

# openGPS ISO 5436-2 XML Reference Manual

## 0.01

Generated by Doxygen 1.4.7

Fri Dec 21 18:21:03 2007

## Contents

<b>1</b>	<b>openGPS ISO 5436-2 XML Library Documentation</b>	<b>1</b>
<b>2</b>	<b>openGPS ISO 5436-2 XML Directory Hierarchy</b>	<b>2</b>
<b>3</b>	<b>openGPS ISO 5436-2 XML Namespace Index</b>	<b>2</b>
<b>4</b>	<b>openGPS ISO 5436-2 XML Hierarchical Index</b>	<b>2</b>
<b>5</b>	<b>openGPS ISO 5436-2 XML Class Index</b>	<b>5</b>
<b>6</b>	<b>openGPS ISO 5436-2 XML File Index</b>	<b>8</b>
<b>7</b>	<b>openGPS ISO 5436-2 XML Directory Documentation</b>	<b>13</b>
<b>8</b>	<b>openGPS ISO 5436-2 XML Namespace Documentation</b>	<b>16</b>
<b>9</b>	<b>openGPS ISO 5436-2 XML Class Documentation</b>	<b>45</b>
<b>10</b>	<b>openGPS ISO 5436-2 XML File Documentation</b>	<b>254</b>

## 1 openGPS ISO 5436-2 XML Library Documentation



Figure 1: width=10cm

The openGPS ISO5436-2 XML Library contains an implementation of the X3P file format according to the ISO 5436-2 standard.

The homepage of openGPS is [www.opengps.eu](http://www.opengps.eu).

## 2 openGPS ISO 5436-2 XML Directory Hierarchy

### 2.1 openGPS ISO 5436-2 XML Directories

This directory hierarchy is sorted roughly, but not completely, alphabetically:

ISO5436_XML	16
trunk	16
src	16
ISO5436_2_XML	15
c	13
cxx	13
ISO5436_2_XML_Demo	15

## 3 openGPS ISO 5436-2 XML Namespace Index

### 3.1 openGPS ISO 5436-2 XML Namespace List

Here is a list of all namespaces with brief descriptions:

<a href="#">OpenGPS</a>	16
<a href="#">std</a> (The Standard namespace for the C++ library )	18
<a href="#">xml_schema</a>	18
<a href="#">xsd</a> (C++ namespace for the <a href="http://www.opengps.eu/xsd/">http://www.opengps.eu/xsd/</a> schema namespace )	25

## 4 openGPS ISO 5436-2 XML Hierarchical Index

### 4.1 openGPS ISO 5436-2 XML Class Hierarchy

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

_OGPS_DATA_POINT	45
_OGPS_ISO5436_2_HANDLE	45
_OGPS_POINT_ITERATOR	45
_OGPS_POINT_VECTOR	46

<b>xsd::AxesType</b>	<b>46</b>
<b>xsd::AxisDescriptionType</b>	<b>55</b>
<b>xsd::AxisType</b>	<b>64</b>
<b>xsd::DataLinkType</b>	<b>79</b>
<b>xsd::DataListType</b>	<b>89</b>
<b>OpenGPS::DataPointImpl</b>	<b>93</b>
<b>OpenGPS::DataPointImpl::_OGPS_DATA_POINT_VALUE</b>	<b>95</b>
<b>OpenGPS::DataPointParser</b>	<b>96</b>
<b>OpenGPS::DoubleDataPointParser</b>	<b>104</b>
<b>OpenGPS::FloatDataPointParser</b>	<b>113</b>
<b>OpenGPS::Int16DataPointParser</b>	<b>124</b>
<b>OpenGPS::Int32DataPointParser</b>	<b>126</b>
<b>OpenGPS::MissingDataPointParser</b>	<b>152</b>
<b>xsd::DataType</b>	<b>97</b>
<b>xsd::Datum</b>	<b>101</b>
<b>OpenGPS::Environment</b>	<b>106</b>
<b>xsd::FeatureType</b>	<b>110</b>
<b>xsd::InstrumentType</b>	<b>115</b>
<b>OpenGPS::ISO5436_2Container</b>	<b>129</b>
<b>xsd::ISO5436_2Type</b>	<b>137</b>
<b>xsd::MatrixDimensionType</b>	<b>146</b>
<b>OpenGPS::PointBuffer</b>	<b>153</b>
<b>OpenGPS::DoublePointBuffer</b>	<b>105</b>
<b>OpenGPS::FloatPointBuffer</b>	<b>114</b>
<b>OpenGPS::Int16PointBuffer</b>	<b>125</b>
<b>OpenGPS::Int32PointBuffer</b>	<b>127</b>
<b>OpenGPS::PointIteratorImpl</b>	<b>156</b>

OpenGPS::PointVectorInputBinaryFileStream	158
OpenGPS::PointVectorInputStringStream	159
OpenGPS::PointVectorInvariantLocale	160
OpenGPS::PointVectorOutputBinaryFileStream	161
OpenGPS::PointVectorOutputStringStream	161
OpenGPS::PointVectorParser	162
OpenGPS::PointVectorParserBuilder	164
OpenGPS::PointVectorProxy	165
OpenGPS::PointVectorProxy::DataPointProxy	167
OpenGPS::PointVectorProxy::DataPointProxyContext	169
OpenGPS::PointVectorProxy::UDataPointProxyContext	171
OpenGPS::PointVectorProxy::VDataPointProxyContext	172
OpenGPS::PointVectorProxy::WDataPointProxyContext	173
OpenGPS::PointVectorProxyContext	174
OpenGPS::PointVectorReaderContext	175
OpenGPS::BinaryPointVectorReaderContext	76
OpenGPS::BinaryLSBPointVectorReaderContext	69
OpenGPS::BinaryMSBPointVectorReaderContext	72
OpenGPS::XmlPointVectorReaderContext	246
OpenGPS::PointVectorWhitespaceFacet	177
OpenGPS::PointVectorWriterContext	178
OpenGPS::BinaryPointVectorWriterContext	77
OpenGPS::BinaryLSBPointVectorWriterContext	71
OpenGPS::BinaryMSBPointVectorWriterContext	74
OpenGPS::XmlPointVectorWriterContext	249
xsd::ProbingSystemType	180
xsd::Record1Type	185

<a href="#">xsd::Record2Type</a>	<a href="#">192</a>
<a href="#">xsd::Record3Type</a>	<a href="#">205</a>
<a href="#">xsd::Record4Type</a>	<a href="#">215</a>
<a href="#">xsd::RotationMatrixElementType</a>	<a href="#">218</a>
<a href="#">xsd::RotationType</a>	<a href="#">221</a>
<a href="#">xsd::Type</a>	<a href="#">237</a>
<a href="#">OpenGPS::ValidBuffer</a>	<a href="#">241</a>
<a href="#">OpenGPS::VectorBuffer</a>	<a href="#">243</a>
<a href="#">OpenGPS::VectorBufferBuilder</a>	<a href="#">245</a>
<a href="#">OpenGPS::ZipOutputStream</a>	<a href="#">252</a>
<a href="#">OpenGPS::ZipStreamBuffer</a>	<a href="#">252</a>

## 5 openGPS ISO 5436-2 XML Class Index

### 5.1 openGPS ISO 5436-2 XML Class List

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

<a href="#">_OGPS_DATA_POINT</a>	<a href="#">45</a>
<a href="#">_OGPS_ISO5436_2_HANDLE</a>	<a href="#">45</a>
<a href="#">_OGPS_POINT_ITERATOR</a>	<a href="#">45</a>
<a href="#">_OGPS_POINT_VECTOR</a>	<a href="#">46</a>
<a href="#">xsd::AxesType</a> (Class corresponding to the AxesType schema type )	<a href="#">46</a>
<a href="#">xsd::AxisDescriptionType</a> (Class corresponding to the AxisDescription-Type schema type )	<a href="#">55</a>
<a href="#">xsd::AxisType</a> (Enumeration class corresponding to the AxisType schema type )	<a href="#">64</a>
<a href="#">OpenGPS::BinaryLSBPointVectorReaderContext</a>	<a href="#">69</a>
<a href="#">OpenGPS::BinaryLSBPointVectorWriterContext</a>	<a href="#">71</a>
<a href="#">OpenGPS::BinaryMSBPointVectorReaderContext</a>	<a href="#">72</a>
<a href="#">OpenGPS::BinaryMSBPointVectorWriterContext</a>	<a href="#">74</a>

<a href="#">OpenGPS::BinaryPointVectorReaderContext</a>	76
<a href="#">OpenGPS::BinaryPointVectorWriterContext</a>	77
<a href="#">xsd::DataLinkType</a> (Class corresponding to the DataLinkType schema type )	79
<a href="#">xsd::DataListType</a> (Class corresponding to the DataListType schema type )	89
<a href="#">OpenGPS::DataPointImpl</a>	93
<a href="#">OpenGPS::DataPointImpl::_OGPS_DATA_POINT_VALUE</a>	95
<a href="#">OpenGPS::DataPointParser</a>	96
<a href="#">xsd::DataType</a> (Enumeration class corresponding to the DataType schema type )	97
<a href="#">xsd::Datum</a> (Class corresponding to the Datum schema type )	101
<a href="#">OpenGPS::DoubleDataPointParser</a>	104
<a href="#">OpenGPS::DoublePointBuffer</a>	105
<a href="#">OpenGPS::Environment</a>	106
<a href="#">xsd::FeatureType</a> (Class corresponding to the FeatureType schema type )	110
<a href="#">OpenGPS::FloatDataPointParser</a>	113
<a href="#">OpenGPS::FloatPointBuffer</a>	114
<a href="#">xsd::InstrumentType</a> (Class corresponding to the InstrumentType schema type )	115
<a href="#">OpenGPS::Int16DataPointParser</a>	124
<a href="#">OpenGPS::Int16PointBuffer</a>	125
<a href="#">OpenGPS::Int32DataPointParser</a>	126
<a href="#">OpenGPS::Int32PointBuffer</a>	127
<a href="#">OpenGPS::ISO5436_2Container</a>	129
<a href="#">xsd::ISO5436_2Type</a> (Class corresponding to the ISO5436_2Type schema type )	137
<a href="#">xsd::MatrixDimensionType</a> (Class corresponding to the MatrixDimension-Type schema type )	146
<a href="#">OpenGPS::MissingDataPointParser</a>	152

<a href="#">OpenGPS::PointBuffer</a>	153
<a href="#">OpenGPS::PointIteratorImpl</a>	156
<a href="#">OpenGPS::PointVectorInputBinaryFileStream</a>	158
<a href="#">OpenGPS::PointVectorInputStringStream</a>	159
<a href="#">OpenGPS::PointVectorInvariantLocale</a>	160
<a href="#">OpenGPS::PointVectorOutputBinaryFileStream</a>	161
<a href="#">OpenGPS::PointVectorOutputStringStream</a>	161
<a href="#">OpenGPS::PointVectorParser</a>	162
<a href="#">OpenGPS::PointVectorParserBuilder</a>	164
<a href="#">OpenGPS::PointVectorProxy</a>	165
<a href="#">OpenGPS::PointVectorProxy::DataPointProxy</a>	167
<a href="#">OpenGPS::PointVectorProxy::DataPointProxyContext</a>	169
<a href="#">OpenGPS::PointVectorProxy::UDataPointProxyContext</a>	171
<a href="#">OpenGPS::PointVectorProxy::VDataPointProxyContext</a>	172
<a href="#">OpenGPS::PointVectorProxy::WDataPointProxyContext</a>	173
<a href="#">OpenGPS::PointVectorProxyContext</a>	174
<a href="#">OpenGPS::PointVectorReaderContext</a>	175
<a href="#">OpenGPS::PointVectorWhitespaceFacet</a>	177
<a href="#">OpenGPS::PointVectorWriterContext</a>	178
<a href="#">xsd::ProbingSystemType</a> (Class corresponding to the ProbingSystemType schema type )	180
<a href="#">xsd::Record1Type</a> (Class corresponding to the Record1Type schema type )	185
<a href="#">xsd::Record2Type</a> (Class corresponding to the Record2Type schema type )	192
<a href="#">xsd::Record3Type</a> (Class corresponding to the Record3Type schema type )	205
<a href="#">xsd::Record4Type</a> (Class corresponding to the Record4Type schema type )	215



<a href="#">xsd::RotationMatrixElementType</a> (Class corresponding to the RotationMatrixElementType schema type )	218
<a href="#">xsd::RotationType</a> (Class corresponding to the RotationType schema type )	221
<a href="#">xsd::Type</a> (Enumeration class corresponding to the Type schema type )	237
<a href="#">OpenGPS::ValidBuffer</a>	241
<a href="#">OpenGPS::VectorBuffer</a>	243
<a href="#">OpenGPS::VectorBufferBuilder</a>	245
<a href="#">OpenGPS::XmlPointVectorReaderContext</a>	246
<a href="#">OpenGPS::XmlPointVectorWriterContext</a>	249
<a href="#">OpenGPS::ZipOutputStream</a>	252
<a href="#">OpenGPS::ZipStreamBuffer</a>	252

## 6 openGPS ISO 5436-2 XML File Index

### 6.1 openGPS ISO 5436-2 XML File List

Here is a list of all files with brief descriptions:

<a href="#">S:/openGPS/ISO5436_XML/trunk/src/Doxygen.cpp</a> (Title page of documentation, no source code )	254
<a href="#">S:/openGPS/ISO5436_XML/trunk/src/ISO5436_2_XML/iso5436_2_-xsd.cxx</a>	285
<a href="#">S:/openGPS/ISO5436_XML/trunk/src/ISO5436_2_XML/iso5436_2_-xsd.hxx</a> (Generated from iso5436_2.xsd )	290
<a href="#">S:/openGPS/ISO5436_XML/trunk/src/ISO5436_2_XML/c/data_-point.cxx</a>	254
<a href="#">S:/openGPS/ISO5436_XML/trunk/src/ISO5436_2_XML/c/data_-point.hxx</a>	255
<a href="#">S:/openGPS/ISO5436_XML/trunk/src/ISO5436_2_XML/c/iso5436_2_-handle.cxx</a>	255
<a href="#">S:/openGPS/ISO5436_XML/trunk/src/ISO5436_2_XML/c/iso5436_2_-handle.hxx</a>	258
<a href="#">S:/openGPS/ISO5436_XML/trunk/src/ISO5436_2_XML/c/point_-iterator.cxx</a>	258

S:/openGPS/ISO5436_XML/trunk/src/ISO5436_2_XML/c/ <a href="#">point_iterator.hxx</a>	260
S:/openGPS/ISO5436_XML/trunk/src/ISO5436_2_XML/c/ <a href="#">point_vector.cxx</a>	260
S:/openGPS/ISO5436_XML/trunk/src/ISO5436_2_XML/c/ <a href="#">point_vector.hxx</a>	264
S:/openGPS/ISO5436_XML/trunk/src/ISO5436_2_XML/cxx/ <a href="#">auto_ptr_types.hxx</a>	264
S:/openGPS/ISO5436_XML/trunk/src/ISO5436_2_XML/cxx/ <a href="#">binary_lsb_point_vector_reader_context.cxx</a>	265
S:/openGPS/ISO5436_XML/trunk/src/ISO5436_2_XML/cxx/ <a href="#">binary_lsb_point_vector_reader_context.hxx</a>	265
S:/openGPS/ISO5436_XML/trunk/src/ISO5436_2_XML/cxx/ <a href="#">binary_lsb_point_vector_writer_context.cxx</a>	265
S:/openGPS/ISO5436_XML/trunk/src/ISO5436_2_XML/cxx/ <a href="#">binary_lsb_point_vector_writer_context.hxx</a>	265
S:/openGPS/ISO5436_XML/trunk/src/ISO5436_2_XML/cxx/ <a href="#">binary_msb_point_vector_reader_context.cxx</a>	266
S:/openGPS/ISO5436_XML/trunk/src/ISO5436_2_XML/cxx/ <a href="#">binary_msb_point_vector_reader_context.hxx</a>	266
S:/openGPS/ISO5436_XML/trunk/src/ISO5436_2_XML/cxx/ <a href="#">binary_msb_point_vector_writer_context.cxx</a>	266
S:/openGPS/ISO5436_XML/trunk/src/ISO5436_2_XML/cxx/ <a href="#">binary_msb_point_vector_writer_context.hxx</a>	267
S:/openGPS/ISO5436_XML/trunk/src/ISO5436_2_XML/cxx/ <a href="#">binary_point_vector_reader_context.hxx</a>	267
S:/openGPS/ISO5436_XML/trunk/src/ISO5436_2_XML/cxx/ <a href="#">binary_point_vector_writer_context.cxx</a>	267
S:/openGPS/ISO5436_XML/trunk/src/ISO5436_2_XML/cxx/ <a href="#">binary_point_vector_writer_context.hxx</a>	267
S:/openGPS/ISO5436_XML/trunk/src/ISO5436_2_XML/cxx/ <a href="#">data_point_impl.cxx</a>	268
S:/openGPS/ISO5436_XML/trunk/src/ISO5436_2_XML/cxx/ <a href="#">data_point_impl.hxx</a>	268

