840_10008_5_1_4_1_2_2_3, 822
uid_1_2_840_10008_5_1_4_1_2_3_1, 822
uid_1_2_840_10008_5_1_4_1_2_3_2, 822
uid_1_2_840_10008_5_1_4_1_2_3_3, 822
uid_1_2_840_10008_5_1_4_31, 822
uid_1_2_840_10008_5_1_4_32, 822
uid_1_2_840_10008_5_1_4_32_1, 822
uid_1_2_840_10008_5_1_4_32_2, 822
uid_1_2_840_10008_5_1_4_32_3, 822
uid_1_2_840_10008_5_1_4_33, 822
uid_1_2_840_10008_5_1_4_34_1, 822
uid_1_2_840_10008_5_1_4_34_2, 822
uid_1_2_840_10008_5_1_4_34_3, 822
uid_1_2_840_10008_5_1_4_34_4, 822
uid_1_2_840_10008_5_1_4_34_4_1, 822
uid_1_2_840_10008_5_1_4_34_4_2, 822
uid_1_2_840_10008_5_1_4_34_4_3, 822
uid_1_2_840_10008_5_1_4_34_4_4, 822
uid_1_2_840_10008_5_1_4_34_5, 822
uid_1_2_840_10008_5_1_4_37_1, 822
uid_1_2_840_10008_5_1_4_37_2, 822
uid_1_2_840_10008_5_1_4_37_3, 822
uid_1_2_840_10008_5_1_4_38_1, 822
uid_1_2_840_10008_5_1_4_38_2, 822
uid_1_2_840_10008_5_1_4_38_3, 822
uid_1_2_840_10008_5_1_4_41, 822
uid_1_2_840_10008_5_1_4_42, 822
UltrasoundImageStorage, 813
UltrasoundImageStorageRetired, 813
UltrasoundMultiframeImageStorage, 813
UltrasoundMultiframeImageStorageRetired, 813
UnifiedProcedureStepEventSOPClass, 815
UnifiedProcedureStepPullSOPClass, 815
UnifiedProcedureStepPushSOPClass, 815
UnifiedProcedureStepWatchSOPClass, 815
UnifiedWorklistandProcedureStepSOPInstance, 816
UnifiedWorklistandProcedureStepServiceClass, 815
VLEndoscopicImageStorage, 814
VLImageStorageTrialRetired, 814
VLMicroscopicImageStorage, 814
VLMultiframeImageStorageTrialRetired, 814
VLPhotographicImageStorage, 814
VLSlideCoordinatesMicroscopicImageStorage, 814
VLWholeSlideMicroscopyImageStorage, 817
VOILUTBoxSOPClass, 813
VerificationSOPClass, 810
VideoEndoscopicImageStorage, 814
VideoMicroscopicImageStorage, 814
VideoPhotographicImageStorage, 814
WaveformStorageTrialRetired, 813
XMLEncoding, 811
XRay3DAngiographicImageStorage, 814
XRay3DCraniofacialImageStorage, 814
XRayAngiographicBiPlaneImageStorageRetired, 814
XRayAngiographicImageStorage, 814
XRayRadiationDoseSRStorage, 815
XRayRadiofluoroscopicImageStorage, 814
gdcm::UNExplicitDataElement, 872
GetLength, 874
Read, 874
ReadPreValue, 874
ReadValue, 874
ReadWithLength, 874
gdcm::UNExplicitImplicitDataElement, 874
GetLength, 876
Read, 876
ReadPreValue, 876
ReadValue, 876
gdcm::UUIDGenerator, 882
Generate, 882
IsValid, 882
gdcm::UI, 802
Internal, 803
operator<<, 803
gdcm::Unpacker12Bits, 876
Pack, 877
Unpack, 877
gdcm::Usage, 877
Conditional, 878
GetUsageString, 878
GetUsageType, 878
Invalid, 878
Mandatory, 878
operator UsageType, 879
operator<<, 879
Usage, 878
UsageType, 878
UserOption, 878
gdcm::UserEvent, 879
gdcm::VMToLength< T >, 895
gdcm::VR16ExplicitDataElement, 900
GetLength, 902
Read, 902
ReadPreValue, 902
ReadValue, 902
ReadWithLength, 902
gdcm::VRToEncoding< T >, 902
gdcm::VRToType< T >, 902
gdcm::VRVLSize< 0 >, 903
Read, 903
Write, 903
gdcm::VRVLSize< 1 >, 904
Read, 904
Write, 904
gdcm::VRVLSize< T >, 903
gdcm::Validate, 883

~Validate, 884
F, 884
GetValidatedFile, 884
SetFile, 884
V, 884
Validate, 884
Validation, 884
gdcm::Value, 884
~Value, 885
Clear, 886
DataElement, 886
GetLength, 886
operator==, 886
SetLength, 886
SetLengthOnly, 886
Value, 885
gdcm::ValueIO< TDE, TSwap, TType >, 886
gdcm::ValueIO
Read, 887
Write, 887
gdcm::Version, 887
~Version, 888
GetBuildVersion, 888
GetMajorVersion, 888
GetMinorVersion, 888
GetVersion, 888
operator<<, 888
Print, 888
Version, 888
gdcm::VL, 888
GetLength, 890
GetVL16Max, 890
GetVL32Max, 890
IsOdd, 890
IsUndefined, 890
operator uint32_t, 890
operator<<, 891
operator++, 890
operator+=, 890
Read, 890
Read16, 890
SetToUndefined, 890
Type, 890
VL, 890
Write, 890
Write16, 890
gdcm::VM, 891
Compatible, 894
GetIndex, 894
GetLength, 895
GetNumberOfElementsFromArray, 895
GetVMString, 895
GetVMType, 895
GetVMTypeFromLength, 895
IsValid, 895
operator VMType, 895
operator<<, 895
VM0, 893
VM1, 893
VM10, 893
VM12, 893
VM16, 893
VM18, 893
VM1_2, 894
VM1_3, 894
VM1_32, 894
VM1_4, 894
VM1_5, 894
VM1_8, 894
VM1_99, 894
VM1_n, 894
VM2, 893
VM24, 894
VM256, 894
VM28, 894
VM2_2n, 894
VM2_n, 894
VM3, 893
VM30_30n, 894
VM32, 894
VM35, 894
VM3_3n, 894
VM3_4, 894
VM3_n, 894
VM4, 893
VM47_47n, 894
VM4_4n, 894
VM5, 893
VM6, 893
VM6_6n, 894
VM7_7n, 894
VM8, 893
VM9, 893
VM99, 894
VM_END, 894
VMType, 893
VM, 894
gdcm::VR, 896
AE, 898
AS, 898
AT, 898
CanDisplay, 899
Compatible, 899
CS, 898
DA, 898
DS, 898
DT, 898
FD, 898

FL, 898
GetLength, 899
GetSize, 899
GetSizeof, 899
GetVRString, 899
GetVRStringFromFile, 899
GetVRType, 899
GetVRTypeFromFile, 899
INVALID, 898
IS, 898
IsASCII2, 899
IsASCII, 899
IsBinary, 899
IsBinary2, 899
IsDual, 899
IsSwap, 899
IsVRFile, 899
IsValid, 899
LO, 898
LT, 898
OB_OW, 898
OB, 898
OD, 898
OF, 898
operator VRType, 899
operator<<, 900
OW, 898
PN, 898
Read, 900
SH, 898
SL, 898
SQ, 898
SS, 898
ST, 898
TM, 898
US_SS_OW, 898
US_SS, 898
UI, 898
UL, 898
UN, 898
US, 898
UT, 898
VL16, 898
VL32, 898
VR_END, 898
VR_VM1, 898
VRALL, 898
VRASCII, 898
VRBINARY, 898
VRType, 898
VR, 899
Write, 900
gdcm::WLMFindQuery, 966
GetAbstractSyntaxUID, 967
GetTagListByLevel, 967
GetValidDataSet, 968
InitializeDataSet, 968
QueryFactory, 968
ValidateQuery, 968
WLMFindQuery, 967
gdcm::Waveform, 965
Waveform, 966
gdcm::Writer, 969
~Writer, 972
CheckFileMetaInformationOff, 972
CheckFileMetaInformationOn, 972
GetFile, 972
GetStreamPtr, 972
Ofstream, 973
SetCheckFileMetaInformation, 972
SetFile, 972
SetFileName, 972
SetStream, 973
SetWriteDataSetOnly, 973
Stream, 973
StreamImageWriter, 973
Write, 973
Writer, 972
gdcm::XMLDictReader, 974
~XMLDictReader, 975
CharacterDataHandler, 975
EndElement, 975
GetDict, 975
HandleDescription, 975
HandleEntry, 975
StartElement, 975
XMLDictReader, 975
gdcm::XMLPrinter, 976
~XMLPrinter, 977
F, 977
GetPrintStyle, 977
HandleBulkData, 977
LOADBULKDATA, 977
OnlyUUID, 977
Print, 977
PrintDataElement, 977
PrintDataSet, 977
PrintSQ, 977
PrintStyle, 977
PrintStyles, 977
SetFile, 977
SetStyle, 977
XMLPrinter, 977
gdcm::XMLPrivateDictReader, 978
~XMLPrivateDictReader, 979
CharacterDataHandler, 979
EndElement, 979
GetPrivateDict, 979

HandleDescription, 979
HandleEntry, 979
StartElement, 979
XMLPrivateDictReader, 979
gdcm::ignore_char, 380
ignore_char, 380
m_char, 380
gdcm::network, 69
cMaxEventID, 75
cMaxStateID, 75
eAABORTPDUReceivedOpen, 74
eAABORTRequest, 74
eAASSOCIATE_RQPDUreceived, 74
eAASSOCIATERequestLocalUser, 74
eAASSOCIATEresponseAccept, 74
eAASSOCIATEresponseReject, 74
eARELEASE_RPPDUReceived, 74
eARELEASE_RQPDUReceivedOpen, 74
eARELEASERequest, 74
eARELEASEResponse, 74
eARTIMTimerExpired, 74
eASSOCIATE_ACPDUreceived, 74
eASSOCIATE_RJPDUreceived, 74
eEventDoesNotExist, 74
EEventID, 74
ePDATATFPDU, 74
ePDATArequest, 74
eSta10ReleaseCollisionAc, 75
eSta11ReleaseCollisionRq, 75
eSta12ReleaseCollisionAcLocal, 75
eSta13AwaitingClose, 75
eSta1Idle, 75
eSta2Open, 75
eSta3WaitLocalAssoc, 75
eSta4LocalAssocDone, 75
eSta5WaitRemoteAssoc, 75
eSta6TransferReady, 75
eSta7WaitRelease, 75
eSta8WaitLocalRelease, 75
eSta9ReleaseCollisionRqLocal, 75
eStaDoesNotExist, 75
EStateID, 74
eTransportConnConfirmLocal, 74
eTransportConnIndicLocal, 74
eTransportConnectionClosed, 74
eUnrecognizedPDUReceived, 74
GetStateIndex, 75
gdcm::network::AAbortPDU, 79
AAbortPDU, 80
IsLastFragment, 80
Print, 80
Read, 80
SetReason, 80
SetSource, 80
Size, 80
Write, 80
gdcm::network::AAssociateACPDU, 81
AAssociateACPDU, 82
AAssociateRQPDU, 83
AddPresentationContextAC, 82
GetNumberOfPresentationContextAC, 82
GetPresentationContextAC, 82
GetUserInformation, 82
InitFromRQ, 82
IsLastFragment, 83
Print, 83
Read, 83
SetCalledAETitle, 83
SetCallingAETitle, 83
Size, 83
SizeType, 82
Write, 83
gdcm::network::AAssociateRJPDU, 84
AAssociateRJPDU, 85
IsLastFragment, 85
Print, 85
Read, 85
Size, 85
Write, 85
gdcm::network::AAssociateRQPDU, 86
AAssociateACPDU, 90
AAssociateRQPDU, 88
AddPresentationContext, 88
GetCalledAETitle, 88
GetCallingAETitle, 88
GetNumberOfPresentationContext, 88
GetPresentationContext, 88
GetPresentationContextByAbstractSyntax, 88
GetPresentationContextByID, 88
GetPresentationContexts, 88
GetReserved43_74, 89
GetUserInformation, 89
IsAETitleValid, 89
IsLastFragment, 89
PresentationContextArrayType, 88
Print, 89
Read, 89
SetCalledAETitle, 89
SetCallingAETitle, 89
SetUserInformation, 90
Size, 90
SizeType, 88
Write, 90
gdcm::network::ARTIMTimer, 109
ARTIMTimer, 110
GetElapsedTime, 110
GetHasExpired, 110
GetTimeout, 110

SetTimeout, 110
Start, 110
Stop, 110
gdcm::network::AReleaseRPPDU, 106
AReleaseRPPDU, 107
IsLastFragment, 107
Print, 107
Read, 107
Size, 107
Write, 107
gdcm::network::AReleaseRQPDU, 108
AReleaseRQPDU, 109
IsLastFragment, 109
Print, 109
Read, 109
Size, 109
Write, 109
gdcm::network::AbstractSyntax, 91
AbstractSyntax, 92
GetAsDataElement, 92
GetName, 92
operator==, 92
Print, 92
Read, 92
SetName, 92
SetNameFromUID, 92
Size, 92
Write, 92
gdcm::network::ApplicationContext, 102
ApplicationContext, 103
GetName, 103
Print, 103
Read, 103
SetName, 103
Size, 103
Write, 103
gdcm::network::AsynchronousOperationsWindowSub, 112
AsynchronousOperationsWindowSub, 112
Print, 112
Read, 112
Size, 112
Write, 112
gdcm::network::BaseCompositeMessage, 139
~BaseCompositeMessage, 140
ConstructPDV, 140
gdcm::network::BaseNormalizedMessage, 140
~BaseNormalizedMessage, 142
ConstructPDV, 142
gdcm::network::BasePDU, 143
~BasePDU, 144
IsLastFragment, 144
Print, 144
Read, 144
Size, 144
Write, 145
gdcm::network::CEchoRSP, 184
ConstructPDVByDataSet, 185
gdcm::network::CEchoRQ, 182
AffectedSOPClassUID, 183
ConstructPDV, 183
MessageID, 183
gdcm::network::CFind, 185
gdcm::network::CFindCancelRQ, 185
ConstructPDVByDataSet, 186
gdcm::network::CFindRSP, 188
ConstructPDVByDataSet, 189
gdcm::network::CFindRQ, 186
ConstructPDV, 188
gdcm::network::CMoveCancelRq, 189
ConstructPDVByDataSet, 190
gdcm::network::CMoveRSP, 192
ConstructPDVByDataSet, 193
gdcm::network::CMoveRQ, 190
ConstructPDV, 191
gdcm::network::CStoreRSP, 229
ConstructPDV, 230
gdcm::network::CStoreRQ, 227
ConstructPDV, 228
gdcm::network::CompositeMessageFactory, 202
ConstructCEchoRQ, 203
ConstructCFindRQ, 203
ConstructCMoveRQ, 203
ConstructCStoreRSP, 203
ConstructCStoreRQ, 203
gdcm::network::DIMSE, 279
C_CANCEL_RQ, 281
C_ECHO_RSP, 280
C_ECHO_RQ, 280
C_FIND_RSP, 280
C_FIND_RQ, 280
C_GET_RSP, 280
C_GET_RQ, 280
C_MOVE_RSP, 280
C_MOVE_RQ, 280
C_STORE_RSP, 280
C_STORE_RQ, 280
CommandTypes, 280
N_ACTION_RSP, 281
N_ACTION_RQ, 281
N_CREATE_RSP, 281
N_CREATE_RQ, 281
N_DELETE_RSP, 281
N_DELETE_RQ, 281
N_EVENT_REPORT_RSP, 281
N_EVENT_REPORT_RQ, 280
N_GET_RSP, 281
N_GET_RQ, 281

N_SET_RSP, 281
N_SET_RQ, 281
gdcm::network::ImplementationClassUIDSub, 424
ImplementationClassUIDSub, 424
Print, 424
Read, 424
Size, 424
Write, 424
gdcm::network::ImplementationUIDSub, 425
ImplementationUIDSub, 425
Write, 425
gdcm::network::ImplementationVersionNameSub, 425
ImplementationVersionNameSub, 426
Print, 426
Read, 426
Size, 426
Write, 426
gdcm::network::MaximumLengthSub, 480
GetMaximumLength, 481
MaximumLengthSub, 481
Print, 481
Read, 481
SetMaximumLength, 481
Size, 481
Write, 481
gdcm::network::NActionRSP, 518
ConstructPDVByDataSet, 519
gdcm::network::NActionRQ, 517
ConstructPDV, 518
gdcm::network::NCreateRSP, 521
ConstructPDVByDataSet, 522
gdcm::network::NCreateRQ, 519
ConstructPDV, 521
gdcm::network::NDeleteRSP, 524
ConstructPDVByDataSet, 525
gdcm::network::NDeleteRQ, 522
ConstructPDV, 523
gdcm::network::NEventReportRSP, 529
ConstructPDVByDataSet, 530
gdcm::network::NEventReportRQ, 528
ConstructPDV, 529
gdcm::network::NGetRSP, 532
ConstructPDVByDataSet, 533
gdcm::network::NGetRQ, 530
ConstructPDV, 531
gdcm::network::NSetRSP, 538
ConstructPDVByDataSet, 539
gdcm::network::NSetRQ, 536
ConstructPDV, 537
gdcm::network::NormalizedMessageFactory, 534
ConstructNAction, 534
ConstructNCreate, 534
ConstructNDelete, 534
ConstructNEventReport, 534
ConstructNGet, 534
ConstructNSet, 534
gdcm::network::PDUFactory, 572
ConstructAbortPDU, 573
ConstructPDU, 573
ConstructReleasePDU, 573
CreateCEchoPDU, 573
CreateCFindPDU, 573
CreateCMovePDU, 573
CreateCStoreRQPDU, 573
CreateCStoreRSPPDU, 573
CreateNActionPDU, 573
CreateNCreatePDU, 573
CreateNDeletePDU, 573
CreateNEventReportPDU, 573
CreateNGetPDU, 573
CreateNSetPDU, 573
DetermineEventByPDU, 573
GetPDVs, 573
gdcm::network::PDataTFPDU, 563
AddPresentationDataValue, 564
GetNumberOfPresentationDataValues, 564
GetPresentationDataValue, 564
IsLastFragment, 564
PDataTFPDU, 564
Print, 564
Read, 565
ReadInto, 565
Size, 565
SizeType, 564
Write, 565
gdcm::network::PresentationContextAC, 604
GetPresentationContextID, 605
GetReason, 605
GetTransferSyntax, 605
PresentationContextAC, 605
Print, 605
Read, 605
SetPresentationContextID, 605
SetReason, 605
SetTransferSyntax, 605
Size, 605
Write, 605
gdcm::network::PresentationContextRQ, 608
AddTransferSyntax, 609
GetAbstractSyntax, 609
GetNumberOfTransferSyntaxes, 609
GetPresentationContextID, 609
GetTransferSyntax, 609
GetTransferSyntaxes, 609
operator==, 610
PresentationContextRQ, 609
Print, 610
Read, 610

SetAbstractSyntax, 610
SetPresentationContextID, 610
Size, 610
SizeType, 609
Write, 610
gdcm::network::PresentationDataValue, 610
ConcatenatePDVBlobs, 611
ConcatenatePDVBlobsAsExplicit, 611
GetBlob, 611
GetIsCommand, 611
GetIsLastFragment, 611
GetMessageHeader, 611
GetPresentationContextID, 611
PresentationDataValue, 611
Print, 611
Read, 611
ReadInto, 611
SetBlob, 611
SetCommand, 611
SetDataSet, 611
SetLastFragment, 611
SetMessageHeader, 612
SetPresentationContextID, 612
Size, 612
Write, 612
gdcm::network::RoleSelectionSub, 657
Print, 658
Read, 658
RoleSelectionSub, 658
SetTuple, 658
Size, 658
Write, 658
gdcm::network::SOPClassExtendedNegociationSub, 709
Print, 710
Read, 710
SOPClassExtendedNegociationSub, 710
SetTuple, 710
Size, 710
Write, 710
gdcm::network::ServiceClassApplicationInformation, 693
Print, 694
Read, 694
ServiceClassApplicationInformation, 694
SetTuple, 694
Size, 694
Write, 694
gdcm::network::TableRow, 773
∼TableRow, 773
TableRow, 773
transitions, 774
gdcm::network::TransferSyntaxSub, 797
GetName, 798
operator==, 798
Print, 798
Read, 798
SetName, 798
SetNameFromUID, 798
Size, 798
TransferSyntaxSub, 798
Write, 798
gdcm::network::Transition, 799
∼Transition, 799
mAction, 800
mEnd, 800
MakeNew, 800
Transition, 799, 800
gdcm::network::ULAction, 825
∼ULAction, 827
PerformAction, 828
ULAction, 827
gdcm::network::ULActionAA1, 828
PerformAction, 829
gdcm::network::ULActionAA2, 829
PerformAction, 830
gdcm::network::ULActionAA3, 830
PerformAction, 831
gdcm::network::ULActionAA4, 831
PerformAction, 832
gdcm::network::ULActionAA5, 832
PerformAction, 833
gdcm::network::ULActionAA6, 833
PerformAction, 834
gdcm::network::ULActionAA7, 834
PerformAction, 835
gdcm::network::ULActionAA8, 835
PerformAction, 836
gdcm::network::ULActionAE1, 836
PerformAction, 837
gdcm::network::ULActionAE2, 837
PerformAction, 838
gdcm::network::ULActionAE3, 838
PerformAction, 839
gdcm::network::ULActionAE4, 839
PerformAction, 840
gdcm::network::ULActionAE5, 840
PerformAction, 841
gdcm::network::ULActionAE6, 841
PerformAction, 842
gdcm::network::ULActionAE7, 842
PerformAction, 843
gdcm::network::ULActionAE8, 843
PerformAction, 844
gdcm::network::ULActionAR1, 844
PerformAction, 845
gdcm::network::ULActionAR10, 845
PerformAction, 846
gdcm::network::ULActionAR2, 846
PerformAction, 847

gdcm::network::ULActionAR3, 847
PerformAction, 848
gdcm::network::ULActionAR4, 848
PerformAction, 849
gdcm::network::ULActionAR5, 849
PerformAction, 850
gdcm::network::ULActionAR6, 850
PerformAction, 851
gdcm::network::ULActionAR7, 851
PerformAction, 852
gdcm::network::ULActionAR8, 852
PerformAction, 853
gdcm::network::ULActionAR9, 853
PerformAction, 854
gdcm::network::ULActionDT1, 854
PerformAction, 855
gdcm::network::ULActionDT2, 855
PerformAction, 856
gdcm::network::ULBasicCallback, 856
~ULBasicCallback, 858
GetDataSets, 858
GetResponses, 858
HandleDataSet, 858
HandleResponse, 858
ULBasicCallback, 858
gdcm::network::ULConnection, 858
~ULConnection, 860
AddAcceptedPresentationContext, 860
FindContext, 860
GetAcceptedPresentationContexts, 860
GetConnectionInfo, 860
GetMaxPDUSize, 860
GetPresentationContextACByID, 860
GetPresentationContextIDFromPresentationContext, 860
GetPresentationContextRQByID, 860
GetPresentationContexts, 860
GetProtocol, 860
GetState, 860
GetTimer, 860
InitializeConnection, 860
InitializeIncomingConnection, 860
SetMaxPDUSize, 861
SetPresentationContexts, 861
SetState, 861
StopProtocol, 861
ULActionAE6, 861
ULConnection, 860
ULConnectionManager, 861
gdcm::network::ULConnectionCallback, 861
~ULConnectionCallback, 862
DataSetHandled, 862
DataSetHandles, 862
HandleDataSet, 862
HandleResponse, 862
mImplicit, 863
ResetHandledDataSet, 863
SetImplicitFlag, 863
ULConnectionCallback, 862
gdcm::network::ULConnectionInfo, 863
GetCalledAETitle, 864
GetCalledComputerName, 864
GetCalledIPAddress, 864
GetCalledIPPort, 864
GetCallingAETitle, 864
GetMaxPDULength, 864
Initialize, 864
SetMaxPDULength, 864
ULConnectionInfo, 864
gdcm::network::ULConnectionManager, 864
~ULConnectionManager, 867
BreakConnection, 867
BreakConnectionNow, 867
EstablishConnection, 867
EstablishConnectionMove, 867
mConnection, 868
mSecondaryConnection, 868
mTransitions, 868
RunEventLoop, 867
RunMoveEventLoop, 867
SendEcho, 867
SendFind, 867
SendMove, 867
SendNAction, 867, 868
SendNCreate, 868
SendNDelete, 868
SendNEventReport, 868
SendNGet, 868
SendNSet, 868
SendStore, 868
ULConnectionManager, 867
gdcm::network::ULEvent, 869
~ULEvent, 869
GetDataSetPos, 869
GetEvent, 869
GetIStream, 869
GetPDUs, 869
SetEvent, 869
SetPDU, 869
ULEvent, 869
gdcm::network::ULTransitionTable, 870
HandleEvent, 870
PrintTable, 870
ULTransitionTable, 870
gdcm::network::ULWritingCallback, 871
~ULWritingCallback, 872
HandleDataSet, 872
HandleResponse, 872

SetDirectory, 872
ULWritingCallback, 872
gdcm::network::UserInformation, 880
~UserInformation, 881
AddRoleSelectionSub, 881
AddSOPClassExtendedNegociationSub, 881
GetMaximumLengthSub, 881
operator=, 881
Print, 881
Read, 882
Size, 882
UserInformation, 881
Write, 882
gdcm::static_assert_test< x >, 720
gdcm::terminal, 76
Attribute, 77
black, 77
blink, 77
blue, 77
bright, 77
CONSOLE, 77
Color, 77
cyan, 77
dim, 77
green, 77
hidden, 77
magenta, 77
Mode, 77
red, 77
reset, 77
reverse, 77
setattribute, 77
setbgcolor, 77
setfgcolor, 77
setmode, 77
underline, 77
VT100, 77
white, 77
yellow, 77
gdcmAAbortPDU.h, 981
gdcmAAssociateACPDU.h, 982
gdcmAAssociateRJPDU.h, 982
gdcmAAssociateRQPDU.h, 983
gdcmARTIMTimer.h, 990
gdcmAReleaseRPPDU.h, 988
gdcmAReleaseRQPDU.h, 989
gdcmASN1.h, 991
gdcmAbstractSyntax.h, 984
gdcmAnonymizeEvent.h, 985
gdcmAnonymizer.h, 986
gdcmApplicationContext.h, 987
gdcmApplicationEntity.h, 988
gdcmAssertAlwaysMacro
gdcmTrace.h, 1217
gdcmAssertMacro
gdcmTrace.h, 1217
gdcmAsynchronousOperationsWindowSub.h, 992
gdcmAttribute.h, 992
gdcmAudioCodec.h, 994
gdcmBase64.h, 994
gdcmBaseCompositeMessage.h, 995
gdcmBaseNormalizedMessage.h, 996
gdcmBasePDU.h, 997
gdcmBaseQuery.h, 998
gdcmBaseRootQuery.h, 999
gdcmBasicOffsetTable.h, 1000
gdcmBitmap.h, 1001
gdcmBitmapToBitmapFilter.h, 1002
gdcmBoxRegion.h, 1003
gdcmByteBuffer.h, 1003
gdcmByteSwap.h, 1005
gdcmByteSwapFilter.h, 1005
gdcmByteValue.h, 1006
gdcmCAPICryptoFactory.h, 1007
gdcmCAPICryptographicMessageSyntax.h, 1008
gdcmCEchoMessages.h, 1008
gdcmCFindMessages.h, 1009
gdcmCMoveMessages.h, 1010
gdcmCP246ExplicitDataElement.h, 1019
gdcmCSAElement.h, 1021
gdcmCSAHeader.h, 1022
gdcmCSAHeaderDict.h, 1023
gdcmCSAHeaderDictEntry.h, 1024
gdcmCStoreMessages.h, 1026
gdcmCodeString.h, 1014
gdcmCodec.h, 1011
gdcmCoder.h, 1012
gdcmCommand.h, 1015
gdcmCommandDataSet.h, 1016
gdcmCompositeMessageFactory.h, 1017
gdcmCompositeNetworkFunctions.h, 1017
gdcmConstCharWrapper.h, 1018
gdcmCryptoFactory.h, 1019
gdcmCryptographicMessageSyntax.h, 1020
gdcmCurve.h, 1027
gdcmDICOMDIR.h, 1036
gdcmDICOMDIRGenerator.h, 1037
gdcmDIMSE.h, 1043
gdcmDataElement.h, 1028
gdcmDataEvent.h, 1029
gdcmDataSet.h, 1030
gdcmDataSetEvent.h, 1031
gdcmDataSetHelper.h, 1031
gdcmDebugMacro
gdcmTrace.h, 1218
gdcmDecoder.h, 1032
gdcmDefinedTerms.h, 1034
gdcmDeflateStream.h, 1034