S:/openGPS/ISO5436_XML/trunk/src/ISO5436_2_XML/cxx/data_ point_parser.hxx	268
S:/openGPS/ISO5436_XML/trunk/src/ISO5436_2_XML/cxx/double_ data_point_parser.cxx	269
S:/openGPS/ISO5436_XML/trunk/src/ISO5436_2_XML/cxx/double_ data_point_parser.hxx	269
S:/openGPS/ISO5436_XML/trunk/src/ISO5436_2_XML/cxx/double_ point_buffer.cxx	269
S:/openGPS/ISO5436_XML/trunk/src/ISO5436_2_XML/cxx/double_ point_buffer.hxx	269
S:/openGPS/ISO5436_XML/trunk/src/ISO5436_2_ XML/cxx/environment.cxx	270
S:/openGPS/ISO5436_XML/trunk/src/ISO5436_2_ XML/cxx/environment.hxx	270
S:/openGPS/ISO5436_XML/trunk/src/ISO5436_2_XML/cxx/float_data_ point_parser.cxx	270
S:/openGPS/ISO5436_XML/trunk/src/ISO5436_2_XML/cxx/float_data_ point_parser.hxx	271
S:/openGPS/ISO5436_XML/trunk/src/ISO5436_2_XML/cxx/float_ point_buffer.cxx	271
S:/openGPS/ISO5436_XML/trunk/src/ISO5436_2_XML/cxx/float_ point_buffer.hxx	271
S:/openGPS/ISO5436_XML/trunk/src/ISO5436_2_XML/cxx/int16_data_ point_parser.cxx	271
S:/openGPS/ISO5436_XML/trunk/src/ISO5436_2_XML/cxx/int16_data_ point_parser.hxx	272
S:/openGPS/ISO5436_XML/trunk/src/ISO5436_2_XML/cxx/int16_ point_buffer.cxx	272
S:/openGPS/ISO5436_XML/trunk/src/ISO5436_2_XML/cxx/int16_ point_buffer.hxx	272
S:/openGPS/ISO5436_XML/trunk/src/ISO5436_2_XML/cxx/int32_data_ point_parser.cxx	272
S:/openGPS/ISO5436_XML/trunk/src/ISO5436_2_XML/cxx/int32_data_ point_parser.hxx	273

S:/openGPS/ISO5436_XML/trunk/src/ISO5436_2_XML/cxx/int32_- point_buffer.cxx	273
S:/openGPS/ISO5436_XML/trunk/src/ISO5436_2_XML/cxx/int32_- point_buffer.hxx	273
S:/openGPS/ISO5436_XML/trunk/src/ISO5436_2_XML/cxx/iso5436_- 2.cxx	273
S:/openGPS/ISO5436_XML/trunk/src/ISO5436_2_XML/cxx/iso5436_2_- container.cxx	274
S:/openGPS/ISO5436_XML/trunk/src/ISO5436_2_XML/cxx/iso5436_2_- container.hxx	275
S:/openGPS/ISO5436_XML/trunk/src/ISO5436_2_XML/cxx/missing_- data_point_parser.cxx	275
S:/openGPS/ISO5436_XML/trunk/src/ISO5436_2_XML/cxx/missing_- data_point_parser.hxx	275
S:/openGPS/ISO5436_XML/trunk/src/ISO5436_2_XML/cxx/point_- buffer.cxx	276
S:/openGPS/ISO5436_XML/trunk/src/ISO5436_2_XML/cxx/point_- buffer.hxx	276
S:/openGPS/ISO5436_XML/trunk/src/ISO5436_2_XML/cxx/point_- iterator_impl.cxx	276
S:/openGPS/ISO5436_XML/trunk/src/ISO5436_2_XML/cxx/point_- iterator_impl.hxx	276
S:/openGPS/ISO5436_XML/trunk/src/ISO5436_2_XML/cxx/point_- vector.cxx	263
S:/openGPS/ISO5436_XML/trunk/src/ISO5436_2_XML/cxx/point_- vector_iostream.cxx	277
S:/openGPS/ISO5436_XML/trunk/src/ISO5436_2_XML/cxx/point_- vector_iostream.hxx	277
S:/openGPS/ISO5436_XML/trunk/src/ISO5436_2_XML/cxx/point_- vector_parser.cxx	277
S:/openGPS/ISO5436_XML/trunk/src/ISO5436_2_XML/cxx/point_- vector_parser.hxx	278
S:/openGPS/ISO5436_XML/trunk/src/ISO5436_2_XML/cxx/point_- vector_parser_builder.cxx	278

S:/openGPS/ISO5436_XML/trunk/src/ISO5436_2_XML/cxx/ <a href="#">point_vector_parser_builder.hxx</a>	278
S:/openGPS/ISO5436_XML/trunk/src/ISO5436_2_XML/cxx/ <a href="#">point_vector_proxy.cxx</a>	278
S:/openGPS/ISO5436_XML/trunk/src/ISO5436_2_XML/cxx/ <a href="#">point_vector_proxy.hxx</a>	279
S:/openGPS/ISO5436_XML/trunk/src/ISO5436_2_XML/cxx/ <a href="#">point_vector_proxy_context.cxx</a>	279
S:/openGPS/ISO5436_XML/trunk/src/ISO5436_2_XML/cxx/ <a href="#">point_vector_proxy_context.hxx</a>	279
S:/openGPS/ISO5436_XML/trunk/src/ISO5436_2_XML/cxx/ <a href="#">point_vector_reader_context.hxx</a>	279
S:/openGPS/ISO5436_XML/trunk/src/ISO5436_2_XML/cxx/ <a href="#">point_vector_writer_context.hxx</a>	280
S:/openGPS/ISO5436_XML/trunk/src/ISO5436_2_XML/cxx/ <a href="#">stdafx.hxx</a>	280
S:/openGPS/ISO5436_XML/trunk/src/ISO5436_2_XML/cxx/ <a href="#">stream_types.hxx</a>	281
S:/openGPS/ISO5436_XML/trunk/src/ISO5436_2_XML/cxx/ <a href="#">string.cxx</a>	281
S:/openGPS/ISO5436_XML/trunk/src/ISO5436_2_XML/cxx/ <a href="#">valid_buffer.cxx</a>	281
S:/openGPS/ISO5436_XML/trunk/src/ISO5436_2_XML/cxx/ <a href="#">valid_buffer.hxx</a>	281
S:/openGPS/ISO5436_XML/trunk/src/ISO5436_2_XML/cxx/ <a href="#">vector_buffer.cxx</a>	282
S:/openGPS/ISO5436_XML/trunk/src/ISO5436_2_XML/cxx/ <a href="#">vector_buffer.hxx</a>	282
S:/openGPS/ISO5436_XML/trunk/src/ISO5436_2_XML/cxx/ <a href="#">vector_buffer_builder.cxx</a>	282
S:/openGPS/ISO5436_XML/trunk/src/ISO5436_2_XML/cxx/ <a href="#">vector_buffer_builder.hxx</a>	282
S:/openGPS/ISO5436_XML/trunk/src/ISO5436_2_XML/cxx/ <a href="#">win32_environment.cxx</a>	283
S:/openGPS/ISO5436_XML/trunk/src/ISO5436_2_XML/cxx/ <a href="#">win32_environment.hxx</a>	283

<a href="#">S:/openGPS/ISO5436_XML/trunk/src/ISO5436_2_XML/cxx/xml_point_-vector_reader_context.cxx</a>	283
<a href="#">S:/openGPS/ISO5436_XML/trunk/src/ISO5436_2_XML/cxx/xml_point_-vector_reader_context.hxx</a>	283
<a href="#">S:/openGPS/ISO5436_XML/trunk/src/ISO5436_2_XML/cxx/xml_point_-vector_writer_context.cxx</a>	283
<a href="#">S:/openGPS/ISO5436_XML/trunk/src/ISO5436_2_XML/cxx/xml_point_-vector_writer_context.hxx</a>	284
<a href="#">S:/openGPS/ISO5436_XML/trunk/src/ISO5436_2_XML/cxx/zip_-stream_buffer.cxx</a>	284
<a href="#">S:/openGPS/ISO5436_XML/trunk/src/ISO5436_2_XML/cxx/zip_-stream_buffer.hxx</a>	284
<a href="#">S:/openGPS/ISO5436_XML/trunk/src/ISO5436_2_XML_-Demo/ISO5436_2_XML_Demo.cpp</a>	299

## 7 openGPS ISO 5436-2 XML Directory Documentation

### 7.1 S:/openGPS/ISO5436\_XML/trunk/src/ISO5436\_2\_XML/c/ Directory Reference

#### Files

- file [data\\_point.cxx](#)
- file [data\\_point.hxx](#)
- file [iso5436\\_2\\_handle.cxx](#)
- file [iso5436\\_2\\_handle.hxx](#)
- file [point\\_iterator.cxx](#)
- file [point\\_iterator.hxx](#)
- file [point\\_vector.cxx](#)
- file [point\\_vector.hxx](#)

### 7.2 S:/openGPS/ISO5436\_XML/trunk/src/ISO5436\_2\_XML/cxx/ Directory Reference

#### Files

- file [auto\\_ptr\\_types.hxx](#)
- file [binary\\_lsb\\_point\\_vector\\_reader\\_context.cxx](#)
- file [binary\\_lsb\\_point\\_vector\\_reader\\_context.hxx](#)
- file [binary\\_lsb\\_point\\_vector\\_writer\\_context.cxx](#)

- file [binary\\_lsb\\_point\\_vector\\_writer\\_context.hxx](#)
- file [binary\\_msb\\_point\\_vector\\_reader\\_context.cxx](#)
- file [binary\\_msb\\_point\\_vector\\_reader\\_context.hxx](#)
- file [binary\\_msb\\_point\\_vector\\_writer\\_context.cxx](#)
- file [binary\\_msb\\_point\\_vector\\_writer\\_context.hxx](#)
- file [binary\\_point\\_vector\\_reader\\_context.hxx](#)
- file [binary\\_point\\_vector\\_writer\\_context.cxx](#)
- file [binary\\_point\\_vector\\_writer\\_context.hxx](#)
- file [data\\_point\\_impl.cxx](#)
- file [data\\_point\\_impl.hxx](#)
- file [data\\_point\\_parser.hxx](#)
- file [double\\_data\\_point\\_parser.cxx](#)
- file [double\\_data\\_point\\_parser.hxx](#)
- file [double\\_point\\_buffer.cxx](#)
- file [double\\_point\\_buffer.hxx](#)
- file [environment.cxx](#)
- file [environment.hxx](#)
- file [float\\_data\\_point\\_parser.cxx](#)
- file [float\\_data\\_point\\_parser.hxx](#)
- file [float\\_point\\_buffer.cxx](#)
- file [float\\_point\\_buffer.hxx](#)
- file [int16\\_data\\_point\\_parser.cxx](#)
- file [int16\\_data\\_point\\_parser.hxx](#)
- file [int16\\_point\\_buffer.cxx](#)
- file [int16\\_point\\_buffer.hxx](#)
- file [int32\\_data\\_point\\_parser.cxx](#)
- file [int32\\_data\\_point\\_parser.hxx](#)
- file [int32\\_point\\_buffer.cxx](#)
- file [int32\\_point\\_buffer.hxx](#)
- file [iso5436\\_2.cxx](#)
- file [iso5436\\_2\\_container.cxx](#)
- file [iso5436\\_2\\_container.hxx](#)
- file [missing\\_data\\_point\\_parser.cxx](#)
- file [missing\\_data\\_point\\_parser.hxx](#)
- file [point\\_buffer.cxx](#)
- file [point\\_buffer.hxx](#)
- file [point\\_iterator\\_impl.cxx](#)
- file [point\\_iterator\\_impl.hxx](#)
- file [point\\_vector.cxx](#)
- file [point\\_vector\\_iostream.cxx](#)
- file [point\\_vector\\_iostream.hxx](#)
- file [point\\_vector\\_parser.cxx](#)
- file [point\\_vector\\_parser.hxx](#)
- file [point\\_vector\\_parser\\_builder.cxx](#)
- file [point\\_vector\\_parser\\_builder.hxx](#)
- file [point\\_vector\\_proxy.cxx](#)

- file [point\\_vector\\_proxy.hxx](#)
- file [point\\_vector\\_proxy\\_context.cxx](#)
- file [point\\_vector\\_proxy\\_context.hxx](#)
- file [point\\_vector\\_reader\\_context.hxx](#)
- file [point\\_vector\\_writer\\_context.hxx](#)
- file [stdafx.hxx](#)
- file [stream\\_types.hxx](#)
- file [string.cxx](#)
- file [valid\\_buffer.cxx](#)
- file [valid\\_buffer.hxx](#)
- file [vector\\_buffer.cxx](#)
- file [vector\\_buffer.hxx](#)
- file [vector\\_buffer\\_builder.cxx](#)
- file [vector\\_buffer\\_builder.hxx](#)
- file [win32\\_environment.cxx](#)
- file [win32\\_environment.hxx](#)
- file [xml\\_point\\_vector\\_reader\\_context.cxx](#)
- file [xml\\_point\\_vector\\_reader\\_context.hxx](#)
- file [xml\\_point\\_vector\\_writer\\_context.cxx](#)
- file [xml\\_point\\_vector\\_writer\\_context.hxx](#)
- file [zip\\_stream\\_buffer.cxx](#)
- file [zip\\_stream\\_buffer.hxx](#)

### **7.3 S:/openGPS/ISO5436\_XML/trunk/src/ISO5436\_2\_XML/ Directory Reference**

#### **Directories**

- directory [c](#)
- directory [cxx](#)

#### **Files**

- file [iso5436\\_2\\_xsd.cxx](#)
- file [iso5436\\_2\\_xsd.hxx](#)

*Generated from iso5436\_2.xsd.*

### **7.4 S:/openGPS/ISO5436\_XML/trunk/src/ISO5436\_2\_XML\_Demo/ Directory Reference**

#### **Files**

- file [ISO5436\\_2\\_XML\\_Demo.cpp](#)



## 7.5 S:/openGPS/ISO5436\_XML/ Directory Reference

### Directories

- directory [trunk](#)

## 7.6 S:/openGPS/ISO5436\_XML/trunk/src/ Directory Reference

### Directories

- directory [ISO5436\\_2\\_XML](#)
- directory [ISO5436\\_2\\_XML\\_Demo](#)

### Files

- file [Doxygen.cpp](#)  
*Title page of documentation, no source code.*

## 7.7 S:/openGPS/ISO5436\_XML/trunk/ Directory Reference

### Directories

- directory [src](#)

# 8 openGPS ISO 5436-2 XML Namespace Documentation

## 8.1 OpenGPS Namespace Reference

### Classes

- class [BinaryLSBPointVectorReaderContext](#)
- class [BinaryLSBPointVectorWriterContext](#)
- class [BinaryMSBPointVectorReaderContext](#)
- class [BinaryMSBPointVectorWriterContext](#)
- class [BinaryPointVectorReaderContext](#)
- class [BinaryPointVectorWriterContext](#)
- class [DataPointImpl](#)
- class [DataPointParser](#)
- class [DoubleDataPointParser](#)
- class [DoublePointBuffer](#)
- class [Environment](#)
- class [FloatDataPointParser](#)

- class [FloatPointBuffer](#)
- class [Int16DataPointParser](#)
- class [Int16PointBuffer](#)
- class [Int32DataPointParser](#)
- class [Int32PointBuffer](#)
- class [ISO5436\\_2Container](#)
- class [MissingDataPointParser](#)
- class [PointBuffer](#)
- class [PointIteratorImpl](#)
- class [PointVectorInputBinaryFileStream](#)
- class [PointVectorInputStringStream](#)
- class [PointVectorInvariantLocale](#)
- class [PointVectorOutputBinaryFileStream](#)
- class [PointVectorOutputStringStream](#)
- class [PointVectorParser](#)
- class [PointVectorParserBuilder](#)
- class [PointVectorProxy](#)
- class [PointVectorProxyContext](#)
- class [PointVectorReaderContext](#)
- class [PointVectorWhitespaceFacet](#)
- class [PointVectorWriterContext](#)
- class [ValidBuffer](#)
- class [VectorBuffer](#)
- class [VectorBufferBuilder](#)
- class [XmlPointVectorReaderContext](#)
- class [XmlPointVectorWriterContext](#)
- class [ZipOutputStream](#)
- class [ZipStreamBuffer](#)

## Typedefs

- typedef std::auto\_ptr< [xsd::ISO5436\\_2Type](#) > [ISO5436\\_2TypeAutoPtr](#)
- typedef std::ostream [OutStringStream](#)
- typedef std::auto\_ptr< [PointIterator](#) > [PointIteratorAutoPtr](#)
- typedef std::auto\_ptr< [PointVectorBase](#) > [PointVectorAutoPtr](#)
- typedef std::auto\_ptr< [PointVectorParserBuilder](#) > [PointVectorParserBuilderAutoPtr](#)
- typedef std::auto\_ptr< [VectorBuffer](#) > [VectorBufferAutoPtr](#)
- typedef std::auto\_ptr< [VectorBufferBuilder](#) > [VectorBufferBuilderAutoPtr](#)

### 8.1.1 Typedef Documentation

**8.1.1.1** typedef std::auto\_ptr<[xsd::ISO5436\\_2Type](#)> [OpenGPS::ISO5436\\_2TypeAutoPtr](#)

**8.1.1.2** typedef std::ostream [OpenGPS::OutStringStream](#)

8.1.1.3 `typedef std::auto_ptr<PointIterator> OpenGPS::PointIteratorAutoPtr`

8.1.1.4 `typedef std::auto_ptr<PointVectorBase> OpenGPS::PointVectorAutoPtr`

8.1.1.5 `typedef std::auto_ptr<PointVectorParserBuilder> OpenGPS::PointVectorParserBuilderAutoPtr`

8.1.1.6 `typedef std::auto_ptr<VectorBuffer> OpenGPS::VectorBufferAutoPtr`

8.1.1.7 `typedef std::auto_ptr<VectorBufferBuilder> OpenGPS::VectorBufferBuilderAutoPtr`

## 8.2 std Namespace Reference

### 8.2.1 Detailed Description

The Standard namespace for the C++ library.