gdcmDefs.h, 1035
gdcmDeltaEncodingCodec.h, 1036
gdcmDict.h, 1038
gdcmDictConverter.h, 1039
gdcmDictEntry.h, 1040
gdcmDictPrinter.h, 1041
gdcmDicts.h, 1042
gdcmDirectionCosines.h, 1043
gdcmDirectory.h, 1044
gdcmDirectoryHelper.h, 1045
gdcmDummyValueGenerator.h, 1046
gdcmDumper.h, 1047
gdcmElement.h, 1047
    VRDS16ILLEGAL, 1049
gdcmEncapsulatedDocument.h, 1049
gdcmEnumeratedValues.h, 1050
gdcmErrorMacro
    gdcmTrace.h, 1218
gdcmEvent.h, 1050
    gdcmEventMacro, 1052
gdcmEventMacro
    gdcmEvent.h, 1052
gdcmException.h, 1052
gdcmExplicitDataElement.h, 1053
gdcmExplicitImplicitDataElement.h, 1054
gdcmFiducials.h, 1054
gdcmFile.h, 1055
gdcmFileAnonymizer.h, 1056
gdcmFileChangeTransferSyntax.h, 1057
gdcmFileDecompressLookupTable.h, 1057
gdcmFileDerivation.h, 1058
gdcmFileExplicitFilter.h, 1059
gdcmFileMetaInformation.h, 1060
gdcmFileNameEvent.h, 1061
gdcmFileSet.h, 1063
gdcmFileStreamer.h, 1064
gdcmFilename.h, 1061
gdcmFilenameGenerator.h, 1062
gdcmFindPatientRootQuery.h, 1065
gdcmFindStudyRootQuery.h, 1066
gdcmFragment.h, 1066
gdcmGlobal.h, 1068
gdcmGroupDict.h, 1069
gdcmIOD.h, 1087
gdcmIODEntry.h, 1089
gdcmIODs.h, 1091
gdcmIPPSorter.h, 1092
gdcmIconImage.h, 1069
gdcmIconImageFilter.h, 1070
gdcmIconImageGenerator.h, 1071
gdcmImage.h, 1072
gdcmImageApplyLookupTable.h, 1073
gdcmImageChangePhotometricInterpretation.h, 1074
gdcmImageChangePlanarConfiguration.h, 1074
gdcmImageChangeTransferSyntax.h, 1075
gdcmImageCodec.h, 1076
gdcmImageConverter.h, 1077
gdcmImageFragmentSplitter.h, 1078
gdcmImageHelper.h, 1079
gdcmImageReader.h, 1080
gdcmImageRegionReader.h, 1081
gdcmImageToImageFilter.h, 1081
gdcmImageWriter.h, 1082
gdcmImplementationClassUIDSub.h, 1083
gdcmImplementationUIDSub.h, 1084
gdcmImplementationVersionNameSub.h, 1085
gdcmImplicitDataElement.h, 1086
gdcmItem.h, 1093
gdcmJPEG12Codec.h, 1094
gdcmJPEG16Codec.h, 1095
gdcmJPEG2000Codec.h, 1095
gdcmJPEG8Codec.h, 1096
gdcmJPEGCodec.h, 1097
gdcmJPEGLSCodec.h, 1099
gdcmJSON.h, 1099
gdcmKAKADUCodec.h, 1100
gdcmLO.h, 1102
gdcmLegacyMacro.h, 1101
    GDCM_LEGACY_BODY, 1102
    GDCM_LEGACY_REPLACED_BODY, 1102
    GDCM_LEGACY, 1102
gdcmLookupTable.h, 1103
gdcmMD5.h, 1110
gdcmMacro.h, 1104
gdcmMacroEntry.h, 1106
    GDCMMACROENTRY_H, 1107
gdcmMacros.h, 1107
gdcmMaximumLengthSub.h, 1109
gdcmMediaStorage.h, 1111
gdcmMeshPrimitive.h, 1112
gdcmModalityPerformedProcedureStepCreateQuery.h, 1113
gdcmModalityPerformedProcedureStepSetQuery.h, 1114
gdcmModule.h, 1114
gdcmModuleEntry.h, 1116
gdcmModules.h, 1118
gdcmMovePatientRootQuery.h, 1119
gdcmMoveStudyRootQuery.h, 1120
gdcmNActionMessages.h, 1120
gdcmNCreateMessages.h, 1121
gdcmNDeleteMessages.h, 1122
gdcmNEventReportMessages.h, 1126
gdcmNGetMessages.h, 1126
gdcmNSetMessages.h, 1129
gdcmNestedModuleEntries.h, 1122
gdcmNetworkEvents.h, 1124
gdcmNetworkStateID.h, 1125
gdcmNormalizedMessageFactory.h, 1127

gdcmNormalizedNetworkFunctions.h, 1128
gdcmObject.h, 1129
gdcmOpenSSLCryptoFactory.h, 1131
gdcmOpenSSLCryptographicMessageSyntax.h, 1131
gdcmOpenSSLP7CryptoFactory.h, 1133
gdcmOpenSSLP7CryptographicMessageSyntax.h, 1133
gdcmOrientation.h, 1135
gdcmOverlay.h, 1135
gdcmPDBElement.h, 1140
gdcmPDBHeader.h, 1141
gdcmPDFCodec.h, 1142
gdcmPDUFactory.h, 1143
gdcmPDataTFPDU.h, 1139
gdcmPGXCodec.h, 1144
gdcmPNMCodec.h, 1151
gdcmPVRGCodec.h, 1161
gdcmParseException.h, 1136
gdcmParser.h, 1138
gdcmPatient.h, 1138
gdcmPersonName.h, 1143
gdcmPhotometricInterpretation.h, 1145
gdcmPixelFormat.h, 1146
gdcmPixmap.h, 1147
gdcmPixmapReader.h, 1148
gdcmPixmapToPixmapFilter.h, 1149
gdcmPixmapWriter.h, 1150
gdcmPreamble.h, 1151
gdcmPresentationContext.h, 1153
gdcmPresentationContextAC.h, 1154
gdcmPresentationContextGenerator.h, 1155
gdcmPresentationContextRQ.h, 1155
gdcmPresentationDataValue.h, 1156
gdcmPrinter.h, 1157
gdcmPrivateTag.h, 1159
gdcmProgressEvent.h, 1160
gdcmPythonFilter.h, 1161
gdcmQueryBase.h, 1162
gdcmQueryFactory.h, 1163
gdcmQueryImage.h, 1164
gdcmQueryPatient.h, 1165
gdcmQuerySeries.h, 1166
gdcmQueryStudy.h, 1167
gdcmRAWCodec.h, 1168
gdcmRLECodec.h, 1172
gdcmReader.h, 1168
gdcmRegion.h, 1170
gdcmRescaler.h, 1171
gdcmRoleSelectionSub.h, 1172
gdcmSHA1.h, 1185
gdcmSOPClassExtendedNegociationSub.h, 1188
gdcmSOPClassUIDToIOD.h, 1189
gdcmScanner.h, 1173
gdcmSegment.h, 1174
gdcmSegmentHelper.h, 1176
gdcmSegmentReader.h, 1177
gdcmSegmentWriter.h, 1179
gdcmSegmentedPaletteColorLookupTable.h, 1175
gdcmSequenceOfFragments.h, 1180
gdcmSequenceOfItems.h, 1180
gdcmSerieHelper.h, 1181
gdcmSeries.h, 1183
gdcmServiceClassApplicationInformation.h, 1184
gdcmServiceClassUser.h, 1185
gdcmSimpleSubjectWatcher.h, 1186
gdcmSmartPointer.h, 1187
gdcmSorter.h, 1190
gdcmSpacing.h, 1192
gdcmSpectroscopy.h, 1192
gdcmSplitMosaicFilter.h, 1193
gdcmStaticAssert.h, 1193
    GDCM_DO_JOIN2, 1194
    GDCM_DO_JOIN, 1194
    GDCM_JOIN, 1194
    GDCM_STATIC_ASSERT, 1194
gdcmStreamImageReader.h, 1195
gdcmStreamImageWriter.h, 1195
gdcmStrictScanner.h, 1196
gdcmString.h, 1197
gdcmStringFilter.h, 1199
gdcmStudy.h, 1199
gdcmSubject.h, 1201
gdcmSurface.h, 1202
gdcmSurfaceHelper.h, 1203
gdcmSurfaceReader.h, 1203
gdcmSurfaceWriter.h, 1204
gdcmSwapCode.h, 1205
gdcmSwapper.h, 1206
gdcmSystem.h, 1207
gdcmTable.h, 1208
gdcmTableEntry.h, 1209
gdcmTableReader.h, 1210
gdcmTag.h, 1211
gdcmTagPath.h, 1212
gdcmTagToVR.h, 1212
gdcmTerminal.h, 1213
gdcmTestDriver.h, 1214
gdcmTesting.h, 1215
gdcmTrace.h, 1216
    GDCM_FUNCTION, 1217
    gdcmAssertAlwaysMacro, 1217
    gdcmAssertMacro, 1217
    gdcmDebugMacro, 1218
    gdcmErrorMacro, 1218
    gdcmWarningMacro, 1219
gdcmTransferSyntax.h, 1219
gdcmTransferSyntaxSub.h, 1221
gdcmType.h, 1222
gdcmTypes.h, 1223

gdcmUIDGenerator.h, 1223
gdcmUIDs.h, 1224
gdcmULAction.h, 1225
gdcmULActionAA.h, 1226
gdcmULActionAE.h, 1226
gdcmULActionAR.h, 1227
gdcmULActionDT.h, 1228
gdcmULBasicCallback.h, 1229
gdcmULConnection.h, 1229
gdcmULConnectionCallback.h, 1230
gdcmULConnectionInfo.h, 1231
gdcmULConnectionManager.h, 1233
gdcmULEvent.h, 1233
gdcmULTransitionTable.h, 1235
gdcmULWritingCallback.h, 1236
gdcmUNExplicitDataElement.h, 1236
gdcmUNExplicitImplicitDataElement.h, 1237
gdcmUUIDGenerator.h, 1241
gdcmUnpacker12Bits.h, 1238
gdcmUsage.h, 1238
gdcmUserInformation.h, 1240
gdcmVL.h, 1244
gdcmVM.h, 1245
  TYPETOLENGTH, 1246
gdcmVR.h, 1246
  TYPETOENCODING, 1248
  VRTypeTemplateCase, 1248
gdcmVR16ExplicitDataElement.h, 1249
gdcmValidate.h, 1241
gdcmValue.h, 1242
gdcmValueIO.h, 1243
gdcmVersion.h, 1243
gdcmWLMFindQuery.h, 1250
gdcmWarningMacro
  gdcmTrace.h, 1219
gdcmWaveform.h, 1249
gdcmWin32.h, 1250
  GDCM_EXPORT, 1250
gdcmWriter.h, 1251
gdcmXMLDictReader.h, 1252
gdcmXMLPrinter.h, 1253
gdcmXMLPrivateDictReader.h, 1253
GeneralECGWaveformStorage
  gdcm::MediaStorage, 487
  gdcm::UIDs, 813
GeneralElectricMagneticResonanceImageStorage
  gdcm::MediaStorage, 487
GeneralPurposePerformedProcedureStepSOPClass
  gdcm::UIDs, 815
GeneralPurposeScheduledProcedureStepSOPClass
  gdcm::UIDs, 815
GeneralPurposeWorklistInformationModelFIND
  gdcm::UIDs, 815
GeneralPurposeWorklistManagementMetaSOPClass
  gdcm::UIDs, 815
GeneralRelevantPatientInformationQuery
  gdcm::UIDs, 816
Generate
  gdcm::DICOMDIRGenerator, 266
  gdcm::DummyValueGenerator, 288
  gdcm::FilenameGenerator, 354
  gdcm::IconImageGenerator, 378
  gdcm::UIDGenerator, 804
  gdcm::UUIDGenerator, 882
GenerateFromFilenames
  gdcm::PresentationContextGenerator, 607
GenerateFromUID
  gdcm::PresentationContextGenerator, 607
GenerateUUID
  gdcm::UIDGenerator, 804
Get
  gdcm::ByteBuffer, 171
GetAETitle
  gdcm::ServiceClassUser, 697
GetALGOType
  gdcm::Segment, 668
GetALGOTypeString
  gdcm::Segment, 668
GetAbbreviation
  gdcm::GroupDict, 374
GetAbstractSyntax
  gdcm::PresentationContext, 603
  gdcm::network::PresentationContextRQ, 609
GetAbstractSyntaxUID
  gdcm::BaseQuery, 147
  gdcm::FindPatientRootQuery, 364
  gdcm::FindStudyRootQuery, 366
  gdcm::ModalityPerformedProcedureStepCreate↩
    Query, 501
  gdcm::ModalityPerformedProcedureStepSetQuery,
    503
  gdcm::MovePatientRootQuery, 513
  gdcm::MoveStudyRootQuery, 516
  gdcm::WLMFindQuery, 967
GetAcceptedPresentationContexts
  gdcm::network::ULConnection, 860
GetAlgorithmFamily
  gdcm::Surface, 749
GetAlgorithmName
  gdcm::Surface, 749
GetAlgorithmVersion
  gdcm::Surface, 749
GetAllFilenamesFromTagToValue
  gdcm::Scanner, 663
  gdcm::StrictScanner, 733
GetAllRequiredTags
  gdcm::QueryBase, 627
GetAllTags

gdcm::QueryBase, 627
GetAnatomicRegion
gdcm::Segment, 668
GetAsDataElement
gdcm::Attribute, 115
gdcm::Attribute< Group, Element, TVR, VM::VM1 >, 120
gdcm::Attribute< Group, Element, TVR, VM::VM1_n >, 127
gdcm::Element, 292
gdcm::Element< TVR, VM::VM1_n >, 296
gdcm::PrivateTag, 619
gdcm::network::AbstractSyntax, 92
GetAsPoints
gdcm::Curve, 232
GetAsString
gdcm::CodeString, 197
GetAxisOfRotation
gdcm::Surface, 749
GetBasicApplicationLevelConfidentialityProfileAttributes
gdcm::Anonymizer, 98
GetBitPosition
gdcm::Overlay, 554
GetBitSample
gdcm::LookupTable, 473
GetBitsAllocated
gdcm::Overlay, 554
gdcm::PixelFormat, 582
GetBitsStored
gdcm::PixelFormat, 582
GetBlob
gdcm::network::PresentationDataValue, 611
GetBuffer
gdcm::Bitmap, 159
gdcm::ByteValue, 176
gdcm::Parser, 562
gdcm::SequenceOfFragments, 681
GetBuffer2
gdcm::Bitmap, 159
GetBufferAsRGBA
gdcm::LookupTable, 473
GetBufferLength
gdcm::Bitmap, 160
gdcm::JPEGLSCodec, 464
gdcm::PNMCodec, 600
gdcm::RLECodec, 656
GetBuildVersion
gdcm::Version, 888
GetByteValue
gdcm::CSAElement, 215
gdcm::DataElement, 237
GetCSADataInfo
gdcm::CSAHeader, 220
GetCSAEEnd
gdcm::CSAHeader, 221
GetCSAElementByName
gdcm::CSAHeader, 221
GetCSAHeaderDict
gdcm::Dicts, 278
GetCSAHeaderDictEntry
gdcm::CSAHeaderDict, 224
GetCSAImageHeaderInfoTag
gdcm::CSAHeader, 221
GetCSASeriesHeaderInfoTag
gdcm::CSAHeader, 221
GetCTImageSeriesUIDs
gdcm::DirectoryHelper, 287
GetCWD
gdcm::System, 766
GetCalledAETitle
gdcm::ServiceClassUser, 697
gdcm::network::AAssociateRQPDU, 88
gdcm::network::ULConnectionInfo, 864
GetCalledComputerName
gdcm::network::ULConnectionInfo, 864
GetCalledIPAddress
gdcm::network::ULConnectionInfo, 864
GetCalledIPPort
gdcm::network::ULConnectionInfo, 864
GetCallingAETitle
gdcm::network::AAssociateRQPDU, 88
gdcm::network::ULConnectionInfo, 864
GetCenterOfRotation
gdcm::Surface, 749
GetCharacterFromCurrentLocale
gdcm::QueryFactory, 629
GetCipherType
gdcm::CAPICryptographicMessageSyntax, 181
gdcm::CryptographicMessageSyntax, 212
gdcm::OpenSSLCryptographicMessageSyntax, 544
gdcm::OpenSSLP7CryptographicMessageSyntax, 548
GetCodec
gdcm::FileChangeTransferSyntax, 333
GetColorLevel
vtkImageColorViewer, 943
GetColorWindow
vtkImageColorViewer, 943
GetColumns
gdcm::Bitmap, 160
gdcm::Overlay, 555
GetCommand
gdcm::Subject, 745
GetConnectionInfo
gdcm::network::ULConnection, 860
GetConstructorString
gdcm::Dicts, 278
GetContourReferencedFrameOfReferenceClassUID

vtkRTStructSetProperties, 962
GetContourReferencedFrameOfReferenceInstanceUID
vtkRTStructSetProperties, 962
GetCryptographicMessageSyntax
gdcm::Anonymizer, 99
GetCurrentByteIndex
gdcm::Parser, 562
GetCurrentDateTime
gdcm::System, 766
GetCurrentModuleFileName
gdcm::System, 766
GetCurrentProcessFileName
gdcm::System, 766
GetCurrentResourcesDirectory
gdcm::System, 766
GetCurve
gdcm::Pixmap, 588
GetCurveDataDescriptor
gdcm::Curve, 232
GetDEEnd
gdcm::DataSet, 251
GetDES
gdcm::DataSet, 251
GetData
gdcm::DataEvent, 245
GetDataElement
gdcm::Bitmap, 160
gdcm::DataSet, 250
gdcm::Item, 441
GetDataExtraRoot
gdcm::Testing, 785
GetDataLength
gdcm::DataEvent, 245
GetDataRoot
gdcm::Testing, 785
GetDataSet
gdcm::CSAHeader, 221
gdcm::DataSetEvent, 255
gdcm::File, 327
GetDataSetPos
gdcm::network::ULEvent, 869
GetDataSetTransferSyntax
gdcm::FileMetaInformation, 345
GetDataSets
gdcm::network::ULBasicCallback, 858
GetDataValueRepresentation
gdcm::Curve, 232
GetDebugFlag
gdcm::Trace, 790
GetDebugStream
gdcm::Trace, 790
GetDecodeLength
gdcm::Base64, 138
GetDefaultTransferSyntax
gdcm::PresentationContextGenerator, 607
GetDefs
gdcm::Global, 371
gdcm::TableReader, 772
GetDescription
gdcm::CSAHeaderDictEntry, 226
gdcm::Exception, 318
gdcm::ModuleEntry, 509
gdcm::Overlay, 555
GetDescriptiveName
vtkGDCMImageReader, 907
vtkGDCMImageReader2, 913
vtkGDCMImageWriter, 920
GetDict
gdcm::XMLDictReader, 975
GetDictEntry
gdcm::Dict, 268
gdcm::Dicts, 278
gdcm::PrivateDict, 617
GetDictEntryByKeyword
gdcm::Dict, 268
GetDictEntryByName
gdcm::Dict, 268
GetDictName
gdcm::DictConverter, 271
GetDictVM
gdcm::Attribute, 115
gdcm::Attribute< Group, Element, TVR, VM::VM1 >, 120
gdcm::Attribute< Group, Element, TVR, VM::VM1_n >, 127
GetDictVR
gdcm::Attribute, 115
gdcm::Attribute< Group, Element, TVR, VM::VM1 >, 120
gdcm::Attribute< Group, Element, TVR, VM::VM1_n >, 127
GetDicts
gdcm::Global, 371, 372
GetDimension
gdcm::Bitmap, 160
GetDimensions
gdcm::Bitmap, 160
gdcm::Curve, 232
gdcm::ImageCodec, 401
GetDimensionsValue
gdcm::ImageHelper, 409
GetDimensionsValueForResolution
gdcm::StreamImageReader, 723
GetDirectionCosines
gdcm::Image, 383
GetDirectionCosinesFromDataSet
gdcm::ImageHelper, 409
GetDirectionCosinesTolerance

gdcm::IPPSorter, 437
GetDirectionCosinesValue
gdcm::ImageHelper, 409
GetDirectories
gdcm::Directory, 285
GetElapsedTime
gdcm::network::ARTIMTimer, 110
GetElement
gdcm::Tag, 776
GetElementTag
gdcm::Tag, 776
GetEncodeLength
gdcm::Base64, 138
GetErrorCode
gdcm::Parser, 562
GetErrorFlag
gdcm::Trace, 790
GetErrorStream
gdcm::Trace, 790
GetErrorString
gdcm::Parser, 562
GetEvent
gdcm::network::ULEvent, 869
GetEventName
gdcm::AnonymizeEvent, 94
gdcm::DataEvent, 245
gdcm::DataSetEvent, 255
gdcm::Event, 316
gdcm::FileNameEvent, 352
gdcm::ProgressEvent, 622
GetExtension
gdcm::Filename, 348
GetFactoryInstance
gdcm::CryptoFactory, 211
GetFile
gdcm::Anonymizer, 99
gdcm::DICOMDIRGenerator, 266
gdcm::FileDecompressLookupTable, 336
gdcm::FileDerivation, 338
gdcm::FileExplicitFilter, 341
gdcm::IconImageFilter, 376
gdcm::PythonFilter, 626
gdcm::Reader, 644
gdcm::SplitMosaicFilter, 718, 719
gdcm::StreamImageReader, 723
gdcm::StringFilter, 741
gdcm::Writer, 972
vtkGDCMMedicalImageProperties, 924
GetFileExtensions
vtkGDCMImageReader, 907
vtkGDCMImageReader2, 913
vtkGDCMImageWriter, 920
GetFileMetaInformationVersion
gdcm::FileMetaInformation, 345
GetFileName
gdcm::FileNameEvent, 352
gdcm::Filename, 349
gdcm::Testing, 785
vtkGDCMImageWriter, 920
vtkGDCMThreadedImageReader2, 937
GetFileNames
gdcm::Testing, 785
GetFilename
gdcm::FilenameGenerator, 354
gdcm::TableReader, 772
GetFilenameFromTagToValue
gdcm::Scanner, 663
gdcm::StrictScanner, 733
GetFilenames
gdcm::Directory, 285
gdcm::FilenameGenerator, 354
gdcm::Scanner, 663
gdcm::Sorter, 714
gdcm::StrictScanner, 733
GetFilenamesFromSeriesUIDs
gdcm::DirectoryHelper, 287
GetFiles
gdcm::FileSet, 356
GetFiniteVolume
gdcm::Surface, 750
GetFirstSingleSerieUIDFileSet
gdcm::SerieHelper, 692
GetForcePixelSpacing
gdcm::ImageHelper, 409
GetForceRescaleInterceptSlope
gdcm::ImageHelper, 409
GetFormat
gdcm::CSAHeader, 221
GetFragBuffer
gdcm::SequenceOfFragments, 681
GetFragment
gdcm::SequenceOfFragments, 681
GetFragmentSizeMax
gdcm::ImageFragmentSplitter, 407
GetFrameOfReference
gdcm::DirectoryHelper, 287
GetFullLength
gdcm::FileMetaInformation, 345
GetGDCMDataRoot
vtkGDCMTesting, 932
GetGDCMImplementationClassUID
gdcm::FileMetaInformation, 345
GetGDCMImplementationVersionName
gdcm::FileMetaInformation, 345
GetGDCMSourceApplicationEntityTitle
gdcm::FileMetaInformation, 345
GetGDCMUID
gdcm::UIDGenerator, 804

GetGroup
gdcm::Curve, 232
gdcm::Overlay, 555
gdcm::Tag, 777
GetHasExpired
gdcm::network::ARTIMTimer, 110
GetHeader
gdcm::File, 327
GetHeaderInfo
gdcm::ImageCodec, 401
gdcm::JPEG12Codec, 445
gdcm::JPEG16Codec, 448
gdcm::JPEG2000Codec, 451
gdcm::JPEG8Codec, 454
gdcm::JPEGCodec, 459
gdcm::JPEGLSCodec, 464
gdcm::PGXCodec, 577
gdcm::PNMCodec, 600
gdcm::RAWCodec, 640
gdcm::RLECodec, 656
GetHierachicalSearchTags
gdcm::QueryBase, 628
gdcm::QueryImage, 631
gdcm::QueryPatient, 633
gdcm::QuerySeries, 635
gdcm::QueryStudy, 637
GetHighBit
gdcm::PixelFormat, 583
GetHostName
gdcm::System, 767
GetIODEntry
gdcm::IOD, 430
GetIODFromFile
gdcm::Defs, 260
GetIODFromSOPClassUID
gdcm::SOPClassUIDToIOD, 711
GetIODNameFromMediaStorage
gdcm::Defs, 260
GetIODs
gdcm::Defs, 260, 261
GetIOD
gdcm::IODs, 435
gdcm::SOPClassUIDToIOD, 711
GetIStream
gdcm::network::ULEvent, 869
GetIconImage
gdcm::IconImageFilter, 376
gdcm::IconImageGenerator, 379
gdcm::Pixmap, 588
vtkGDCMImageReader, 907
vtkGDCMImageReader2, 914
GetIconImagePort
vtkGDCMImageReader2, 914
GetIE
gdcm::IODEntry, 432
GetImage
gdcm::ImageReader, 414
gdcm::ImageWriter, 423
gdcm::PixmapWriter, 596
gdcm::SplitMosaicFilter, 719
GetImplementationClassUID
gdcm::FileMetaInformation, 345
GetImplementationVersionName
gdcm::FileMetaInformation, 345
GetIndex
gdcm::SwapCode, 762
gdcm::VM, 894
GetInitialized
gdcm::CAPICryptographicMessageSyntax, 181
GetInput
gdcm::ImageToImageFilter, 421
gdcm::PixmapToPixmapFilter, 594
vtkImageColorViewer, 943
GetInputFilename
gdcm::DictConverter, 271
GetInstance
gdcm::Global, 372
GetIntercept
gdcm::Image, 383
gdcm::Rescaler, 652
GetInterfile
gdcm::CSAHeader, 222
GetInternal
gdcm::Preamble, 601
GetIsCommand
gdcm::network::PresentationDataValue, 611
GetIsLastFragment
gdcm::network::PresentationDataValue, 611
GetItem
gdcm::SequenceOfItems, 687
GetKey
gdcm::CSAElement, 215
GetKeys
gdcm::Scanner, 663
gdcm::StrictScanner, 734
GetKeyword
gdcm::DictEntry, 273
GetKeywordFromTag
gdcm::Dict, 269
GetLUTDescriptor
gdcm::LookupTable, 474
GetLUTLength
gdcm::LookupTable, 474
GetLUT
gdcm::Bitmap, 160
gdcm::ImageCodec, 401
gdcm::ImageHelper, 409
gdcm::LookupTable, 474

GetLabel
gdcm::Orientation, 550
GetLastElement
gdcm::ParseException, 559
GetLastSystemError
gdcm::System, 767
GetLength
gdcm::ByteValue, 176
gdcm::CP246ExplicitDataElement, 209
gdcm::DataElement, 237
gdcm::DataSet, 251
gdcm::Element, 292
gdcm::Element< TVR, VM::VM1_n >, 296
gdcm::Element< VR::AS, VM::VM5 >, 304
gdcm::ExplicitDataElement, 322
gdcm::ExplicitImplicitDataElement, 324
gdcm::Fragment, 369
gdcm::ImplicitDataElement, 428
gdcm::Item, 442
gdcm::Preamble, 601
gdcm::SequenceOfFragments, 681
gdcm::SequenceOfItems, 687
gdcm::Tag, 777
gdcm::UNExplicitDataElement, 874
gdcm::UNExplicitImplicitDataElement, 876
gdcm::VR16ExplicitDataElement, 902
gdcm::Value, 886
gdcm::VL, 890
gdcm::VM, 895
gdcm::VR, 899
GetLocaleCharset
gdcm::System, 767
GetLossless
gdcm::JPEGCodec, 459
gdcm::JPEGLSCodec, 464
GetLossyFlag
gdcm::ImageCodec, 401
GetLossyFlagFromFile
gdcm::Testing, 786
GetMD5DataImage
gdcm::Testing, 786
GetMD5DataImages
gdcm::Testing, 786
GetMD5FromBrokenFile
gdcm::Testing, 786
GetMD5FromFile
gdcm::Testing, 786
GetMD5MetaImage
vtkGDCMTesting, 932
GetMHDMD5FromFile
vtkGDCMTesting, 932
GetMPType
gdcm::MeshPrimitive, 497
GetMPTypeString
gdcm::MeshPrimitive, 497
GetMRImageSeriesUIDs
gdcm::DirectoryHelper, 287
GetMSString
gdcm::MediaStorage, 489
GetMSType
gdcm::MediaStorage, 489
GetMTime
vtkImageMapToColors16, 948
GetMacro
gdcm::Macros, 480
GetMacroEntry
gdcm::Macro, 478
GetMacros
gdcm::Defs, 261
GetMajorAxisFromPatientRelativeDirectionCosine
gdcm::Orientation, 550
GetMajorVersion
gdcm::Version, 888
GetManifold
gdcm::Surface, 750
GetMapping
gdcm::Scanner, 663
gdcm::StrictScanner, 734
GetMappingFromTagToValue
gdcm::Scanner, 663
gdcm::StrictScanner, 734
GetMappings
gdcm::Scanner, 664
gdcm::StrictScanner, 734
GetMax
gdcm::PixelFormat, 583
GetMaxLength
gdcm::PersonName, 574
GetMaxPDULength
gdcm::network::ULConnectionInfo, 864
GetMaxPDUSize
gdcm::network::ULConnection, 860
GetMaximumLength
gdcm::network::MaximumLengthSub, 481
GetMaximumLengthSub
gdcm::network::UserInformation, 881
GetMaximumPointDistance
gdcm::Surface, 750
GetMeanPointDistance
gdcm::Surface, 750
GetMediaStorage
gdcm::DataSet, 251
gdcm::FileMetaInformation, 345
GetMediaStorageAsString
gdcm::FileMetaInformation, 345
GetMediaStorageDataFile
gdcm::Testing, 786
GetMediaStorageDataFiles

gdcm::Testing, 786
GetMediaStorageFromFile
gdcm::Testing, 786
GetMeshPrimitive
gdcm::Surface, 750
GetMessageHeader
gdcm::network::PresentationDataValue, 611
GetMetaInformationTS
gdcm::FileMetaInformation, 345
GetMin
gdcm::PixelFormat, 583
GetMinorVersion
gdcm::Version, 888
GetModality
gdcm::MediaStorage, 489
GetModalityDimension
gdcm::MediaStorage, 489
GetModule
gdcm::Modules, 511
GetModuleEntry
gdcm::NestedModuleEntries, 527
GetModuleEntryInMacros
gdcm::Module, 506
GetModules
gdcm::Defs, 261
GetName
gdcm::CSAElement, 216
gdcm::CSAHeaderDictEntry, 226
gdcm::DictEntry, 273
gdcm::Filename, 349
gdcm::GroupDict, 374
gdcm::IODEntry, 432
gdcm::Macro, 478
gdcm::Module, 506
gdcm::ModuleEntry, 509
gdcm::PDBElement, 566
gdcm::QueryBase, 628
gdcm::QueryImage, 631
gdcm::QueryPatient, 633
gdcm::QuerySeries, 635
gdcm::QueryStudy, 637
gdcm::UIDs, 824
gdcm::network::AbstractSyntax, 92
gdcm::network::ApplicationContext, 103
gdcm::network::TransferSyntaxSub, 798
GetNeedByteSwap
gdcm::Bitmap, 161
gdcm::ImageCodec, 401
GetNegociatedType
gdcm::TransferSyntax, 795
GetNestedDataSet
gdcm::Item, 442
GetNextSingleSerieUIDFileSet
gdcm::SerieHelper, 692
GetNoOfItems
gdcm::CSAElement, 216
GetNumberOfComponents
gdcm::PersonName, 574
GetNumberOfContourReferencedFrameOfReferences
vtkRTStructSetProperties, 963
GetNumberOfCurves
gdcm::Curve, 232
gdcm::Pixmap, 588
GetNumberOfDimensions
gdcm::Bitmap, 161
gdcm::ImageCodec, 401
GetNumberOfElementsFromArray
gdcm::VM, 895
GetNumberOfFileNames
gdcm::Testing, 786
GetNumberOfFilenames
gdcm::FilenameGenerator, 354
GetNumberOfFragments
gdcm::SequenceOfFragments, 681
GetNumberOfIODs
gdcm::IOD, 430
GetNumberOfIconImages
gdcm::IconImageFilter, 376
GetNumberOfItems
gdcm::SequenceOfItems, 687
GetNumberOfMD5DataImages
gdcm::Testing, 786
GetNumberOfMD5MetaImages
vtkGDCMTesting, 932
GetNumberOfMSString
gdcm::MediaStorage, 489
GetNumberOfMSType
gdcm::MediaStorage, 489
GetNumberOfMediaStorageDataFiles
gdcm::Testing, 787
GetNumberOfModality
gdcm::MediaStorage, 489
GetNumberOfModuleEntries
gdcm::NestedModuleEntries, 527
GetNumberOfOverlays
gdcm::Pixmap, 588
GetNumberOfPoints
gdcm::Curve, 232
GetNumberOfPresentationContext
gdcm::network::AAssociateRQPDU, 88
GetNumberOfPresentationContextAC
gdcm::network::AAssociateACPDU, 82
GetNumberOfPresentationDataValues
gdcm::network::PDataTFPDU, 564
GetNumberOfPrimitivesData
gdcm::MeshPrimitive, 497
GetNumberOfReferencedFrameOfReferences
vtkRTStructSetProperties, 963

GetNumberOfSOPClassToIOD
gdcm::SOPClassUIDToIOD, 711
GetNumberOfSegments
gdcm::SegmentWriter, 677
GetNumberOfStructureSetROIs
vtkRTStructSetProperties, 963
GetNumberOfSurfacePoints
gdcm::Surface, 750
GetNumberOfSurfaces
gdcm::SurfaceReader, 758
gdcm::SurfaceWriter, 760
GetNumberOfTransferSyntaxStrings
gdcm::UIDs, 824
GetNumberOfTransferSyntaxes
gdcm::PresentationContext, 603
gdcm::network::PresentationContextRQ, 609
GetNumberOfValues
gdcm::Attribute, 115
gdcm::Attribute< Group, Element, TVR, VM::VM1 >, 120
gdcm::Attribute< Group, Element, TVR, VM::VM1_n >, 127
GetNumberOfVectors
gdcm::Surface, 750
GetObliquityThresholdCosineValue
gdcm::Orientation, 550
GetOffScreenRendering
vtkImageColorViewer, 943
GetOptionalTags
gdcm::QueryBase, 628
gdcm::QueryImage, 631
gdcm::QueryPatient, 633
gdcm::QuerySeries, 635
gdcm::QueryStudy, 637
GetOrderedValues
gdcm::Scanner, 664
gdcm::StrictScanner, 734
GetOrigin
gdcm::Image, 383
gdcm::Overlay, 555
GetOriginValue
gdcm::ImageHelper, 409
GetOuput
gdcm::ImageConverter, 405
GetOutput
gdcm::BitmapToBitmapFilter, 166
gdcm::ImageToImageFilter, 421
gdcm::PixmapToPixmapFilter, 594
GetOutputAsBitmap
gdcm::BitmapToBitmapFilter, 166
GetOutputAsPixmap
gdcm::PixmapToPixmapFilter, 594
GetOutputFilename
gdcm::DictConverter, 271
GetOutputType
gdcm::DictConverter, 271
GetOverlay
gdcm::Pixmap, 588
vtkGDCMImageReader, 907
vtkGDCMImageReader2, 914
GetOverlayData
gdcm::Overlay, 555
GetOverlayPort
vtkGDCMImageReader2, 914
GetOverlayTypeAsString
gdcm::Overlay, 555
GetOverlayTypeFromString
gdcm::Overlay, 555
GetOverlayVisibility
vtkImageColorViewer, 943
GetOwner
gdcm::PrivateTag, 619
GetPDBEEnd
gdcm::PDBHeader, 568
GetPDBElementByName
gdcm::PDBHeader, 569
GetPDBInfoTag
gdcm::PDBHeader, 569
GetPDUs
gdcm::network::ULEvent, 869
GetPDVs
gdcm::network::PDUFactory, 573
GetPIString
gdcm::PhotometricInterpretation, 579
GetPIType
gdcm::PhotometricInterpretation, 579
GetPMSRescaleInterceptSlope
gdcm::ImageHelper, 410
GetPath
gdcm::Filename, 349
GetPattern
gdcm::FilenameGenerator, 354
GetPermissions
gdcm::System, 767
GetPhotometricInterpretation
gdcm::Bitmap, 161
gdcm::ImageChangePhotometricInterpretation, 389
gdcm::ImageCodec, 401
GetPhotometricInterpretationValue
gdcm::ImageHelper, 409
GetPixelFormat
gdcm::Bitmap, 161
gdcm::ImageCodec, 401
GetPixelFormatValue
gdcm::ImageHelper, 410
GetPixelRepresentation
gdcm::PixelFormat, 583
GetPixelSize

gdcm::PixelFormat, 583
GetPixelSpacingDataRoot
gdcm::Testing, 787
GetPixmap
gdcm::FileDecompressLookupTable, 336
gdcm::IconImageGenerator, 379
gdcm::PixmapReader, 591
gdcm::PixmapWriter, 596, 597
GetPlanarConfiguration
gdcm::Bitmap, 161
gdcm::ImageChangePlanarConfiguration, 392
gdcm::ImageCodec, 402
GetPlanarConfigurationValue
gdcm::ImageHelper, 410
GetPointCoordinatesData
gdcm::Surface, 750
GetPointPositionAccuracy
gdcm::Surface, 750
GetPointer
gdcm::ByteValue, 177
gdcm::LookupTable, 474
gdcm::SmartPointer, 708
vtkLookupTable16, 959
GetPointerFromElement
gdcm::ImageHelper, 410
GetPointsBoundingBoxCoordinates
gdcm::Surface, 750
GetPosition
vtkImageColorViewer, 943
GetPreamble
gdcm::FileMetaInformation, 345
GetPrefix
gdcm::FilenameGenerator, 355
GetPresentationContext
gdcm::network::AAssociateRQPDU, 88
GetPresentationContextACByID
gdcm::network::ULConnection, 860
GetPresentationContextAC
gdcm::network::AAssociateACPDU, 82
GetPresentationContextByAbstractSyntax
gdcm::network::AAssociateRQPDU, 88
GetPresentationContextByID
gdcm::network::AAssociateRQPDU, 88
GetPresentationContextIDFromPresentationContext
gdcm::network::ULConnection, 860
GetPresentationContextID
gdcm::PresentationContext, 603
gdcm::network::PresentationContextAC, 605
gdcm::network::PresentationContextRQ, 609
gdcm::network::PresentationDataValue, 611
GetPresentationContextRQByID
gdcm::network::ULConnection, 860
GetPresentationContexts
gdcm::PresentationContextGenerator, 607
gdcm::network::AAssociateRQPDU, 88
gdcm::network::ULConnection, 860
GetPresentationDataValue
gdcm::network::PDataTFPDU, 564
GetPrettyPrint
gdcm::JSON, 466
GetPrimitiveData
gdcm::MeshPrimitive, 497, 498
GetPrimitiveType
gdcm::MeshPrimitive, 498
GetPrimitivesData
gdcm::MeshPrimitive, 498
GetPrintStyle
gdcm::Printer, 614
gdcm::XMLPrinter, 977
GetPrivateCreator
gdcm::DataSet, 251
gdcm::Tag, 777
GetPrivateDict
gdcm::Dicts, 279
gdcm::XMLPrivateDictReader, 979
GetProcessingAlgorithm
gdcm::Surface, 750, 751
GetProgress
gdcm::ProgressEvent, 622
GetPropertyCategory
gdcm::Segment, 668, 669
GetPropertyType
gdcm::Segment, 669
GetProtocol
gdcm::network::ULConnection, 860
GetPublicDict
gdcm::Dicts, 279
GetQuality
gdcm::JPEG2000Codec, 451
gdcm::JPEGCodec, 459
GetQueryDataSet
gdcm::BaseQuery, 147
GetQueryLevel
gdcm::QueryBase, 628
gdcm::QueryImage, 632
gdcm::QueryPatient, 634
gdcm::QuerySeries, 636
gdcm::QueryStudy, 638
GetQueryLevelFromQueryRoot
gdcm::BaseRootQuery, 150
GetQueryLevelFromString
gdcm::BaseRootQuery, 150
GetQueryLevelString
gdcm::BaseRootQuery, 150
GetRAWMD5FromFile
vtkGDCMTesting, 933
GetRTStructSeriesUIDs
gdcm::DirectoryHelper, 287

GetRate
gdcm::JPEG2000Codec, 451
GetRealWorldValueMappingContent
gdcm::ImageHelper, 410
GetReason
gdcm::network::PresentationContextAC, 605
GetRecommendedDisplayCIELabValue
gdcm::Surface, 751
GetRecommendedDisplayGrayscaleValue
gdcm::Surface, 751
GetRecommendedPresentationOpacity
gdcm::Surface, 751
GetRecommendedPresentationType
gdcm::Surface, 751
GetRef
gdcm::IODEntry, 432
GetReferencedFrameOfReferenceClassUID
vtkRTStructSetProperties, 963
GetReferencedFrameOfReferenceInstanceUID
vtkRTStructSetProperties, 963
GetRegion
gdcm::ImageRegionReader, 418
GetRequiredDataSet
gdcm::ModalityPerformedProcedureStepCreate↩
Query, 501
gdcm::ModalityPerformedProcedureStepSetQuery, 503
GetRequiredTags
gdcm::QueryBase, 628
gdcm::QueryImage, 632
gdcm::QueryPatient, 634
gdcm::QuerySeries, 636
gdcm::QueryStudy, 638
GetRescaleInterceptSlopeValue
gdcm::ImageHelper, 410
GetReserved43_74
gdcm::network::AAssociateRQPDU, 89
GetResponses
gdcm::network::ULBasicCallback, 858
GetRetired
gdcm::DictEntry, 273
GetRoot
gdcm::UIDGenerator, 804
GetRows
gdcm::Bitmap, 161
gdcm::Overlay, 555
GetSOPClassUIDFromIOD
gdcm::SOPClassUIDToIOD, 711
GetSOPClassUIDToIODs
gdcm::SOPClassUIDToIOD, 712
GetSOPClassUIDToIOD
gdcm::SOPClassUIDToIOD, 712
GetSOPClassUID
gdcm::DirectoryHelper, 287
GetSOPInstanceUID
gdcm::BaseQuery, 147
GetSTATESString
gdcm::Surface, 751
GetSTATES
gdcm::Surface, 751
GetSamplesPerPixel
gdcm::PhotometricInterpretation, 579
gdcm::PixelFormat, 583
GetScalarType
gdcm::PixelFormat, 584
GetScalarTypeAsString
gdcm::PixelFormat, 584
GetScanner
gdcm::DICOMDIRGenerator, 266
GetSegment
gdcm::SegmentWriter, 677
GetSegmentAlgorithmName
gdcm::Segment, 669
GetSegmentAlgorithmType
gdcm::Segment, 669
GetSegmentDescription
gdcm::Segment, 669
GetSegmentLabel
gdcm::Segment, 669
GetSegmentNumber
gdcm::Segment, 669
GetSegments
gdcm::SegmentReader, 675
gdcm::SegmentWriter, 677
GetSelectedPrivateGroupOffsetFromFile
gdcm::Testing, 787
GetSelectedTagsOffsetFromFile
gdcm::Testing, 787
GetSequenceOfFragments
gdcm::DataElement, 238
GetSeriesUIDsBySOPClassUID
gdcm::DirectoryHelper, 287
GetSize
gdcm::VR, 899
vtkImageColorViewer, 943
GetSizeof
gdcm::VR, 899
GetSliceMax
vtkImageColorViewer, 943
GetSliceMin
vtkImageColorViewer, 943
GetSliceRange
vtkImageColorViewer, 943
GetSlope
gdcm::Image, 383
gdcm::Rescaler, 652
GetSourceApplicationEntityTitle
gdcm::FileMetaInformation, 346

GetSourceDirectory
gdcm::Testing, 787
GetSpacing
gdcm::Image, 383
GetSpacingTagFromMediaStorage
gdcm::ImageHelper, 410
GetSpacingValue
gdcm::ImageHelper, 410
GetStart
gdcm::ByteBuffer, 171
GetState
gdcm::network::ULConnection, 860
GetStateIndex
gdcm::network, 75
GetStream
gdcm::Trace, 790
GetStreamCurrentPosition
gdcm::Reader, 644
GetStreamOffsetFromFile
gdcm::Testing, 787
GetStreamPtr
gdcm::Reader, 644
gdcm::Writer, 972
GetString
gdcm::MediaStorage, 489
gdcm::PhotometricInterpretation, 579
gdcm::TransferSyntax, 796
gdcm::UIDs, 824
GetStringValueFromTag
gdcm::DirectoryHelper, 287
GetStructureSetObservationNumber
vtkRTStructSetProperties, 963
GetStructureSetROIDescription
vtkRTStructSetProperties, 963
GetStructureSetROIGenerationAlgorithm
vtkRTStructSetProperties, 963
GetStructureSetROIName
vtkRTStructSetProperties, 963
GetStructureSetROINumber
vtkRTStructSetProperties, 963
GetStructureSetROIObservationLabel
vtkRTStructSetProperties, 963
GetStructureSetROIRefFrameRefUID
vtkRTStructSetProperties, 963
GetStructureSetRTROIInterpretedType
vtkRTStructSetProperties, 963
GetSurface
gdcm::Segment, 669
GetSurfaceComments
gdcm::Surface, 751
GetSurfaceCount
gdcm::Segment, 669
GetSurfaceNumber
gdcm::Surface, 751
GetSurfaceProcessing
gdcm::Surface, 751
GetSurfaceProcessingDescription
gdcm::Surface, 751
GetSurfaceProcessingRatio
gdcm::Surface, 751
GetSurfaces
gdcm::Segment, 669
GetSwapCode
gdcm::TransferSyntax, 796
GetSwapCodeString
gdcm::SwapCode, 762
GetSyngoDT
gdcm::CSAElement, 216
GetTSString
gdcm::TransferSyntax, 796
GetTSType
gdcm::TransferSyntax, 796
GetTable
gdcm::SequenceOfFragments, 682
GetTableEntry
gdcm::Table, 769
GetTag
gdcm::AnonymizeEvent, 94
gdcm::Attribute, 115
gdcm::Attribute< Group, Element, TVR, VM::VM1 >, 120
gdcm::Attribute< Group, Element, TVR, VM::VM1_n >, 127
gdcm::DataElement, 238
GetTagListByLevel
gdcm::BaseRootQuery, 150
gdcm::FindPatientRootQuery, 364
gdcm::FindStudyRootQuery, 366
gdcm::MovePatientRootQuery, 513
gdcm::MoveStudyRootQuery, 516
gdcm::WLMFindQuery, 967
GetTempDirectory
gdcm::Testing, 787
GetTempDirectoryW
gdcm::Testing, 787
GetTempFilename
gdcm::Testing, 787
GetTempFilenameW
gdcm::Testing, 788
GetTimeout
gdcm::ServiceClassUser, 697
gdcm::network::ARTIMTimer, 110
GetTimer
gdcm::network::ULConnection, 860
GetTimezoneOffsetFromUTC
gdcm::System, 767
GetToplevel
gdcm::Directory, 285

GetTransferSyntax
gdcm::Bitmap, 161
gdcm::ImageChangeTransferSyntax, 395
gdcm::PresentationContext, 603
gdcm::network::PresentationContextAC, 605
gdcm::network::PresentationContextRQ, 609
GetTransferSyntaxString
gdcm::UIDs, 824
GetTransferSyntaxStrings
gdcm::UIDs, 824
GetTransferSyntaxes
gdcm::network::PresentationContextRQ, 609
GetType
gdcm::ModuleEntry, 509
gdcm::Orientation, 550
gdcm::Overlay, 555
gdcm::PhotometricInterpretation, 579
GetTypeAsEnum
gdcm::Overlay, 555
GetTypeFromTag
gdcm::Defs, 261
gdcm::IOD, 431
GetTypeOfData
gdcm::Curve, 232
GetTypeOfDataDescription
gdcm::Curve, 232
GetTypeString
gdcm::Type, 802
GetTypeType
gdcm::Type, 802
GetUIDName
gdcm::UIDs, 824
GetUIDString
gdcm::UIDs, 824
GetUniqueTags
gdcm::QueryBase, 628
gdcm::QueryImage, 632
gdcm::QueryPatient, 634
gdcm::QuerySeries, 636
gdcm::QueryStudy, 638
GetUnpackBuffer
gdcm::Overlay, 556
GetUnpackBufferLength
gdcm::Overlay, 556
GetUsage
gdcm::IODEntry, 432
GetUsageString
gdcm::Usage, 878
GetUsageType
gdcm::IODEntry, 433
gdcm::Usage, 878
GetUserData
gdcm::Parser, 562
GetUserInformation
gdcm::network::AAssociateACPDU, 82
gdcm::network::AAssociateRQPDU, 89
GetVIEWType
gdcm::Surface, 751
GetVIEWTypeString
gdcm::Surface, 751
GetVL16Max
gdcm::VL, 890
GetVL32Max
gdcm::VL, 890
GetVMString
gdcm::VM, 895
GetVMType
gdcm::VM, 895
GetVMTypeFromLength
gdcm::VM, 895
GetVRFromTag
gdcm, 63
GetVRString
gdcm::VR, 899
GetVRStringFromFile
gdcm::VR, 899
GetVRType
gdcm::VR, 899
GetVRTypeFromFile
gdcm::VR, 899
GetVTKDataRoot
vtkGDCMTesting, 933
GetValidDataSet
gdcm::WLMFindQuery, 968
GetValidatedFile
gdcm::Validate, 884
GetValue
gdcm::Attribute, 116
gdcm::Attribute< Group, Element, TVR, VM::VM1 >, 121
gdcm::Attribute< Group, Element, TVR, VM::VM1_n >, 127
gdcm::CSAElement, 216
gdcm::DataElement, 238
gdcm::Element, 292
gdcm::Element< TVR, VM::VM1_n >, 297
gdcm::PDBElement, 566
gdcm::Scanner, 664
gdcm::StrictScanner, 734
GetValueAsSQ
gdcm::DataElement, 239
GetValues
gdcm::Attribute, 116
gdcm::Attribute< Group, Element, TVR, VM::VM1 >, 121
gdcm::Attribute< Group, Element, TVR, VM::VM1_n >, 127
gdcm::Element, 292

gdcm::Scanner, 664
gdcm::StrictScanner, 734
GetVectorAccuracy
gdcm::Surface, 751
GetVectorCoordinateData
gdcm::Surface, 751
GetVectorDimensionality
gdcm::Surface, 751
GetVersion
gdcm::Version, 888
GetVL
gdcm::DataElement, 239
GetVM
gdcm::Attribute, 116
gdcm::Attribute< Group, Element, TVR, VM::VM1 >, 121
gdcm::Attribute< Group, Element, TVR, VM::VM1_3 >, 124
gdcm::Attribute< Group, Element, TVR, VM::VM1_8 >, 125
gdcm::Attribute< Group, Element, TVR, VM::VM1_n >, 127
gdcm::Attribute< Group, Element, TVR, VM::VM2↩ _2n >, 130
gdcm::Attribute< Group, Element, TVR, VM::VM2_n >, 132
gdcm::Attribute< Group, Element, TVR, VM::VM3↩ _3n >, 133
gdcm::Attribute< Group, Element, TVR, VM::VM3_n >, 135
gdcm::CSAElement, 216
gdcm::CSAHeaderDictEntry, 226
gdcm::DictEntry, 273
gdcm::Element, 292
gdcm::Element< TVR, VM::VM1_n >, 297
GetVR
gdcm::Attribute, 116
gdcm::Attribute< Group, Element, TVR, VM::VM1 >, 121
gdcm::Attribute< Group, Element, TVR, VM::VM1_n >, 127
gdcm::CSAElement, 216
gdcm::CSAHeaderDictEntry, 226
gdcm::DataElement, 239
gdcm::DictEntry, 273
gdcm::Element, 293
gdcm::Element< TVR, VM::VM1_n >, 297
GetWarningFlag
gdcm::Trace, 790
GetWarningStream
gdcm::Trace, 790
GetWindowName
vtkImageColorViewer, 943
GetXMax
gdcm::BoxRegion, 169
GetXMin
gdcm::BoxRegion, 169
GetYMax
gdcm::BoxRegion, 169
GetYMin
gdcm::BoxRegion, 170
GetZMax
gdcm::BoxRegion, 170
GetZMin
gdcm::BoxRegion, 170
GetZSpacing
gdcm::IPPSorter, 437
GetZSpacingTagFromMediaStorage
gdcm::ImageHelper, 410
GetZSpacingTolerance
gdcm::IPPSorter, 437
Global
gdcm::Defs, 261
gdcm::Dicts, 279
gdcm::Global, 371
GlobalInstance
gdcm, 69
GrabOverlayFromPixelData
gdcm::Overlay, 556
Graphics
gdcm::Overlay, 554
GrayscaleSoftcopyPresentationStateStorageSOPClass
gdcm::MediaStorage, 487
gdcm::UIDs, 814
green
gdcm::terminal, 77
group
gdcm::SerieHelper::Rule, 659
GroupDict
gdcm::GroupDict, 374
GroupStringVector
gdcm::GroupDict, 374
GuessFromModality
gdcm::MediaStorage, 489

HSV
gdcm::PhotometricInterpretation, 578
HandleBulkData
gdcm::XMLPrinter, 977
HandleDataSet
gdcm::network::ULBasicCallback, 858
gdcm::network::ULConnectionCallback, 862
gdcm::network::ULWritingCallback, 872
HandleDescription
gdcm::XMLDictReader, 975
gdcm::XMLPrivateDictReader, 979
HandleEntry
gdcm::XMLDictReader, 975

gdcm::XMLPrivateDictReader, 979
HandleEvent
gdcm::network::ULTransitionTable, 870
HandleIODEntry
gdcm::TableReader, 772
HandleIOD
gdcm::TableReader, 772
HandleMacro
gdcm::TableReader, 772
HandleMacroEntry
gdcm::TableReader, 772
HandleMacroEntryDescription
gdcm::TableReader, 772
HandleModule
gdcm::TableReader, 772
HandleModuleEntry
gdcm::TableReader, 772
HandleModuleEntryDescription
gdcm::TableReader, 772
HandleModuleInclude
gdcm::TableReader, 772
HandleResponse
gdcm::network::ULBasicCallback, 858
gdcm::network::ULConnectionCallback, 862
gdcm::network::ULWritingCallback, 872
HangingProtocolInformationModelFIND
gdcm::UIDs, 816
HangingProtocolInformationModelMOVE
gdcm::UIDs, 816
HangingProtocolStorage
gdcm::MediaStorage, 488
gdcm::UIDs, 816
HardcopyColorImageStorageSOPClassRetired
gdcm::UIDs, 813
HardcopyGrayscaleImageStorage
gdcm::MediaStorage, 487
HardcopyGrayscaleImageStorageSOPClassRetired
gdcm::UIDs, 813
HasObserver
gdcm::Subject, 745
HemodynamicWaveformStorage
gdcm::MediaStorage, 487
gdcm::UIDs, 814
hidden
gdcm::terminal, 77

ICBM452T1FrameofReference
gdcm::UIDs, 812
ICBMSingleSubjectMRIFrameofReference
gdcm::UIDs, 812
INT12
gdcm::PixelFormat, 582
INT16
gdcm::PixelFormat, 582
INT32
gdcm::PixelFormat, 582
INT64
gdcm::PixelFormat, 582
INT8
gdcm::PixelFormat, 582
INTERFILE
gdcm::CSAHeader, 220
INVALID
gdcm::VR, 898
IODEntry
gdcm::IODEntry, 432
IODMapType
gdcm::IODs, 434
IODMapTypeConstIterator
gdcm::IODs, 434
IODName
gdcm::IODs, 434
IODs
gdcm::IODs, 434
IOD
gdcm::IOD, 430
IPPSorter
gdcm::IPPSorter, 437
Icon
gdcm::Pixmap, 589
IconDataScalarType
vtkGDCMImageReader, 910
vtkGDCMImageReader2, 916
IconImage
gdcm, 61
IconImageDataExtent
vtkGDCMImageReader, 910
vtkGDCMImageReader2, 916
IconImageFilter
gdcm::IconImageFilter, 376
IconImageGenerator
gdcm::IconImageGenerator, 378
IconNumberOfScalarComponents
vtkGDCMImageReader, 910
vtkGDCMImageReader2, 916
ID
gdcm::PresentationContext, 604
ignore_char
gdcm::ignore_char, 380
Image
gdcm::Image, 383
ImageActor
vtkImageColorViewer, 946
ImageApplyLookupTable
gdcm::ImageApplyLookupTable, 387
ImageChangePhotometricInterpretation
gdcm::ImageChangePhotometricInterpretation, 389
gdcm::ImageCodec, 403

ImageChangePlanarConfiguration
gdcm::ImageChangePlanarConfiguration, 392
ImageChangeTransferSyntax
gdcm::Bitmap, 164
gdcm::ImageChangeTransferSyntax, 395
ImageCodec
gdcm::ImageCodec, 399
ImageConverter
gdcm::ImageConverter, 405
ImageFormat
vtkGDCMImageReader, 910
vtkGDCMImageReader2, 916
ImageFragmentSplitter
gdcm::ImageFragmentSplitter, 407
ImageOrientationPatient
vtkGDCMImageReader, 910
vtkGDCMImageReader2, 916
ImageOverlayBoxSOPClassRetired
gdcm::UIDs, 813
ImagePositionPatient
vtkGDCMImageReader, 910
vtkGDCMImageReader2, 916
ImagePositionPatientOrdering
gdcm::SerieHelper, 692
ImageReader
gdcm::ImageReader, 414
ImageRegionReader
gdcm::ImageRegionReader, 418
gdcm::JPEG2000Codec, 452
gdcm::JPEGCodec, 460
gdcm::JPEGLSCodec, 465
gdcm::RLECodec, 657
ImageToImageFilter
gdcm::ImageToImageFilter, 421
ImageWriter
gdcm::ImageWriter, 423
ImplementationClassUIDSub
gdcm::network::ImplementationClassUIDSub, 424
ImplementationUIDSub
gdcm::network::ImplementationUIDSub, 425
ImplementationVersionNameSub
gdcm::network::ImplementationVersionNameSub, 426
Implicit
gdcm::TransferSyntax, 794
ImplicitVRBigEndianACRNEMA
gdcm::TransferSyntax, 795
ImplicitVRBigEndianPrivateGE
gdcm::TransferSyntax, 795
ImplicitVRLittleEndian
gdcm::TransferSyntax, 795
ImplicitVRLittleEndianDefaultTransferSyntaxforDICOM
gdcm::UIDs, 810
IncompleteLUT
gdcm::LookupTable, 475
InitFromRQ
gdcm::network::AAssociateACPDU, 82
InitOpenSSL
gdcm::OpenSSLCryptoFactory, 542
Initialize
gdcm::network::ULConnectionInfo, 864
InitializeBlueLUT
gdcm::LookupTable, 474
InitializeConnection
gdcm::ServiceClassUser, 697
gdcm::network::ULConnection, 860
InitializeDataSet
gdcm::BaseRootQuery, 150
gdcm::FindPatientRootQuery, 364
gdcm::FindStudyRootQuery, 367
gdcm::MovePatientRootQuery, 514
gdcm::MoveStudyRootQuery, 516
gdcm::WLMFindQuery, 968
InitializeGreenLUT
gdcm::LookupTable, 474
InitializeIncomingConnection
gdcm::network::ULConnection, 860
InitializeLUT
gdcm::LookupTable, 474
InitializeRTStructSet
vtkGDCMPolyDataWriter, 929
InitializeRedLUT
gdcm::LookupTable, 474
Initialized
gdcm::LookupTable, 474
Input
gdcm::BitmapToBitmapFilter, 166
Insert
gdcm::CommandDataSet, 201
gdcm::DataSet, 251
gdcm::FileMetaInformation, 346
gdcm::GroupDict, 374
InsertDataElement
gdcm::DataSet, 251
gdcm::Item, 442
InsertEntry
gdcm::Table, 769
InstallPipeline
vtkImageColorViewer, 943
InstanceAvailabilityNotificationSOPClass
gdcm::UIDs, 815
Interactor
vtkImageColorViewer, 946
InteractorStyle
vtkImageColorViewer, 946
Internal
gdcm::ApplicationEntity, 105
gdcm::Attribute, 118

gdcm::Attribute< Group, Element, TVR, VM::VM1 >, 122
gdcm::Element, 293
gdcm::Element< VR::AS, VM::VM5 >, 304
gdcm::LookupTable, 475
gdcm::UI, 803
InternalCode
gdcm::Coder, 195
gdcm::JPEG12Codec, 445
gdcm::JPEG16Codec, 448
gdcm::JPEG8Codec, 454
Internals
vtkRTStructSetProperties, 964
Invalid
gdcm::Overlay, 554
gdcm::Usage, 878
InverseRescale
gdcm::Rescaler, 652
InverseRescaleFunctionIntoBestFit
gdcm::Rescaler, 652
InvokeEvent
gdcm::Subject, 745
IS
gdcm::VR, 898
IsAETitleValid
gdcm::network::AAssociateRQPDU, 89
IsASCII2
gdcm::VR, 899
IsASCII
gdcm::VR, 899
IsBinary
gdcm::VR, 899
IsBinary2
gdcm::VR, 899
IsCompatible
gdcm::PixelFormat, 584
IsDual
gdcm::VR, 899
IsEmpty
gdcm::Bitmap, 161
gdcm::ByteValue, 177
gdcm::CSAElement, 216
gdcm::CSAHeaderDict, 224
gdcm::Curve, 232
gdcm::DataElement, 239
gdcm::DataSet, 252
gdcm::Defs, 261
gdcm::Dict, 269
gdcm::Dicts, 279
gdcm::Filename, 349
gdcm::Macros, 480
gdcm::Modules, 511
gdcm::Overlay, 556
gdcm::Preamble, 601
gdcm::PrivateDict, 617
gdcm::SegmentHelper::BasicCodedEntry, 153
IsEncapsulated
gdcm::TransferSyntax, 796
IsEncoded
gdcm::TransferSyntax, 796
IsExplicit
gdcm::TransferSyntax, 796
IsFrameEncoder
gdcm::ImageCodec, 402
gdcm::JPEG2000Codec, 451
gdcm::JPEGCodec, 459
gdcm::JPEGLSCodec, 464
gdcm::RLECodec, 656
IsGroupLength
gdcm::Tag, 777
IsGroupXX
gdcm::Tag, 777
IsIdentical
gdcm::Filename, 349
IsIllegal
gdcm::Tag, 777
IsImage
gdcm::MediaStorage, 490
IsImplicit
gdcm::TransferSyntax, 796
IsInPixelData
gdcm::Overlay, 556
IsKey
gdcm::Scanner, 664
gdcm::StrictScanner, 735
IsLastFragment
gdcm::network::AAbortPDU, 80
gdcm::network::AAssociateACPDU, 83
gdcm::network::AAssociateRJPDU, 85
gdcm::network::AAssociateRQPDU, 89
gdcm::network::AReleaseRPPDU, 107
gdcm::network::AReleaseRQPDU, 109
gdcm::network::BasePDU, 144
gdcm::network::PDataTFPDU, 564
IsLossless
gdcm::PhotometricInterpretation, 579
gdcm::TransferSyntax, 796
IsLossy
gdcm::Bitmap, 162
gdcm::ImageCodec, 402
gdcm::PhotometricInterpretation, 579
gdcm::TransferSyntax, 796
IsOdd
gdcm::VL, 890
IsPresentationContextAccepted
gdcm::ServiceClassUser, 697
IsPrintable
gdcm::ByteValue, 177