## 8.3 xml\_schema Namespace Reference

### Typedefs

- `typedef ::xsd::cxx::tree::base64_binary< wchar_t, simple_type > base64\_binary`
- `typedef bool boolean`
- `typedef ::xsd::cxx::tree::bounds< wchar_t > bounds`
- `typedef ::xsd::cxx::tree::buffer< wchar_t > buffer`
- `typedef signed char byte`
- `typedef ::xsd::cxx::tree::date< wchar_t, simple_type > date`
- `typedef ::xsd::cxx::tree::date_time< wchar_t, simple_type > date\_time`
- `typedef ::xsd::cxx::tree::day< wchar_t, simple_type > day`
- `typedef double decimal`
- `typedef ::xsd::cxx::tree::diagnostics< wchar_t > diagnostics`
- `typedef double double\_`
- `typedef ::xsd::cxx::tree::duplicate_id< wchar_t > duplicate\_id`
- `typedef ::xsd::cxx::tree::duration< wchar_t, simple_type > duration`
- `typedef ::xsd::cxx::tree::entities< wchar_t, simple_type, entity > entities`
- `typedef ::xsd::cxx::tree::entity< wchar_t, ncname > entity`
- `typedef ::xsd::cxx::tree::error< wchar_t > error`
- `typedef ::xsd::cxx::xml::error_handler< wchar_t > error\_handler`
- `typedef ::xsd::cxx::tree::exception< wchar_t > exception`

- typedef ::xsd::cxx::tree::expected\_attribute< wchar\_t > expected\_attribute
- typedef ::xsd::cxx::tree::expected\_element< wchar\_t > expected\_element
- typedef ::xsd::cxx::tree::expected\_text\_content< wchar\_t > expected\_text\_content
- typedef ::xsd::cxx::tree::flags flags
- typedef float float\_
- typedef ::xsd::cxx::tree::hex\_binary< wchar\_t, simple\_type > hex\_binary
- typedef ::xsd::cxx::tree::id< wchar\_t, ncname > id
- typedef ::xsd::cxx::tree::idref< type, wchar\_t, ncname > idref
- typedef ::xsd::cxx::tree::idrefs< wchar\_t, simple\_type, idref > idrefs
- typedef int int\_
- typedef long long integer
- typedef ::xsd::cxx::tree::language< wchar\_t, token > language
- typedef long long long\_
- typedef ::xsd::cxx::tree::month< wchar\_t, simple\_type > month
- typedef ::xsd::cxx::tree::month\_day< wchar\_t, simple\_type > month\_day
- typedef ::xsd::cxx::tree::name< wchar\_t, token > name
- typedef ::xsd::cxx::xml::dom::namespace\_info< wchar\_t > namespace\_info
- typedef ::xsd::cxx::xml::dom::namespace\_infomap< wchar\_t > namespace\_infomap
- typedef ::xsd::cxx::tree::ncname< wchar\_t, name > ncname
- typedef integer negative\_integer
- typedef ::xsd::cxx::tree::nmtoken< wchar\_t, token > nmtoken
- typedef ::xsd::cxx::tree::nmtokens< wchar\_t, simple\_type, nmtoken > nmtokens
- typedef ::xsd::cxx::tree::no\_namespace\_mapping< wchar\_t > no\_namespace\_mapping
- typedef ::xsd::cxx::tree::no\_prefix\_mapping< wchar\_t > no\_prefix\_mapping
- typedef ::xsd::cxx::tree::no\_type\_info< wchar\_t > no\_type\_info
- typedef integer non\_negative\_integer
- typedef integer non\_positive\_integer
- typedef ::xsd::cxx::tree::normalized\_string< wchar\_t, string > normalized\_string
- typedef ::xsd::cxx::tree::not\_derived< wchar\_t > not\_derived
- typedef ::xsd::cxx::tree::parsing< wchar\_t > parsing
- typedef integer positive\_integer
- typedef ::xsd::cxx::tree::properties< wchar\_t > properties
- typedef ::xsd::cxx::tree::qname< wchar\_t, simple\_type, uri, ncname > qname
- typedef ::xsd::cxx::tree::serialization< wchar\_t > serialization
- typedef ::xsd::cxx::tree::severity severity
- typedef short short\_
- typedef ::xsd::cxx::tree::simple\_type< type > simple\_type
- typedef ::xsd::cxx::tree::string< wchar\_t, simple\_type > string
- typedef ::xsd::cxx::tree::time< wchar\_t, simple\_type > time
- typedef ::xsd::cxx::tree::token< wchar\_t, normalized\_string > token
- typedef ::xsd::cxx::tree::type type
- typedef ::xsd::cxx::tree::unexpected\_element< wchar\_t > unexpected\_element

- typedef `::xsd::cxx::tree::unexpected_enumerator< wchar_t > unexpected_enumerator`
- typedef unsigned char `unsigned_byte`
- typedef unsigned int `unsigned_int`
- typedef unsigned long long `unsigned_long`
- typedef unsigned short `unsigned_short`
- typedef `::xsd::cxx::tree::uri< wchar_t, simple_type > uri`
- typedef `::xsd::cxx::tree::xsi_already_in_use< wchar_t > xsi_already_in_use`
- typedef `::xsd::cxx::tree::year< wchar_t, simple_type > year`
- typedef `::xsd::cxx::tree::year_month< wchar_t, simple_type > year_month`

#### Variables

- const XMLCh \*const `tree_node_key` = `::xsd::cxx::tree::user_data_keys::node`

#### 8.3.1 Typedef Documentation

**8.3.1.1** typedef `::xsd::cxx::tree::base64_binary< wchar_t, simple_type > xml_schema::base64_binary`

**8.3.1.2** typedef bool `xml_schema::boolean`

**8.3.1.3** typedef `::xsd::cxx::tree::bounds< wchar_t > xml_schema::bounds`

**8.3.1.4** typedef `::xsd::cxx::tree::buffer< wchar_t > xml_schema::buffer`

**8.3.1.5** typedef signed char `xml_schema::byte`

**8.3.1.6** typedef `::xsd::cxx::tree::date< wchar_t, simple_type > xml_schema::date`

**8.3.1.7** typedef `::xsd::cxx::tree::date_time< wchar_t, simple_type > xml_schema::date_time`

**8.3.1.8** typedef `::xsd::cxx::tree::day< wchar_t, simple_type > xml_schema::day`

**8.3.1.9** typedef double `xml_schema::decimal`

**8.3.1.10** typedef `::xsd::cxx::tree::diagnostics< wchar_t > xml_schema::diagnostics`

8.3.1.11 typedef double [xml\\_schema::double\\_](#)

8.3.1.12 typedef [::xsd::cxx::tree::duplicate\\_id](#)< [wchar\\_t](#) > [xml\\_schema::duplicate\\_id](#)

8.3.1.13 typedef [::xsd::cxx::tree::duration](#)< [wchar\\_t](#), [simple\\_type](#) > [xml\\_schema::duration](#)

8.3.1.14 typedef [::xsd::cxx::tree::entities](#)< [wchar\\_t](#), [simple\\_type](#), [entity](#) > [xml\\_schema::entities](#)

8.3.1.15 typedef [::xsd::cxx::tree::entity](#)< [wchar\\_t](#), [ncname](#) > [xml\\_schema::entity](#)

8.3.1.16 typedef [::xsd::cxx::tree::error](#)< [wchar\\_t](#) > [xml\\_schema::error](#)

8.3.1.17 typedef [::xsd::cxx::xml::error\\_handler](#)< [wchar\\_t](#) > [xml\\_schema::error\\_handler](#)

8.3.1.18 typedef [::xsd::cxx::tree::exception](#)< [wchar\\_t](#) > [xml\\_schema::exception](#)

8.3.1.19 typedef [::xsd::cxx::tree::expected\\_attribute](#)< [wchar\\_t](#) > [xml\\_schema::expected\\_attribute](#)

8.3.1.20 typedef [::xsd::cxx::tree::expected\\_element](#)< [wchar\\_t](#) > [xml\\_schema::expected\\_element](#)

8.3.1.21 typedef [::xsd::cxx::tree::expected\\_text\\_content](#)< [wchar\\_t](#) > [xml\\_schema::expected\\_text\\_content](#)

8.3.1.22 typedef [::xsd::cxx::tree::flags](#) [xml\\_schema::flags](#)

8.3.1.23 typedef float [xml\\_schema::float\\_](#)

8.3.1.24 typedef [::xsd::cxx::tree::hex\\_binary](#)< [wchar\\_t](#), [simple\\_type](#) > [xml\\_schema::hex\\_binary](#)

8.3.1.25 typedef [::xsd::cxx::tree::id](#)< [wchar\\_t](#), [ncname](#) > [xml\\_schema::id](#)

8.3.1.26 typedef `::xsd::cxx::tree::idref`< `type`, `wchar_t`, `ncname` > `xml_schema::idref`

8.3.1.27 typedef `::xsd::cxx::tree::idrefs`< `wchar_t`, `simple_type`, `idref` > `xml_schema::idrefs`

8.3.1.28 typedef int `xml_schema::int_`

8.3.1.29 typedef long long `xml_schema::integer`

8.3.1.30 typedef `::xsd::cxx::tree::language`< `wchar_t`, `token` > `xml_schema::language`

8.3.1.31 typedef long long `xml_schema::long_`

8.3.1.32 typedef `::xsd::cxx::tree::month`< `wchar_t`, `simple_type` > `xml_schema::month`

8.3.1.33 typedef `::xsd::cxx::tree::month_day`< `wchar_t`, `simple_type` > `xml_schema::month_day`

8.3.1.34 typedef `::xsd::cxx::tree::name`< `wchar_t`, `token` > `xml_schema::name`

8.3.1.35 typedef `::xsd::cxx::xml::dom::namespace_info`< `wchar_t` > `xml_schema::namespace_info`

8.3.1.36 typedef `::xsd::cxx::xml::dom::namespace_infomap`< `wchar_t` > `xml_schema::namespace_infomap`

8.3.1.37 typedef `::xsd::cxx::tree::ncname`< `wchar_t`, `name` > `xml_schema::ncname`

8.3.1.38 typedef `integer` `xml_schema::negative_integer`

8.3.1.39 typedef `::xsd::cxx::tree::nmtoken`< `wchar_t`, `token` > `xml_schema::nmtoken`

8.3.1.40 typedef `::xsd::cxx::tree::nmtokens`< `wchar_t`, `simple_type`, `nmtoken`> `xml_schema::nmtokens`

8.3.1.41 typedef `::xsd::cxx::tree::no_namespace_mapping< wchar_t > xml_schema::no_namespace_mapping`

8.3.1.42 typedef `::xsd::cxx::tree::no_prefix_mapping< wchar_t > xml_schema::no_prefix_mapping`

8.3.1.43 typedef `::xsd::cxx::tree::no_type_info< wchar_t > xml_schema::no_type_info`

8.3.1.44 typedef `integer xml_schema::non_negative_integer`

8.3.1.45 typedef `integer xml_schema::non_positive_integer`

8.3.1.46 typedef `::xsd::cxx::tree::normalized_string< wchar_t, string > xml_schema::normalized_string`

8.3.1.47 typedef `::xsd::cxx::tree::not_derived< wchar_t > xml_schema::not_derived`

8.3.1.48 typedef `::xsd::cxx::tree::parsing< wchar_t > xml_schema::parsing`

8.3.1.49 typedef `integer xml_schema::positive_integer`

8.3.1.50 typedef `::xsd::cxx::tree::properties< wchar_t > xml_schema::properties`

8.3.1.51 typedef `::xsd::cxx::tree::qname< wchar_t, simple_type, uri, ncname > xml_schema::qname`

8.3.1.52 typedef `::xsd::cxx::tree::serialization< wchar_t > xml_schema::serialization`

8.3.1.53 typedef `::xsd::cxx::tree::severity xml_schema::severity`

8.3.1.54 typedef `short xml_schema::short_`

8.3.1.55 typedef `::xsd::cxx::tree::simple_type<type> xml_schema::simple_type`



8.3.1.56 typedef `::xsd::cxx::tree::string`< `wchar_t`, `simple_type` > `xml_schema::string`

8.3.1.57 typedef `::xsd::cxx::tree::time`< `wchar_t`, `simple_type` > `xml_schema::time`

8.3.1.58 typedef `::xsd::cxx::tree::token`< `wchar_t`, `normalized_string` > `xml_schema::token`

8.3.1.59 typedef `::xsd::cxx::tree::type` `xml_schema::type`

8.3.1.60 typedef `::xsd::cxx::tree::unexpected_element`< `wchar_t` > `xml_schema::unexpected_element`

8.3.1.61 typedef `::xsd::cxx::tree::unexpected_enumerator`< `wchar_t` > `xml_schema::unexpected_enumerator`

8.3.1.62 typedef unsigned char `xml_schema::unsigned_byte`

8.3.1.63 typedef unsigned int `xml_schema::unsigned_int`

8.3.1.64 typedef unsigned long long `xml_schema::unsigned_long`

8.3.1.65 typedef unsigned short `xml_schema::unsigned_short`

8.3.1.66 typedef `::xsd::cxx::tree::uri`< `wchar_t`, `simple_type` > `xml_schema::uri`

8.3.1.67 typedef `::xsd::cxx::tree::xsi_already_in_use`< `wchar_t` > `xml_schema::xsi_already_in_use`

8.3.1.68 typedef `::xsd::cxx::tree::year`< `wchar_t`, `simple_type` > `xml_schema::year`

8.3.1.69 typedef `::xsd::cxx::tree::year_month`< `wchar_t`, `simple_type` > `xml_schema::year_month`

### 8.3.2 Variable Documentation

8.3.2.1 const XMLCh\* const `xml_schema::tree_node_key` = `::xsd::cxx::tree::user_data_keys::node`

## 8.4 xsd Namespace Reference

### 8.4.1 Detailed Description

C++ namespace for the <http://www.opengps.eu/xsd/> schema namespace.