IsPrivate
gdcm::Tag, 777
IsPrivateCreator
gdcm::Tag, 778
IsPublic
gdcm::Tag, 778
IsRetired
gdcm::PhotometricInterpretation, 579
IsRowEncoder
gdcm::ImageCodec, 402
gdcm::JPEG2000Codec, 451
gdcm::JPEGCodec, 459
gdcm::JPEGLSCodec, 464
gdcm::RLECodec, 656
IsSameColorSpace
gdcm::PhotometricInterpretation, 579
IsStateSuspension
gdcm::JPEG12Codec, 446
gdcm::JPEG16Codec, 448
gdcm::JPEG8Codec, 455
gdcm::JPEGCodec, 459
IsSwap
gdcm::VR, 899
IsTransferSyntaxCompatible
gdcm::Bitmap, 162
IsUndefined
gdcm::MediaStorage, 490
gdcm::VL, 890
IsUndefinedLength
gdcm::DataElement, 240
gdcm::SequenceOfItems, 687
IsUnique
gdcm::DictEntry, 273
IsVRFile
gdcm::VR, 899
IsValid
gdcm::ApplicationEntity, 105
gdcm::BoxRegion, 170
gdcm::CodeString, 197
gdcm::DirectionCosines, 282
gdcm::FileMetaInformation, 346
gdcm::ImageCodec, 402
gdcm::JPEGCodec, 459
gdcm::LO, 470
gdcm::PixelFormat, 584
gdcm::Preamble, 601
gdcm::Region, 649
gdcm::String, 739
gdcm::TagPath, 782
gdcm::TransferSyntax, 796
gdcm::UIDGenerator, 804
gdcm::UUIDGenerator, 882
gdcm::VM, 895
gdcm::VR, 899
IsZero
gdcm::Overlay, 556
ItFileSetHt
gdcm::SerieHelper, 692
Item
gdcm::Item, 441
ItemVector
gdcm::SequenceOfItems, 686
Items
gdcm::SequenceOfItems, 689
Iterator
gdcm::CSAHeaderDict, 224
gdcm::DataSet, 249
gdcm::Dict, 268
gdcm::SequenceOfFragments, 680
gdcm::SequenceOfItems, 686
iterator
gdcm::CodeString, 197
gdcm::LO, 470
gdcm::String, 738

JPEG12Codec
gdcm::JPEG12Codec, 445
JPEG16Codec
gdcm::JPEG16Codec, 447
JPEG2000
gdcm::TransferSyntax, 795
JPEG2000_COMPRESSION
vtkGDCMImageWriter, 919
JPEG2000Codec
gdcm::JPEG2000Codec, 450
JPEG2000ImageCompression
gdcm::UIDs, 811
JPEG2000ImageCompressionLosslessOnly
gdcm::UIDs, 811
JPEG2000Lossless
gdcm::TransferSyntax, 795
JPEG2000Part2
gdcm::TransferSyntax, 795
JPEG2000Part2Lossless
gdcm::TransferSyntax, 795
JPEG2000Part2MulticomponentImageCompression
gdcm::UIDs, 811
JPEG2000Part2MulticomponentImageCompression↩
LosslessOnly
gdcm::UIDs, 811
JPEG8Codec
gdcm::JPEG8Codec, 454
JPEG_COMPRESSION
vtkGDCMImageWriter, 919
JPEGBaselineProcess1
gdcm::TransferSyntax, 795
JPEGBaselineProcess1DefaultTransferSyntaxforLossyJ↩
PEG8BitImageCompression

gdcm::UIDs, 810
JPEGCodec
gdcm::JPEGCodec, 457
JPEGExtendedHierarchicalProcess1618Retired
gdcm::UIDs, 811
JPEGExtendedHierarchicalProcess1719Retired
gdcm::UIDs, 811
JPEGExtendedProcess24DefaultTransferSyntaxfor↩ LossyJPEG12BitImageCompressionProcess4only
gdcm::UIDs, 810
JPEGExtendedProcess2_4
gdcm::TransferSyntax, 795
JPEGExtendedProcess35Retired
gdcm::UIDs, 810
JPEGExtendedProcess3_5
gdcm::TransferSyntax, 795
JPEGFullProgressionHierarchicalProcess2426Retired
gdcm::UIDs, 811
JPEGFullProgressionHierarchicalProcess2527Retired
gdcm::UIDs, 811
JPEGFullProgressionNonHierarchicalProcess1012↩ Retired
gdcm::UIDs, 811
JPEGFullProgressionNonHierarchicalProcess1113↩ Retired
gdcm::UIDs, 811
JPEGFullProgressionProcess10_12
gdcm::TransferSyntax, 795
JPEGLS_COMPRESSION
vtkGDCMImageWriter, 919
JPEGLSCodec
gdcm::JPEGLSCodec, 463
JPEGLSLossless
gdcm::TransferSyntax, 795
JPEGLSLosslessImageCompression
gdcm::UIDs, 811
JPEGLSLossyNearLosslessImageCompression
gdcm::UIDs, 811
JPEGLSNearLossless
gdcm::TransferSyntax, 795
JPEGLosslessHierarchicalProcess28Retired
gdcm::UIDs, 811
JPEGLosslessHierarchicalProcess29Retired
gdcm::UIDs, 811
JPEGLosslessNonHierarchicalFirstOrderPrediction↩ Process14SelectionValue1DefaultTransfer↩ SyntaxforLosslessJPEGImageCompression
gdcm::UIDs, 811
JPEGLosslessNonHierarchicalProcess14
gdcm::UIDs, 811
JPEGLosslessNonHierarchicalProcess15Retired
gdcm::UIDs, 811
JPEGLosslessProcess14
gdcm::TransferSyntax, 795
JPEGLosslessProcess14_1
gdcm::TransferSyntax, 795
JPEGSpectralSelectionHierarchicalProcess2022Retired
gdcm::UIDs, 811
JPEGSpectralSelectionHierarchicalProcess2123Retired
gdcm::UIDs, 811
JPEGSpectralSelectionNonHierarchicalProcess68Retired
gdcm::UIDs, 810
JPEGSpectralSelectionNonHierarchicalProcess79Retired
gdcm::UIDs, 810
JPEGSpectralSelectionProcess6_8
gdcm::TransferSyntax, 795
JPIPReferenced
gdcm::TransferSyntax, 795
gdcm::UIDs, 811
JPIPReferencedDeflate
gdcm::UIDs, 811
JSON
gdcm::JSON, 465
Join
gdcm::Filename, 349
JunkAfterDocElementError
gdcm::Parser, 561

KAKADUCodec
gdcm::KAKADUCodec, 467
KeyField
gdcm::CSAElement, 217
KeyObjectSelectionDocument
gdcm::MediaStorage, 488
KeyObjectSelectionDocumentStorage
gdcm::UIDs, 815
KeyValuePairArrayType
gdcm::CompositeNetworkFunctions, 204
KeyValuePairType
gdcm::CompositeNetworkFunctions, 204

LD_ALL
gdcm, 63
LD_NOSEQ
gdcm, 63
LD_NOSHADOWSEQ
gdcm, 63
LD_NOSHADOW
gdcm, 63
LINE
gdcm::MeshPrimitive, 497
LOADBULKDATA
gdcm::XMLPrinter, 977
LOComp
gdcm, 61
LTComp
gdcm, 61
LUTPtr
gdcm::Bitmap, 159

gdcm::ImageCodec, 399
LUT
gdcm::Bitmap, 164
gdcm::ImageCodec, 403
LeadECGWaveformStorage
gdcm::MediaStorage, 487
Level
vtkImageMapToWindowLevelColors2, 952
ListCharSets
gdcm::QueryFactory, 629
LittleEndian
gdcm::SwapCode, 761
LO
gdcm::LO, 470
gdcm::VR, 898
Load
gdcm::Directory, 285
LoadDefault
gdcm::CSAHeaderDict, 224
gdcm::Dict, 269
gdcm::PrivateDict, 617
LoadDefaults
gdcm::Defs, 261
gdcm::Dicts, 279
LoadFromDataElement
gdcm::CSAHeader, 222
gdcm::PDBHeader, 569
LoadFromFile
gdcm::Defs, 261
LoadIconImage
vtkGDCMImageReader, 910
vtkGDCMImageReader2, 916
LoadImageFromFiles
gdcm::DirectoryHelper, 287
LoadOverlays
vtkGDCMImageReader, 910
vtkGDCMImageReader2, 917
LoadResourcesFiles
gdcm::Global, 372
LoadSingleFile
vtkGDCMImageReader, 907
vtkGDCMImageReader2, 914
Locate
gdcm::Global, 372
LodModeType
gdcm, 63
LookupTable
gdcm::LookupTable, 473
vtkImageMapToColors16, 949
LookupTableType
gdcm::LookupTable, 473
LossyFlag
gdcm::Bitmap, 164
gdcm::ImageCodec, 403
vtkGDCMImageReader, 910
vtkGDCMImageReader2, 917
LT
gdcm::VR, 898

m_ConstMemberFunction
gdcm::MemberCommand, 494
m_MemberFunction
gdcm::MemberCommand, 494
gdcm::SimpleMemberCommand, 704
m_This
gdcm::MemberCommand, 494
gdcm::SimpleMemberCommand, 704
m_char
gdcm::ignore_char, 380
MAGNIFIED
gdcm::Spacing, 717
MANUAL
gdcm::Segment, 668
mAction
gdcm::network::Transition, 800
mConnection
gdcm::network::ULConnectionManager, 868
MD5
gdcm::MD5, 482
MD5DataImagesType
gdcm::Testing, 784
MD5MetaImagesType
vtkGDCMTesting, 932
mDataSet
gdcm::BaseQuery, 148
mElementOffsets
gdcm::StreamImageWriter, 729
mElementOffsets1
gdcm::StreamImageWriter, 729
mEnd
gdcm::network::Transition, 800
mHelpDescription
gdcm::BaseQuery, 148
gdcm::BaseRootQuery, 151
mImage
gdcm::BaseRootQuery, 151
mImplicit
gdcm::network::ULConnectionCallback, 863
MONOCHROME1
gdcm::PhotometricInterpretation, 578
MONOCHROME2
gdcm::PhotometricInterpretation, 578
MPEG2MainProfile
gdcm::TransferSyntax, 795
MPEG2MainProfileHighLevel
gdcm::TransferSyntax, 795
MPEG2MainProfileMainLevel
gdcm::UIDs, 811

MPEG4AVCH264BDcompatibleHighProfileLevel4_1
gdcm::TransferSyntax, 795
MPEG4AVCH264HighProfileLevel4_1
gdcm::TransferSyntax, 795
MPType
gdcm::MeshPrimitive, 497
MPType_END
gdcm::MeshPrimitive, 497
mPatient
gdcm::BaseRootQuery, 151
MRImageStorage
gdcm::MediaStorage, 486
gdcm::UIDs, 813
MRSpectroscopyStorage
gdcm::MediaStorage, 486
gdcm::UIDs, 813
mRootType
gdcm::BaseRootQuery, 151
MS_END
gdcm::MediaStorage, 488
MSType
gdcm::MediaStorage, 486
mSecondaryConnection
gdcm::network::ULConnectionManager, 868
mSeries
gdcm::BaseRootQuery, 151
mSopInstanceUID
gdcm::BaseQuery, 148
mStudy
gdcm::BaseRootQuery, 151
mTransitions
gdcm::network::ULConnectionManager, 868
mWriter
gdcm::StreamImageWriter, 729
mXMax
gdcm::StreamImageWriter, 729
mXMin
gdcm::StreamImageWriter, 729
mYMax
gdcm::StreamImageWriter, 729
mYMin
gdcm::StreamImageWriter, 729
mZMax
gdcm::StreamImageWriter, 729
mZMin
gdcm::StreamImageWriter, 729
Macro
gdcm::Macro, 477
MacroEntry
gdcm, 61
Macros
gdcm::Macros, 479
magenta
gdcm::terminal, 77
MakeDirectory
gdcm::System, 767
MakeNew
gdcm::network::Transition, 800
MakeObject
gdcm::AnonymizeEvent, 95
gdcm::DataEvent, 245
gdcm::DataSetEvent, 255
gdcm::Event, 316
gdcm::FileNameEvent, 352
gdcm::ProgressEvent, 622
MammographyCADSRStorage
gdcm::UIDs, 815
MammographyCADSR
gdcm::MediaStorage, 488
Mandatory
gdcm::Usage, 878
MapCSAHeaderDictEntry
gdcm::CSAHeaderDict, 224
MapDictEntry
gdcm::Dict, 268
MapIODEntry
gdcm::IOD, 430
MapModuleEntry
gdcm::Macro, 477
gdcm::Module, 506
MapScalarsThroughTable2
vtkLookupTable16, 959
MapTableEntry
gdcm::Table, 769
MappingType
gdcm::Scanner, 662
gdcm::StrictScanner, 732
MaxLength
gdcm::ApplicationEntity, 105
gdcm::PersonName, 575
MaxNumberOfComponents
gdcm::ApplicationEntity, 105
gdcm::PersonName, 575
MaxPrintLength
gdcm::Printer, 616
MaximumLengthSub
gdcm::network::MaximumLengthSub, 481
MediaCreationManagementSOPClassUID
gdcm::UIDs, 813
MediaStorage
gdcm::MediaStorage, 489
MediaStorageDataFilesType
gdcm::Testing, 784
MediaStorageDirectoryStorage
gdcm::MediaStorage, 486
gdcm::UIDs, 811
MedicalImageProperties
vtkGDCMImageReader, 910

vtkGDCMPolyDataReader, 927
vtkGDCMPolyDataWriter, 930
MemberCommand
gdcm::MemberCommand, 493
MeshPrimitive
gdcm::MeshPrimitive, 497
MessageID
gdcm::network::CEchoRQ, 183
MetaInformationTS
gdcm::FileMetaInformation, 347
ModalityPerformedProcedureStepCreateQuery
gdcm::ModalityPerformedProcedureStepCreate↩
Query, 501
ModalityPerformedProcedureStepNotificationSOPClass
gdcm::UIDs, 812
ModalityPerformedProcedureStepRetrieveSOPClass
gdcm::UIDs, 812
ModalityPerformedProcedureStepSOPClass
gdcm::MediaStorage, 488
gdcm::UIDs, 812
ModalityPerformedProcedureStepSetQuery
gdcm::ModalityPerformedProcedureStepSetQuery,
503
ModalityWorklistInformationModelFIND
gdcm::UIDs, 815
Mode
gdcm::terminal, 77
Module
gdcm::Module, 506
ModuleEntry
gdcm::ModuleEntry, 509
ModuleMapType
gdcm::Macros, 479
gdcm::Modules, 511
Modules
gdcm::Modules, 511
MovePatientRootQuery
gdcm::MovePatientRootQuery, 513
MoveStudyRootQuery
gdcm::MoveStudyRootQuery, 516
mspFile
gdcm::StreamImageWriter, 729
MultiframeGrayscaleByteSecondaryCaptureImage↩
Storage
gdcm::MediaStorage, 487
gdcm::UIDs, 813
MultiframeGrayscaleWordSecondaryCaptureImage↩
Storage
gdcm::MediaStorage, 487
gdcm::UIDs, 813
MultiframeSingleBitSecondaryCaptureImageStorage
gdcm::MediaStorage, 487
gdcm::UIDs, 813
MultiframeTrueColorSecondaryCaptureImageStorage
gdcm::MediaStorage, 487
gdcm::UIDs, 813

N_ACTION_RSP
gdcm::network::DIMSE, 281
N_ACTION_RQ
gdcm::network::DIMSE, 281
N_CREATE_RSP
gdcm::network::DIMSE, 281
N_CREATE_RQ
gdcm::network::DIMSE, 281
N_DELETE_RSP
gdcm::network::DIMSE, 281
N_DELETE_RQ
gdcm::network::DIMSE, 281
N_EVENT_REPORT_RSP
gdcm::network::DIMSE, 281
N_EVENT_REPORT_RQ
gdcm::network::DIMSE, 280
N_GET_RSP
gdcm::network::DIMSE, 281
N_GET_RQ
gdcm::network::DIMSE, 281
N_SET_RSP
gdcm::network::DIMSE, 281
N_SET_RQ
gdcm::network::DIMSE, 281
NAction
gdcm::NormalizedNetworkFunctions, 536
NCreate
gdcm::NormalizedNetworkFunctions, 536
NDelete
gdcm::NormalizedNetworkFunctions, 536
NEventReport
gdcm::NormalizedNetworkFunctions, 536
NGet
gdcm::NormalizedNetworkFunctions, 536
NO_COMPRESSION
vtkGDCMImageWriter, 919
NOMAGIC
gdcm::CSAHeader, 220
NSet
gdcm::NormalizedNetworkFunctions, 536
Name
gdcm::ModuleEntry, 510
NameField
gdcm::CSAElement, 218
gdcm::PDBElement, 567
NeedByteSwap
gdcm::Bitmap, 164
gdcm::ImageCodec, 403
NeedOverlayCleanup
gdcm::ImageCodec, 403
NegociatedType

gdcm::TransferSyntax, 794
NestedMacroEntries
gdcm, 61
NestedModuleEntries
gdcm::NestedModuleEntries, 527
New
gdcm::Anonymizer, 99
gdcm::FileChangeTransferSyntax, 334
gdcm::FileStreamer, 360
gdcm::MemberCommand, 494
gdcm::Scanner, 664
gdcm::SequenceOfFragments, 682
gdcm::SequenceOfItems, 688
gdcm::ServiceClassUser, 697
gdcm::SimpleMemberCommand, 704
gdcm::StrictScanner, 735
vtkGDCMImageReader, 907
vtkGDCMImageReader2, 914
vtkGDCMImageWriter, 920
vtkGDCMMedicalImageProperties, 924
vtkGDCMPolyDataReader, 926
vtkGDCMPolyDataWriter, 929
vtkGDCMTesting, 933
vtkGDCMThreadedImageReader, 935
vtkGDCMThreadedImageReader2, 937
vtkImageColorViewer, 943
vtkImageMapToColors16, 948
vtkImageMapToWindowLevelColors2, 951
vtkImagePlanarComponentsToComponents, 953
vtkImageRGBToYBR, 955
vtkImageYBRToRGB, 957
vtkLookupTable16, 959
vtkRTStructSetProperties, 963
NO
gdcm::Surface, 749
NoElementsError
gdcm::Parser, 561
NoError
gdcm::Parser, 561
NoMemoryError
gdcm::Parser, 561
NoObject
gdcm::MediaStorage, 488
NoOfItemsField
gdcm::CSAElement, 218
Normalize
gdcm::DirectionCosines, 282
NuclearMedicineImageStorage
gdcm::MediaStorage, 487
gdcm::UIDs, 814
NuclearMedicineImageStorageRetired
gdcm::MediaStorage, 486
gdcm::UIDs, 813
NumberOfDimensions
gdcm::Bitmap, 164
gdcm::ImageCodec, 404
NumberOfIconImages
vtkGDCMImageReader, 910
vtkGDCMImageReader2, 917
NumberOfOverlays
vtkGDCMImageReader, 910
vtkGDCMImageReader2, 917
NumberOfSurfaces
gdcm::SurfaceWriter, 760

OB_OW
gdcm::VR, 898
OBLIQUE
gdcm::Orientation, 550
OPENSSLP7
gdcm::CryptoFactory, 210
OPENSSL
gdcm::CryptoFactory, 210
OB
gdcm::VR, 898
Object
gdcm::Object, 540
ObjectEnd
gdcm::MediaStorage, 488
ObjectType
gdcm::MediaStorage, 488
OD
gdcm::VR, 898
OF
gdcm::VR, 898
Ofstream
gdcm::Writer, 973
OnlyUUID
gdcm::XMLPrinter, 977
op
gdcm::SerieHelper::Rule, 659
OpenSSLCryptoFactory
gdcm::OpenSSLCryptoFactory, 542
OpenSSLCryptographicMessageSyntax
gdcm::OpenSSLCryptographicMessageSyntax, 544
OpenSSLP7CryptoFactory
gdcm::OpenSSLP7CryptoFactory, 546
OpenSSLP7CryptographicMessageSyntax
gdcm::OpenSSLP7CryptographicMessageSyntax, 548
operator const char ∗
gdcm::ConstCharWrapper, 207
gdcm::Filename, 349
gdcm::String, 739
operator const double ∗
gdcm::DirectionCosines, 283
operator const std::vector< char > &
gdcm::ByteValue, 177

operator MSType
gdcm::MediaStorage, 490
operator ObjectType *
gdcm::SmartPointer, 708
operator PIType
gdcm::PhotometricInterpretation, 579
operator ScalarType
gdcm::PixelFormat, 584
operator SwapCode::SwapCodeType
gdcm::SwapCode, 762
operator TSType
gdcm::TransferSyntax, 797
gdcm::UIDs, 824
operator TypeType
gdcm::Type, 802
operator uint32_t
gdcm::VL, 890
operator UsageType
gdcm::Usage, 879
operator VMType
gdcm::VM, 895
operator VRType
gdcm::VR, 899
operator!=
gdcm, 63
gdcm::Attribute, 116
gdcm::Attribute< Group, Element, TVR, VM::VM1 >, 121
gdcm::CodeString, 198
gdcm::PixelFormat, 584
gdcm::Tag, 778
operator<
gdcm::Attribute, 116
gdcm::Attribute< Group, Element, TVR, VM::VM1 >, 121
gdcm::CSAElement, 217
gdcm::CSAHeaderDictEntry, 226
gdcm::DataElement, 240
gdcm::PrivateTag, 619
gdcm::Tag, 778
operator<<
gdcm, 63–68
gdcm::BasicOffsetTable, 156
gdcm::CSAElement, 217
gdcm::CSAHeader, 222
gdcm::CSAHeaderDict, 224
gdcm::CSAHeaderDictEntry, 226
gdcm::CodeString, 198
gdcm::CommandDataSet, 202
gdcm::DataElement, 242
gdcm::DataSet, 253
gdcm::Dict, 269
gdcm::DictEntry, 274
gdcm::Dicts, 279
gdcm::Directory, 286
gdcm::File, 328
gdcm::FileMetaInformation, 347
gdcm::FileSet, 357
gdcm::Fragment, 370
gdcm::Global, 373
gdcm::GroupDict, 374
gdcm::IODEntry, 433
gdcm::IODs, 435
gdcm::IOD, 431
gdcm::Item, 442
gdcm::Macro, 478
gdcm::Macros, 480
gdcm::MediaStorage, 491
gdcm::Module, 507
gdcm::ModuleEntry, 510
gdcm::Modules, 512
gdcm::NestedModuleEntries, 527
gdcm::Object, 541
gdcm::Orientation, 551
gdcm::PDBElement, 567
gdcm::PDBHeader, 569
gdcm::PhotometricInterpretation, 579
gdcm::PixelFormat, 585
gdcm::Preamble, 601
gdcm::PrivateDict, 617
gdcm::PrivateTag, 620
gdcm::Scanner, 665
gdcm::Sorter, 715
gdcm::StrictScanner, 735
gdcm::SwapCode, 762
gdcm::Table, 770
gdcm::Tag, 781
gdcm::TransferSyntax, 797
gdcm::Type, 802
gdcm::UI, 803
gdcm::Usage, 879
gdcm::Version, 888
gdcm::VL, 891
gdcm::VM, 895
gdcm::VR, 900
operator<=
gdcm::Tag, 778
operator>>
gdcm, 68, 69
gdcm::Tag, 781
operator*
gdcm::SmartPointer, 709
operator()
gdcm::DataSet, 252
gdcm::Scanner::ltstr, 476
gdcm::StrictScanner::ltstr, 476
operator++
gdcm::VL, 890

operator+=
gdcm::VL, 890
operator->
gdcm::SmartPointer, 709
operator=
gdcm::BoxRegion, 170
gdcm::ByteValue, 177
gdcm::CSAElement, 217
gdcm::DataElement, 240
gdcm::DataSet, 252
gdcm::Element< TVR, VM::VM1_n >, 297
gdcm::Object, 541
gdcm::Overlay, 556
gdcm::ParseException, 559
gdcm::Preamble, 601
gdcm::SequenceOfItems, 688
gdcm::SmartPointer, 709
gdcm::Tag, 778
gdcm::network::UserInformation, 881
operator==
gdcm, 68
gdcm::Attribute, 116
gdcm::Attribute< Group, Element, TVR, VM::VM1 >, 121
gdcm::ByteValue, 177
gdcm::CSAElement, 217
gdcm::CodeString, 198
gdcm::DataElement, 240
gdcm::PDBElement, 566
gdcm::PixelFormat, 584
gdcm::PresentationContext, 603
gdcm::SequenceOfFragments, 682
gdcm::SequenceOfItems, 688
gdcm::Tag, 778
gdcm::Value, 886
gdcm::network::AbstractSyntax, 92
gdcm::network::PresentationContextRQ, 610
gdcm::network::TransferSyntaxSub, 798
operator[]
gdcm::Attribute, 116, 117
gdcm::Attribute< Group, Element, TVR, VM::VM1_n >, 128
gdcm::DataSet, 252
gdcm::Element, 293
gdcm::Element< TVR, VM::VM1_n >, 297
gdcm::Tag, 779
OphthalmicPhotography16BitImageStorage
gdcm::UIDs, 814
OphthalmicPhotography8BitImageStorage
gdcm::MediaStorage, 488
gdcm::UIDs, 814
OphthalmicTomographyImageStorage
gdcm::MediaStorage, 488
gdcm::UIDs, 814
OrderFileList
gdcm::SerieHelper, 692
Orientation
gdcm::Orientation, 550
OrientationType
gdcm::Orientation, 550
Output
gdcm::BitmapToBitmapFilter, 166
OutputFormat
vtkImageMapToColors16, 949
OutputTypes
gdcm::DictConverter, 270
Overlay
gdcm::Overlay, 554
OverlayImageActor
vtkImageColorViewer, 946
OverlayType
gdcm::Overlay, 554
Overlays
gdcm::Pixmap, 589
OW
gdcm::VR, 898

PALETTE_COLOR
gdcm::PhotometricInterpretation, 578
PDBElement
gdcm::PDBElement, 566
PDBHeader
gdcm::PDBHeader, 568
PDFCodec
gdcm::PDFCodec, 571
PDataTFPDU
gdcm::network::PDataTFPDU, 564
PDF
gdcm::MediaStorage, 488
PETImageStorage
gdcm::MediaStorage, 487
PGXCodec
gdcm::PGXCodec, 576
PHILIPS
gdcm::Dicts, 278
PI_END
gdcm::PhotometricInterpretation, 578
PIType
gdcm::PhotometricInterpretation, 578
PNComp
gdcm, 61
PNMCodec
gdcm::PNMCodec, 599
POINTS
gdcm::Surface, 749
PVRGCodec
gdcm::PVRGCodec, 624
Pack

gdcm::Unpacker12Bits, 877
Padding
gdcm::ApplicationEntity, 105
gdcm::PersonName, 575
Parent
gdcm::Element< TVR, VM::VM1_2 >, 295
gdcm::Element< TVR, VM::VM2_2n >, 299
gdcm::Element< TVR, VM::VM2_n >, 301
gdcm::Element< TVR, VM::VM3_3n >, 302
gdcm::Element< TVR, VM::VM3_n >, 304
Parse
gdcm::Parser, 562
ParseBuffer
gdcm::Parser, 562
ParseCertificateFile
gdcm::CAPICryptographicMessageSyntax, 182
gdcm::CryptographicMessageSyntax, 213
gdcm::OpenSSLCryptographicMessageSyntax, 544
gdcm::OpenSSLP7CryptographicMessageSyntax, 548
ParseDateTime
gdcm::System, 767
ParseDump
gdcm::ASN1, 111
ParseDumpFile
gdcm::ASN1, 111
ParseException
gdcm::ParseException, 559
ParseKeyFile
gdcm::CAPICryptographicMessageSyntax, 182
gdcm::CryptographicMessageSyntax, 213
gdcm::OpenSSLCryptographicMessageSyntax, 544
gdcm::OpenSSLP7CryptographicMessageSyntax, 548
Parser
gdcm::Parser, 562
PassAlphaToOutput
vtkImageMapToColors16, 949
Patient
gdcm::Patient, 563
PatientRootQueryRetrieveInformationModelFIND
gdcm::UIDs, 815
PatientRootQueryRetrieveInformationModelGET
gdcm::UIDs, 815
PatientRootQueryRetrieveInformationModelMOVE
gdcm::UIDs, 815
PatientStudyOnlyQueryRetrieveInformationModelFIND↩ Retired
gdcm::UIDs, 815
PatientStudyOnlyQueryRetrieveInformationModelGET↩ Retired
gdcm::UIDs, 815
PatientStudyOnlyQueryRetrieveInformationModelMOV↩ ERetired
gdcm::UIDs, 815
PerformAction
gdcm::network::ULAction, 828
gdcm::network::ULActionAA1, 829
gdcm::network::ULActionAA2, 830
gdcm::network::ULActionAA3, 831
gdcm::network::ULActionAA4, 832
gdcm::network::ULActionAA5, 833
gdcm::network::ULActionAA6, 834
gdcm::network::ULActionAA7, 835
gdcm::network::ULActionAA8, 836
gdcm::network::ULActionAE1, 837
gdcm::network::ULActionAE2, 838
gdcm::network::ULActionAE3, 839
gdcm::network::ULActionAE4, 840
gdcm::network::ULActionAE5, 841
gdcm::network::ULActionAE6, 842
gdcm::network::ULActionAE7, 843
gdcm::network::ULActionAE8, 844
gdcm::network::ULActionAR1, 845
gdcm::network::ULActionAR10, 846
gdcm::network::ULActionAR2, 847
gdcm::network::ULActionAR3, 848
gdcm::network::ULActionAR4, 849
gdcm::network::ULActionAR5, 850
gdcm::network::ULActionAR6, 851
gdcm::network::ULActionAR7, 852
gdcm::network::ULActionAR8, 853
gdcm::network::ULActionAR9, 854
gdcm::network::ULActionDT1, 855
gdcm::network::ULActionDT2, 856
PF
gdcm::Bitmap, 164
gdcm::ImageCodec, 404
Philips3D
gdcm::MediaStorage, 487
PhilipsPrivateMRSyntheticImageStorage
gdcm::MediaStorage, 488
PhotometricInterpretation
gdcm::PhotometricInterpretation, 579
PI
gdcm::Bitmap, 164
gdcm::ImageCodec, 404
PixelData
gdcm::Bitmap, 164
gdcm::PixmapReader, 592
gdcm::PixmapWriter, 597
PixelFormat
gdcm::PixelFormat, 582
Pixmap
gdcm::Pixmap, 588
PixmapReader
gdcm::Bitmap, 164
gdcm::PixmapReader, 591

PixmapToPixmapFilter
gdcm::PixmapToPixmapFilter, 594
PixmapWriter
gdcm::PixmapWriter, 596
PlanarConfiguration
gdcm::Bitmap, 164
gdcm::ImageCodec, 404
vtkGDCMImageReader, 910
vtkGDCMImageReader2, 917
PN
gdcm::VR, 898
pointer
gdcm::CodeString, 197
gdcm::LO, 470
gdcm::String, 738
PositronEmissionTomographyImageStorage
gdcm::UIDs, 815
Preamble
gdcm::Preamble, 601
PrepareWrite
gdcm::PixmapWriter, 597
gdcm::SegmentWriter, 677
gdcm::SurfaceWriter, 760
PrepareWritePointMacro
gdcm::SurfaceWriter, 760
Prepend
gdcm::Global, 372
PresentationContext
gdcm::PresentationContext, 603
PresentationContextAC
gdcm::network::PresentationContextAC, 605
PresentationContextArrayType
gdcm::PresentationContextGenerator, 607
gdcm::network::AAssociateRQPDU, 88
PresentationContextGenerator
gdcm::PresentationContextGenerator, 607
PresentationContextRQ
gdcm::network::PresentationContextRQ, 609
PresentationDataValue
gdcm::network::PresentationDataValue, 611
PresentationLUTSOPClass
gdcm::UIDs, 813
PrettyPrintOff
gdcm::JSON, 466
PrettyPrintOn
gdcm::JSON, 466
PrimitiveData
gdcm::MeshPrimitive, 498
PrimitiveType
gdcm::MeshPrimitive, 498
PrimitivesData
gdcm::MeshPrimitive, 497
Print
gdcm::ApplicationEntity, 105
gdcm::Attribute, 117
gdcm::Attribute< Group, Element, TVR, VM::VM1 >, 121
gdcm::Attribute< Group, Element, TVR, VM::VM1_n >, 128
gdcm::BaseQuery, 147
gdcm::Bitmap, 162
gdcm::BoxRegion, 170
gdcm::ByteValue, 177
gdcm::CSAHeader, 222
gdcm::Curve, 232
gdcm::DataSet, 252
gdcm::DictPrinter, 276
gdcm::DirectionCosines, 283
gdcm::Directory, 285
gdcm::Element, 293
gdcm::Element< TVR, VM::VM1_n >, 297
gdcm::Element< VR::AS, VM::VM5 >, 304
gdcm::Event, 316
gdcm::Image, 384
gdcm::LookupTable, 474
gdcm::Object, 541
gdcm::Orientation, 550
gdcm::Overlay, 556
gdcm::PDBHeader, 569
gdcm::PersonName, 574
gdcm::PixelFormat, 584
gdcm::Pixmap, 588
gdcm::Preamble, 601
gdcm::PresentationContext, 603
gdcm::Printer, 614
gdcm::Region, 649
gdcm::Scanner, 665
gdcm::SegmentedPaletteColorLookupTable, 672
gdcm::SequenceOfFragments, 682
gdcm::SequenceOfItems, 688
gdcm::Sorter, 714
gdcm::StrictScanner, 735
gdcm::TagPath, 782
gdcm::Testing, 788
gdcm::Version, 888
gdcm::XMLPrinter, 977
gdcm::network::AAbortPDU, 80
gdcm::network::AAssociateACPDU, 83
gdcm::network::AAssociateRJPDU, 85
gdcm::network::AAssociateRQPDU, 89
gdcm::network::AReleaseRPPDU, 107
gdcm::network::AReleaseRQPDU, 109
gdcm::network::AbstractSyntax, 92
gdcm::network::ApplicationContext, 103
gdcm::network::AsynchronousOperationsWindowSub, 112
gdcm::network::BasePDU, 144
gdcm::network::ImplementationClassUIDSub, 424

gdcm::network::ImplementationVersionNameSub, 426
gdcm::network::MaximumLengthSub, 481
gdcm::network::PDataTFPDU, 564
gdcm::network::PresentationContextAC, 605
gdcm::network::PresentationContextRQ, 610
gdcm::network::PresentationDataValue, 611
gdcm::network::RoleSelectionSub, 658
gdcm::network::SOPClassExtendedNegociationSub, 710
gdcm::network::ServiceClassApplicationInformation, 694
gdcm::network::TransferSyntaxSub, 798
gdcm::network::UserInformation, 881
PrintASCIIXML
gdcm::ByteValue, 178
PrintASCII
gdcm::ByteValue, 178
PrintAsContinuousString
gdcm::Tag, 779
PrintAsContinuousUpperCaseString
gdcm::Tag, 779
PrintAsPipeSeparatedString
gdcm::Tag, 779
PrintDataElement
gdcm::Printer, 615
gdcm::XMLPrinter, 977
PrintDataElement2
gdcm::DictPrinter, 276
PrintDataSet
gdcm::Printer, 615
gdcm::XMLPrinter, 977
PrintDataSet2
gdcm::DictPrinter, 276
PrintGroupLength
gdcm::ByteValue, 178
PrintHex
gdcm::ByteValue, 178
PrintHexXML
gdcm::ByteValue, 178
PrintJobSOPClass
gdcm::UIDs, 812
PrintPNXML
gdcm::ByteValue, 178
PrintQueueManagementSOPClassRetired
gdcm::UIDs, 813
PrintQueueSOPInstanceRetired
gdcm::UIDs, 813
PrintSelf
vtkGDCMImageReader, 907
vtkGDCMImageReader2, 914
vtkGDCMImageWriter, 920
vtkGDCMMedicalImageProperties, 924
vtkGDCMPolyDataReader, 926
vtkGDCMPolyDataWriter, 929
vtkGDCMTesting, 933
vtkGDCMThreadedImageReader, 935
vtkGDCMThreadedImageReader2, 938
vtkImageColorViewer, 943
vtkImageMapToColors16, 948
vtkImageMapToWindowLevelColors2, 951
vtkImagePlanarComponentsToComponents, 953
vtkImageRGBToYBR, 955
vtkImageYBRToRGB, 957
vtkLookupTable16, 959
vtkRTStructSetProperties, 963
PrintSQ
gdcm::Printer, 615
gdcm::XMLPrinter, 977
PrintStyle
gdcm::Printer, 616
gdcm::XMLPrinter, 977
PrintStyles
gdcm::Printer, 614
gdcm::XMLPrinter, 977
PrintTable
gdcm::network::ULTransitionTable, 870
PrintXML
gdcm::PrivateDict, 617
Printer
gdcm::Printer, 614
PrinterConfigurationRetrievalSOPClass
gdcm::UIDs, 812
PrinterConfigurationRetrievalSOPInstance
gdcm::UIDs, 813
PrinterSOPClass
gdcm::UIDs, 812
PrinterSOPInstance
gdcm::UIDs, 812
PrivateDict
gdcm::PrivateDict, 617
PrivateTag
gdcm::PrivateTag, 619
ProceduralEventLoggingSOPClass
gdcm::UIDs, 812
ProceduralEventLoggingSOPInstance
gdcm::UIDs, 812
ProcedureLogStorage
gdcm::UIDs, 815
Process
gdcm::Parser, 562
ProcessDataSet
gdcm::FileExplicitFilter, 341
ProcessPublicTag
gdcm::Scanner, 665
gdcm::StrictScanner, 735
ProcessRequest
vtkGDCMImageReader2, 914

ProduceCharacterSetDataElement
gdcm::QueryFactory, 629
ProduceQuery
gdcm::QueryFactory, 630
ProductCharacteristicsQuerySOPClass
gdcm::UIDs, 816
ProgressEvent
gdcm::ProgressEvent, 622
PropertyCategory
gdcm::Segment, 670
PropertyType
gdcm::Segment, 670
PseudoColorSoftcopyPresentationStateStorageSOP↩
Class
gdcm::UIDs, 814
PullPrintRequestSOPClassRetired
gdcm::UIDs, 813
PullStoredPrintManagementMetaSOPClassRetired
gdcm::UIDs, 813
Push
gdcm::TagPath, 783
PushBackFile
vtkGDCMMedicalImageProperties, 924
PythonFilter
gdcm::PythonFilter, 626

Quality
gdcm::JPEGCodec, 460
QueryFactory
gdcm::BaseQuery, 148
gdcm::BaseRootQuery, 151
gdcm::FindPatientRootQuery, 365
gdcm::FindStudyRootQuery, 367
gdcm::ModalityPerformedProcedureStepCreate↩
Query, 501
gdcm::ModalityPerformedProcedureStepSetQuery,
503
gdcm::MovePatientRootQuery, 514
gdcm::MoveStudyRootQuery, 517
gdcm::WLMFindQuery, 968

RAWCodec
gdcm::RAWCodec, 639
README.txt, 1254
RED
gdcm::LookupTable, 473
RFC2557MIMEencapsulation
gdcm::UIDs, 811
RGB2YBR
gdcm::ImageChangePhotometricInterpretation, 389
RGBPixelsToRGBPlanes
gdcm::ImageChangePlanarConfiguration, 392
RGBPlanesToRGBPixels
gdcm::ImageChangePlanarConfiguration, 392
RGBToRecommendedDisplayCIELab
gdcm::SurfaceHelper, 755
RGBToRecommendedDisplayGrayscale
gdcm::SurfaceHelper, 755
RGB
gdcm::PhotometricInterpretation, 578
RLE_COMPRESSION
vtkGDCMImageWriter, 919
RLECodec
gdcm::RLECodec, 655
RLELossless
gdcm::TransferSyntax, 795
gdcm::UIDs, 811
ROI
gdcm::Overlay, 554
RTBeamsDeliveryInstructionStorageSupplement74↩
FrozenDraft
gdcm::UIDs, 815
RTBeamsTreatmentRecordStorage
gdcm::UIDs, 815
RTBrachyTreatmentRecordStorage
gdcm::UIDs, 815
RTConventionalMachineVerificationSupplement74↩
FrozenDraft
gdcm::UIDs, 815
RTDoseStorage
gdcm::MediaStorage, 487
gdcm::UIDs, 815
RTImageStorage
gdcm::MediaStorage, 487
gdcm::UIDs, 815
RTIonBeamsTreatmentRecordStorage
gdcm::MediaStorage, 488
gdcm::UIDs, 815
RTIonMachineVerificationSupplement74FrozenDraft
gdcm::UIDs, 815
RTIonPlanStorage
gdcm::MediaStorage, 488
gdcm::UIDs, 815
RTPlanStorage
gdcm::MediaStorage, 487
gdcm::UIDs, 815
RTStructSetProperties
vtkGDCMPolyDataReader, 927
vtkGDCMPolyDataWriter, 930
RTStructureSetStorage
gdcm::MediaStorage, 487
gdcm::UIDs, 815
RTTreatmentSummaryRecordStorage
gdcm::MediaStorage, 488
gdcm::UIDs, 815
RawDataStorage
gdcm::MediaStorage, 487
gdcm::UIDs, 814
Read

gdcm::BasicOffsetTable, 156
gdcm::ByteValue, 178
gdcm::CP246ExplicitDataElement, 209
gdcm::CSAHeader, 222
gdcm::CommandDataSet, 201
gdcm::DataElement, 240
gdcm::DataSet, 252
gdcm::Element, 293
gdcm::Element< TVR, VM::VM1_n >, 297
gdcm::EncodingImplementation< VR::VRASCII >, 311
gdcm::EncodingImplementation< VR::VRBINARY >, 312
gdcm::ExplicitDataElement, 322
gdcm::ExplicitImplicitDataElement, 324
gdcm::File, 327
gdcm::FileMetaInformation, 346
gdcm::Fragment, 369
gdcm::ImageReader, 415
gdcm::ImageRegionReader, 418
gdcm::ImplicitDataElement, 428
gdcm::Item, 442
gdcm::PGXCodec, 577
gdcm::PNMCodec, 600
gdcm::PixmapReader, 591
gdcm::Preamble, 601
gdcm::Reader, 645
gdcm::SegmentReader, 675
gdcm::SequenceOfFragments, 682
gdcm::SequenceOfItems, 688
gdcm::StreamImageReader, 723
gdcm::SurfaceReader, 758
gdcm::TableReader, 772
gdcm::Tag, 779
gdcm::UNExplicitDataElement, 874
gdcm::UNExplicitImplicitDataElement, 876
gdcm::VR16ExplicitDataElement, 902
gdcm::VRVLSize< 0 >, 903
gdcm::VRVLSize< 1 >, 904
gdcm::ValueIO, 887
gdcm::VL, 890
gdcm::VR, 900
gdcm::network::AAbortPDU, 80
gdcm::network::AAssociateACPDU, 83
gdcm::network::AAssociateRJPDU, 85
gdcm::network::AAssociateRQPDU, 89
gdcm::network::AReleaseRPPDU, 107
gdcm::network::AReleaseRQPDU, 109
gdcm::network::AbstractSyntax, 92
gdcm::network::ApplicationContext, 103
gdcm::network::AsynchronousOperationsWindowSub, 112
gdcm::network::BasePDU, 144
gdcm::network::ImplementationClassUIDSub, 424
gdcm::network::ImplementationVersionNameSub, 426
gdcm::network::MaximumLengthSub, 481
gdcm::network::PDataTFPDU, 565
gdcm::network::PresentationContextAC, 605
gdcm::network::PresentationContextRQ, 610
gdcm::network::PresentationDataValue, 611
gdcm::network::RoleSelectionSub, 658
gdcm::network::SOPClassExtendedNegociationSub, 710
gdcm::network::ServiceClassApplicationInformation, 694
gdcm::network::TransferSyntaxSub, 798
gdcm::network::UserInformation, 882
Read16
  gdcm::VL, 890
ReadACRNEMAImage
  gdcm::ImageReader, 415
  gdcm::PixmapReader, 591
ReadBacktrack
  gdcm::Fragment, 369
ReadCompat
  gdcm::FileMetaInformation, 346
ReadCompatInternal
  gdcm::FileMetaInformation, 346
ReadComputeLength
  gdcm::EncodingImplementation< VR::VRASCII >, 311
  gdcm::EncodingImplementation< VR::VRBINARY >, 312
ReadDataSet
  gdcm::Reader, 645
ReadFiles
  vtkGDCMThreadedImageReader, 935
ReadFromCommaSeparatedString
  gdcm::PrivateTag, 619
  gdcm::Tag, 779
ReadFromContinuousString
  gdcm::Tag, 779
ReadFromPipeSeparatedString
  gdcm::Tag, 780
ReadImage
  gdcm::ImageReader, 415
  gdcm::PixmapReader, 592
ReadImageInformation
  gdcm::StreamImageReader, 723
ReadImageInternal
  gdcm::PixmapReader, 592
ReadInformation
  gdcm::ImageRegionReader, 418
ReadInto
  gdcm::network::PDataTFPDU, 565
  gdcm::network::PresentationDataValue, 611
ReadIntoBuffer

gdcm::ImageRegionReader, 419
ReadMetaInformation
gdcm::Reader, 645
ReadNested
gdcm::DataSet, 252
ReadNoSwap
gdcm::EncodingImplementation< VR::VRASCII >, 311
gdcm::EncodingImplementation< VR::VRBINARY >, 312
ReadOrSkip
gdcm::DataElement, 240
ReadPointMacro
gdcm::SurfaceReader, 758
ReadPreValue
gdcm::CP246ExplicitDataElement, 209
gdcm::DataElement, 240
gdcm::ExplicitDataElement, 322
gdcm::ExplicitImplicitDataElement, 324
gdcm::Fragment, 369
gdcm::ImplicitDataElement, 428
gdcm::SequenceOfFragments, 682
gdcm::UNExplicitDataElement, 874
gdcm::UNExplicitImplicitDataElement, 876
gdcm::VR16ExplicitDataElement, 902
ReadPreamble
gdcm::Reader, 645
ReadSegment
gdcm::SegmentReader, 675
ReadSegments
gdcm::SegmentReader, 675
ReadSelectedPrivateTags
gdcm::DataSet, 252
gdcm::Reader, 645
ReadSelectedPrivateTagsWithLength
gdcm::DataSet, 252
ReadSelectedTags
gdcm::DataSet, 252
gdcm::Reader, 645
ReadSelectedTagsWithLength
gdcm::DataSet, 252
ReadSurface
gdcm::SurfaceReader, 758
ReadSurfaces
gdcm::SurfaceReader, 758
ReadUpToTag
gdcm::DataSet, 252
gdcm::Reader, 645
ReadUpToTagWithLength
gdcm::DataSet, 252
ReadValue
gdcm::CP246ExplicitDataElement, 209
gdcm::DataElement, 240
gdcm::ExplicitDataElement, 322
gdcm::ExplicitImplicitDataElement, 324
gdcm::Fragment, 370
gdcm::ImplicitDataElement, 428
gdcm::SequenceOfFragments, 682
gdcm::UNExplicitDataElement, 874
gdcm::UNExplicitImplicitDataElement, 876
gdcm::VR16ExplicitDataElement, 902
ReadValueWithLength
gdcm::DataElement, 240
gdcm::ImplicitDataElement, 428
ReadVM
gdcm::DictConverter, 271
ReadVR
gdcm::DictConverter, 271
ReadWithLength
gdcm::CP246ExplicitDataElement, 209
gdcm::DataElement, 240
gdcm::DataSet, 252
gdcm::ExplicitDataElement, 322
gdcm::ExplicitImplicitDataElement, 324
gdcm::ImplicitDataElement, 428
gdcm::UNExplicitDataElement, 874
gdcm::VR16ExplicitDataElement, 902
Reader
gdcm::Reader, 644
Readuint16
gdcm::DictConverter, 271
RealWorldValueIntercept
gdcm::RealWorldValueMappingContent, 647
RealWorldValueMappingStorage
gdcm::UIDs, 814
RealWorldValueSlope
gdcm::RealWorldValueMappingContent, 647
RecommendedDisplayCIELabToRGB
gdcm::SurfaceHelper, 754
RecurseDataSet
gdcm::Anonymizer, 99
red
gdcm::terminal, 77
reference
gdcm::CodeString, 197
gdcm::LO, 470
gdcm::String, 738
ReferenceFrameOfReferenceUID
vtkRTStructSetProperties, 965
ReferenceSeriesInstanceUID
vtkRTStructSetProperties, 965
ReferencedColorPrintManagementMetaSOPClassRetired
gdcm::UIDs, 813
ReferencedGrayscalePrintManagementMetaSOPClassRetired
gdcm::UIDs, 812
ReferencedImageBoxSOPClassRetired
gdcm::UIDs, 812

Region
gdcm::Region, 649
Register
gdcm::Object, 541
Remove
gdcm::Anonymizer, 99
gdcm::DataSet, 252
gdcm::FileAnonymizer, 330
gdcm::Preamble, 601
RemoveAllObservers
gdcm::Subject, 745
RemoveDictEntry
gdcm::PrivateDict, 617
RemoveFile
gdcm::System, 768
RemoveGroupLength
gdcm::Anonymizer, 99
RemoveItemByIndex
gdcm::SequenceOfItems, 688
RemoveObserver
gdcm::Subject, 745
RemoveOverlay
gdcm::Pixmap, 588
RemovePrivateTags
gdcm::Anonymizer, 99
RemoveRetired
gdcm::Anonymizer, 99
Render
vtkImageColorViewer, 944
RenderWindow
vtkImageColorViewer, 946
Renderer
vtkImageColorViewer, 946
Replace
gdcm::Anonymizer, 99, 100
gdcm::CommandDataSet, 201
gdcm::DataSet, 252
gdcm::FileAnonymizer, 330
gdcm::FileMetaInformation, 346
ReplaceEmpty
gdcm::DataSet, 253
RequestData
vtkGDCMImageReader2, 914
vtkGDCMPolyDataReader, 927
vtkImageMapToColors16, 948
vtkImageMapToWindowLevelColors2, 951
vtkImagePlanarComponentsToComponents, 954
RequestData_HemodynamicWaveformStorage
vtkGDCMPolyDataReader, 927
RequestData_RTStructureSetStorage
vtkGDCMPolyDataReader, 927
RequestDataCompat
vtkGDCMImageReader, 908
vtkGDCMImageReader2, 914
vtkGDCMThreadedImageReader, 935
RequestInformation
vtkGDCMImageReader2, 914
vtkGDCMPolyDataReader, 927
vtkGDCMThreadedImageReader2, 938
vtkImageMapToColors16, 949
vtkImageMapToWindowLevelColors2, 951
RequestInformation_HemodynamicWaveformStorage
vtkGDCMPolyDataReader, 927
RequestInformation_RTStructureSetStorage
vtkGDCMPolyDataReader, 927
RequestInformationCompat
vtkGDCMImageReader, 908
vtkGDCMImageReader2, 914
RequestPaddedCompositePixelCode
gdcm::ImageCodec, 404
RequestPlanarConfiguration
gdcm::ImageCodec, 404
Rescale
gdcm::Rescaler, 652
RescaleFunctionIntoBestFit
gdcm::Rescaler, 652
Rescaler
gdcm::Rescaler, 651
ReserveDataElement
gdcm::FileStreamer, 360
ReserveGroupDataElement
gdcm::FileStreamer, 360
reset
gdcm::terminal, 77
ResetHandledDataSet
gdcm::network::ULConnectionCallback, 863
RetrieveSOPInstanceUIDFromIndex
gdcm::DirectoryHelper, 287
RetrieveSOPInstanceUIDFromZPosition
gdcm::DirectoryHelper, 287
reverse
gdcm::terminal, 77
reverse_iterator
gdcm::CodeString, 197
gdcm::LO, 470
gdcm::String, 738
RoleSelectionSub
gdcm::network::RoleSelectionSub, 658
RunEventLoop
gdcm::network::ULConnectionManager, 867
RunMoveEventLoop
gdcm::network::ULConnectionManager, 867

SAGITTAL
gdcm::Orientation, 550
SHA1
gdcm::SHA1, 701
SHComp

gdcm, 61
SIEMENS
gdcm::Dicts, 278
SINGLEBIT
gdcm::PixelFormat, 582
SLICE_ORIENTATION_XY
vtkImageColorViewer, 942
SLICE_ORIENTATION_XZ
vtkImageColorViewer, 942
SLICE_ORIENTATION_YZ
vtkImageColorViewer, 942
SOPClassExtendedNegociationSub
gdcm::network::SOPClassExtendedNegociationSub, 710
SOPInstanceUID
vtkRTStructSetProperties, 965
SPM2AVG152PDFrameofReference
gdcm::UIDs, 812
SPM2AVG152T1FrameofReference
gdcm::UIDs, 811
SPM2AVG152T2FrameofReference
gdcm::UIDs, 812
SPM2AVG305T1FrameofReference
gdcm::UIDs, 811
SPM2BRAINMASKFrameofReference
gdcm::UIDs, 811
SPM2CSFFrameofReference
gdcm::UIDs, 811
SPM2EPIFrameofReference
gdcm::UIDs, 811
SPM2FILT1FrameofReference
gdcm::UIDs, 811
SPM2GRAYFrameofReference
gdcm::UIDs, 811
SPM2PDFrameofReference
gdcm::UIDs, 811
SPM2PETFrameofReference
gdcm::UIDs, 811
SPM2SINGLESUBJT1FrameofReference
gdcm::UIDs, 812
SPM2SPECTFrameofReference
gdcm::UIDs, 811
SPM2T1FrameofReference
gdcm::UIDs, 811
SPM2T2FrameofReference
gdcm::UIDs, 811
SPM2TRANSMFrameofReference
gdcm::UIDs, 811
SPM2WHITEFrameofReference
gdcm::UIDs, 811
STATES_END
gdcm::Surface, 749
STATES
gdcm::Surface, 749
STComp
gdcm, 61
SURFACE
gdcm::Surface, 749
SV10
gdcm::CSAHeader, 220
ScalarType
gdcm::PixelFormat, 582
Scale
vtkGDCMImageReader, 910
vtkGDCMImageReader2, 917
Scan
gdcm::Scanner, 665
gdcm::StrictScanner, 735
Scanner
gdcm::Scanner, 662
SecondaryCaptureImageStorage
gdcm::MediaStorage, 486
gdcm::UIDs, 813
Segment
gdcm::Segment, 668
SegmentAlgorithmName
gdcm::Segment, 670
SegmentAlgorithmType
gdcm::Segment, 670
SegmentDescription
gdcm::Segment, 670
SegmentLabel
gdcm::Segment, 670
SegmentMap
gdcm::SegmentReader, 675
SegmentNumber
gdcm::Segment, 670
SegmentReader
gdcm::SegmentReader, 675
SegmentVector
gdcm::SegmentReader, 675
gdcm::SegmentWriter, 677
SegmentWriter
gdcm::SegmentWriter, 677
Segmentation
gdcm::MediaStorage, 488
SegmentationStorage
gdcm::MediaStorage, 488
gdcm::UIDs, 814
SegmentedPaletteColorLookupTable
gdcm::SegmentedPaletteColorLookupTable, 672
Segments
gdcm::SegmentReader, 675
gdcm::SegmentWriter, 678
Selection
gdcm::Sorter, 715
SelectionMap
gdcm::Sorter, 714

Self
gdcm::AnonymizeEvent, 94
gdcm::DataEvent, 245
gdcm::DataSetEvent, 255
gdcm::FileNameEvent, 352
gdcm::MemberCommand, 493
gdcm::ProgressEvent, 622
gdcm::SimpleMemberCommand, 703
SendEcho
gdcm::ServiceClassUser, 697
gdcm::network::ULConnectionManager, 867
SendFind
gdcm::ServiceClassUser, 697
gdcm::network::ULConnectionManager, 867
SendMove
gdcm::ServiceClassUser, 697, 698
gdcm::network::ULConnectionManager, 867
SendNAction
gdcm::network::ULConnectionManager, 867, 868
SendNCreate
gdcm::network::ULConnectionManager, 868
SendNDelete
gdcm::network::ULConnectionManager, 868
SendNEventReport
gdcm::network::ULConnectionManager, 868
SendNGet
gdcm::network::ULConnectionManager, 868
SendNSet
gdcm::network::ULConnectionManager, 868
SendStore
gdcm::ServiceClassUser, 698
gdcm::network::ULConnectionManager, 868
Separator
gdcm::ApplicationEntity, 105
gdcm::PersonName, 575
SequenceLengthField
gdcm::SequenceOfItems, 689
SequenceOfFragments
gdcm::SequenceOfFragments, 680
SequenceOfItems
gdcm::SequenceOfItems, 686
SerieHelper
gdcm::SerieHelper, 691
SerieRestrictions
gdcm::SerieHelper, 691
Series
gdcm::Series, 693
SeriesInstanceUID
vtkRTStructSetProperties, 965
ServiceClassApplicationInformation
gdcm::network::ServiceClassApplicationInformation, 694
ServiceClassUser
gdcm::ServiceClassUser, 696
Set
gdcm::Attribute, 117
gdcm::Attribute< Group, Element, TVR, VM::VM1 >, 121
gdcm::Attribute< Group, Element, TVR, VM::VM1_n >, 128
gdcm::Element, 293
gdcm::Element< TVR, VM::VM1_n >, 297
SetAETitle
gdcm::ServiceClassUser, 698
SetAbstractSyntax
gdcm::PresentationContext, 604
gdcm::network::PresentationContextRQ, 610
SetAlgorithmFamily
gdcm::Surface, 751
SetAlgorithmName
gdcm::Surface, 751
SetAlgorithmVersion
gdcm::Surface, 752
SetAnatomicRegion
gdcm::Segment, 669
SetArray
gdcm::Element< TVR, VM::VM1_n >, 297
SetAxisOfRotation
gdcm::Surface, 752
SetBitPosition
gdcm::Overlay, 556
SetBitSample
gdcm::JPEGCodec, 459
SetBitsAllocated
gdcm::Overlay, 556
gdcm::PixelFormat, 584
SetBitsStored
gdcm::PixelFormat, 585
SetBlob
gdcm::ApplicationEntity, 105
gdcm::PersonName, 574
gdcm::network::PresentationDataValue, 611
SetBlueLUT
gdcm::LookupTable, 475
SetBufferLength
gdcm::JPEGLSCodec, 464
gdcm::PNMCodec, 600
gdcm::RLECodec, 656
SetByteSwapTag
gdcm::ByteSwapFilter, 173
SetByteValue
gdcm::Attribute, 117
gdcm::Attribute< Group, Element, TVR, VM::VM1 >, 121
gdcm::Attribute< Group, Element, TVR, VM::VM1_n >, 128
gdcm::CSAElement, 217
gdcm::DataElement, 240