#### Classes

- class [AxesType](#)  
*Class corresponding to the AxesType schema type.*
- class [AxisDescriptionType](#)  
*Class corresponding to the AxisDescriptionType schema type.*
- class [AxisType](#)  
*Enumeration class corresponding to the AxisType schema type.*
- class [DataLinkType](#)  
*Class corresponding to the DataLinkType schema type.*
- class [DataListType](#)  
*Class corresponding to the DataListType schema type.*
- class [DataType](#)  
*Enumeration class corresponding to the DataType schema type.*
- class [Datum](#)  
*Class corresponding to the Datum schema type.*
- class [FeatureType](#)  
*Class corresponding to the FeatureType schema type.*
- class [InstrumentType](#)  
*Class corresponding to the InstrumentType schema type.*
- class [ISO5436\\_2Type](#)  
*Class corresponding to the ISO5436\_2Type schema type.*
- class [MatrixDimensionType](#)  
*Class corresponding to the MatrixDimensionType schema type.*
- class [ProbingSystemType](#)  
*Class corresponding to the ProbingSystemType schema type.*
- class [Record1Type](#)  
*Class corresponding to the Record1Type schema type.*

- class [Record2Type](#)  
*Class corresponding to the Record2Type schema type.*
- class [Record3Type](#)  
*Class corresponding to the Record3Type schema type.*
- class [Record4Type](#)  
*Class corresponding to the Record4Type schema type.*
- class [RotationMatrixElementType](#)  
*Class corresponding to the RotationMatrixElementType schema type.*
- class [RotationType](#)  
*Class corresponding to the RotationType schema type.*
- class [Type](#)  
*Enumeration class corresponding to the Type schema type.*

#### Serialization functions for the ISO5436\_2 document root.

The only global element: The root node

- `::xsd::cxx::xml::dom::auto_ptr< ::xercesc::DOMDocument > ISO5436_2 (const ::xsd::ISO5436_2Type &x, const ::xml_schema::namespace_infomap &m, ::xml_schema::flags f=0)`  
*Serialize to a new Xerces-C++ DOM document.*
- `void ISO5436_2 (::xercesc::DOMDocument &d, const ::xsd::ISO5436_2Type &x, ::xml_schema::flags f=0)`  
*Serialize to an existing Xerces-C++ DOM document.*
- `void ISO5436_2 (::xercesc::XMLFormatTarget &ft, const ::xsd::ISO5436_2Type &x, const ::xml_schema::namespace_infomap &m, ::xercesc::DOMErrorHandler &eh, const ::std::wstring &e=L"UTF-8", ::xml_schema::flags f=0)`  
*Serialize to a Xerces-C++ XML format target with a Xerces-C++ DOM error handler.*
- `void ISO5436_2 (::xercesc::XMLFormatTarget &ft, const ::xsd::ISO5436_2Type &x, const ::xml_schema::namespace_infomap &m, ::xml_schema::error_handler &eh, const ::std::wstring &e=L"UTF-8", ::xml_schema::flags f=0)`  
*Serialize to a Xerces-C++ XML format target with an error handler.*
- `void ISO5436_2 (::xercesc::XMLFormatTarget &ft, const ::xsd::ISO5436_2Type &x, const ::xml_schema::namespace_infomap &m, const ::std::wstring &e=L"UTF-8", ::xml_schema::flags f=0)`

*Serialize to a Xerces-C++ XML format target.*

- void [ISO5436\\_2](#) (::std::ostream &os, const ::xsd::ISO5436\_2Type &x, const ::xml\_schema::namespace\_infomap &m, ::xercesc::DOMErrorHandler &eh, const ::std::wstring &e=L"UTF-8", ::xml\_schema::flags f=0)

*Serialize to a standard output stream with a Xerces-C++ DOM error handler.*

- void [ISO5436\\_2](#) (::std::ostream &os, const ::xsd::ISO5436\_2Type &x, const ::xml\_schema::namespace\_infomap &m, ::xml\_schema::error\_handler &eh, const ::std::wstring &e=L"UTF-8", ::xml\_schema::flags f=0)

*Serialize to a standard output stream with an error handler.*

- void [ISO5436\\_2](#) (::std::ostream &os, const ::xsd::ISO5436\_2Type &x, const ::xml\_schema::namespace\_infomap &m, const ::std::wstring &e=L"UTF-8", ::xml\_schema::flags f=0)

*Serialize to a standard output stream.*

#### Parsing functions for the ISO5436\_2 document root.

The only global element: The root node

- ::std::auto\_ptr< ::xsd::ISO5436\_2Type > [ISO5436\\_2](#) (::xercesc::DOMDocument \*d, ::xml\_schema::flags f=0, const ::xml\_schema::properties &p=::xml\_schema::properties())

*Parse a Xerces-C++ DOM document.*

- ::std::auto\_ptr< ::xsd::ISO5436\_2Type > [ISO5436\\_2](#) (const ::xercesc::DOMDocument &d, ::xml\_schema::flags f=0, const ::xml\_schema::properties &p=::xml\_schema::properties())

*Parse a Xerces-C++ DOM document.*

- ::std::auto\_ptr< ::xsd::ISO5436\_2Type > [ISO5436\\_2](#) (const ::xercesc::DOMInputSource &is, ::xercesc::DOMErrorHandler &eh, ::xml\_schema::flags f=0, const ::xml\_schema::properties &p=::xml\_schema::properties())

*Parse a Xerces-C++ DOM input source with a Xerces-C++ DOM error handler.*

- ::std::auto\_ptr< ::xsd::ISO5436\_2Type > [ISO5436\\_2](#) (const ::xercesc::DOMInputSource &is, ::xml\_schema::error\_handler &eh, ::xml\_schema::flags f=0, const ::xml\_schema::properties &p=::xml\_schema::properties())

*Parse a Xerces-C++ DOM input source with an error handler.*

- ::std::auto\_ptr< ::xsd::ISO5436\_2Type > [ISO5436\\_2](#) (const ::xercesc::DOMInputSource &is, ::xml\_schema::flags f=0, const ::xml\_schema::properties &p=::xml\_schema::properties())

*Parse a Xerces-C++ DOM input source.*

- `::std::auto_ptr< ::xsd::ISO5436_2Type > ISO5436_2 (::std::istream &is, const ::std::wstring &id, ::xercesc::DOMErrorHandler &eh, ::xml_schema::flags f=0, const ::xml_schema::properties &p=::xml_schema::properties())`

*Parse a standard input stream with a resource id and a Xerces-C++ DOM error handler.*

- `::std::auto_ptr< ::xsd::ISO5436_2Type > ISO5436_2 (::std::istream &is, const ::std::wstring &id, ::xml_schema::error_handler &eh, ::xml_schema::flags f=0, const ::xml_schema::properties &p=::xml_schema::properties())`

*Parse a standard input stream with a resource id and an error handler.*

- `::std::auto_ptr< ::xsd::ISO5436_2Type > ISO5436_2 (::std::istream &is, const ::std::wstring &id, ::xml_schema::flags f=0, const ::xml_schema::properties &p=::xml_schema::properties())`

*Parse a standard input stream with a resource id.*

- `::std::auto_ptr< ::xsd::ISO5436_2Type > ISO5436_2 (::std::istream &is, ::xercesc::DOMErrorHandler &eh, ::xml_schema::flags f=0, const ::xml_schema::properties &p=::xml_schema::properties())`

*Parse a standard input stream with a Xerces-C++ DOM error handler.*

- `::std::auto_ptr< ::xsd::ISO5436_2Type > ISO5436_2 (::std::istream &is, ::xml_schema::error_handler &eh, ::xml_schema::flags f=0, const ::xml_schema::properties &p=::xml_schema::properties())`

*Parse a standard input stream with an error handler.*

- `::std::auto_ptr< ::xsd::ISO5436_2Type > ISO5436_2 (::std::istream &is, ::xml_schema::flags f=0, const ::xml_schema::properties &p=::xml_schema::properties())`

*Parse a standard input stream.*

- `::std::auto_ptr< ::xsd::ISO5436_2Type > ISO5436_2 (const ::std::wstring &uri, ::xercesc::DOMErrorHandler &eh, ::xml_schema::flags f=0, const ::xml_schema::properties &p=::xml_schema::properties())`

*Parse a URI or a local file with a Xerces-C++ DOM error handler.*

- `::std::auto_ptr< ::xsd::ISO5436_2Type > ISO5436_2 (const ::std::wstring &uri, ::xml_schema::error_handler &eh, ::xml_schema::flags f=0, const ::xml_schema::properties &p=::xml_schema::properties())`

*Parse a URI or a local file with an error handler.*

- `::std::auto_ptr< ::xsd::ISO5436_2Type > ISO5436_2 (const ::std::wstring &uri, ::xml_schema::flags f=0, const ::xml_schema::properties &p=::xml_schema::properties())`

*Parse a URI or a local file.*

## Functions

- bool `operator!=` (const [RotationType](#) &x, const [RotationType](#) &y)
- bool `operator!=` (const [MatrixDimensionType](#) &x, const [MatrixDimensionType](#) &y)
- bool `operator!=` (const [DataLinkType](#) &x, const [DataLinkType](#) &y)
- bool `operator!=` (const [DataListType](#) &x, const [DataListType](#) &y)
- bool `operator!=` (const [ProbingSystemType](#) &x, const [ProbingSystemType](#) &y)
- bool `operator!=` (const [InstrumentType](#) &x, const [InstrumentType](#) &y)
- bool `operator!=` (const [AxisDescriptionType](#) &x, const [AxisDescriptionType](#) &y)
- bool `operator!=` (const [AxesType](#) &x, const [AxesType](#) &y)
- bool `operator!=` (const [Record4Type](#) &x, const [Record4Type](#) &y)
- bool `operator!=` (const [Record3Type](#) &x, const [Record3Type](#) &y)
- bool `operator!=` (const [Record2Type](#) &x, const [Record2Type](#) &y)
- bool `operator!=` (const [ISO5436\\_2Type](#) &x, const [ISO5436\\_2Type](#) &y)
- bool `operator!=` (const [Record1Type](#) &x, const [Record1Type](#) &y)
- void `operator<<` (::xsd::cxx::tree::list\_stream< wchar\_t > &l, const [Datum](#) &i)
- void `operator<<` (::xercesc::DOMAttr &a, const [Datum](#) &i)
- void `operator<<` (::xercesc::DOMELEMENT &e, const [Datum](#) &i)
- void `operator<<` (::xsd::cxx::tree::list\_stream< wchar\_t > &l, const [Type](#) &i)
- void `operator<<` (::xercesc::DOMAttr &a, const [Type](#) &i)
- void `operator<<` (::xercesc::DOMELEMENT &e, const [Type](#) &i)
- void `operator<<` (::xsd::cxx::tree::list\_stream< wchar\_t > &l, const [DataType](#) &i)
- void `operator<<` (::xercesc::DOMAttr &a, const [DataType](#) &i)
- void `operator<<` (::xercesc::DOMELEMENT &e, const [DataType](#) &i)
- void `operator<<` (::xsd::cxx::tree::list\_stream< wchar\_t > &l, const [AxisType](#) &i)
- void `operator<<` (::xercesc::DOMAttr &a, const [AxisType](#) &i)
- void `operator<<` (::xercesc::DOMELEMENT &e, const [AxisType](#) &i)
- void `operator<<` (::xsd::cxx::tree::list\_stream< wchar\_t > &l, const [FeatureType](#) &i)
- void `operator<<` (::xercesc::DOMAttr &a, const [FeatureType](#) &i)
- void `operator<<` (::xercesc::DOMELEMENT &e, const [FeatureType](#) &i)
- void `operator<<` (::xsd::cxx::tree::list\_stream< wchar\_t > &l, const [RotationMatrixElementType](#) &i)
- void `operator<<` (::xercesc::DOMAttr &a, const [RotationMatrixElementType](#) &i)
- void `operator<<` (::xercesc::DOMELEMENT &e, const [RotationMatrixElementType](#) &i)
- void `operator<<` (::xercesc::DOMELEMENT &e, const [RotationType](#) &i)
- void `operator<<` (::xercesc::DOMELEMENT &e, const [MatrixDimensionType](#) &i)
- void `operator<<` (::xercesc::DOMELEMENT &e, const [DataLinkType](#) &i)
- void `operator<<` (::xercesc::DOMELEMENT &e, const [DataListType](#) &i)
- void `operator<<` (::xercesc::DOMELEMENT &e, const [ProbingSystemType](#) &i)
- void `operator<<` (::xercesc::DOMELEMENT &e, const [InstrumentType](#) &i)

- void [operator<<](#) (::xercesc::DOMElement &e, const [AxisDescriptionType](#) &i)
- void [operator<<](#) (::xercesc::DOMElement &e, const [AxesType](#) &i)
- void [operator<<](#) (::xercesc::DOMElement &e, const [Record4Type](#) &i)
- void [operator<<](#) (::xercesc::DOMElement &e, const [Record3Type](#) &i)
- void [operator<<](#) (::xercesc::DOMElement &e, const [Record2Type](#) &i)
- void [operator<<](#) (::xercesc::DOMElement &e, const [ISO5436\\_2Type](#) &i)
- void [operator<<](#) (::xercesc::DOMElement &e, const [Record1Type](#) &i)
- ::std::wostream & [operator<<](#) (::std::wostream &o, const [Datum](#) &i)
- ::std::wostream & [operator<<](#) (::std::wostream &o, const [Type](#) &i)
- ::std::wostream & [operator<<](#) (::std::wostream &o, [Type::value](#) i)
- ::std::wostream & [operator<<](#) (::std::wostream &o, const [DataType](#) &i)
- ::std::wostream & [operator<<](#) (::std::wostream &o, [DataType::value](#) i)
- ::std::wostream & [operator<<](#) (::std::wostream &o, const [AxisType](#) &i)
- ::std::wostream & [operator<<](#) (::std::wostream &o, [AxisType::value](#) i)
- ::std::wostream & [operator<<](#) (::std::wostream &o, const [FeatureType](#) &i)
- ::std::wostream & [operator<<](#) (::std::wostream &o, const [RotationMatrix-ElementType](#) &i)
- ::std::wostream & [operator<<](#) (::std::wostream &o, const [RotationType](#) &i)
- ::std::wostream & [operator<<](#) (::std::wostream &o, const [MatrixDimension-Type](#) &i)
- ::std::wostream & [operator<<](#) (::std::wostream &o, const [DataLinkType](#) &i)
- ::std::wostream & [operator<<](#) (::std::wostream &o, const [DataListType](#) &i)
- ::std::wostream & [operator<<](#) (::std::wostream &o, const [ProbingSystemType](#) &i)
- ::std::wostream & [operator<<](#) (::std::wostream &o, const [InstrumentType](#) &i)
- ::std::wostream & [operator<<](#) (::std::wostream &o, const [AxisDescriptionType](#) &i)
- ::std::wostream & [operator<<](#) (::std::wostream &o, const [AxesType](#) &i)
- ::std::wostream & [operator<<](#) (::std::wostream &o, const [Record4Type](#) &i)
- ::std::wostream & [operator<<](#) (::std::wostream &o, const [Record3Type](#) &i)
- ::std::wostream & [operator<<](#) (::std::wostream &o, const [Record2Type](#) &i)
- ::std::wostream & [operator<<](#) (::std::wostream &o, const [ISO5436\\_2Type](#) &i)
- ::std::wostream & [operator<<](#) (::std::wostream &o, const [Record1Type](#) &i)
- bool [operator==](#) (const [RotationType](#) &x, const [RotationType](#) &y)
- bool [operator==](#) (const [MatrixDimensionType](#) &x, const [MatrixDimensionType](#) &y)
- bool [operator==](#) (const [DataLinkType](#) &x, const [DataLinkType](#) &y)
- bool [operator==](#) (const [DataListType](#) &x, const [DataListType](#) &y)
- bool [operator==](#) (const [ProbingSystemType](#) &x, const [ProbingSystemType](#) &y)
- bool [operator==](#) (const [InstrumentType](#) &x, const [InstrumentType](#) &y)
- bool [operator==](#) (const [AxisDescriptionType](#) &x, const [AxisDescriptionType](#) &y)
- bool [operator==](#) (const [AxesType](#) &x, const [AxesType](#) &y)
- bool [operator==](#) (const [Record4Type](#) &x, const [Record4Type](#) &y)
- bool [operator==](#) (const [Record3Type](#) &x, const [Record3Type](#) &y)
- bool [operator==](#) (const [Record2Type](#) &x, const [Record2Type](#) &y)
- bool [operator==](#) (const [ISO5436\\_2Type](#) &x, const [ISO5436\\_2Type](#) &y)
- bool [operator==](#) (const [Record1Type](#) &x, const [Record1Type](#) &y)

### 8.4.2 Function Documentation

**8.4.2.1** `xsd::cxx::xml::dom::auto_ptr<::xercesc::DOMDocument >  
xsd::ISO5436_2 (const ::xsd::ISO5436_2Type & x, const ::xml_  
schema::namespace_infomap & m, ::xml_schema::flags f = 0)`

Serialize to a new Xerces-C++ DOM document.

**Parameters:**

- x* An object model to serialize.
- m* A namespace information map.
- f* Serialization flags.

**Returns:**

A pointer to the new Xerces-C++ DOM document.

**8.4.2.2** `void xsd::ISO5436_2 (::xercesc::DOMDocument & d, const  
::xsd::ISO5436_2Type & x, ::xml_schema::flags f = 0)`

Serialize to an existing Xerces-C++ DOM document.

**Parameters:**

- d* A Xerces-C++ DOM document.
- x* An object model to serialize.
- f* Serialization flags.

Note that it is your responsibility to create the DOM document with the correct root element as well as set the necessary namespace mapping attributes.

**8.4.2.3** `void xsd::ISO5436_2 (::xercesc::XMLFormatTarget & ft, const  
::xsd::ISO5436_2Type & x, const ::xml_schema::namespace_infomap & m,  
::xercesc::DOMErrorHandler & eh, const ::std::wstring & e = L"UTF-8",  
::xml_schema::flags f = 0)`

Serialize to a Xerces-C++ XML format target with a Xerces-C++ DOM error handler.

**Parameters:**

- ft* A Xerces-C++ XML format target.
- x* An object model to serialize.
- m* A namespace information map.
- eh* A Xerces-C++ DOM error handler.
- e* A character encoding to produce XML in.
- f* Serialization flags.

This function reports serialization errors by calling the error handler.