SetByteValueNoSwap
gdcm::Attribute, 117
gdcm::Attribute< Group, Element, TVR, VM::VM1 >, 122
SetCallbackFunction
gdcm::MemberCommand, 494
gdcm::SimpleMemberCommand, 704
SetCalledAETitle
gdcm::ServiceClassUser, 698
gdcm::network::AAssociateACPDU, 83
gdcm::network::AAssociateRQPDU, 89
SetCallingAETitle
gdcm::network::AAssociateACPDU, 83
gdcm::network::AAssociateRQPDU, 89
SetCenterOfRotation
gdcm::Surface, 752
SetChangePrivateTags
gdcm::FileExplicitFilter, 341
SetCheckFileMetaInformation
gdcm::Writer, 972
SetCipherType
gdcm::CAPICryptographicMessageSyntax, 182
gdcm::CryptographicMessageSyntax, 213
gdcm::OpenSSLCryptographicMessageSyntax, 544
gdcm::OpenSSLP7CryptographicMessageSyntax, 548
SetColor
gdcm::Printer, 615
SetColorLevel
vtkImageColorViewer, 944
SetColorWindow
vtkImageColorViewer, 944
SetColumns
gdcm::Bitmap, 162
gdcm::Overlay, 557
SetCommand
gdcm::network::PresentationDataValue, 611
SetComponents
gdcm::PersonName, 574, 575
SetCompressIconImage
gdcm::ImageChangeTransferSyntax, 396
SetComputeZSpacing
gdcm::IPPSorter, 438
SetCoordinateStartValue
gdcm::Curve, 232
SetCoordinateStepValue
gdcm::Curve, 233
SetCryptographicMessageSyntax
gdcm::Anonymizer, 100
SetCurve
gdcm::Curve, 233
vtkGDCMImageReader, 908
vtkGDCMImageReader2, 914
SetCurveDataDescriptor
gdcm::Curve, 233
SetCurveDescription
gdcm::Curve, 233
SetData
gdcm::DataEvent, 246
SetDataElement
gdcm::Bitmap, 162
SetDataSet
gdcm::File, 328
gdcm::network::PresentationDataValue, 611
SetDataSetTransferSyntax
gdcm::FileMetaInformation, 346
SetDataValueRepresentation
gdcm::Curve, 233
SetDebug
gdcm::Trace, 790
SetDebugStream
gdcm::Trace, 790
SetDefaultTransferSyntax
gdcm::PresentationContextGenerator, 607
SetDerivationCodeSequenceCodeValue
gdcm::FileDerivation, 339
SetDerivationDescription
gdcm::FileDerivation, 339
SetDescription
gdcm::CSAHeaderDictEntry, 226
gdcm::ModuleEntry, 509
gdcm::Overlay, 557
SetDescriptor
gdcm::DICOMDIRGenerator, 266
SetDictName
gdcm::DictConverter, 271
SetDicts
gdcm::PythonFilter, 626
gdcm::StringFilter, 741
SetDimension
gdcm::Bitmap, 162
SetDimensions
gdcm::Bitmap, 162
gdcm::Curve, 233
gdcm::ImageCodec, 402
SetDimensionsValue
gdcm::ImageHelper, 410
SetDirectionCosines
gdcm::Image, 384
vtkGDCMImageWriter, 920
SetDirectionCosinesFromImageOrientationPatient
vtkGDCMImageWriter, 920
SetDirectionCosinesTolerance
gdcm::IPPSorter, 438
SetDirectionCosinesValue
gdcm::ImageHelper, 410
SetDirectory
gdcm::SerieHelper, 692

gdcm::network::ULWritingCallback, 872
SetDisplayId
vtkImageColorViewer, 944
SetDomain
gdcm::BoxRegion, 170
SetDropDuplicatePositions
gdcm::IPPSorter, 438
SetElement
gdcm::Tag, 780
SetElementHandler
gdcm::Parser, 562
SetElementTag
gdcm::Tag, 780
SetElementXX
gdcm::DictEntry, 274
SetError
gdcm::Trace, 791
SetErrorStream
gdcm::Trace, 791
SetEvent
gdcm::network::ULEvent, 869
SetFile
gdcm::Anonymizer, 100
gdcm::DICOMDIRGenerator, 266
gdcm::FileDecompressLookupTable, 336
gdcm::FileDerivation, 339
gdcm::FileExplicitFilter, 341
gdcm::IconImageFilter, 377
gdcm::Printer, 615
gdcm::PythonFilter, 626
gdcm::Reader, 646
gdcm::SplitMosaicFilter, 719
gdcm::StreamImageWriter, 727
gdcm::StringFilter, 741
gdcm::Validate, 884
gdcm::Writer, 972
gdcm::XMLPrinter, 977
SetFileName
gdcm::FileNameEvent, 352
gdcm::Reader, 646
gdcm::StreamImageReader, 723
gdcm::StreamImageWriter, 727
gdcm::Writer, 972
vtkGDCMThreadedImageReader2, 938
SetFileNames
vtkGDCMImageReader, 908
vtkGDCMImageWriter, 920
vtkGDCMThreadedImageReader2, 938
SetFilePattern
vtkGDCMImageReader, 908
vtkGDCMImageReader2, 914
SetFilePrefix
vtkGDCMImageReader, 908
vtkGDCMImageReader2, 914
SetFilename
gdcm::TableReader, 772
SetFilenames
gdcm::DICOMDIRGenerator, 266
SetFiles
gdcm::FileSet, 357
SetFiniteVolume
gdcm::Surface, 752
SetForce
gdcm::ImageChangeTransferSyntax, 396
gdcm::ImageFragmentSplitter, 407
SetForcePixelSpacing
gdcm::ImageHelper, 410
SetForceRescaleInterceptSlope
gdcm::ImageHelper, 411
SetFragmentSizeMax
gdcm::ImageFragmentSplitter, 407
SetFrameOrigin
gdcm::Overlay, 557
SetFromDataElement
gdcm::Attribute, 117
gdcm::Attribute< Group, Element, TVR, VM::VM1 >, 122
gdcm::Attribute< Group, Element, TVR, VM::VM1_n >, 128
gdcm::Element, 293
gdcm::Element< TVR, VM::VM1_n >, 297
SetFromDataSet
gdcm::Attribute, 117
gdcm::Attribute< Group, Element, TVR, VM::VM1 >, 122
gdcm::Attribute< Group, Element, TVR, VM::VM1_n >, 128
gdcm::MediaStorage, 490
SetFromFile
gdcm::MediaStorage, 490
SetFromHeader
gdcm::MediaStorage, 490
SetFromModality
gdcm::MediaStorage, 491
SetFromSourceImageSequence
gdcm::MediaStorage, 491
SetFromString
gdcm::DirectionCosines, 283
SetFromUID
gdcm::UIDs, 824
SetGreenLUT
gdcm::LookupTable, 475
SetGroup
gdcm::Curve, 233
gdcm::Overlay, 557
gdcm::Tag, 780
SetGroupXX
gdcm::DictEntry, 274

SetHeader
gdcm::File, 328
SetHighBit
gdcm::PixelFormat, 585
SetHostname
gdcm::ServiceClassUser, 698
SetIconImage
gdcm::Pixmap, 589
SetIE
gdcm::IODEntry, 433
SetImage
gdcm::PixmapWriter, 597
gdcm::SplitMosaicFilter, 719
SetImplementationClassUID
gdcm::FileMetaInformation, 346
SetImplementationVersionName
gdcm::FileMetaInformation, 346
SetImplicitFlag
gdcm::network::ULConnectionCallback, 863
SetInput
gdcm::BitmapToBitmapFilter, 166
gdcm::ImageConverter, 405
vtkImageColorViewer, 944
SetInputConnection
vtkImageColorViewer, 944
SetInputFileName
gdcm::DictConverter, 271
gdcm::FileAnonymizer, 331
gdcm::FileChangeTransferSyntax, 334
SetIntercept
gdcm::Image, 384
gdcm::Rescaler, 652
SetKey
gdcm::CSAElement, 217
SetKeyword
gdcm::DictEntry, 274
SetLUT
gdcm::Bitmap, 162
gdcm::ImageCodec, 402
gdcm::LookupTable, 475
gdcm::SegmentedPaletteColorLookupTable, 672
SetLastElement
gdcm::ParseException, 559
SetLastFragment
gdcm::network::PresentationDataValue, 611
SetLength
gdcm::ByteValue, 178
gdcm::Element< TVR, VM::VM1_2 >, 295
gdcm::Element< TVR, VM::VM1_n >, 297
gdcm::Element< TVR, VM::VM2_2n >, 299
gdcm::Element< TVR, VM::VM2_n >, 301
gdcm::Element< TVR, VM::VM3_3n >, 302
gdcm::Element< TVR, VM::VM3_n >, 304
gdcm::RLECodec, 657
gdcm::SequenceOfFragments, 682
gdcm::SequenceOfItems, 688
gdcm::Value, 886
SetLengthOnly
gdcm::ByteValue, 178
gdcm::Value, 886
SetLengthToUndefined
gdcm::SequenceOfItems, 689
SetLoadMode
gdcm::SerieHelper, 692
SetLookupTable
vtkImageMapToColors16, 949
SetLossless
gdcm::JPEGCodec, 460
gdcm::JPEGLSCodec, 464
SetLossyError
gdcm::JPEGLSCodec, 464
SetLossyFlag
gdcm::Bitmap, 162
gdcm::ImageCodec, 402
gdcm::PVRGCodec, 625
SetManifold
gdcm::Surface, 752
SetMaxPDULength
gdcm::network::ULConnectionInfo, 864
SetMaxPDUSize
gdcm::network::ULConnection, 861
SetMaximumLength
gdcm::network::MaximumLengthSub, 481
SetMaximumPointDistance
gdcm::Surface, 752
SetMeanPointDistance
gdcm::Surface, 752
SetMedicalImageProperties
vtkGDCMImageReader, 908
vtkGDCMImageReader2, 914
vtkGDCMImageWriter, 920
vtkGDCMPolyDataWriter, 930
SetMergeModeToAbstractSyntax
gdcm::PresentationContextGenerator, 608
SetMergeModeToTransferSyntax
gdcm::PresentationContextGenerator, 608
SetMeshPrimitive
gdcm::Surface, 752
SetMessageHeader
gdcm::network::PresentationDataValue, 612
SetMinMaxForPixelType
gdcm::Rescaler, 652
SetName
gdcm::CSAElement, 217
gdcm::CSAHeaderDictEntry, 226
gdcm::DictEntry, 274
gdcm::IODEntry, 433
gdcm::Macro, 478

gdcm::Module, 507
gdcm::ModuleEntry, 510
gdcm::PDBElement, 567
gdcm::network::AbstractSyntax, 92
gdcm::network::ApplicationContext, 103
gdcm::network::TransferSyntaxSub, 798
SetNameFromUID
gdcm::network::AbstractSyntax, 92
gdcm::network::TransferSyntaxSub, 798
SetNeedByteSwap
gdcm::Bitmap, 163
gdcm::ImageCodec, 402
SetNeedOverlayCleanup
gdcm::ImageCodec, 402
SetNestedDataSet
gdcm::Item, 442
SetNoOfItems
gdcm::CSAElement, 217
SetNoSwap
gdcm::Element, 293
gdcm::Element< TVR, VM::VM1_n >, 297
SetNumberOfCurves
gdcm::Pixmap, 589
SetNumberOfDimensions
gdcm::Bitmap, 163
gdcm::ImageCodec, 402
SetNumberOfFilenames
gdcm::FilenameGenerator, 355
SetNumberOfFrames
gdcm::Overlay, 557
SetNumberOfInputPorts
vtkGDCMPolyDataWriter, 930
SetNumberOfItems
gdcm::SequenceOfItems, 689
SetNumberOfOverlays
gdcm::Pixmap, 589
SetNumberOfPoints
gdcm::Curve, 233
SetNumberOfResolutions
gdcm::JPEG2000Codec, 452
SetNumberOfSegments
gdcm::SegmentWriter, 677
SetNumberOfSurfacePoints
gdcm::Surface, 752
SetNumberOfSurfaces
gdcm::SurfaceWriter, 760
SetNumberOfTableValues
vtkLookupTable16, 959
SetNumberOfValues
gdcm::Attribute< Group, Element, TVR, VM::VM1_n >, 128
SetNumberOfVectors
gdcm::Surface, 752
SetObliquityThresholdCosineValue
gdcm::Orientation, 551
SetOffScreenRendering
vtkImageColorViewer, 944
SetOrigin
gdcm::Image, 384
gdcm::Overlay, 557
SetOriginValue
gdcm::ImageHelper, 411
SetOutputDimensions
gdcm::IconImageGenerator, 379
SetOutputFileName
gdcm::DictConverter, 271
gdcm::FileAnonymizer, 331
gdcm::FileChangeTransferSyntax, 334
gdcm::FileStreamer, 360
SetOutputFormatToLuminance
vtkImageMapToColors16, 949
SetOutputFormatToLuminanceAlpha
vtkImageMapToColors16, 949
SetOutputFormatToRGBA
vtkImageMapToColors16, 949
SetOutputFormatToRGB
vtkImageMapToColors16, 949
SetOutputType
gdcm::DictConverter, 271
SetOutsideValuePixel
gdcm::IconImageGenerator, 379
SetOverlay
gdcm::Overlay, 557
SetOverlayVisibility
vtkImageColorViewer, 944
SetOwner
gdcm::PrivateTag, 619
SetPDU
gdcm::network::ULEvent, 869
SetPMSRescaleInterceptSlope
gdcm::ImageHelper, 411
SetParentId
vtkImageColorViewer, 944
SetPassword
gdcm::CAPICryptographicMessageSyntax, 182
gdcm::CryptographicMessageSyntax, 213
gdcm::OpenSSLCryptographicMessageSyntax, 545
gdcm::OpenSSLP7CryptographicMessageSyntax, 548
SetPattern
gdcm::FilenameGenerator, 355
SetPermissions
gdcm::System, 768
SetPhotometricInterpretation
gdcm::Bitmap, 163
gdcm::ImageChangePhotometricInterpretation, 389
gdcm::ImageCodec, 402
SetPixelFormat

gdcm::Bitmap, 163
gdcm::ImageCodec, 402
gdcm::JPEGCodec, 460
gdcm::Rescaler, 652
SetPixelMinMax
gdcm::IconImageGenerator, 379
SetPixelRepresentation
gdcm::PixelFormat, 585
SetPixmap
gdcm::FileDecompressLookupTable, 336
gdcm::IconImageGenerator, 379
gdcm::PixmapWriter, 597
SetPlanarConfiguration
gdcm::Bitmap, 163
gdcm::ImageChangePlanarConfiguration, 392
gdcm::ImageCodec, 403
SetPointCoordinatesData
gdcm::Surface, 752
SetPointPositionAccuracy
gdcm::Surface, 752
SetPointsBoundingBoxCoordinates
gdcm::Surface, 752
SetPort
gdcm::ServiceClassUser, 699
SetPortSCP
gdcm::ServiceClassUser, 699
SetPosition
vtkImageColorViewer, 944
SetPreamble
gdcm::FileMetaInformation, 347
SetPrefix
gdcm::FilenameGenerator, 355
SetPresentationContextID
gdcm::PresentationContext, 604
gdcm::network::PresentationContextAC, 605
gdcm::network::PresentationContextRQ, 610
gdcm::network::PresentationDataValue, 612
SetPresentationContexts
gdcm::ServiceClassUser, 699
gdcm::network::ULConnection, 861
SetPrettyPrint
gdcm::JSON, 466
SetPrimitiveData
gdcm::MeshPrimitive, 498
SetPrimitiveType
gdcm::MeshPrimitive, 498
SetPrimitivesData
gdcm::MeshPrimitive, 498
SetPrivateCreator
gdcm::Tag, 780
SetProcessingAlgorithm
gdcm::Surface, 752
SetProgress
gdcm::ProgressEvent, 622
SetPropertyCategory
gdcm::Segment, 669
SetPropertyType
gdcm::Segment, 669
SetPurposeOfReferenceCodeSequenceCodeValue
gdcm::FileDerivation, 339
SetQuality
gdcm::JPEG2000Codec, 452
gdcm::JPEGCodec, 460
SetRTStructSetProperties
vtkGDCMPolyDataWriter, 930
SetRate
gdcm::JPEG2000Codec, 452
SetReason
gdcm::network::AAbortPDU, 80
gdcm::network::PresentationContextAC, 605
SetRecommendedDisplayCIELabValue
gdcm::Surface, 752
SetRecommendedDisplayGrayscaleValue
gdcm::Surface, 752
SetRecommendedPresentationOpacity
gdcm::Surface, 752
SetRecommendedPresentationType
gdcm::Surface, 752
SetRecomputeItemLength
gdcm::FileExplicitFilter, 341
SetRecomputeSequenceLength
gdcm::FileExplicitFilter, 341
SetRedLUT
gdcm::LookupTable, 475
SetRef
gdcm::IODEntry, 433
SetRegion
gdcm::ImageRegionReader, 419
SetRenderWindow
vtkImageColorViewer, 944
SetRenderer
vtkImageColorViewer, 944
SetRescaleInterceptSlopeValue
gdcm::ImageHelper, 411
SetRetired
gdcm::DictEntry, 274
SetReversible
gdcm::JPEG2000Codec, 452
SetRoot
gdcm::UIDGenerator, 805
SetRootDirectory
gdcm::DICOMDIRGenerator, 266
SetRows
gdcm::Bitmap, 163
gdcm::Overlay, 557
SetSOPInstanceUID
gdcm::BaseQuery, 148
SetSamplesPerPixel

gdcm::PixelFormat, 585
SetScalarType
gdcm::PixelFormat, 585
SetSearchParameter
gdcm::BaseQuery, 147, 148
SetSegmentAlgorithmName
gdcm::Segment, 669
SetSegmentAlgorithmType
gdcm::Segment, 669
SetSegmentDescription
gdcm::Segment, 669
SetSegmentLabel
gdcm::Segment, 669
SetSegmentNumber
gdcm::Segment, 669
SetSegments
gdcm::SegmentWriter, 677
SetSize
vtkImageColorViewer, 944
SetSlice
vtkImageColorViewer, 945
SetSliceOrientation
vtkImageColorViewer, 945
SetSliceOrientationToXY
vtkImageColorViewer, 945
SetSliceOrientationToXZ
vtkImageColorViewer, 945
SetSliceOrientationToYZ
vtkImageColorViewer, 945
SetSlope
gdcm::Image, 384
gdcm::Rescaler, 652
SetSortFunction
gdcm::Sorter, 714
SetSource
gdcm::network::AAbortPDU, 80
SetSourceApplicationEntityTitle
gdcm::FileMetaInformation, 347
SetSpacing
gdcm::Image, 384
SetSpacingValue
gdcm::ImageHelper, 411
SetState
gdcm::network::ULConnection, 861
SetStream
gdcm::Reader, 646
gdcm::StreamImageReader, 724
gdcm::StreamImageWriter, 728
gdcm::Trace, 791
gdcm::Writer, 973
SetStreamToFile
gdcm::Trace, 791
SetStyle
gdcm::Printer, 615
gdcm::XMLPrinter, 977
SetSurfaceComments
gdcm::Surface, 752
SetSurfaceCount
gdcm::Segment, 670
SetSurfaceNumber
gdcm::Surface, 753
SetSurfaceProcessing
gdcm::Surface, 753
SetSurfaceProcessingDescription
gdcm::Surface, 753
SetSurfaceProcessingRatio
gdcm::Surface, 753
SetSyngoDT
gdcm::CSAElement, 217
SetTag
gdcm::AnonymizeEvent, 95
gdcm::DataElement, 241
SetTargetPixelType
gdcm::Rescaler, 652
SetTemplateFileName
gdcm::FileStreamer, 360
SetTileSize
gdcm::JPEG2000Codec, 452
SetTimeout
gdcm::ServiceClassUser, 699
gdcm::network::ARTIMTimer, 110
SetToUndefined
gdcm::VL, 890
SetTransferSyntax
gdcm::Bitmap, 163
gdcm::FileChangeTransferSyntax, 334
gdcm::ImageChangeTransferSyntax, 396
gdcm::network::PresentationContextAC, 605
SetTuple
gdcm::network::RoleSelectionSub, 658
gdcm::network::SOPClassExtendedNegociationSub, 710
gdcm::network::ServiceClassApplicationInformation, 694
SetType
gdcm::ModuleEntry, 510
gdcm::Overlay, 557
SetTypeOfData
gdcm::Curve, 233
SetUsage
gdcm::IODEntry, 433
SetUseSeriesDetails
gdcm::SerieHelper, 692
SetUseTargetPixelType
gdcm::Rescaler, 653
SetUseVRUN
gdcm::FileExplicitFilter, 342
SetUserCodec

gdcm::ImageChangeTransferSyntax, 396
SetUserData
gdcm::Parser, 562
SetUserInformation
gdcm::network::AAssociateRQPDU, 90
SetVLToUndefined
gdcm::DataElement, 241
SetValue
gdcm::Attribute, 117
gdcm::Attribute< Group, Element, TVR, VM::VM1 >, 122
gdcm::Attribute< Group, Element, TVR, VM::VM1_n >, 128
gdcm::CSAElement, 217
gdcm::DataElement, 241
gdcm::Element, 293
gdcm::Element< TVR, VM::VM1_n >, 297
gdcm::PDBElement, 567
SetValueFieldLength
gdcm::DataElement, 241
SetValues
gdcm::Attribute, 118
gdcm::Attribute< Group, Element, TVR, VM::VM1_n >, 128
SetVectorAccuracy
gdcm::Surface, 753
SetVectorCoordinateData
gdcm::Surface, 753
SetVectorDimensionality
gdcm::Surface, 753
SetVL
gdcm::DataElement, 241
SetVM
gdcm::CSAElement, 217
gdcm::CSAHeaderDictEntry, 226
gdcm::DictEntry, 274
SetVR
gdcm::CSAElement, 217
gdcm::CSAHeaderDictEntry, 226
gdcm::DataElement, 242
gdcm::DictEntry, 274
SetWarning
gdcm::Trace, 791
SetWarningStream
gdcm::Trace, 791
SetWindowId
vtkImageColorViewer, 945
SetWriteDataSetOnly
gdcm::Writer, 973
SetZSpacingTolerance
gdcm::IPPSorter, 438
setattribute
gdcm::terminal, 77
setbgcolor
gdcm::terminal, 77
setfgcolor
gdcm::terminal, 77
setmode
gdcm::terminal, 77
SetupInteractor
vtkImageColorViewer, 945
SH
gdcm::VR, 898
Shift
vtkGDCMImageReader, 910
vtkGDCMImageReader2, 917
ShiftEnd
gdcm::ByteBuffer, 171
ShowAbort
gdcm::SimpleSubjectWatcher, 705
ShowAnonymization
gdcm::SimpleSubjectWatcher, 705
ShowData
gdcm::SimpleSubjectWatcher, 705
ShowDataSet
gdcm::SimpleSubjectWatcher, 705
ShowFileName
gdcm::SimpleSubjectWatcher, 705
ShowIteration
gdcm::SimpleSubjectWatcher, 705
ShowProgress
gdcm::SimpleSubjectWatcher, 706
SimpleMemberCommand
gdcm::SimpleMemberCommand, 703
SimpleSubjectWatcher
gdcm::SimpleSubjectWatcher, 705
SingleSerieUIDFileSetHT
gdcm::SerieHelper, 692
SingleSerieUIDFileSetmap
gdcm::SerieHelper, 691
Size
gdcm::CodeString, 198
gdcm::DataSet, 253
gdcm::GroupDict, 374
gdcm::network::AAbortPDU, 80
gdcm::network::AAssociateACPDU, 83
gdcm::network::AAssociateRJPDU, 85
gdcm::network::AAssociateRQPDU, 90
gdcm::network::AReleaseRPPDU, 107
gdcm::network::AReleaseRQPDU, 109
gdcm::network::AbstractSyntax, 92
gdcm::network::ApplicationContext, 103
gdcm::network::AsynchronousOperationsWindowSub, 112
gdcm::network::BasePDU, 144
gdcm::network::ImplementationClassUIDSub, 424
gdcm::network::ImplementationVersionNameSub, 426

gdcm::network::MaximumLengthSub, 481
gdcm::network::PDataTFPDU, 565
gdcm::network::PresentationContextAC, 605
gdcm::network::PresentationContextRQ, 610
gdcm::network::PresentationDataValue, 612
gdcm::network::RoleSelectionSub, 658
gdcm::network::SOPClassExtendedNegociationSub, 710
gdcm::network::ServiceClassApplicationInformation, 694
gdcm::network::TransferSyntaxSub, 798
gdcm::network::UserInformation, 882
size_type
gdcm::CodeString, 197
gdcm::LO, 470
gdcm::String, 738
SizeType
gdcm::DataSet, 249
gdcm::FilenameGenerator, 354
gdcm::IOD, 430
gdcm::NestedModuleEntries, 527
gdcm::PresentationContext, 603
gdcm::PresentationContextGenerator, 607
gdcm::SequenceOfFragments, 680
gdcm::SequenceOfItems, 686
gdcm::network::AAssociateACPDU, 82
gdcm::network::AAssociateRQPDU, 88
gdcm::network::PDataTFPDU, 564
gdcm::network::PresentationContextRQ, 609
SL
gdcm::VR, 898
Slice
vtkImageColorViewer, 946
SliceOrientation
vtkImageColorViewer, 946
SmartPointer
gdcm::Object, 541
gdcm::SmartPointer, 708
Sort
gdcm::IPPSorter, 438
gdcm::Sorter, 714
SortFunc
gdcm::Sorter, 715
SortFunction
gdcm::Sorter, 714
Sorter
gdcm::Sorter, 714
SpacialFiducialsStorage
gdcm::MediaStorage, 487
SpacialRegistrationStorage
gdcm::MediaStorage, 487
Spacing
gdcm::Spacing, 717
SpacingType
gdcm::Spacing, 717
SpatialFiducialsStorage
gdcm::UIDs, 814
SpatialRegistrationStorage
gdcm::UIDs, 814
Spectroscopy
gdcm::Spectroscopy, 718
Split
gdcm::ImageFragmentSplitter, 407
gdcm::SplitMosaicFilter, 719
SplitExtent
vtkGDCMThreadedImageReader2, 938
SplitMosaicFilter
gdcm::SplitMosaicFilter, 718
SQ
gdcm::VR, 898
Squeeze
gdcm::ApplicationEntity, 105
SS
gdcm::VR, 898
ST
gdcm::VR, 898
StableSort
gdcm::Sorter, 715
StandaloneCurveStorage
gdcm::MediaStorage, 487
StandaloneCurveStorageRetired
gdcm::UIDs, 813
StandaloneModalityLUTStorage
gdcm::MediaStorage, 487
StandaloneModalityLUTStorageRetired
gdcm::UIDs, 814
StandaloneOverlayStorage
gdcm::MediaStorage, 487
StandaloneOverlayStorageRetired
gdcm::UIDs, 813
StandalonePETCurveStorageRetired
gdcm::UIDs, 815
StandaloneVOILUTStorage
gdcm::MediaStorage, 487
StandaloneVOILUTStorageRetired
gdcm::UIDs, 814
Start
gdcm::network::ARTIMTimer, 110
StartAssociation
gdcm::ServiceClassUser, 699
StartDataElement
gdcm::FileStreamer, 360
StartElement
gdcm::TableReader, 772
gdcm::XMLDictReader, 975
gdcm::XMLPrivateDictReader, 979
StartElementHandler
gdcm::Parser, 561

StartEncode
gdcm::ImageCodec, 403
gdcm::JPEG2000Codec, 452
gdcm::JPEGCodec, 460
gdcm::JPEGLSCodec, 464
gdcm::RLECodec, 657
StartFilter
gdcm::SimpleSubjectWatcher, 706
StartGroupDataElement
gdcm::FileStreamer, 360
StereometricRelationshipStorage
gdcm::UIDs, 814
Stop
gdcm::network::ARTIMTimer, 110
StopAssociation
gdcm::ServiceClassUser, 699
StopDataElement
gdcm::FileStreamer, 361
StopEncode
gdcm::ImageCodec, 403
gdcm::JPEG2000Codec, 452
gdcm::JPEGCodec, 460
gdcm::JPEGLSCodec, 464
gdcm::RLECodec, 657
StopGroupDataElement
gdcm::FileStreamer, 361
StopProtocol
gdcm::network::ULConnection, 861
StorageCommitmentPullModelSOPClassRetired
gdcm::UIDs, 812
StorageCommitmentPullModelSOPInstanceRetired
gdcm::UIDs, 812
StorageCommitmentPushModelSOPClass
gdcm::UIDs, 812
StorageCommitmentPushModelSOPInstance
gdcm::UIDs, 812
StorageServiceClass
gdcm::UIDs, 812
StoredPrintStorageSOPClassRetired
gdcm::UIDs, 813
StrCaseCmp
gdcm::System, 768
StrNCaseCmp
gdcm::System, 768
StrSep
gdcm::System, 768
StrTokR
gdcm::System, 768
Stream
gdcm::Writer, 973
StreamImageReader
gdcm::Reader, 646
gdcm::StreamImageReader, 722
StreamImageWriter
gdcm::StreamImageWriter, 727
gdcm::Writer, 973
StrictScanner
gdcm::StrictScanner, 733
String
gdcm::String, 738, 739
StringFilter
gdcm::StringFilter, 741
StructureSetDate
vtkRTStructSetProperties, 965
StructureSetLabel
vtkRTStructSetProperties, 965
StructureSetName
vtkRTStructSetProperties, 965
StructureSetTime
vtkRTStructSetProperties, 965
Study
gdcm::Study, 743
StudyComponentManagementSOPClass
gdcm::MediaStorage, 487
StudyComponentManagementSOPClassRetired
gdcm::UIDs, 812
StudyInstanceUID
vtkRTStructSetProperties, 965
StudyRootQueryRetrieveInformationModelFIND
gdcm::UIDs, 815
StudyRootQueryRetrieveInformationModelGET
gdcm::UIDs, 815
StudyRootQueryRetrieveInformationModelMOVE
gdcm::UIDs, 815
Subject
gdcm::Subject, 745
SubstanceAdministrationLoggingSOPClass
gdcm::UIDs, 812
SubstanceAdministrationLoggingSOPInstance
gdcm::UIDs, 812
SubstanceApprovalQuerySOPClass
gdcm::UIDs, 816
Superclass
gdcm::AnonymizeEvent, 94
gdcm::DataEvent, 245
gdcm::DataSetEvent, 255
gdcm::FileNameEvent, 352
gdcm::LO, 470
gdcm::ProgressEvent, 622
Surface
gdcm::Surface, 749
SurfaceCount
gdcm::Segment, 670
SurfaceReader
gdcm::SurfaceReader, 758
SurfaceSegmentationStorage
gdcm::MediaStorage, 488
gdcm::UIDs, 817