**8.4.2.4** `void xsd::ISO5436_2 (::xercesc::XMLFormatTarget & ft, const ::xsd::ISO5436_2Type & x, const ::xml_schema::namespace_infomap & m, ::xml_schema::error_handler & eh, const ::std::wstring & e = L"UTF-8", ::xml_schema::flags f = 0)`

Serialize to a Xerces-C++ XML format target with an error handler.

**Parameters:**

- ft* A Xerces-C++ XML format target.
- x* An object model to serialize.
- m* A namespace information map.
- eh* An error handler.
- e* A character encoding to produce XML in.
- f* Serialization flags.

This function reports serialization errors by calling the error handler.

**8.4.2.5** `void xsd::ISO5436_2 (::xercesc::XMLFormatTarget & ft, const ::xsd::ISO5436_2Type & x, const ::xml_schema::namespace_infomap & m, const ::std::wstring & e = L"UTF-8", ::xml_schema::flags f = 0)`

Serialize to a Xerces-C++ XML format target.

**Parameters:**

- ft* A Xerces-C++ XML format target.
- x* An object model to serialize.
- m* A namespace information map.
- e* A character encoding to produce XML in.
- f* Serialization flags.

This function uses exceptions to report serialization errors.

**8.4.2.6** `void xsd::ISO5436_2 (::std::ostream & os, const ::xsd::ISO5436_2Type & x, const ::xml_schema::namespace_infomap & m, ::xercesc::DOMErrorHandler & eh, const ::std::wstring & e = L"UTF-8", ::xml_schema::flags f = 0)`

Serialize to a standard output stream with a Xerces-C++ DOM error handler.

**Parameters:**

- os* A standrad output stream.
- x* An object model to serialize.
- m* A namespace information map.
- eh* A Xerces-C++ DOM error handler.

*e* A character encoding to produce XML in.  
*f* Serialization flags.

This function reports serialization errors by calling the error handler.

**8.4.2.7** `void xsd::ISO5436_2 (::std::ostream & os, const ::xsd::ISO5436_2Type & x, const ::xml_schema::namespace_infomap & m, ::xml_schema::error_handler & eh, const ::std::wstring & e = L"UTF-8", ::xml_schema::flags f = 0)`

Serialize to a standard output stream with an error handler.

**Parameters:**

*os* A standrad output stream.  
*x* An object model to serialize.  
*m* A namespace information map.  
*eh* An error handler.  
*e* A character encoding to produce XML in.  
*f* Serialization flags.

This function reports serialization errors by calling the error handler.

**8.4.2.8** `void xsd::ISO5436_2 (::std::ostream & os, const ::xsd::ISO5436_2Type & x, const ::xml_schema::namespace_infomap & m, const ::std::wstring & e = L"UTF-8", ::xml_schema::flags f = 0)`

Serialize to a standard output stream.

**Parameters:**

*os* A standrad output stream.  
*x* An object model to serialize.  
*m* A namespace information map.  
*e* A character encoding to produce XML in.  
*f* Serialization flags.

This function uses exceptions to report serialization errors.

**8.4.2.9** `std::auto_ptr<::xsd::ISO5436_2Type> xsd::ISO5436_2 (::xercesc::DOMDocument * d, ::xml_schema::flags f = 0, const ::xml_schema::properties & p = ::xml_schema::properties())`

Parse a Xerces-C++ DOM document.

**Parameters:**

*d* A pointer to the Xerces-C++ DOM document.

*f* Parsing flags.

*p* Parsing properties.

#### Returns:

A pointer to the root of the object model.

This function is normally used together with the `keep_dom` and `own_dom` parsing flags to assign ownership of the DOM document to the object model.

**8.4.2.10** `std::auto_ptr<::xsd::ISO5436_2Type > xsd::ISO5436_2 (const ::xercesc::DOMDocument & d, ::xml_schema::flags f = 0, const ::xml_schema::properties & p = ::xml_schema::properties())`

Parse a Xerces-C++ DOM document.

#### Parameters:

*d* A Xerces-C++ DOM document.

*f* Parsing flags.

*p* Parsing properties.

#### Returns:

A pointer to the root of the object model.

**8.4.2.11** `std::auto_ptr<::xsd::ISO5436_2Type > xsd::ISO5436_2 (const ::xercesc::DOMInputSource & is, ::xercesc::DOMErrorHandler & eh, ::xml_schema::flags f = 0, const ::xml_schema::properties & p = ::xml_schema::properties())`

Parse a Xerces-C++ DOM input source with a Xerces-C++ DOM error handler.

#### Parameters:

*is* A Xerces-C++ DOM input source.

*eh* A Xerces-C++ DOM error handler.

*f* Parsing flags.

*p* Parsing properties.

#### Returns:

A pointer to the root of the object model.

This function reports parsing errors by calling the error handler.

**8.4.2.12** `std::auto_ptr<::xsd::ISO5436_2Type > xsd::ISO5436_2 (const ::xercesc::DOMInputSource & is, ::xml_schema::error_handler & eh, ::xml_schema::flags f = 0, const ::xml_schema::properties & p = ::xml_schema::properties())`

Parse a Xerces-C++ DOM input source with an error handler.

**Parameters:**

*is* A Xerces-C++ DOM input source.

*eh* An error handler.

*f* Parsing flags.

*p* Parsing properties.

**Returns:**

A pointer to the root of the object model.

This function reports parsing errors by calling the error handler.

**8.4.2.13** `std::auto_ptr<::xsd::ISO5436_2Type > xsd::ISO5436_2 (const ::xercesc::DOMInputSource & is, ::xml_schema::flags f = 0, const ::xml_schema::properties & p = ::xml_schema::properties())`

Parse a Xerces-C++ DOM input source.

**Parameters:**

*is* A Xerces-C++ DOM input source.

*f* Parsing flags.

*p* Parsing properties.

**Returns:**

A pointer to the root of the object model.

This function uses exceptions to report parsing errors.

**8.4.2.14** `std::auto_ptr<::xsd::ISO5436_2Type > xsd::ISO5436_2 (::std::istream & is, const ::std::wstring & id, ::xercesc::DOMErrorHandler & eh, ::xml_schema::flags f = 0, const ::xml_schema::properties & p = ::xml_schema::properties())`

Parse a standard input stream with a resource id and a Xerces-C++ DOM error handler.

**Parameters:**

*is* A standrad input stream.

*id* A resource id.

*eh* A Xerces-C++ DOM error handler.

*f* Parsing flags.

*p* Parsing properties.

#### Returns:

A pointer to the root of the object model.

The resource id is used to identify the document being parsed in diagnostics as well as to resolve relative paths.

This function reports parsing errors by calling the error handler.

**8.4.2.15** `std::auto_ptr<::xsd::ISO5436_2Type> xsd::ISO5436_2  
(::std::istream & is, const ::std::wstring & id, ::xml_schema::error_handler  
& eh, ::xml_schema::flags f = 0, const ::xml_schema::properties & p = ::xml_  
schema::properties())`

Parse a standard input stream with a resource id and an error handler.

#### Parameters:

*is* A standrad input stream.

*id* A resource id.

*eh* An error handler.

*f* Parsing flags.

*p* Parsing properties.

#### Returns:

A pointer to the root of the object model.

The resource id is used to identify the document being parsed in diagnostics as well as to resolve relative paths.

This function reports parsing errors by calling the error handler.

**8.4.2.16** `std::auto_ptr<::xsd::ISO5436_2Type> xsd::ISO5436_2  
(::std::istream & is, const ::std::wstring & id, ::xml_schema::flags f = 0,  
const ::xml_schema::properties & p = ::xml_schema::properties())`

Parse a standard input stream with a resource id.

#### Parameters:

*is* A standrad input stream.

*id* A resource id.

*f* Parsing flags.

*p* Parsing properties.

**Returns:**

A pointer to the root of the object model.

The resource id is used to identify the document being parsed in diagnostics as well as to resolve relative paths.

This function uses exceptions to report parsing errors.

**8.4.2.17** `std::auto_ptr<::xsd::ISO5436_2Type> xsd::ISO5436_2  
(::std::istream & is, ::xercesc::DOMErrorHandler & eh, ::xml_schema::flags f =  
0, const ::xml_schema::properties & p = ::xml_schema::properties())`

Parse a standard input stream with a Xerces-C++ DOM error handler.

**Parameters:**

*is* A standrad input stream.

*eh* A Xerces-C++ DOM error handler.

*f* Parsing flags.

*p* Parsing properties.

**Returns:**

A pointer to the root of the object model.

This function reports parsing errors by calling the error handler.

**8.4.2.18** `std::auto_ptr<::xsd::ISO5436_2Type> xsd::ISO5436_2  
(::std::istream & is, ::xml_schema::error_handler & eh, ::xml_schema::flags f =  
0, const ::xml_schema::properties & p = ::xml_schema::properties())`

Parse a standard input stream with an error handler.

**Parameters:**

*is* A standrad input stream.

*eh* An error handler.

*f* Parsing flags.

*p* Parsing properties.

**Returns:**

A pointer to the root of the object model.

This function reports parsing errors by calling the error handler.

**8.4.2.19** `std::auto_ptr<::xsd::ISO5436_2Type > xsd::ISO5436_2 (::std::istream & is, ::xml_schema::flags f = 0, const ::xml_schema::properties & p = ::xml_schema::properties())`

Parse a standard input stream.

**Parameters:**

*is* A standrad input stream.

*f* Parsing flags.

*p* Parsing properties.

**Returns:**

A pointer to the root of the object model.

This function uses exceptions to report parsing errors.

**8.4.2.20** `std::auto_ptr<::xsd::ISO5436_2Type > xsd::ISO5436_2 (const ::std::wstring & uri, ::xercesc::DOMErrorHandler & eh, ::xml_schema::flags f = 0, const ::xml_schema::properties & p = ::xml_schema::properties())`

Parse a URI or a local file with a Xerces-C++ DOM error handler.

**Parameters:**

*uri* A URI or a local file name.

*eh* A Xerces-C++ DOM error handler.

*f* Parsing flags.

*p* Parsing properties.

**Returns:**

A pointer to the root of the object model.

This function reports parsing errors by calling the error handler.

**8.4.2.21** `std::auto_ptr<::xsd::ISO5436_2Type > xsd::ISO5436_2 (const ::std::wstring & uri, ::xml_schema::error_handler & eh, ::xml_schema::flags f = 0, const ::xml_schema::properties & p = ::xml_schema::properties())`

Parse a URI or a local file with an error handler.

**Parameters:**

*uri* A URI or a local file name.

*eh* An error handler.

*f* Parsing flags.

*p* Parsing properties.

**Returns:**

A pointer to the root of the object model.

This function reports parsing errors by calling the error handler.

**8.4.2.22** `std::auto_ptr<::xsd::ISO5436_2Type > xsd::ISO5436_2 (const ::std::wstring & uri, ::xml_schema::flags f = 0, const ::xml_schema::properties & p = : :xml_schema::properties())`

Parse a URI or a local file.

**Parameters:**

*uri* A URI or a local file name.

*f* Parsing flags.

*p* Parsing properties.

**Returns:**

A pointer to the root of the object model.

This function uses exceptions to report parsing errors.

**8.4.2.23** `bool xsd::operator!= (const RotationType & x, const RotationType & y)`

**8.4.2.24** `bool xsd::operator!= (const MatrixDimensionType & x, const MatrixDimensionType & y)`

**8.4.2.25** `bool xsd::operator!= (const DataLinkType & x, const DataLinkType & y)`

**8.4.2.26** `bool xsd::operator!= (const DataListType & x, const DataListType & y)`

**8.4.2.27** `bool xsd::operator!= (const ProbingSystemType & x, const ProbingSystemType & y)`

**8.4.2.28** `bool xsd::operator!= (const InstrumentType & x, const InstrumentType & y)`

**8.4.2.29** `bool xsd::operator!= (const AxisDescriptionType & x, const AxisDescriptionType & y)`



8.4.2.30 **bool** xsd::operator!= (const AxesType & *x*, const AxesType & *y*)

8.4.2.31 **bool** xsd::operator!= (const Record4Type & *x*, const Record4Type & *y*)

8.4.2.32 **bool** xsd::operator!= (const Record3Type & *x*, const Record3Type & *y*)

8.4.2.33 **bool** xsd::operator!= (const Record2Type & *x*, const Record2Type & *y*)

8.4.2.34 **bool** xsd::operator!= (const ISO5436\_2Type & *x*, const ISO5436\_2Type & *y*)

8.4.2.35 **bool** xsd::operator!= (const Record1Type & *x*, const Record1Type & *y*)

8.4.2.36 **void** xsd::operator<< (::xsd::cxx::tree::list\_stream< wchar\_t > & *l*, const Datum & *i*)

8.4.2.37 **void** xsd::operator<< (::xercesc::DOMAttr & *a*, const Datum & *i*)

8.4.2.38 **void** xsd::operator<< (::xercesc::DOMElement & *e*, const Datum & *i*)

8.4.2.39 **void** xsd::operator<< (::xsd::cxx::tree::list\_stream< wchar\_t > & *l*, const Type & *i*)

8.4.2.40 **void** xsd::operator<< (::xercesc::DOMAttr & *a*, const Type & *i*)

8.4.2.41 **void** xsd::operator<< (::xercesc::DOMElement & *e*, const Type & *i*)

8.4.2.42 **void** xsd::operator<< (::xsd::cxx::tree::list\_stream< wchar\_t > & *l*, const DataType & *i*)

8.4.2.43 **void** xsd::operator<< (::xercesc::DOMAttr & *a*, const DataType & *i*)

8.4.2.44 **void** xsd::operator<< (::xercesc::DOMElement & *e*, const DataType & *i*)

**8.4.2.45** void xsd::operator<< (::xsd::cxx::tree::list\_stream< wchar\_t > & *l*, const AxisType & *i*)

**8.4.2.46** void xsd::operator<< (::xercesc::DOMAttr & *a*, const AxisType & *i*)

**8.4.2.47** void xsd::operator<< (::xercesc::DOMElement & *e*, const AxisType & *i*)

**8.4.2.48** void xsd::operator<< (::xsd::cxx::tree::list\_stream< wchar\_t > & *l*, const FeatureType & *i*)

**8.4.2.49** void xsd::operator<< (::xercesc::DOMAttr & *a*, const FeatureType & *i*)

**8.4.2.50** void xsd::operator<< (::xercesc::DOMElement & *e*, const FeatureType & *i*)

**8.4.2.51** void xsd::operator<< (::xsd::cxx::tree::list\_stream< wchar\_t > & *l*, const RotationMatrixElementType & *i*)