SurfaceVector
gdcm::Segment, 668
SurfaceWriter
gdcm::SurfaceWriter, 760
Surfaces
gdcm::Segment, 670
Swap
gdcm::ByteSwap, 172
gdcm::SwapperDoOp, 763
gdcm::SwapperNoOp, 763
SwapArray
gdcm::SwapperDoOp, 763
gdcm::SwapperNoOp, 763
SwapCode
gdcm::SwapCode, 762
SwapCodeType
gdcm::SwapCode, 761
SwapFromSwapCodeIntoSystem
gdcm::ByteSwap, 172
SwapRange
gdcm::ByteSwap, 172
SwapRangeFromSwapCodeIntoSystem
gdcm::ByteSwap, 172
SyngoDTField
gdcm::CSAElement, 218
SyntaxError
gdcm::Parser, 561
SystemIsBigEndian
gdcm::ByteSwap, 172
SystemIsLittleEndian
gdcm::ByteSwap, 172

T1
gdcm::Type, 801
T1C
gdcm::Type, 801
T2
gdcm::Type, 801
T2C
gdcm::Type, 801
T3
gdcm::Type, 801
TConstMemberFunctionPointer
gdcm::MemberCommand, 493
TMComp
gdcm, 61
TMemberFunctionPointer
gdcm::MemberCommand, 493
gdcm::SimpleMemberCommand, 703
TRIANGLE_FAN
gdcm::MeshPrimitive, 497
TRIANGLE_STRIP
gdcm::MeshPrimitive, 497
TRIANGLE
gdcm::MeshPrimitive, 497
TS_END
gdcm::TransferSyntax, 795
TSName
gdcm::UIDs, 810
TSType
gdcm::TransferSyntax, 794
gdcm::UIDs, 817
TYPETOENCODING
gdcm, 69
gdcmVR.h, 1248
TYPETOLENGTH
gdcmVM.h, 1246
Table
gdcm::Table, 769
Table16
vtkLookupTable16, 959
TableEntry
gdcm::TableEntry, 770
TableReader
gdcm::TableReader, 772
TableRow
gdcm::network::TableRow, 773
Tag
gdcm::Tag, 776
tag
gdcm::Tag, 781
TagField
gdcm::DataElement, 242
TagMismatchError
gdcm::Parser, 561
TagPath
gdcm::TagPath, 782
TagToValue
gdcm::Scanner, 662
gdcm::StrictScanner, 732
TagToValueValueType
gdcm::Scanner, 662
gdcm::StrictScanner, 732
tags
gdcm::Tag, 781
TalairachBrainAtlasFrameofReference
gdcm::UIDs, 811
TestAbortOff
gdcm::SimpleSubjectWatcher, 706
TestAbortOn
gdcm::SimpleSubjectWatcher, 706
TestPBKDF2
gdcm::ASN1, 111
Testing
gdcm::Testing, 785
TestsList.txt, 1254
TextSRStorageTrialRetired
gdcm::UIDs, 814

ThreadedExecute
vtkImageRGBToYBR, 955
vtkImageYBRToRGB, 957
ThreadedRequestData
vtkGDCMThreadedImageReader2, 938
vtkImageMapToColors16, 949
vtkImageMapToWindowLevelColors2, 951
TM
gdcm::VR, 898
to_string
gdcm, 69
ToPyObject
gdcm::PythonFilter, 626
ToString
gdcm::StringFilter, 741, 742
ToStringPair
gdcm::StringFilter, 742
ToUnixSlashes
gdcm::Filename, 349
ToWindowsSlashes
gdcm::Filename, 350
ToshibaPrivateDataStorage
gdcm::MediaStorage, 488
Trace
gdcm::Trace, 790
TransferSyntax
gdcm::TransferSyntax, 795
TransferSyntaxArrayType
gdcm::PresentationContext, 603
TransferSyntaxStringsType
gdcm::UIDs, 810
TransferSyntaxSub
gdcm::network::TransferSyntaxSub, 798
TransferSyntaxes
gdcm::PresentationContext, 604
Transition
gdcm::network::Transition, 799, 800
transitions
gdcm::network::TableRow, 774
Trim
gdcm::String, 739
TrimInternal
gdcm::CodeString, 198
Truncate
gdcm::String, 739
TryJPEG2000Codec
gdcm::Bitmap, 163
gdcm::ImageChangeTransferSyntax, 396
TryJPEG2000Codec2
gdcm::Bitmap, 164
TryJPEGCodec
gdcm::Bitmap, 164
gdcm::ImageChangeTransferSyntax, 396
TryJPEGCodec2
gdcm::Bitmap, 164
TryJPEGLSCodec
gdcm::Bitmap, 164
gdcm::ImageChangeTransferSyntax, 396
TryKAKADUCodec
gdcm::Bitmap, 164
TryPVRGCodec
gdcm::Bitmap, 164
TryRAWCodec
gdcm::Bitmap, 164
gdcm::ImageChangeTransferSyntax, 396
TryRLECodec
gdcm::Bitmap, 164
gdcm::ImageChangeTransferSyntax, 396
TS
gdcm::Bitmap, 164
Type
gdcm::Element, 292
gdcm::Element< TVR, VM::VM1_n >, 296
gdcm::Type, 802
gdcm::VL, 890
TypeType
gdcm::Type, 801

UIComp
gdcm, 61
UIDGenerator
gdcm::UIDGenerator, 804
UINT12
gdcm::PixelFormat, 582
UINT16
gdcm::PixelFormat, 582
UINT32
gdcm::PixelFormat, 582
UINT64
gdcm::PixelFormat, 582
UINT8
gdcm::PixelFormat, 582
ULAction
gdcm::network::ULAction, 827
ULActionAE6
gdcm::network::ULConnection, 861
ULBasicCallback
gdcm::network::ULBasicCallback, 858
ULConnection
gdcm::network::ULConnection, 860
ULConnectionCallback
gdcm::network::ULConnectionCallback, 862
ULConnectionInfo
gdcm::network::ULConnectionInfo, 864
ULConnectionManager
gdcm::network::ULConnection, 861
gdcm::network::ULConnectionManager, 867
ULEvent

gdcm::network::ULEvent, 869
ULTransitionTable
gdcm::network::ULTransitionTable, 870
ULWritingCallback
gdcm::network::ULWritingCallback, 872
UNKNOWN
gdcm::CSAHeader, 220
gdcm::LookupTable, 473
gdcm::Orientation, 550
gdcm::PhotometricInterpretation, 578
gdcm::PixelFormat, 582
gdcm::Spacing, 717
gdcm::Surface, 749
gdcm::Type, 801
URI
gdcm::MediaStorage, 488
US_SS_OW
gdcm::VR, 898
US_SS
gdcm::VR, 898
UTComp
gdcm, 61
UI
gdcm::VR, 898
uid_1_2_840_10008_15_0_3_1
gdcm::UIDs, 822
uid_1_2_840_10008_15_0_3_10
gdcm::UIDs, 823
uid_1_2_840_10008_15_0_3_11
gdcm::UIDs, 823
uid_1_2_840_10008_15_0_3_12
gdcm::UIDs, 823
uid_1_2_840_10008_15_0_3_13
gdcm::UIDs, 823
uid_1_2_840_10008_15_0_3_14
gdcm::UIDs, 823
uid_1_2_840_10008_15_0_3_15
gdcm::UIDs, 823
uid_1_2_840_10008_15_0_3_16
gdcm::UIDs, 823
uid_1_2_840_10008_15_0_3_17
gdcm::UIDs, 823
uid_1_2_840_10008_15_0_3_18
gdcm::UIDs, 823
uid_1_2_840_10008_15_0_3_19
gdcm::UIDs, 823
uid_1_2_840_10008_15_0_3_2
gdcm::UIDs, 822
uid_1_2_840_10008_15_0_3_20
gdcm::UIDs, 823
uid_1_2_840_10008_15_0_3_21
gdcm::UIDs, 823
uid_1_2_840_10008_15_0_3_22
gdcm::UIDs, 823
uid_1_2_840_10008_15_0_3_23
gdcm::UIDs, 823
uid_1_2_840_10008_15_0_3_24
gdcm::UIDs, 823
uid_1_2_840_10008_15_0_3_25
gdcm::UIDs, 823
uid_1_2_840_10008_15_0_3_26
gdcm::UIDs, 823
uid_1_2_840_10008_15_0_3_27
gdcm::UIDs, 823
uid_1_2_840_10008_15_0_3_28
gdcm::UIDs, 823
uid_1_2_840_10008_15_0_3_29
gdcm::UIDs, 823
uid_1_2_840_10008_15_0_3_3
gdcm::UIDs, 822
uid_1_2_840_10008_15_0_3_30
gdcm::UIDs, 823
uid_1_2_840_10008_15_0_3_31
gdcm::UIDs, 823
uid_1_2_840_10008_15_0_3_4
gdcm::UIDs, 822
uid_1_2_840_10008_15_0_3_5
gdcm::UIDs, 822
uid_1_2_840_10008_15_0_3_6
gdcm::UIDs, 822
uid_1_2_840_10008_15_0_3_7
gdcm::UIDs, 822
uid_1_2_840_10008_15_0_3_8
gdcm::UIDs, 822
uid_1_2_840_10008_15_0_3_9
gdcm::UIDs, 823
uid_1_2_840_10008_15_0_4_1
gdcm::UIDs, 823
uid_1_2_840_10008_15_0_4_2
gdcm::UIDs, 823
uid_1_2_840_10008_15_0_4_3
gdcm::UIDs, 823
uid_1_2_840_10008_15_0_4_4
gdcm::UIDs, 823
uid_1_2_840_10008_15_0_4_5
gdcm::UIDs, 823
uid_1_2_840_10008_15_0_4_6
gdcm::UIDs, 823
uid_1_2_840_10008_15_0_4_7
gdcm::UIDs, 823
uid_1_2_840_10008_15_0_4_8
gdcm::UIDs, 823
uid_1_2_840_10008_1_1
gdcm::UIDs, 817
uid_1_2_840_10008_1_2
gdcm::UIDs, 817
uid_1_2_840_10008_1_20_1
gdcm::UIDs, 818

uid_1_2_840_10008_1_20_1_1
gdcm::UIDs, 818
uid_1_2_840_10008_1_20_2
gdcm::UIDs, 818
uid_1_2_840_10008_1_20_2_1
gdcm::UIDs, 818
uid_1_2_840_10008_1_2_1
gdcm::UIDs, 817
uid_1_2_840_10008_1_2_1_99
gdcm::UIDs, 817
uid_1_2_840_10008_1_2_2
gdcm::UIDs, 817
uid_1_2_840_10008_1_2_4_100
gdcm::UIDs, 818
uid_1_2_840_10008_1_2_4_101
gdcm::UIDs, 823
uid_1_2_840_10008_1_2_4_102
gdcm::UIDs, 823
uid_1_2_840_10008_1_2_4_103
gdcm::UIDs, 823
uid_1_2_840_10008_1_2_4_50
gdcm::UIDs, 817
uid_1_2_840_10008_1_2_4_51
gdcm::UIDs, 817
uid_1_2_840_10008_1_2_4_52
gdcm::UIDs, 817
uid_1_2_840_10008_1_2_4_53
gdcm::UIDs, 817
uid_1_2_840_10008_1_2_4_54
gdcm::UIDs, 817
uid_1_2_840_10008_1_2_4_55
gdcm::UIDs, 817
uid_1_2_840_10008_1_2_4_56
gdcm::UIDs, 817
uid_1_2_840_10008_1_2_4_57
gdcm::UIDs, 817
uid_1_2_840_10008_1_2_4_58
gdcm::UIDs, 817
uid_1_2_840_10008_1_2_4_59
gdcm::UIDs, 817
uid_1_2_840_10008_1_2_4_60
gdcm::UIDs, 817
uid_1_2_840_10008_1_2_4_61
gdcm::UIDs, 817
uid_1_2_840_10008_1_2_4_62
gdcm::UIDs, 817
uid_1_2_840_10008_1_2_4_63
gdcm::UIDs, 817
uid_1_2_840_10008_1_2_4_64
gdcm::UIDs, 817
uid_1_2_840_10008_1_2_4_65
gdcm::UIDs, 817
uid_1_2_840_10008_1_2_4_66
gdcm::UIDs, 817
uid_1_2_840_10008_1_2_4_70
gdcm::UIDs, 817
uid_1_2_840_10008_1_2_4_80
gdcm::UIDs, 817
uid_1_2_840_10008_1_2_4_81
gdcm::UIDs, 817
uid_1_2_840_10008_1_2_4_90
gdcm::UIDs, 817
uid_1_2_840_10008_1_2_4_91
gdcm::UIDs, 817
uid_1_2_840_10008_1_2_4_92
gdcm::UIDs, 818
uid_1_2_840_10008_1_2_4_93
gdcm::UIDs, 818
uid_1_2_840_10008_1_2_4_94
gdcm::UIDs, 818
uid_1_2_840_10008_1_2_4_95
gdcm::UIDs, 818
uid_1_2_840_10008_1_2_5
gdcm::UIDs, 818
uid_1_2_840_10008_1_2_6_1
gdcm::UIDs, 818
uid_1_2_840_10008_1_2_6_2
gdcm::UIDs, 818
uid_1_2_840_10008_1_3_10
gdcm::UIDs, 818
uid_1_2_840_10008_1_40
gdcm::UIDs, 818
uid_1_2_840_10008_1_40_1
gdcm::UIDs, 818
uid_1_2_840_10008_1_42
gdcm::UIDs, 818
uid_1_2_840_10008_1_42_1
gdcm::UIDs, 818
uid_1_2_840_10008_1_4_1_1
gdcm::UIDs, 818
uid_1_2_840_10008_1_4_1_10
gdcm::UIDs, 818
uid_1_2_840_10008_1_4_1_11
gdcm::UIDs, 818
uid_1_2_840_10008_1_4_1_12
gdcm::UIDs, 818
uid_1_2_840_10008_1_4_1_13
gdcm::UIDs, 818
uid_1_2_840_10008_1_4_1_14
gdcm::UIDs, 818
uid_1_2_840_10008_1_4_1_15
gdcm::UIDs, 818
uid_1_2_840_10008_1_4_1_16
gdcm::UIDs, 818
uid_1_2_840_10008_1_4_1_17
gdcm::UIDs, 818
uid_1_2_840_10008_1_4_1_18
gdcm::UIDs, 818

uid_1_2_840_10008_1_4_1_2
gdcm::UIDs, 818
uid_1_2_840_10008_1_4_1_3
gdcm::UIDs, 818
uid_1_2_840_10008_1_4_1_4
gdcm::UIDs, 818
uid_1_2_840_10008_1_4_1_5
gdcm::UIDs, 818
uid_1_2_840_10008_1_4_1_6
gdcm::UIDs, 818
uid_1_2_840_10008_1_4_1_7
gdcm::UIDs, 818
uid_1_2_840_10008_1_4_1_8
gdcm::UIDs, 818
uid_1_2_840_10008_1_4_1_9
gdcm::UIDs, 818
uid_1_2_840_10008_1_4_2_1
gdcm::UIDs, 818
uid_1_2_840_10008_1_4_2_2
gdcm::UIDs, 818
uid_1_2_840_10008_1_9
gdcm::UIDs, 818
uid_1_2_840_10008_2_16_4
gdcm::UIDs, 818
uid_1_2_840_10008_2_6_1
gdcm::UIDs, 818
uid_1_2_840_10008_3_1_1_1
gdcm::UIDs, 818
uid_1_2_840_10008_3_1_2_1_1
gdcm::UIDs, 818
uid_1_2_840_10008_3_1_2_1_4
gdcm::UIDs, 819
uid_1_2_840_10008_3_1_2_2_1
gdcm::UIDs, 819
uid_1_2_840_10008_3_1_2_3_1
gdcm::UIDs, 819
uid_1_2_840_10008_3_1_2_3_2
gdcm::UIDs, 819
uid_1_2_840_10008_3_1_2_3_3
gdcm::UIDs, 819
uid_1_2_840_10008_3_1_2_3_4
gdcm::UIDs, 819
uid_1_2_840_10008_3_1_2_3_5
gdcm::UIDs, 819
uid_1_2_840_10008_3_1_2_5_1
gdcm::UIDs, 819
uid_1_2_840_10008_3_1_2_5_4
gdcm::UIDs, 819
uid_1_2_840_10008_3_1_2_5_5
gdcm::UIDs, 819
uid_1_2_840_10008_3_1_2_6_1
gdcm::UIDs, 819
uid_1_2_840_10008_4_2
gdcm::UIDs, 819
uid_1_2_840_10008_5_1_1_1
gdcm::UIDs, 819
uid_1_2_840_10008_5_1_1_14
gdcm::UIDs, 819
uid_1_2_840_10008_5_1_1_15
gdcm::UIDs, 819
uid_1_2_840_10008_5_1_1_16
gdcm::UIDs, 819
uid_1_2_840_10008_5_1_1_16_376
gdcm::UIDs, 819
uid_1_2_840_10008_5_1_1_17
gdcm::UIDs, 819
uid_1_2_840_10008_5_1_1_17_376
gdcm::UIDs, 819
uid_1_2_840_10008_5_1_1_18
gdcm::UIDs, 819
uid_1_2_840_10008_5_1_1_18_1
gdcm::UIDs, 819
uid_1_2_840_10008_5_1_1_2
gdcm::UIDs, 819
uid_1_2_840_10008_5_1_1_22
gdcm::UIDs, 819
uid_1_2_840_10008_5_1_1_23
gdcm::UIDs, 819
uid_1_2_840_10008_5_1_1_24
gdcm::UIDs, 819
uid_1_2_840_10008_5_1_1_24_1
gdcm::UIDs, 819
uid_1_2_840_10008_5_1_1_25
gdcm::UIDs, 819
uid_1_2_840_10008_5_1_1_26
gdcm::UIDs, 819
uid_1_2_840_10008_5_1_1_27
gdcm::UIDs, 819
uid_1_2_840_10008_5_1_1_29
gdcm::UIDs, 819
uid_1_2_840_10008_5_1_1_30
gdcm::UIDs, 819
uid_1_2_840_10008_5_1_1_31
gdcm::UIDs, 819
uid_1_2_840_10008_5_1_1_32
gdcm::UIDs, 819
uid_1_2_840_10008_5_1_1_33
gdcm::UIDs, 819
uid_1_2_840_10008_5_1_1_4
gdcm::UIDs, 819
uid_1_2_840_10008_5_1_1_4_1
gdcm::UIDs, 819
uid_1_2_840_10008_5_1_1_4_2
gdcm::UIDs, 819
uid_1_2_840_10008_5_1_1_9
gdcm::UIDs, 819
uid_1_2_840_10008_5_1_1_9_1
gdcm::UIDs, 819

uid_1_2_840_10008_5_1_4_1_1_1
gdcm::UIDs, 819
uid_1_2_840_10008_5_1_4_1_1_10
gdcm::UIDs, 820
uid_1_2_840_10008_5_1_4_1_1_104_1
gdcm::UIDs, 821
uid_1_2_840_10008_5_1_4_1_1_104_2
gdcm::UIDs, 821
uid_1_2_840_10008_5_1_4_1_1_11
gdcm::UIDs, 820
uid_1_2_840_10008_5_1_4_1_1_11_1
gdcm::UIDs, 820
uid_1_2_840_10008_5_1_4_1_1_11_2
gdcm::UIDs, 820
uid_1_2_840_10008_5_1_4_1_1_11_3
gdcm::UIDs, 820
uid_1_2_840_10008_5_1_4_1_1_11_4
gdcm::UIDs, 820
uid_1_2_840_10008_5_1_4_1_1_128
gdcm::UIDs, 821
uid_1_2_840_10008_5_1_4_1_1_129
gdcm::UIDs, 821
uid_1_2_840_10008_5_1_4_1_1_12_1
gdcm::UIDs, 820
uid_1_2_840_10008_5_1_4_1_1_12_1_1
gdcm::UIDs, 820
uid_1_2_840_10008_5_1_4_1_1_12_2
gdcm::UIDs, 820
uid_1_2_840_10008_5_1_4_1_1_12_2_1
gdcm::UIDs, 820
uid_1_2_840_10008_5_1_4_1_1_12_3
gdcm::UIDs, 820
uid_1_2_840_10008_5_1_4_1_1_13_1_1
gdcm::UIDs, 820
uid_1_2_840_10008_5_1_4_1_1_13_1_2
gdcm::UIDs, 820
uid_1_2_840_10008_5_1_4_1_1_13_1_3
gdcm::UIDs, 823
uid_1_2_840_10008_5_1_4_1_1_1_1
gdcm::UIDs, 819
uid_1_2_840_10008_5_1_4_1_1_1_1_1
gdcm::UIDs, 819
uid_1_2_840_10008_5_1_4_1_1_1_2
gdcm::UIDs, 820
uid_1_2_840_10008_5_1_4_1_1_1_2_1
gdcm::UIDs, 820
uid_1_2_840_10008_5_1_4_1_1_1_3
gdcm::UIDs, 820
uid_1_2_840_10008_5_1_4_1_1_1_3_1
gdcm::UIDs, 820
uid_1_2_840_10008_5_1_4_1_1_2
gdcm::UIDs, 820
uid_1_2_840_10008_5_1_4_1_1_20
gdcm::UIDs, 820
uid_1_2_840_10008_5_1_4_1_1_2_1
gdcm::UIDs, 820
uid_1_2_840_10008_5_1_4_1_1_3
gdcm::UIDs, 820
uid_1_2_840_10008_5_1_4_1_1_3_1
gdcm::UIDs, 820
uid_1_2_840_10008_5_1_4_1_1_4
gdcm::UIDs, 820
uid_1_2_840_10008_5_1_4_1_1_481_1
gdcm::UIDs, 821
uid_1_2_840_10008_5_1_4_1_1_481_2
gdcm::UIDs, 821
uid_1_2_840_10008_5_1_4_1_1_481_3
gdcm::UIDs, 821
uid_1_2_840_10008_5_1_4_1_1_481_4
gdcm::UIDs, 821
uid_1_2_840_10008_5_1_4_1_1_481_5
gdcm::UIDs, 821
uid_1_2_840_10008_5_1_4_1_1_481_6
gdcm::UIDs, 821
uid_1_2_840_10008_5_1_4_1_1_481_7
gdcm::UIDs, 821
uid_1_2_840_10008_5_1_4_1_1_481_8
gdcm::UIDs, 822
uid_1_2_840_10008_5_1_4_1_1_481_9
gdcm::UIDs, 822
uid_1_2_840_10008_5_1_4_1_1_4_1
gdcm::UIDs, 820
uid_1_2_840_10008_5_1_4_1_1_4_2
gdcm::UIDs, 820
uid_1_2_840_10008_5_1_4_1_1_5
gdcm::UIDs, 820
uid_1_2_840_10008_5_1_4_1_1_6
gdcm::UIDs, 820
uid_1_2_840_10008_5_1_4_1_1_66
gdcm::UIDs, 821
uid_1_2_840_10008_5_1_4_1_1_66_1
gdcm::UIDs, 821
uid_1_2_840_10008_5_1_4_1_1_66_2
gdcm::UIDs, 821
uid_1_2_840_10008_5_1_4_1_1_66_3
gdcm::UIDs, 821
uid_1_2_840_10008_5_1_4_1_1_66_4
gdcm::UIDs, 821
uid_1_2_840_10008_5_1_4_1_1_66_5
gdcm::UIDs, 823
uid_1_2_840_10008_5_1_4_1_1_67
gdcm::UIDs, 821
uid_1_2_840_10008_5_1_4_1_1_6_1
gdcm::UIDs, 820
uid_1_2_840_10008_5_1_4_1_1_6_2
gdcm::UIDs, 823
uid_1_2_840_10008_5_1_4_1_1_7
gdcm::UIDs, 820

uid_1_2_840_10008_5_1_4_1_1_77_1
gdcm::UIDs, 821
uid_1_2_840_10008_5_1_4_1_1_77_1_1
gdcm::UIDs, 821
uid_1_2_840_10008_5_1_4_1_1_77_1_1_1
gdcm::UIDs, 821
uid_1_2_840_10008_5_1_4_1_1_77_1_2
gdcm::UIDs, 821
uid_1_2_840_10008_5_1_4_1_1_77_1_2_1
gdcm::UIDs, 821
uid_1_2_840_10008_5_1_4_1_1_77_1_3
gdcm::UIDs, 821
uid_1_2_840_10008_5_1_4_1_1_77_1_4
gdcm::UIDs, 821
uid_1_2_840_10008_5_1_4_1_1_77_1_4_1
gdcm::UIDs, 821
uid_1_2_840_10008_5_1_4_1_1_77_1_5_1
gdcm::UIDs, 821
uid_1_2_840_10008_5_1_4_1_1_77_1_5_2
gdcm::UIDs, 821
uid_1_2_840_10008_5_1_4_1_1_77_1_5_3
gdcm::UIDs, 821
uid_1_2_840_10008_5_1_4_1_1_77_1_5_4
gdcm::UIDs, 821
uid_1_2_840_10008_5_1_4_1_1_77_1_6
gdcm::UIDs, 823
uid_1_2_840_10008_5_1_4_1_1_77_2
gdcm::UIDs, 821
uid_1_2_840_10008_5_1_4_1_1_7_1
gdcm::UIDs, 820
uid_1_2_840_10008_5_1_4_1_1_7_2
gdcm::UIDs, 820
uid_1_2_840_10008_5_1_4_1_1_7_3
gdcm::UIDs, 820
uid_1_2_840_10008_5_1_4_1_1_7_4
gdcm::UIDs, 820
uid_1_2_840_10008_5_1_4_1_1_8
gdcm::UIDs, 820
uid_1_2_840_10008_5_1_4_1_1_88_1
gdcm::UIDs, 821
uid_1_2_840_10008_5_1_4_1_1_88_11
gdcm::UIDs, 821
uid_1_2_840_10008_5_1_4_1_1_88_2
gdcm::UIDs, 821
uid_1_2_840_10008_5_1_4_1_1_88_22
gdcm::UIDs, 821
uid_1_2_840_10008_5_1_4_1_1_88_3
gdcm::UIDs, 821
uid_1_2_840_10008_5_1_4_1_1_88_33
gdcm::UIDs, 821
uid_1_2_840_10008_5_1_4_1_1_88_4
gdcm::UIDs, 821
uid_1_2_840_10008_5_1_4_1_1_88_40
gdcm::UIDs, 821
uid_1_2_840_10008_5_1_4_1_1_88_50
gdcm::UIDs, 821
uid_1_2_840_10008_5_1_4_1_1_88_59
gdcm::UIDs, 821
uid_1_2_840_10008_5_1_4_1_1_88_65
gdcm::UIDs, 821
uid_1_2_840_10008_5_1_4_1_1_88_67
gdcm::UIDs, 821
uid_1_2_840_10008_5_1_4_1_1_9
gdcm::UIDs, 820
uid_1_2_840_10008_5_1_4_1_1_9_1
gdcm::UIDs, 820
uid_1_2_840_10008_5_1_4_1_1_9_1_1
gdcm::UIDs, 820
uid_1_2_840_10008_5_1_4_1_1_9_1_2
gdcm::UIDs, 820
uid_1_2_840_10008_5_1_4_1_1_9_1_3
gdcm::UIDs, 820
uid_1_2_840_10008_5_1_4_1_1_9_2_1
gdcm::UIDs, 820
uid_1_2_840_10008_5_1_4_1_1_9_3_1
gdcm::UIDs, 820
uid_1_2_840_10008_5_1_4_1_1_9_4_1
gdcm::UIDs, 820
uid_1_2_840_10008_5_1_4_1_2_1_1
gdcm::UIDs, 822
uid_1_2_840_10008_5_1_4_1_2_1_2
gdcm::UIDs, 822
uid_1_2_840_10008_5_1_4_1_2_1_3
gdcm::UIDs, 822
uid_1_2_840_10008_5_1_4_1_2_2_1
gdcm::UIDs, 822
uid_1_2_840_10008_5_1_4_1_2_2_2
gdcm::UIDs, 822
uid_1_2_840_10008_5_1_4_1_2_2_3
gdcm::UIDs, 822
uid_1_2_840_10008_5_1_4_1_2_3_1
gdcm::UIDs, 822
uid_1_2_840_10008_5_1_4_1_2_3_2
gdcm::UIDs, 822
uid_1_2_840_10008_5_1_4_1_2_3_3
gdcm::UIDs, 822
uid_1_2_840_10008_5_1_4_31
gdcm::UIDs, 822
uid_1_2_840_10008_5_1_4_32
gdcm::UIDs, 822
uid_1_2_840_10008_5_1_4_32_1
gdcm::UIDs, 822
uid_1_2_840_10008_5_1_4_32_2
gdcm::UIDs, 822
uid_1_2_840_10008_5_1_4_32_3
gdcm::UIDs, 822
uid_1_2_840_10008_5_1_4_33
gdcm::UIDs, 822

uid_1_2_840_10008_5_1_4_34_1
gdcm::UIDs, 822
uid_1_2_840_10008_5_1_4_34_2
gdcm::UIDs, 822
uid_1_2_840_10008_5_1_4_34_3
gdcm::UIDs, 822
uid_1_2_840_10008_5_1_4_34_4
gdcm::UIDs, 822
uid_1_2_840_10008_5_1_4_34_4_1
gdcm::UIDs, 822
uid_1_2_840_10008_5_1_4_34_4_2
gdcm::UIDs, 822
uid_1_2_840_10008_5_1_4_34_4_3
gdcm::UIDs, 822
uid_1_2_840_10008_5_1_4_34_4_4
gdcm::UIDs, 822
uid_1_2_840_10008_5_1_4_34_5
gdcm::UIDs, 822
uid_1_2_840_10008_5_1_4_37_1
gdcm::UIDs, 822
uid_1_2_840_10008_5_1_4_37_2
gdcm::UIDs, 822
uid_1_2_840_10008_5_1_4_37_3
gdcm::UIDs, 822
uid_1_2_840_10008_5_1_4_38_1
gdcm::UIDs, 822
uid_1_2_840_10008_5_1_4_38_2
gdcm::UIDs, 822
uid_1_2_840_10008_5_1_4_38_3
gdcm::UIDs, 822
uid_1_2_840_10008_5_1_4_41
gdcm::UIDs, 822
uid_1_2_840_10008_5_1_4_42
gdcm::UIDs, 822
UL
gdcm::VR, 898
UltrasoundImageStorage
gdcm::MediaStorage, 486
gdcm::UIDs, 813
UltrasoundImageStorageRetired
gdcm::MediaStorage, 486
gdcm::UIDs, 813
UltrasoundMultiFrameImageStorage
gdcm::MediaStorage, 486
UltrasoundMultiFrameImageStorageRetired
gdcm::MediaStorage, 486
UltrasoundMultiframeImageStorage
gdcm::UIDs, 813
UltrasoundMultiframeImageStorageRetired
gdcm::UIDs, 813
UN
gdcm::VR, 898
UnInstallPipeline
vtkImageColorViewer, 945
UnRegister
gdcm::Object, 541
UndefinedEntityError
gdcm::Parser, 561
underline
gdcm::terminal, 77
UnexpectedStateError
gdcm::Parser, 561
UnifiedProcedureStepEventSOPClass
gdcm::UIDs, 815
UnifiedProcedureStepPullSOPClass
gdcm::UIDs, 815
UnifiedProcedureStepPushSOPClass
gdcm::UIDs, 815
UnifiedProcedureStepWatchSOPClass
gdcm::UIDs, 815
UnifiedWorklistandProcedureStepSOPInstance
gdcm::UIDs, 816
UnifiedWorklistandProcedureStepServiceClass
gdcm::UIDs, 815
Unknown
gdcm::SwapCode, 761
gdcm::TransferSyntax, 794
Unpack
gdcm::Unpacker12Bits, 877
Update
gdcm::Curve, 233
gdcm::Overlay, 557
UpdateDisplayExtent
vtkImageColorViewer, 945
UpdateOrientation
vtkImageColorViewer, 945
UpdatePosition
gdcm::ByteBuffer, 171
US
gdcm::VR, 898
Usage
gdcm::Usage, 878
UsageType
gdcm::Usage, 878
UseDictAlways
gdcm::PythonFilter, 626
gdcm::StringFilter, 742
UserInformation
gdcm::network::UserInformation, 881
UserOption
gdcm::Usage, 878
UserOrdering
gdcm::SerieHelper, 692
UT
gdcm::VR, 898