**8.4.2.52** void xsd::operator<< (::xercesc::DOMAttr & *a*, const RotationMatrixElementType & *i*)

**8.4.2.53** void xsd::operator<< (::xercesc::DOMElement & *e*, const RotationMatrixElementType & *i*)

**8.4.2.54** void xsd::operator<< (::xercesc::DOMElement & *e*, const RotationType & *i*)

**8.4.2.55** void xsd::operator<< (::xercesc::DOMElement & *e*, const MatrixDimensionType & *i*)

**8.4.2.56** void xsd::operator<< (::xercesc::DOMElement & *e*, const DataLinkType & *i*)

**8.4.2.57** void xsd::operator<< (::xercesc::DOMElement & *e*, const DataListType & *i*)

**8.4.2.58** void xsd::operator<< (::xercesc::DOMElement & *e*, const ProbingSystemType & *i*)

**8.4.2.59** void xsd::operator<< (::xercesc::DOMElement & *e*, const InstrumentType & *i*)

**8.4.2.60** void xsd::operator<< (::xercesc::DOMElement & *e*, const AxisDescriptionType & *i*)

**8.4.2.61** void xsd::operator<< (::xercesc::DOMElement & *e*, const AxesType & *i*)

**8.4.2.62** void xsd::operator<< (::xercesc::DOMElement & *e*, const Record4Type & *i*)

**8.4.2.63** void xsd::operator<< (::xercesc::DOMElement & *e*, const Record3Type & *i*)

**8.4.2.64** void xsd::operator<< (::xercesc::DOMElement & *e*, const Record2Type & *i*)

**8.4.2.65** void xsd::operator<< (::xercesc::DOMElement & *e*, const ISO5436\_2Type & *i*)

**8.4.2.66** void xsd::operator<< (::xercesc::DOMElement & *e*, const Record1Type & *i*)

**8.4.2.67** std::wostream & xsd::operator<< (std::wostream & *o*, const Datum & *i*)

**8.4.2.68** std::wostream & xsd::operator<< (std::wostream & *o*, const Type & *i*)

**8.4.2.69** std::wostream & xsd::operator<< (std::wostream & *o*, Type::value *i*)

**8.4.2.70** std::wostream & xsd::operator<< (std::wostream & *o*, const DataType & *i*)

**8.4.2.71** std::wostream & xsd::operator<< (std::wostream & *o*, DataType::value *i*)

**8.4.2.72** std::wostream & xsd::operator<< (std::wostream & *o*, const AxisType & *i*)

**8.4.2.73** `std::wostream & xsd::operator<< (::std::wostream & o, AxisType::value i)`

**8.4.2.74** `std::wostream & xsd::operator<< (::std::wostream & o, const FeatureType & i)`

**8.4.2.75** `std::wostream & xsd::operator<< (::std::wostream & o, const RotationMatrixElementType & i)`

**8.4.2.76** `std::wostream & xsd::operator<< (::std::wostream & o, const RotationType & i)`

**8.4.2.77** `std::wostream & xsd::operator<< (::std::wostream & o, const MatrixDimensionType & i)`

**8.4.2.78** `std::wostream & xsd::operator<< (::std::wostream & o, const DataLinkType & i)`

**8.4.2.79** `std::wostream & xsd::operator<< (::std::wostream & o, const DataListType & i)`

**8.4.2.80** `std::wostream & xsd::operator<< (::std::wostream & o, const ProbingSystemType & i)`

**8.4.2.81** `std::wostream & xsd::operator<< (::std::wostream & o, const InstrumentType & i)`

**8.4.2.82** `std::wostream & xsd::operator<< (::std::wostream & o, const AxisDescriptionType & i)`

**8.4.2.83** `std::wostream & xsd::operator<< (::std::wostream & o, const AxesType & i)`

**8.4.2.84** `std::wostream & xsd::operator<< (::std::wostream & o, const Record4Type & i)`

**8.4.2.85** `std::wostream & xsd::operator<< (::std::wostream & o, const Record3Type & i)`

**8.4.2.86** `std::wostream & xsd::operator<< (::std::wostream & o, const Record2Type & i)`

**8.4.2.87** `std::wostream & xsd::operator<< (::std::wostream & o, const ISO5436_2Type & i)`

**8.4.2.88** `std::wostream & xsd::operator<< (::std::wostream & o, const Record1Type & i)`

**8.4.2.89** `bool xsd::operator== (const RotationType & x, const RotationType & y)`

**8.4.2.90** `bool xsd::operator== (const MatrixDimensionType & x, const MatrixDimensionType & y)`

**8.4.2.91** `bool xsd::operator== (const DataLinkType & x, const DataLinkType & y)`

**8.4.2.92** `bool xsd::operator== (const DataListType & x, const DataListType & y)`

**8.4.2.93** `bool xsd::operator== (const ProbingSystemType & x, const ProbingSystemType & y)`

**8.4.2.94** `bool xsd::operator== (const InstrumentType & x, const InstrumentType & y)`

**8.4.2.95** `bool xsd::operator== (const AxisDescriptionType & x, const AxisDescriptionType & y)`

**8.4.2.96** `bool xsd::operator== (const AxesType & x, const AxesType & y)`

**8.4.2.97** `bool xsd::operator== (const Record4Type & x, const Record4Type & y)`

**8.4.2.98** `bool xsd::operator== (const Record3Type & x, const Record3Type & y)`

**8.4.2.99** `bool xsd::operator== (const Record2Type & x, const Record2Type & y)`

**8.4.2.100** `bool xsd::operator== (const ISO5436_2Type & x, const ISO5436_2Type & y)`

8.4.2.101 `bool xsd::operator==(const Record1Type & x, const Record1Type & y)`

## 9 openGPS ISO 5436-2 XML Class Documentation

### 9.1 \_OGPS\_DATA\_POINT Struct Reference

```
#include <data_point.hxx>
```

#### Public Attributes

- `OpenGPS::DataPoint * instance`

#### 9.1.1 Member Data Documentation

##### 9.1.1.1 `OpenGPS::DataPoint* \_OGPS\_DATA\_POINT::instance`

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

- `S:/openGPS/ISO5436_XML/trunk/src/ISO5436_2_XML/c/data\_point.hxx`

### 9.2 \_OGPS\_ISO5436\_2\_HANDLE Struct Reference

```
#include <iso5436_2_handle.hxx>
```

#### Public Attributes

- `OpenGPS::ISO5436_2 * instance`

#### 9.2.1 Member Data Documentation

##### 9.2.1.1 `OpenGPS::ISO5436_2* \_OGPS\_ISO5436\_2\_HANDLE::instance`

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

- `S:/openGPS/ISO5436_XML/trunk/src/ISO5436_2_XML/c/iso5436\_2\_handle.hxx`

### 9.3 \_OGPS\_POINT\_ITERATOR Struct Reference

```
#include <point_iterator.hxx>
```

#### Public Attributes

- `const OpenGPS::PointIteratorAutoPtr * instance`

### 9.3.1 Member Data Documentation

#### 9.3.1.1 `const OpenGPS::PointIteratorAutoPtr* _OGPS_POINT_ITERATOR::instance`

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

- `S:/openGPS/ISO5436_XML/trunk/src/ISO5436_2_XML/c/point_iterator.hxx`

## 9.4 \_OGPS\_POINT\_VECTOR Struct Reference

```
#include <point_vector.hxx>
```

### Public Attributes

- `OpenGPS::PointVector instance`
- `OGPS_DataPointPtr x`
- `OGPS_DataPointPtr y`
- `OGPS_DataPointPtr z`

### 9.4.1 Member Data Documentation

#### 9.4.1.1 `OpenGPS::PointVector _OGPS_POINT_VECTOR::instance`

#### 9.4.1.2 `OGPS_DataPointPtr _OGPS_POINT_VECTOR::x`

#### 9.4.1.3 `OGPS_DataPointPtr _OGPS_POINT_VECTOR::y`

#### 9.4.1.4 `OGPS_DataPointPtr _OGPS_POINT_VECTOR::z`

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

- `S:/openGPS/ISO5436_XML/trunk/src/ISO5436_2_XML/c/point_vector.hxx`

## 9.5 xsd::AxesType Class Reference

```
#include <iso5436_2_xsd.hxx>
```

### 9.5.1 Detailed Description

Class corresponding to the AxesType schema type.

## CX

Accessor and modifier functions for the CX required element.

Description of X-Axis

- `typedef ::xsd::cxx::tree::traits< CX_type, wchar_t > CX_traits`  
*Element traits type.*
- `typedef ::xsd::AxisDescriptionType CX_type`  
*Element type.*
- `void CX (::std::auto_ptr< CX_type > p)`  
*Set the element value without copying.*
- `void CX (const CX_type &x)`  
*Set the element value.*
- `CX_type & CX ()`  
*Return a read-write reference to the element.*
- `const CX_type & CX () const`  
*Return a read-only (constant) reference to the element.*

## CY

Accessor and modifier functions for the CY required element.

Description of Y-Axis

- `typedef ::xsd::cxx::tree::traits< CY_type, wchar_t > CY_traits`  
*Element traits type.*
- `typedef ::xsd::AxisDescriptionType CY_type`  
*Element type.*
- `void CY (::std::auto_ptr< CY_type > p)`  
*Set the element value without copying.*
- `void CY (const CY_type &x)`  
*Set the element value.*
- `CY_type & CY ()`  
*Return a read-write reference to the element.*
- `const CY_type & CY () const`  
*Return a read-only (constant) reference to the element.*



**CZ**

Accessor and modifier functions for the CZ required element.

Description of Z-Axis

- typedef ::xsd::cxx::tree::traits< CZ\_type, wchar\_t > CZ\_traits  
*Element traits type.*
- typedef ::xsd::AxisDescriptionType CZ\_type  
*Element type.*
- void CZ (::std::auto\_ptr< CZ\_type > p)  
*Set the element value without copying.*
- void CZ (const CZ\_type &x)  
*Set the element value.*
- CZ\_type & CZ ()  
*Return a read-write reference to the element.*
- const CZ\_type & CZ () const  
*Return a read-only (constant) reference to the element.*

**Rotation**

Accessor and modifier functions for the Rotation optional element.

An optional rotation of the data points. If this element is missing a unit transformation is assumed.

- typedef ::xsd::cxx::tree::optional< Rotation\_type > Rotation\_optional  
*Element optional container type.*
- typedef ::xsd::cxx::tree::traits< Rotation\_type, wchar\_t > Rotation\_traits  
*Element traits type.*
- typedef ::xsd::RotationType Rotation\_type  
*Element type.*
- void Rotation (::std::auto\_ptr< Rotation\_type > p)  
*Set the element value without copying.*
- void Rotation (const Rotation\_optional &x)  
*Set the element value.*

- void [Rotation](#) (const [Rotation\\_type](#) &x)  
*Set the element value.*
- [Rotation\\_optional](#) & [Rotation](#) ()  
*Return a read-write reference to the element container.*
- const [Rotation\\_optional](#) & [Rotation](#) () const  
*Return a read-only (constant) reference to the element container.*

### Constructors

- virtual [AxesType](#) \* [\\_clone](#) (::xml\_schema::flags f=0,::xml\_schema::type \*c=0)  
const  
*Copy the object polymorphically.*
- [AxesType](#) (const [AxesType](#) &x,::xml\_schema::flags f=0,::xml\_schema::type \*c=0)  
*Copy constructor.*
- [AxesType](#) (const ::xercesc::DOMElement &e,::xml\_schema::flags f=0,::xml\_schema::type \*c=0)  
*Construct an instance from a DOM element.*
- [AxesType](#) (const [CX\\_type](#) &, const [CY\\_type](#) &, const [CZ\\_type](#) &)  
*Construct an instance from the ultimate base and initializers for required elements and attributes.*

### Protected Member Functions

- void [parse](#) (::xsd::cxx::xml::dom::parser< wchar\_t > &,::xml\_schema::flags)

### Private Attributes

- ::xsd::cxx::tree::one< [CX\\_type](#) > [CX\\_](#)
- ::xsd::cxx::tree::one< [CY\\_type](#) > [CY\\_](#)
- ::xsd::cxx::tree::one< [CZ\\_type](#) > [CZ\\_](#)
- [Rotation\\_optional](#) [Rotation\\_](#)

## 9.5.2 Member Typedef Documentation

### 9.5.2.1 typedef ::xsd::cxx::tree::traits< [CX\\_type](#), wchar\_t > [xsd::AxesType::CX\\_traits](#)

Element traits type.

**9.5.2.2 typedef ::xsd::AxisDescriptionType xsd::AxesType::CX\_type**

Element type.

**9.5.2.3 typedef ::xsd::cxx::tree::traits< CY\_type, wchar\_t > xsd::AxesType::CY\_traits**

Element traits type.

**9.5.2.4 typedef ::xsd::AxisDescriptionType xsd::AxesType::CY\_type**

Element type.

**9.5.2.5 typedef ::xsd::cxx::tree::traits< CZ\_type, wchar\_t > xsd::AxesType::CZ\_traits**

Element traits type.

**9.5.2.6 typedef ::xsd::AxisDescriptionType xsd::AxesType::CZ\_type**

Element type.

**9.5.2.7 typedef ::xsd::cxx::tree::optional< Rotation\_type > xsd::AxesType::Rotation\_optional**

Element optional container type.

**9.5.2.8 typedef ::xsd::cxx::tree::traits< Rotation\_type, wchar\_t > xsd::AxesType::Rotation\_traits**

Element traits type.

**9.5.2.9 typedef ::xsd::RotationType xsd::AxesType::Rotation\_type**

Element type.

**9.5.3 Constructor & Destructor Documentation****9.5.3.1 xsd::AxesType::AxesType (const CX\_type &, const CY\_type &, const CZ\_type &)**

Construct an instance from the ultimate base and initializers for required elements and attributes.

**9.5.3.2 xsd::AxesType::AxesType (const ::xercesc::DOMElement & e, ::xml\_schema::flags f = 0, ::xml\_schema::type \* c = 0)**

Construct an instance from a DOM element.

**Parameters:**

- e* A DOM element to extract the data from.
- f* Flags to construct the new instance with.
- c* A pointer to the object that will contain the new instance.

**9.5.3.3** `xsd::AxesType::AxesType (const AxesType & x, ::xml_schema::flags f = 0, ::xml_schema::type * c = 0)`

Copy constructor.

**Parameters:**

- x* An instance to make a copy of.
- f* Flags to construct the copy with.
- c* A pointer to the object that will contain the copy.

For polymorphic object models use the `_clone` function instead.

**9.5.4 Member Function Documentation**

**9.5.4.1** `AxesType * xsd::AxesType::_clone (::xml_schema::flags f = 0, ::xml_schema::type * c = 0) const` `[virtual]`

Copy the object polymorphically.

**Parameters:**

- f* Flags to construct the copy with.
- c* A pointer to the object that will contain the copy.

**Returns:**

A pointer to the dynamically allocated copy.

This function ensures that the dynamic type of an instance is used for copying and should be used for polymorphic object models instead of the copy constructor.

**9.5.4.2** `void xsd::AxesType::CX (::std::auto_ptr< CX_type > p)`

Set the element value without copying.

**Parameters:**

- p* A new value to use.

This function will try to use the passed value directly instead of making a copy.

**9.5.4.3 void xsd::AxesType::CX (const CX\_type & x)**

Set the element value.

**Parameters:**

*x* A new value to set.

This function makes a copy of its argument and sets it as the new value of the element.

**9.5.4.4 CX\_type& xsd::AxesType::CX ()**

Return a read-write reference to the element.

**Returns:**

A reference to the element.

**9.5.4.5 const AxesType::CX\_type & xsd::AxesType::CX () const**

Return a read-only (constant) reference to the element.

**Returns:**

A constant reference to the element.

**9.5.4.6 void xsd::AxesType::CY (::std::auto\_ptr< CY\_type > p)**

Set the element value without copying.

**Parameters:**

*p* A new value to use.

This function will try to use the passed value directly instead of making a copy.

**9.5.4.7 void xsd::AxesType::CY (const CY\_type & x)**

Set the element value.

**Parameters:**

*x* A new value to set.

This function makes a copy of its argument and sets it as the new value of the element.

**9.5.4.8** `CY_type& xsd::AxesType::CY ()`

Return a read-write reference to the element.

**Returns:**

A reference to the element.

**9.5.4.9** `const CY_type& xsd::AxesType::CY () const`

Return a read-only (constant) reference to the element.

**Returns:**

A constant reference to the element.

**9.5.4.10** `void xsd::AxesType::CZ (::std::auto_ptr< CZ_type > p)`

Set the element value without copying.

**Parameters:**

*p* A new value to use.

This function will try to use the passed value directly instead of making a copy.

**9.5.4.11** `void xsd::AxesType::CZ (const CZ_type & x)`

Set the element value.

**Parameters:**

*x* A new value to set.

This function makes a copy of its argument and sets it as the new value of the element.

**9.5.4.12** `CZ_type& xsd::AxesType::CZ ()`

Return a read-write reference to the element.

**Returns:**

A reference to the element.

**9.5.4.13** `const CZ_type& xsd::AxesType::CZ () const`

Return a read-only (constant) reference to the element.

**Returns:**

A constant reference to the element.

**9.5.4.14** void xsd::AxesType::parse (::xsd::cxx::xml::dom::parser< wchar\_t > &, ::xml\_schema::flags) [protected]

**9.5.4.15** void xsd::AxesType::Rotation (::std::auto\_ptr< Rotation\_type > p)

Set the element value without copying.

**Parameters:**

*p* A new value to use.

This function will try to use the passed value directly instead of making a copy.

**9.5.4.16** void xsd::AxesType::Rotation (const Rotation\_optional & x)

Set the element value.

**Parameters:**

*x* An optional container with the new value to set.

If the value is present in *x* then this function makes a copy of this value and sets it as the new value of the element. Otherwise the element container is set the 'not present' state.

**9.5.4.17** void xsd::AxesType::Rotation (const Rotation\_type & x)

Set the element value.

**Parameters:**

*x* A new value to set.

This function makes a copy of its argument and sets it as the new value of the element.

**9.5.4.18** Rotation\_optional& xsd::AxesType::Rotation ()

Return a read-write reference to the element container.

**Returns:**

A reference to the optional container.

**9.5.4.19** const Rotation\_optional& xsd::AxesType::Rotation () const

Return a read-only (constant) reference to the element container.

**Returns:**

A constant reference to the optional container.

### 9.5.5 Member Data Documentation

9.5.5.1 `::xsd::cxx::tree::one< CX\_type > xsd::AxesType::CX\_` [private]

9.5.5.2 `::xsd::cxx::tree::one< CY\_type > xsd::AxesType::CY\_` [private]

9.5.5.3 `::xsd::cxx::tree::one< CZ\_type > xsd::AxesType::CZ\_` [private]

9.5.5.4 [Rotation\\_optional xsd::AxesType::Rotation\\_](#) [private]

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

- S:/openGPS/ISO5436\_XML/trunk/src/ISO5436\_2\_XML/[iso5436\\_2\\_xsd.hxx](#)
- S:/openGPS/ISO5436\_XML/trunk/src/ISO5436\_2\_XML/[iso5436\\_2\\_xsd.cxx](#)

## 9.6 xsd::AxisDescriptionType Class Reference

```
#include <iso5436_2_xsd.hxx>
```

### 9.6.1 Detailed Description

Class corresponding to the AxisDescriptionType schema type.

#### AxisType

Accessor and modifier functions for the AxisType required element.

Type of axis can be "I" for Incremental, "A" for Absolute. The z-axis must be absolute!

- `typedef ::xsd::cxx::tree::traits< AxisType\_type, wchar_t > AxisType\_traits`  
*Element traits type.*
- `typedef ::xsd::AxisType AxisType\_type`  
*Element type.*
- `void AxisType (::std::auto_ptr< AxisType\_type > p)`  
*Set the element value without copying.*
- `void AxisType (const AxisType\_type &x)`  
*Set the element value.*
- `AxisType\_type & AxisType ()`



*Return a read-write reference to the element.*

- const [AxisType\\_type](#) & [AxisType](#) () const  
*Return a read-only (constant) reference to the element.*

## DataType

Accessor and modifier functions for the DataType optional element.

Data type for absolute axis: "I" for int16, "L" for int32, "F" for float32, "D" for float64. Incremental axes do not have/need a data type

- typedef ::xsd::cxx::tree::optional< [DataType\\_type](#) > [DataType\\_optional](#)  
*Element optional container type.*
- typedef ::xsd::cxx::tree::traits< [DataType\\_type](#), wchar\_t > [DataType\\_traits](#)  
*Element traits type.*
- typedef ::xsd::DataType [DataType\\_type](#)  
*Element type.*
- void [DataType](#) (::std::auto\_ptr< [DataType\\_type](#) > p)  
*Set the element value without copying.*
- void [DataType](#) (const [DataType\\_optional](#) &x)  
*Set the element value.*
- void [DataType](#) (const [DataType\\_type](#) &x)  
*Set the element value.*
- [DataType\\_optional](#) & [DataType](#) ()  
*Return a read-write reference to the element container.*
- const [DataType\\_optional](#) & [DataType](#) () const  
*Return a read-only (constant) reference to the element container.*

## Increment

Accessor and modifier functions for the Increment optional element.

Needed for incremental axis and integer data types: Increment is the multiplier of the integer coordinate for the computation of the real coordinate:  $X_{real} = X_{offset} + X_{integer} * X_{increment}$ . The unit of increment and offset is metre.

- typedef ::xsd::cxx::tree::optional< [Increment\\_type](#) > [Increment\\_optional](#)

*Element optional container type.*

- typedef ::xsd::cxx::tree::traits< [Increment\\_type](#), wchar\_t > [Increment\\_traits](#)  
*Element traits type.*
- typedef ::xml\_schema::double\_ [Increment\\_type](#)  
*Element type.*
- void [Increment](#) (const [Increment\\_optional](#) &x)  
*Set the element value.*
- void [Increment](#) (const [Increment\\_type](#) &x)  
*Set the element value.*
- [Increment\\_optional](#) & [Increment](#) ()  
*Return a read-write reference to the element container.*
- const [Increment\\_optional](#) & [Increment](#) () const  
*Return a read-only (constant) reference to the element container.*

## Offset

Accessor and modifier functions for the Offset optional element.

The offset of axis in meter.

- typedef ::xsd::cxx::tree::optional< [Offset\\_type](#) > [Offset\\_optional](#)  
*Element optional container type.*
- typedef ::xsd::cxx::tree::traits< [Offset\\_type](#), wchar\_t > [Offset\\_traits](#)  
*Element traits type.*
- typedef ::xml\_schema::double\_ [Offset\\_type](#)  
*Element type.*
- void [Offset](#) (const [Offset\\_optional](#) &x)  
*Set the element value.*
- void [Offset](#) (const [Offset\\_type](#) &x)  
*Set the element value.*
- [Offset\\_optional](#) & [Offset](#) ()  
*Return a read-write reference to the element container.*
- const [Offset\\_optional](#) & [Offset](#) () const  
*Return a read-only (constant) reference to the element container.*

### Constructors

- virtual `AxisDescriptionType * _clone (::xml_schema::flags f=0, ::xml_schema::type *c=0) const`  
*Copy the object polymorphically.*
- `AxisDescriptionType (const AxisDescriptionType &x, ::xml_schema::flags f=0, ::xml_schema::type *c=0)`  
*Copy constructor.*
- `AxisDescriptionType (const ::xercesc::DOMElement &e, ::xml_schema::flags f=0, ::xml_schema::type *c=0)`  
*Construct an instance from a DOM element.*
- `AxisDescriptionType (const AxisType_type &)`  
*Construct an instance from the ultimate base and initializers for required elements and attributes.*

### Protected Member Functions

- void `parse (::xsd::cxx::xml::dom::parser< wchar_t > &, ::xml_schema::flags)`

### Private Attributes

- `::xsd::cxx::tree::one< AxisType_type > AxisType_`
- `DataType_optional DataType_`
- `Increment_optional Increment_`
- `Offset_optional Offset_`

## 9.6.2 Member Typedef Documentation

### 9.6.2.1 typedef ::xsd::cxx::tree::traits< AxisType\_type, wchar\_t > xsd::AxisDescriptionType::AxisType\_traits

Element traits type.

### 9.6.2.2 typedef ::xsd::AxisType xsd::AxisDescriptionType::AxisType\_type

Element type.

### 9.6.2.3 typedef ::xsd::cxx::tree::optional< DataType\_type > xsd::AxisDescriptionType::DataType\_optional

Element optional container type.

**9.6.2.4** `typedef ::xsd::cxx::tree::traits< DataType\_type, wchar_t > xsd::AxisDescriptionType::DataType\_traits`

Element traits type.

**9.6.2.5** `typedef ::xsd::DataType xsd::AxisDescriptionType::DataType\_type`

Element type.

**9.6.2.6** `typedef ::xsd::cxx::tree::optional< Increment\_type > xsd::AxisDescriptionType::Increment\_optional`

Element optional container type.

**9.6.2.7** `typedef ::xsd::cxx::tree::traits< Increment\_type, wchar_t > xsd::AxisDescriptionType::Increment\_traits`

Element traits type.

**9.6.2.8** `typedef ::xml\_schema::double\_ xsd::AxisDescriptionType::Increment\_type`

Element type.

**9.6.2.9** `typedef ::xsd::cxx::tree::optional< Offset\_type > xsd::AxisDescriptionType::Offset\_optional`

Element optional container type.

**9.6.2.10** `typedef ::xsd::cxx::tree::traits< Offset\_type, wchar_t > xsd::AxisDescriptionType::Offset\_traits`

Element traits type.

**9.6.2.11** `typedef ::xml\_schema::double\_ xsd::AxisDescriptionType::Offset\_type`

Element type.

### 9.6.3 Constructor & Destructor Documentation

**9.6.3.1** `xsd::AxisDescriptionType::AxisDescriptionType (const AxisType\_type &)`

Construct an instance from the ultimate base and initializers for required elements and attributes.

**9.6.3.2 xsd::AxisDescriptionType::AxisDescriptionType** (const [::xercesc::DOMElement](#) & *e*, [::xml\\_schema::flags](#) *f* = 0, [::xml\\_schema::type](#) \* *c* = 0)

Construct an instance from a DOM element.

**Parameters:**

- e* A DOM element to extract the data from.
- f* Flags to construct the new instance with.
- c* A pointer to the object that will contain the new instance.

**9.6.3.3 xsd::AxisDescriptionType::AxisDescriptionType** (const [AxisDescriptionType](#) & *x*, [::xml\\_schema::flags](#) *f* = 0, [::xml\\_schema::type](#) \* *c* = 0)

Copy constructor.

**Parameters:**

- x* An instance to make a copy of.
- f* Flags to construct the copy with.
- c* A pointer to the object that will contain the copy.

For polymorphic object models use the `_clone` function instead.

## 9.6.4 Member Function Documentation

**9.6.4.1 [AxisDescriptionType](#) \* xsd::AxisDescriptionType::\_clone** ([::xml\\_schema::flags](#) *f* = 0, [::xml\\_schema::type](#) \* *c* = 0) const [virtual]

Copy the object polymorphically.

**Parameters:**

- f* Flags to construct the copy with.
- c* A pointer to the object that will contain the copy.

**Returns:**

A pointer to the dynamically allocated copy.

This function ensures that the dynamic type of an instance is used for copying and should be used for polymorphic object models instead of the copy constructor.

**9.6.4.2 void xsd::AxisDescriptionType::AxisType** ([::std::auto\\_ptr](#)< [AxisType\\_type](#) > *p*)

Set the element value without copying.

**Parameters:**

*p* A new value to use.

This function will try to use the passed value directly instead of making a copy.

**9.6.4.3 void xsd::AxisDescriptionType::AxisType (const AxisType\_type & x)**

Set the element value.

**Parameters:**

*x* A new value to set.

This function makes a copy of its argument and sets it as the new value of the element.

**9.6.4.4 AxisType\_type& xsd::AxisDescriptionType::AxisType ()**

Return a read-write reference to the element.

**Returns:**

A reference to the element.

**9.6.4.5 const AxisType\_type& xsd::AxisDescriptionType::AxisType () const**

Return a read-only (constant) reference to the element.

**Returns:**

A constant reference to the element.

**9.6.4.6 void xsd::AxisDescriptionType::DataType (::std::auto\_ptr< DataType\_type > p)**

Set the element value without copying.

**Parameters:**

*p* A new value to use.

This function will try to use the passed value directly instead of making a copy.

**9.6.4.7 void xsd::AxisDescriptionType::DataType (const DataType\_optional & x)**

Set the element value.

**Parameters:**

*x* An optional container with the new value to set.

If the value is present in *x* then this function makes a copy of this value and sets it as the new value of the element. Otherwise the element container is set the 'not present' state.

**9.6.4.8 void xsd::AxisDescriptionType::DataType (const [DataType\\_type](#) & *x*)**

Set the element value.

**Parameters:**

*x* A new value to set.

This function makes a copy of its argument and sets it as the new value of the element.

**9.6.4.9 [DataType\\_optional&](#) xsd::AxisDescriptionType::DataType ()**

Return a read-write reference to the element container.

**Returns:**

A reference to the optional container.

**9.6.4.10 const [DataType\\_optional&](#) xsd::AxisDescriptionType::DataType ()  
const**

Return a read-only (constant) reference to the element container.

**Returns:**

A constant reference to the optional container.

**9.6.4.11 void xsd::AxisDescriptionType::Increment (const [Increment\\_optional](#) & *x*)**

Set the element value.

**Parameters:**

*x* An optional container with the new value to set.

If the value is present in *x* then this function makes a copy of this value and sets it as the new value of the element. Otherwise the element container is set the 'not present' state.

**9.6.4.12 void xsd::AxisDescriptionType::Increment (const Increment\_type & x)**

Set the element value.

**Parameters:**

*x* A new value to set.

This function makes a copy of its argument and sets it as the new value of the element.

**9.6.4.13 Increment\_optional& xsd::AxisDescriptionType::Increment ()**

Return a read-write reference to the element container.

**Returns:**

A reference to the optional container.

**9.6.4.14 const Increment\_optional& xsd::AxisDescriptionType::Increment ()  
const**

Return a read-only (constant) reference to the element container.

**Returns:**

A constant reference to the optional container.

**9.6.4.15 void xsd::AxisDescriptionType::Offset (const Offset\_optional & x)**

Set the element value.

**Parameters:**

*x* An optional container with the new value to set.

If the value is present in *x* then this function makes a copy of this value and sets it as the new value of the element. Otherwise the element container is set the 'not present' state.

**9.6.4.16 void xsd::AxisDescriptionType::Offset (const Offset\_type & x)**

Set the element value.

**Parameters:**

*x* A new value to set.

This function makes a copy of its argument and sets it as the new value of the element.



**9.6.4.17** [Offset\\_optional](#)& xsd::AxisDescriptionType::Offset ()

Return a read-write reference to the element container.

**Returns:**

A reference to the optional container.

**9.6.4.18** [const Offset\\_optional](#)& xsd::AxisDescriptionType::Offset () const

Return a read-only (constant) reference to the element container.

**Returns:**

A constant reference to the optional container.

**9.6.4.19** void xsd::AxisDescriptionType::parse (::xsd::cxx::xml::dom::parser< wchar\_t> &, [::xml\\_schema::flags](#)) [protected]**9.6.5** Member Data Documentation**9.6.5.1** ::xsd::cxx::tree::one< [AxisType\\_type](#) > [xsd::AxisDescriptionType::AxisType\\_](#) [private]**9.6.5.2** [DataType\\_optional](#) [xsd::AxisDescriptionType::DataType\\_](#) [private]**9.6.5.3** [Increment\\_optional](#) [xsd::AxisDescriptionType::Increment\\_](#) [private]**9.6.5.4** [Offset\\_optional](#) [xsd::AxisDescriptionType::Offset\\_](#) [private]

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

- S:/openGPS/ISO5436\_XML/trunk/src/ISO5436\_2\_XML/[iso5436\\_2\\_xsd.hxx](#)
- S:/openGPS/ISO5436\_XML/trunk/src/ISO5436\_2\_XML/[iso5436\\_2\\_xsd.cxx](#)

**9.7** xsd::AxisType Class Reference

```
#include <iso5436_2_xsd.hxx>
```

**9.7.1** Detailed Description

Enumeration class corresponding to the AxisType schema type.

### Public Types

- [A](#)
  - [I](#)
  - enum [value](#) { [A](#), [I](#) }
- Underlying enum type.*

### Public Member Functions

- virtual [AxisType](#) \* [\\_clone](#) (::xml\_schema::flags f=0, ::xml\_schema::type \*c=0) const  
*Copy the object polymorphically.*
- [AxisType](#) (const [AxisType](#) &x, ::xml\_schema::flags f=0, ::xml\_schema::type \*c=0)  
*Copy constructor.*
- [AxisType](#) (const ::std::wstring &s, const ::xercesc::DOMElement \*e, ::xml\_schema::flags f=0, ::xml\_schema::type \*c=0)  
*Construct an instance from a string fragment.*
- [AxisType](#) (const ::xercesc::DOMAttr &a, ::xml\_schema::flags f=0, ::xml\_schema::type \*c=0)  
*Construct an instance from a DOM attribute.*
- [AxisType](#) (const ::xercesc::DOMElement &e, ::xml\_schema::flags f=0, ::xml\_schema::type \*c=0)  
*Construct an instance from a DOM element.*
- [AxisType](#) (const ::xml\_schema::token &v)  
*Construct an instance from the base value.*
- [AxisType](#) ([value](#) v)  
*Construct an instance from the underlying enum value.*
- virtual [operator value](#) () const  
*Implicit conversion operator to the underlying enum value.*
- [AxisType](#) & [operator=](#) ([value](#) v)  
*Assign the underlying enum value.*

### Static Public Attributes

- static const [value](#) [\\_xsd\\_AxisType\\_indexes\\_](#) [2]
- static const wchar\_t \*const [\\_xsd\\_AxisType\\_literals\\_](#) [2]

### Protected Member Functions

- [value \\_xsd\\_AxisType\\_convert](#) () const

### 9.7.2 Member Enumeration Documentation

#### 9.7.2.1 enum [xsd::AxisType::value](#)

Underlying enum type.

#### Enumerator:

*A*

*I*

### 9.7.3 Constructor & Destructor Documentation

#### 9.7.3.1 [xsd::AxisType::AxisType](#) ([value](#) *v*)

Construct an instance from the underlying enum value.

#### Parameters:

*v* A enum value.

#### 9.7.3.2 [xsd::AxisType::AxisType](#) (const [::xml\\_schema::token](#) & *v*)

Construct an instance from the base value.

#### Parameters:

*v* A base value.

#### 9.7.3.3 [xsd::AxisType::AxisType](#) (const [::xercesc::DOMElement](#) & *e*, [::xml\\_schema::flags](#) *f* = 0, [::xml\\_schema::type](#) \* *c* = 0)

Construct an instance from a DOM element.

#### Parameters:

*e* A DOM element to extract the data from.

*f* Flags to construct the new instance with.

*c* A pointer to the object that will contain the new instance.

**9.7.3.4** `xsd::AxisType::AxisType (const ::xercesc::DOMAttr & a, ::xml_schema::flags f = 0, ::xml_schema::type * c = 0)`

Construct an instance from a DOM attribute.

**Parameters:**

- a* A DOM attribute to extract the data from.
- f* Flags to construct the new instance with.
- c* A pointer to the object that will contain the new instance.

**9.7.3.5** `xsd::AxisType::AxisType (const ::std::wstring & s, const ::xercesc::DOMElement * e, ::xml_schema::flags f = 0, ::xml_schema::type * c = 0)`

Construct an instance from a string fragment.

**Parameters:**

- s* A string fragment to extract the data from.
- e* A DOM element containing the string fragment.
- f* Flags to construct the new instance with.
- c* A pointer to the object that will contain the new instance.

**9.7.3.6** `xsd::AxisType::AxisType (const AxisType & x, ::xml_schema::flags f = 0, ::xml_schema::type * c = 0)`

Copy constructor.