V
gdcm::Validate, 884

VERBOSE_STYLE
gdcm::Printer, 614
VERTEX
gdcm::MeshPrimitive, 497
VIEWType
gdcm::Surface, 749
VIEWType_END
gdcm::Surface, 749
VL16
gdcm::VR, 898
VL32
gdcm::VR, 898
VLEndoscopicImageStorage
gdcm::MediaStorage, 488
gdcm::UIDs, 814
VLImageStorageTrialRetired
gdcm::UIDs, 814
VLMicroscopicImageStorage
gdcm::MediaStorage, 488
gdcm::UIDs, 814
VLMultiframeImageStorageTrialRetired
gdcm::UIDs, 814
VLPhotographicImageStorage
gdcm::MediaStorage, 488
gdcm::UIDs, 814
VLSlideCoordinatesMicroscopicImageStorage
gdcm::UIDs, 814
VLWholeSlideMicroscopyImageStorage
gdcm::MediaStorage, 488
gdcm::UIDs, 817
VM0
gdcm::VM, 893
VM1
gdcm::VM, 893
VM10
gdcm::VM, 893
VM12
gdcm::VM, 893
VM16
gdcm::VM, 893
VM18
gdcm::VM, 893
VM1_2
gdcm::VM, 894
VM1_3
gdcm::VM, 894
VM1_32
gdcm::VM, 894
VM1_4
gdcm::VM, 894
VM1_5
gdcm::VM, 894
VM1_8
gdcm::VM, 894
VM1_99
gdcm::VM, 894
VM1_n
gdcm::VM, 894
VM2
gdcm::VM, 893
VM24
gdcm::VM, 894
VM256
gdcm::VM, 894
VM28
gdcm::VM, 894
VM2_2n
gdcm::VM, 894
VM2_n
gdcm::VM, 894
VM3
gdcm::VM, 893
VM30_30n
gdcm::VM, 894
VM32
gdcm::VM, 894
VM35
gdcm::VM, 894
VM3_3n
gdcm::VM, 894
VM3_4
gdcm::VM, 894
VM3_n
gdcm::VM, 894
VM4
gdcm::VM, 893
VM47_47n
gdcm::VM, 894
VM4_4n
gdcm::VM, 894
VM5
gdcm::VM, 893
VM6
gdcm::VM, 893
VM6_6n
gdcm::VM, 894
VM7_7n
gdcm::VM, 894
VM8
gdcm::VM, 893
VM9
gdcm::VM, 893
VM99
gdcm::VM, 894
VM_END
gdcm::VM, 894
VMType
gdcm::Attribute, 115

gdcm::Attribute< Group, Element, TVR, VM::VM1 >, 120
gdcm::VM, 893
VOILUTBoxSOPClass
gdcm::UIDs, 813
VR_END
gdcm::VR, 898
VR_VM1
gdcm::VR, 898
VRALL
gdcm::VR, 898
VRASCII
gdcm::VR, 898
VRBINARY
gdcm, 69
gdcm::VR, 898
VRDS16ILLEGAL
gdcmElement.h, 1049
VRField
gdcm::CSAElement, 218
gdcm::DataElement, 243
VRType
gdcm::VR, 898
VRTypeTemplateCase
gdcmVR.h, 1248
VT100
gdcm::terminal, 77
VTK_CMYK
vtkGDCMImageReader.h, 1255
vtkGDCMImageReader2.h, 1256
VTK_INVERSE_LUMINANCE
vtkGDCMImageReader.h, 1255
vtkGDCMImageReader2.h, 1256
VTK_LEGACY
vtkImageColorViewer, 945, 946
VTK_LOOKUP_TABLE
vtkGDCMImageReader.h, 1255
vtkGDCMImageReader2.h, 1256
VTK_YBR
vtkGDCMImageReader.h, 1255
vtkGDCMImageReader2.h, 1256
Valid
gdcm::Preamble, 601
ValidDataSet
gdcm::BaseQuery, 148
Validate
gdcm::PixelFormat, 585
gdcm::Validate, 884
ValidateQuery
gdcm::BaseQuery, 148
gdcm::BaseRootQuery, 151
gdcm::FindPatientRootQuery, 364
gdcm::FindStudyRootQuery, 367
gdcm::ModalityPerformedProcedureStepCreateQuery, 501
gdcm::ModalityPerformedProcedureStepSetQuery, 503
gdcm::MovePatientRootQuery, 514
gdcm::MoveStudyRootQuery, 516
gdcm::WLMFindQuery, 968
Validation
gdcm::Validate, 884
Value
gdcm::Value, 885
value
gdcm::STATIC_ASSERTION_FAILURE< true >, 721
gdcm::SerieHelper::Rule, 659
value_type
gdcm::CodeString, 197
gdcm::LO, 470
gdcm::String, 738
ValueField
gdcm::DataElement, 242
gdcm::PDBElement, 567
ValueLengthField
gdcm::DataElement, 242
ValueMultiplicityField
gdcm::CSAElement, 218
ValuePtr
gdcm::DataElement, 237
ValuesType
gdcm::Scanner, 662
gdcm::StrictScanner, 733
VerificationSOPClass
gdcm::UIDs, 810
Verify
gdcm::Defs, 261
gdcm::Macro, 478
gdcm::Module, 507
Version
gdcm::Version, 888
Video
gdcm::MediaStorage, 488
VideoEndoscopicImageStorage
gdcm::MediaStorage, 487
gdcm::UIDs, 814
VideoMicroscopicImageStorage
gdcm::UIDs, 814
VideoPhotographicImageStorage
gdcm::MediaStorage, 488
gdcm::UIDs, 814
VL
gdcm::VL, 890
VM
gdcm::VM, 894
VR

gdcm::VR, 899
vtkBooleanMacro
vtkGDCMImageReader, 908
vtkGDCMImageReader2, 914, 915
vtkGDCMImageWriter, 921
vtkGDCMThreadedImageReader, 935
vtkGDCMThreadedImageReader2, 938
vtkImageColorViewer, 946
vtkImageMapToColors16, 949
vtkGDCMImageReader, 904
~vtkGDCMImageReader, 907
ApplyInverseVideo, 910
ApplyLookupTable, 910
ApplyPlanarConfiguration, 910
ApplyShiftScale, 910
ApplyYBRToRGB, 910
CanReadFile, 907
Curve, 910
DirectionCosines, 910
ExecuteData, 907
ExecuteInformation, 907
FileNames, 910
FillMedicalImageInformation, 907
ForceRescale, 910
GetDescriptiveName, 907
GetFileExtensions, 907
GetIconImage, 907
GetOverlay, 907
IconDataScalarType, 910
IconImageDataExtent, 910
IconNumberOfScalarComponents, 910
ImageFormat, 910
ImageOrientationPatient, 910
ImagePositionPatient, 910
LoadIconImage, 910
LoadOverlays, 910
LoadSingleFile, 907
LossyFlag, 910
MedicalImageProperties, 910
New, 907
NumberOfIconImages, 910
NumberOfOverlays, 910
PlanarConfiguration, 910
PrintSelf, 907
RequestDataCompat, 908
RequestInformationCompat, 908
Scale, 910
SetCurve, 908
SetFileNames, 908
SetFilePattern, 908
SetFilePrefix, 908
SetMedicalImageProperties, 908
Shift, 910
vtkBooleanMacro, 908
vtkGDCMImageReader, 907
vtkGDCMMedicalImageProperties, 924
vtkGetMacro, 908, 909
vtkGetObjectMacro, 909
vtkGetStringMacro, 909
vtkGetVector3Macro, 909
vtkGetVector6Macro, 909
vtkSetMacro, 909
vtkSetVector6Macro, 909
vtkTypeMacro, 909
vtkGDCMImageReader.h, 1254
VTK_CMYK, 1255
VTK_INVERSE_LUMINANCE, 1255
VTK_LOOKUP_TABLE, 1255
VTK_YBR, 1255
vtkGDCMImageReader2, 911
~vtkGDCMImageReader2, 913
ApplyInverseVideo, 916
ApplyLookupTable, 916
ApplyPlanarConfiguration, 916
ApplyShiftScale, 916
ApplyYBRToRGB, 916
CanReadFile, 913
Curve, 916
DirectionCosines, 916
FillMedicalImageInformation, 913
ForceRescale, 916
GetDescriptiveName, 913
GetFileExtensions, 913
GetIconImage, 914
GetIconImagePort, 914
GetOverlay, 914
GetOverlayPort, 914
IconDataScalarType, 916
IconImageDataExtent, 916
IconNumberOfScalarComponents, 916
ImageFormat, 916
ImageOrientationPatient, 916
ImagePositionPatient, 916
LoadIconImage, 916
LoadOverlays, 917
LoadSingleFile, 914
LossyFlag, 917
New, 914
NumberOfIconImages, 917
NumberOfOverlays, 917
PlanarConfiguration, 917
PrintSelf, 914
ProcessRequest, 914
RequestData, 914
RequestDataCompat, 914
RequestInformation, 914
RequestInformationCompat, 914
Scale, 917

SetCurve, 914
SetFilePattern, 914
SetFilePrefix, 914
SetMedicalImageProperties, 914
Shift, 917
vtkBooleanMacro, 914, 915
vtkGDCMImageReader2, 913
vtkGDCMMedicalImageProperties, 924
vtkGetMacro, 915
vtkGetObjectMacro, 915
vtkGetStringMacro, 915
vtkGetVector3Macro, 915
vtkGetVector6Macro, 915
vtkSetMacro, 915, 916
vtkSetVector6Macro, 916
vtkTypeMacro, 916
vtkGDCMImageReader2.h, 1255
VTK_CMYK, 1256
VTK_INVERSE_LUMINANCE, 1256
VTK_LOOKUP_TABLE, 1256
VTK_YBR, 1256
vtkGDCMImageWriter, 917
∼vtkGDCMImageWriter, 920
CompressionTypes, 919
GetDescriptiveName, 920
GetFileExtensions, 920
GetFileName, 920
JPEG2000_COMPRESSION, 919
JPEG_COMPRESSION, 919
JPEGLS_COMPRESSION, 919
NO_COMPRESSION, 919
New, 920
PrintSelf, 920
RLE_COMPRESSION, 919
SetDirectionCosines, 920
SetDirectionCosinesFromImageOrientationPatient, 920
SetFileNames, 920
SetMedicalImageProperties, 920
vtkBooleanMacro, 921
vtkGDCMImageWriter, 920
vtkGDCMMedicalImageProperties, 924
vtkGetMacro, 921
vtkGetObjectMacro, 921
vtkGetStringMacro, 921
vtkSetMacro, 921, 922
vtkSetStringMacro, 922
vtkTypeMacro, 922
Write, 922
WriteGDCMData, 922
WriteSlice, 922
vtkGDCMImageWriter.h, 1256
vtkGDCMMedicalImageProperties, 922
∼vtkGDCMMedicalImageProperties, 924
Clear, 924
GetFile, 924
New, 924
PrintSelf, 924
PushBackFile, 924
vtkGDCMImageReader, 924
vtkGDCMImageReader2, 924
vtkGDCMImageWriter, 924
vtkGDCMMedicalImageProperties, 924
vtkTypeMacro, 924
vtkGDCMMedicalImageProperties.h, 1257
vtkGDCMPolyDataReader, 925
∼vtkGDCMPolyDataReader, 926
FileName, 927
FillMedicalImageInformation, 926
MedicalImageProperties, 927
New, 926
PrintSelf, 926
RTStructSetProperties, 927
RequestData, 927
RequestData_HemodynamicWaveformStorage, 927
RequestData_RTStructureSetStorage, 927
RequestInformation, 927
RequestInformation_HemodynamicWaveform↩ Storage, 927
RequestInformation_RTStructureSetStorage, 927
vtkGDCMPolyDataReader, 926
vtkGetObjectMacro, 927
vtkGetStringMacro, 927
vtkSetStringMacro, 927
vtkTypeMacro, 927
vtkGDCMPolyDataReader.h, 1257
vtkGDCMPolyDataWriter, 928
∼vtkGDCMPolyDataWriter, 929
InitializeRTStructSet, 929
MedicalImageProperties, 930
New, 929
PrintSelf, 929
RTStructSetProperties, 930
SetMedicalImageProperties, 930
SetNumberOfInputPorts, 930
SetRTStructSetProperties, 930
vtkGDCMPolyDataWriter, 929
vtkTypeMacro, 930
WriteData, 930
WriteRTSTRUCTData, 930
WriteRTSTRUCTInfo, 930
vtkGDCMPolyDataWriter.h, 1258
vtkGDCMTesting, 931
∼vtkGDCMTesting, 932
GetGDCMDataRoot, 932
GetMD5MetaImage, 932
GetMHDMD5FromFile, 932
GetNumberOfMD5MetaImages, 932

GetRAWMD5FromFile, 933
GetVTKDataRoot, 933
MD5MetaImagesType, 932
New, 933
PrintSelf, 933
vtkGDCMTesting, 932
vtkTypeMacro, 933
vtkGDCMTesting.h, 1259
vtkGDCMThreadedImageReader, 933
~vtkGDCMThreadedImageReader, 935
ExecuteData, 935
ExecuteInformation, 935
New, 935
PrintSelf, 935
ReadFiles, 935
RequestDataCompat, 935
vtkBooleanMacro, 935
vtkGDCMThreadedImageReader, 935
vtkGetMacro, 935
vtkSetMacro, 935
vtkTypeMacro, 935
vtkGDCMThreadedImageReader.h, 1259
vtkGDCMThreadedImageReader2, 936
~vtkGDCMThreadedImageReader2, 937
GetFileName, 937
New, 937
PrintSelf, 938
RequestInformation, 938
SetFileName, 938
SetFileNames, 938
SplitExtent, 938
ThreadedRequestData, 938
vtkBooleanMacro, 938
vtkGDCMThreadedImageReader2, 937
vtkGetMacro, 938
vtkGetObjectMacro, 938
vtkGetVector3Macro, 938
vtkGetVector6Macro, 939
vtkSetMacro, 939
vtkSetVector3Macro, 939
vtkSetVector6Macro, 939
vtkTypeMacro, 939
vtkGDCMThreadedImageReader2.h, 1260
vtkGetMacro
vtkGDCMImageReader, 908, 909
vtkGDCMImageReader2, 915
vtkGDCMImageWriter, 921
vtkGDCMThreadedImageReader, 935
vtkGDCMThreadedImageReader2, 938
vtkImageColorViewer, 946
vtkImageMapToColors16, 949
vtkImageMapToWindowLevelColors2, 951
vtkGetObjectMacro
vtkGDCMImageReader, 909
vtkGDCMImageReader2, 915
vtkGDCMImageWriter, 921
vtkGDCMPolyDataReader, 927
vtkGDCMThreadedImageReader2, 938
vtkImageColorViewer, 946
vtkImageMapToColors16, 949
vtkGetStringMacro
vtkGDCMImageReader, 909
vtkGDCMImageReader2, 915
vtkGDCMImageWriter, 921
vtkGDCMPolyDataReader, 927
vtkRTStructSetProperties, 964
vtkGetVector3Macro
vtkGDCMImageReader, 909
vtkGDCMImageReader2, 915
vtkGDCMThreadedImageReader2, 938
vtkGetVector6Macro
vtkGDCMImageReader, 909
vtkGDCMImageReader2, 915
vtkGDCMThreadedImageReader2, 939
vtkImageColorViewer, 940
~vtkImageColorViewer, 943
AddInput, 943
AddInputConnection, 943
FirstRender, 946
GetColorLevel, 943
GetColorWindow, 943
GetInput, 943
GetOffScreenRendering, 943
GetOverlayVisibility, 943
GetPosition, 943
GetSize, 943
GetSliceMax, 943
GetSliceMin, 943
GetSliceRange, 943
GetWindowName, 943
ImageActor, 946
InstallPipeline, 943
Interactor, 946
InteractorStyle, 946
New, 943
OverlayImageActor, 946
PrintSelf, 943
Render, 944
RenderWindow, 946
Renderer, 946
SLICE_ORIENTATION_XY, 942
SLICE_ORIENTATION_XZ, 942
SLICE_ORIENTATION_YZ, 942
SetColorLevel, 944
SetColorWindow, 944
SetDisplayId, 944
SetInput, 944
SetInputConnection, 944

SetOffScreenRendering, 944
SetOverlayVisibility, 944
SetParentId, 944
SetPosition, 944
SetRenderWindow, 944
SetRenderer, 944
SetSize, 944
SetSlice, 945
SetSliceOrientation, 945
SetSliceOrientationToXY, 945
SetSliceOrientationToXZ, 945
SetSliceOrientationToYZ, 945
SetWindowId, 945
SetupInteractor, 945
Slice, 946
SliceOrientation, 946
UnInstallPipeline, 945
UpdateDisplayExtent, 945
UpdateOrientation, 945
VTK_LEGACY, 945, 946
vtkBooleanMacro, 946
vtkGetMacro, 946
vtkGetObjectMacro, 946
vtkImageColorViewer, 943
vtkImageColorViewerCallback, 946
vtkTypeMacro, 946
WindowLevel, 946
vtkImageColorViewer.h, 1260
vtkImageColorViewerCallback
vtkImageColorViewer, 946
vtkImageMapToColors16, 947
~vtkImageMapToColors16, 948
ActiveComponent, 949
DataWasPassed, 949
GetMTime, 948
LookupTable, 949
New, 948
OutputFormat, 949
PassAlphaToOutput, 949
PrintSelf, 948
RequestData, 948
RequestInformation, 949
SetLookupTable, 949
SetOutputFormatToLuminance, 949
SetOutputFormatToLuminanceAlpha, 949
SetOutputFormatToRGBA, 949
SetOutputFormatToRGB, 949
ThreadedRequestData, 949
vtkBooleanMacro, 949
vtkGetMacro, 949
vtkGetObjectMacro, 949
vtkImageMapToColors16, 948
vtkSetMacro, 949
vtkTypeMacro, 949
vtkImageMapToColors16.h, 1261
vtkImageMapToWindowLevelColors2, 950
~vtkImageMapToWindowLevelColors2, 951
Level, 952
New, 951
PrintSelf, 951
RequestData, 951
RequestInformation, 951
ThreadedRequestData, 951
vtkGetMacro, 951
vtkImageMapToWindowLevelColors2, 951
vtkSetMacro, 952
vtkTypeMacro, 952
Window, 952
vtkImageMapToWindowLevelColors2.h, 1261
vtkImagePlanarComponentsToComponents, 952
~vtkImagePlanarComponentsToComponents, 953
New, 953
PrintSelf, 953
RequestData, 954
vtkImagePlanarComponentsToComponents, 953
vtkTypeMacro, 954
vtkImagePlanarComponentsToComponents.h, 1262
vtkImageRGBToYBR.h, 1262
vtkImageRGBToYBR, 954
~vtkImageRGBToYBR, 955
New, 955
PrintSelf, 955
ThreadedExecute, 955
vtkImageRGBToYBR, 955
vtkTypeMacro, 955
vtkImageYBRToRGB.h, 1263
vtkImageYBRToRGB, 956
~vtkImageYBRToRGB, 957
New, 957
PrintSelf, 957
ThreadedExecute, 957
vtkImageYBRToRGB, 957
vtkTypeMacro, 957
vtkLookupTable16, 957
~vtkLookupTable16, 959
Build, 959
GetPointer, 959
MapScalarsThroughTable2, 959
New, 959
PrintSelf, 959
SetNumberOfTableValues, 959
Table16, 959
vtkLookupTable16, 959
vtkTypeMacro, 959
WritePointer, 959
vtkLookupTable16.h, 1263
vtkRTStructSetProperties, 960
~vtkRTStructSetProperties, 962

AddContourReferencedFrameOfReference, 962
AddReferencedFrameOfReference, 962
AddStructureSetROIObservation, 962
AddStructureSetROI, 962
Clear, 962
DeepCopy, 962
GetContourReferencedFrameOfReferenceClassUID, 962
GetContourReferencedFrameOfReferenceInstanceUID, 962
GetNumberOfContourReferencedFrameOfReferences, 963
GetNumberOfReferencedFrameOfReferences, 963
GetNumberOfStructureSetROIs, 963
GetReferencedFrameOfReferenceClassUID, 963
GetReferencedFrameOfReferenceInstanceUID, 963
GetStructureSetObservationNumber, 963
GetStructureSetROIDescription, 963
GetStructureSetROIGenerationAlgorithm, 963
GetStructureSetROIName, 963
GetStructureSetROINumber, 963
GetStructureSetROIObservationLabel, 963
GetStructureSetROIRefFrameRefUID, 963
GetStructureSetRTROIInterpretedType, 963
Internals, 964
New, 963
PrintSelf, 963
ReferenceFrameOfReferenceUID, 965
ReferenceSeriesInstanceUID, 965
SOPInstanceUID, 965
SeriesInstanceUID, 965
StructureSetDate, 965
StructureSetLabel, 965
StructureSetName, 965
StructureSetTime, 965
StudyInstanceUID, 965
vtkGetStringMacro, 964
vtkRTStructSetProperties, 962
vtkSetStringMacro, 964
vtkTypeMacro, 964
vtkRTStructSetProperties.h, 1264
vtkSetMacro
vtkGDCMImageReader, 909
vtkGDCMImageReader2, 915, 916
vtkGDCMImageWriter, 921, 922
vtkGDCMThreadedImageReader, 935
vtkGDCMThreadedImageReader2, 939
vtkImageMapToColors16, 949
vtkImageMapToWindowLevelColors2, 952
vtkSetStringMacro
vtkGDCMImageWriter, 922
vtkGDCMPolyDataReader, 927
vtkRTStructSetProperties, 964
vtkSetVector3Macro
vtkGDCMThreadedImageReader2, 939
vtkSetVector6Macro
vtkGDCMImageReader, 909
vtkGDCMImageReader2, 916
vtkGDCMThreadedImageReader2, 939
vtkTypeMacro
vtkGDCMImageReader, 909
vtkGDCMImageReader2, 916
vtkGDCMImageWriter, 922
vtkGDCMMedicalImageProperties, 924
vtkGDCMPolyDataReader, 927
vtkGDCMPolyDataWriter, 930
vtkGDCMTesting, 933
vtkGDCMThreadedImageReader, 935
vtkGDCMThreadedImageReader2, 939
vtkImageColorViewer, 946
vtkImageMapToColors16, 949
vtkImageMapToWindowLevelColors2, 952
vtkImagePlanarComponentsToComponents, 954
vtkImageRGBToYBR, 955
vtkImageYBRToRGB, 957
vtkLookupTable16, 959
vtkRTStructSetProperties, 964

WIREFRAME
gdcm::Surface, 749
WLMFindQuery
gdcm::WLMFindQuery, 967
WarningOff
gdcm::Trace, 791
WarningOn
gdcm::Trace, 792
Waveform
gdcm::MediaStorage, 488
gdcm::Waveform, 966
WaveformStorageTrialRetired
gdcm::UIDs, 813
what
gdcm::Exception, 318
white
gdcm::terminal, 77
Window
vtkImageMapToWindowLevelColors2, 952
WindowLevel
vtkImageColorViewer, 946
Write
gdcm::ByteValue, 178
gdcm::CSAHeader, 222
gdcm::CommandDataSet, 202
gdcm::DataElement, 242
gdcm::DataSet, 253
gdcm::Element, 293
gdcm::Element< TVR, VM::VM1_n >, 298

gdcm::EncodingImplementation< VR::VRASCII >, 311
gdcm::EncodingImplementation< VR::VRBINARY >, 312
gdcm::ExplicitDataElement, 322
gdcm::File, 328
gdcm::FileAnonymizer, 331
gdcm::FileMetaInformation, 347
gdcm::Fragment, 370
gdcm::ImageWriter, 423
gdcm::ImplicitDataElement, 428
gdcm::Item, 442
gdcm::PGXCodec, 577
gdcm::PNMCodec, 600
gdcm::PixmapWriter, 597
gdcm::Preamble, 601
gdcm::SegmentWriter, 677
gdcm::SequenceOfFragments, 682
gdcm::SequenceOfItems, 689
gdcm::StreamImageWriter, 728
gdcm::SurfaceWriter, 760
gdcm::Tag, 781
gdcm::VRVLSize< 0 >, 903
gdcm::VRVLSize< 1 >, 904
gdcm::ValueIO, 887
gdcm::VL, 890
gdcm::VR, 900
gdcm::Writer, 973
gdcm::network::AAbortPDU, 80
gdcm::network::AAssociateACPDU, 83
gdcm::network::AAssociateRJPDU, 85
gdcm::network::AAssociateRQPDU, 90
gdcm::network::AReleaseRPPDU, 107
gdcm::network::AReleaseRQPDU, 109
gdcm::network::AbstractSyntax, 92
gdcm::network::ApplicationContext, 103
gdcm::network::AsynchronousOperationsWindowSub, 112
gdcm::network::BasePDU, 145
gdcm::network::ImplementationClassUIDSub, 424
gdcm::network::ImplementationUIDSub, 425
gdcm::network::ImplementationVersionNameSub, 426
gdcm::network::MaximumLengthSub, 481
gdcm::network::PDataTFPDU, 565
gdcm::network::PresentationContextAC, 605
gdcm::network::PresentationContextRQ, 610
gdcm::network::PresentationDataValue, 612
gdcm::network::RoleSelectionSub, 658
gdcm::network::SOPClassExtendedNegociationSub, 710
gdcm::network::ServiceClassApplicationInformation, 694
gdcm::network::TransferSyntaxSub, 798
gdcm::network::UserInformation, 882
vtkGDCMImageWriter, 922
Write16
gdcm::VL, 890
WriteASCII
gdcm::Element< TVR, VM::VM1_n >, 298
WriteBuffer
gdcm::ByteValue, 178
gdcm::SequenceOfFragments, 683
WriteBufferAsRGBA
gdcm::LookupTable, 475
WriteData
vtkGDCMPolyDataWriter, 930
WriteFooter
gdcm::DictConverter, 271
WriteGDCMData
vtkGDCMImageWriter, 922
WriteHeader
gdcm::DictConverter, 271
WriteHelpFile
gdcm::BaseQuery, 148
WriteImageInformation
gdcm::StreamImageWriter, 728
WriteImageSubregionRAW
gdcm::StreamImageWriter, 728
WritePointer
vtkLookupTable16, 959
WriteQuery
gdcm::BaseQuery, 148
WriteRTSTRUCTData
vtkGDCMPolyDataWriter, 930
WriteRTSTRUCTInfo
vtkGDCMPolyDataWriter, 930
WriteRawHeader
gdcm::StreamImageWriter, 728
WriteSlice
vtkGDCMImageWriter, 922
Writer
gdcm::Writer, 972

XMLDictReader
gdcm::XMLDictReader, 975
XMLEncoding
gdcm::UIDs, 811
XMLPrinter
gdcm::XMLPrinter, 977
XMLPrivateDictReader
gdcm::XMLPrivateDictReader, 979
XML
gdcm::Printer, 614
XRay3DAngiographicImageStorage
gdcm::MediaStorage, 488
gdcm::UIDs, 814
XRay3DCraniofacialImageStorage

gdcm::UIDs, 814
XRayAngiographicBiPlaneImageStorageRetired
gdcm::MediaStorage, 487
gdcm::UIDs, 814
XRayAngiographicImageStorage
gdcm::MediaStorage, 487
gdcm::UIDs, 814
XRayRadiationDoseSRStorage
gdcm::UIDs, 815
XRayRadiationDoseSR
gdcm::MediaStorage, 488
XRayRadiofluoroscopicImageStorage
gdcm::UIDs, 814
XRayRadiofluoroscopingImageStorage
gdcm::MediaStorage, 487

YBR2RGB
gdcm::ImageChangePhotometricInterpretation, 390
YBR_FULL_422
gdcm::PhotometricInterpretation, 578
YBR_FULL
gdcm::PhotometricInterpretation, 578
YBR_ICT
gdcm::PhotometricInterpretation, 578
YBR_PARTIAL_420
gdcm::PhotometricInterpretation, 578
YBR_PARTIAL_422
gdcm::PhotometricInterpretation, 578
YBR_RCT
gdcm::PhotometricInterpretation, 578
YES
gdcm::Surface, 749
yellow
gdcm::terminal, 77

ZEROED_OUT
gdcm::CSAHeader, 220
ZSpacing
gdcm::IPPSorter, 439
ZTolerance
gdcm::IPPSorter, 439