**Parameters:**

- x* An instance to make a copy of.
- f* Flags to construct the copy with.
- c* A pointer to the object that will contain the copy.

For polymorphic object models use the `_clone` function instead.

## 9.7.4 Member Function Documentation

**9.7.4.1** `AxisType * xsd::AxisType::_clone (::xml_schema::flags f = 0, ::xml_schema::type * c = 0) const [virtual]`

Copy the object polymorphically.

**Parameters:**

- f* Flags to construct the copy with.

*c* A pointer to the object that will contain the copy.

#### Returns:

A pointer to the dynamically allocated copy.

This function ensures that the dynamic type of an instance is used for copying and should be used for polymorphic object models instead of the copy constructor.

**9.7.4.2** [AxisType::value](#) `xsd::AxisType::_xsd_AxisType_convert () const` [protected]

**9.7.4.3** `virtual xsd::AxisType::operator value () const` [inline, virtual]

Implicit conversion operator to the underlying enum value.

#### Returns:

A enum value.

**9.7.4.4** [AxisType&](#) `xsd::AxisType::operator= (value v)`

Assign the underlying enum value.

#### Parameters:

*v* A enum value.

#### Returns:

A reference to the instance.

### 9.7.5 Member Data Documentation

**9.7.5.1** `const value xsd::AxisType::_xsd_AxisType_indexes [2]` [static]

**9.7.5.2** `const wchar_t* const xsd::AxisType::_xsd_AxisType_literals [2]` [static]

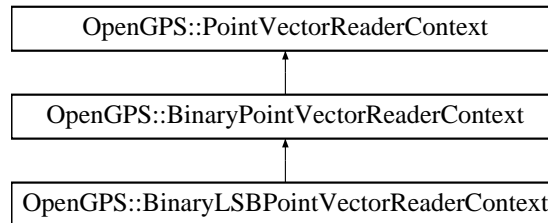
The documentation for this class was generated from the following files:

- S:/openGPS/ISO5436\_XML/trunk/src/ISO5436\_2\_XML/iso5436\_2\_xsd.hxx
- S:/openGPS/ISO5436\_XML/trunk/src/ISO5436\_2\_XML/iso5436\_2\_xsd.cxx

## 9.8 OpenGPS::BinaryLSBPointVectorReaderContext Class Reference

```
#include <binary_lsb_point_vector_reader_context.hpp>
```

Inheritance diagram for OpenGPS::BinaryLSBPointVectorReaderContext:



### Public Member Functions

- [BinaryLSBPointVectorReaderContext](#) (const String &filePath)
- virtual OGPS\_Boolean [Close](#) ()
- virtual OGPS\_Boolean [IsValid](#) () const
- virtual OGPS\_Boolean [MoveNext](#) ()
- virtual OGPS\_Boolean [Read](#) (double \*value)
- virtual OGPS\_Boolean [Read](#) (float \*value)
- virtual OGPS\_Boolean [Read](#) (int \*value)
- virtual OGPS\_Boolean [Read](#) (short \*value)
- virtual OGPS\_Boolean [Skip](#) ()
- virtual [~BinaryLSBPointVectorReaderContext](#) ()

### Protected Member Functions

- virtual OGPS\_Boolean [IsGood](#) () const

### Private Attributes

- [PointVectorInputBinaryFileStream](#) \* [m\\_Stream](#)

### 9.8.1 Constructor & Destructor Documentation

#### 9.8.1.1 BinaryLSBPointVectorReaderContext::BinaryLSBPointVectorReaderContext (const String &filePath)

#### 9.8.1.2 BinaryLSBPointVectorReaderContext::~~BinaryLSBPointVectorReaderContext () [virtual]

## 9.8.2 Member Function Documentation

**9.8.2.1 OGPS\_Boolean BinaryLSBPointVectorReaderContext::Close ()**  
[virtual]

Implements [OpenGPS::BinaryPointVectorReaderContext](#).

**9.8.2.2 OGPS\_Boolean BinaryLSBPointVectorReaderContext::IsGood () const**  
[protected, virtual]

**9.8.2.3 OGPS\_Boolean BinaryLSBPointVectorReaderContext::IsValid () const**  
[virtual]

Implements [OpenGPS::PointVectorReaderContext](#).

**9.8.2.4 OGPS\_Boolean BinaryLSBPointVectorReaderContext::MoveNext ()**  
[virtual]

Implements [OpenGPS::PointVectorReaderContext](#).

**9.8.2.5 OGPS\_Boolean BinaryLSBPointVectorReaderContext::Read (double \*  
*value*)** [virtual]

Implements [OpenGPS::PointVectorReaderContext](#).

**9.8.2.6 OGPS\_Boolean BinaryLSBPointVectorReaderContext::Read (float \*  
*value*)** [virtual]

Implements [OpenGPS::PointVectorReaderContext](#).

**9.8.2.7 OGPS\_Boolean BinaryLSBPointVectorReaderContext::Read (int \*  
*value*)** [virtual]

Implements [OpenGPS::PointVectorReaderContext](#).

**9.8.2.8 OGPS\_Boolean BinaryLSBPointVectorReaderContext::Read (short \*  
*value*)** [virtual]

Implements [OpenGPS::PointVectorReaderContext](#).

**9.8.2.9 OGPS\_Boolean BinaryLSBPointVectorReaderContext::Skip ()**  
[virtual]

Implements [OpenGPS::PointVectorReaderContext](#).

### 9.8.3 Member Data Documentation

#### 9.8.3.1 PointVectorInputBinaryFileStream\* OpenGPS::BinaryLSBPointVectorReaderContext::m\_Stream [private]

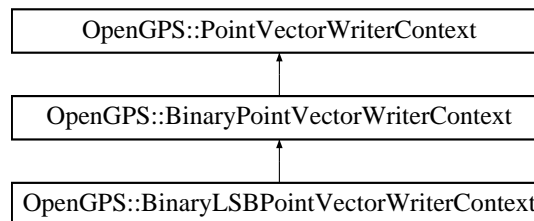
The documentation for this class was generated from the following files:

- S:/openGPS/ISO5436\_XML/trunk/src/ISO5436\_2\_XML/cxx/binary\_lsb\_point\_vector\_reader\_context.hxx
- S:/openGPS/ISO5436\_XML/trunk/src/ISO5436\_2\_XML/cxx/binary\_lsb\_point\_vector\_reader\_context.cxx

## 9.9 OpenGPS::BinaryLSBPointVectorWriterContext Class Reference

```
#include <binary_lsb_point_vector_writer_context.hxx>
```

Inheritance diagram for OpenGPS::BinaryLSBPointVectorWriterContext::



### Public Member Functions

- [BinaryLSBPointVectorWriterContext](#) (zipFile handle)
- virtual OGPS\_Boolean [Write](#) (const double \*value)
- virtual OGPS\_Boolean [Write](#) (const float \*value)
- virtual OGPS\_Boolean [Write](#) (const int \*value)
- virtual OGPS\_Boolean [Write](#) (const short \*value)
- virtual [~BinaryLSBPointVectorWriterContext](#) ()

### 9.9.1 Constructor & Destructor Documentation

#### 9.9.1.1 BinaryLSBPointVectorWriterContext::BinaryLSBPointVectorWriterContext (zipFile *handle*)

#### 9.9.1.2 BinaryLSBPointVectorWriterContext::~~BinaryLSBPointVectorWriterContext () [virtual]



## 9.9.2 Member Function Documentation

**9.9.2.1 OGPS\_Boolean BinaryLSBPointVectorWriterContext::Write** (const double \* *value*) [virtual]

Implements [OpenGPS::PointVectorWriterContext](#).

**9.9.2.2 OGPS\_Boolean BinaryLSBPointVectorWriterContext::Write** (const float \* *value*) [virtual]

Implements [OpenGPS::PointVectorWriterContext](#).

**9.9.2.3 OGPS\_Boolean BinaryLSBPointVectorWriterContext::Write** (const int \* *value*) [virtual]

Implements [OpenGPS::PointVectorWriterContext](#).

**9.9.2.4 OGPS\_Boolean BinaryLSBPointVectorWriterContext::Write** (const short \* *value*) [virtual]

Implements [OpenGPS::PointVectorWriterContext](#).

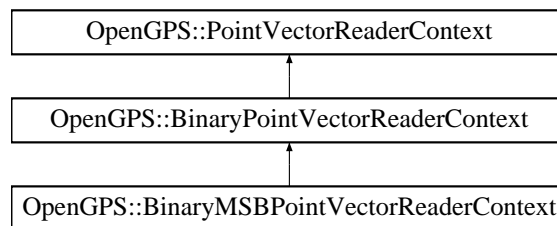
The documentation for this class was generated from the following files:

- S:/openGPS/ISO5436\_XML/trunk/src/ISO5436\_2\_XML/cxx/binary\_lsb\_point\_vector\_writer\_context.hxx
- S:/openGPS/ISO5436\_XML/trunk/src/ISO5436\_2\_XML/cxx/binary\_lsb\_point\_vector\_writer\_context.cxx

## 9.10 OpenGPS::BinaryMSBPointVectorReaderContext Class Reference

```
#include <binary_msb_point_vector_reader_context.hxx>
```

Inheritance diagram for OpenGPS::BinaryMSBPointVectorReaderContext::



### Public Member Functions

- [BinaryMSBPointVectorReaderContext](#) (const String &filePath)
- virtual OGPS\_Boolean [Close](#) ()

- virtual OGPS\_Boolean [IsValid](#) () const
- virtual OGPS\_Boolean [MoveNext](#) ()
- virtual OGPS\_Boolean [Read](#) (double \*value)
- virtual OGPS\_Boolean [Read](#) (float \*value)
- virtual OGPS\_Boolean [Read](#) (int \*value)
- virtual OGPS\_Boolean [Read](#) (short \*value)
- virtual OGPS\_Boolean [Skip](#) ()
- virtual [~BinaryMSBPointVectorReaderContext](#) ()

### Protected Member Functions

- virtual OGPS\_Boolean [IsGood](#) () const

### Private Attributes

- [PointVectorInputBinaryFileStream](#) \* [m\\_Stream](#)

## 9.10.1 Constructor & Destructor Documentation

**9.10.1.1 BinaryMSBPointVectorReaderContext::BinaryMSBPointVectorReaderContext (const String & *filePath*)**

**9.10.1.2 BinaryMSBPointVectorReaderContext::~BinaryMSBPointVectorReaderContext ()** [\[virtual\]](#)

## 9.10.2 Member Function Documentation

**9.10.2.1 OGPS\_Boolean BinaryMSBPointVectorReaderContext::Close ()**  
[\[virtual\]](#)

Implements [OpenGPS::BinaryPointVectorReaderContext](#).

**9.10.2.2 OGPS\_Boolean BinaryMSBPointVectorReaderContext::IsGood ()**  
**const** [\[protected, virtual\]](#)

**9.10.2.3 OGPS\_Boolean BinaryMSBPointVectorReaderContext::IsValid ()**  
**const** [\[virtual\]](#)

Implements [OpenGPS::PointVectorReaderContext](#).

**9.10.2.4 OGPS\_Boolean BinaryMSBPointVectorReaderContext::MoveNext ()**  
[\[virtual\]](#)

Implements [OpenGPS::PointVectorReaderContext](#).

**9.10.2.5 OGPS\_Boolean BinaryMSBPointVectorReaderContext::Read (double \* *value*)** [virtual]

Implements [OpenGPS::PointVectorReaderContext](#).

**9.10.2.6 OGPS\_Boolean BinaryMSBPointVectorReaderContext::Read (float \* *value*)** [virtual]

Implements [OpenGPS::PointVectorReaderContext](#).

**9.10.2.7 OGPS\_Boolean BinaryMSBPointVectorReaderContext::Read (int \* *value*)** [virtual]

Implements [OpenGPS::PointVectorReaderContext](#).

**9.10.2.8 OGPS\_Boolean BinaryMSBPointVectorReaderContext::Read (short \* *value*)** [virtual]

Implements [OpenGPS::PointVectorReaderContext](#).

**9.10.2.9 OGPS\_Boolean BinaryMSBPointVectorReaderContext::Skip ()** [virtual]

Implements [OpenGPS::PointVectorReaderContext](#).

### 9.10.3 Member Data Documentation

**9.10.3.1 PointVectorInputBinaryFileStream\* OpenGPS::BinaryMSBPointVectorReaderContext::m\_Stream** [private]

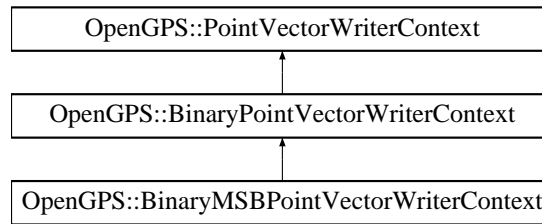
The documentation for this class was generated from the following files:

- S:/openGPS/ISO5436\_XML/trunk/src/ISO5436\_2\_XML/cxx/[binary\\_msb\\_point\\_vector\\_reader\\_context.hxx](#)
- S:/openGPS/ISO5436\_XML/trunk/src/ISO5436\_2\_XML/cxx/[binary\\_msb\\_point\\_vector\\_reader\\_context.cxx](#)

## 9.11 OpenGPS::BinaryMSBPointVectorWriterContext Class Reference

```
#include <binary_msb_point_vector_writer_context.hxx>
```

Inheritance diagram for OpenGPS::BinaryMSBPointVectorWriterContext::



### Public Member Functions

- [BinaryMSBPointVectorWriterContext](#) (zipFile handle)
- virtual OGPS\_Boolean [Write](#) (const double \*value)
- virtual OGPS\_Boolean [Write](#) (const float \*value)
- virtual OGPS\_Boolean [Write](#) (const int \*value)
- virtual OGPS\_Boolean [Write](#) (const short \*value)
- virtual [~BinaryMSBPointVectorWriterContext](#) ()

### 9.11.1 Constructor & Destructor Documentation

**9.11.1.1** [BinaryMSBPointVectorWriterContext::BinaryMSBPointVectorWriterContext](#) (zipFile *handle*)

**9.11.1.2** [BinaryMSBPointVectorWriterContext::~~BinaryMSBPointVectorWriterContext](#) () [virtual]

### 9.11.2 Member Function Documentation

**9.11.2.1** OGPS\_Boolean [BinaryMSBPointVectorWriterContext::Write](#) (const double \* *value*) [virtual]

Implements [OpenGPS::PointVectorWriterContext](#).

**9.11.2.2** OGPS\_Boolean [BinaryMSBPointVectorWriterContext::Write](#) (const float \* *value*) [virtual]

Implements [OpenGPS::PointVectorWriterContext](#).

**9.11.2.3** OGPS\_Boolean [BinaryMSBPointVectorWriterContext::Write](#) (const int \* *value*) [virtual]

Implements [OpenGPS::PointVectorWriterContext](#).

#### 9.11.2.4 OGPS\_Boolean BinaryMSBPointVectorWriterContext::Write (const short \* value) [virtual]

Implements [OpenGPS::PointVectorWriterContext](#).

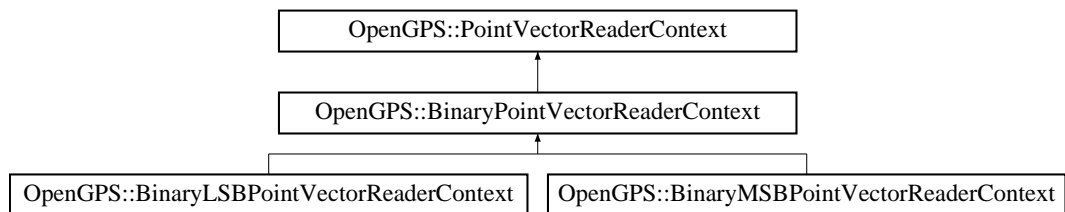
The documentation for this class was generated from the following files:

- S:/openGPS/ISO5436\_XML/trunk/src/ISO5436\_2\_XML/cxx/binary\_msb\_point\_vector\_writer\_context.hxx
- S:/openGPS/ISO5436\_XML/trunk/src/ISO5436\_2\_XML/cxx/binary\_msb\_point\_vector\_writer\_context.cxx

## 9.12 OpenGPS::BinaryPointVectorReaderContext Class Reference

```
#include <binary_point_vector_reader_context.hxx>
```

Inheritance diagram for OpenGPS::BinaryPointVectorReaderContext::



### Public Member Functions

- [BinaryPointVectorReaderContext](#) ()
- virtual OGPS\_Boolean [Close](#) ()=0

### Protected Member Functions

- virtual [~BinaryPointVectorReaderContext](#) ()

### 9.12.1 Constructor & Destructor Documentation

#### 9.12.1.1 BinaryPointVectorReaderContext::BinaryPointVectorReaderContext ()

#### 9.12.1.2 BinaryPointVectorReaderContext::~~BinaryPointVectorReaderContext () [protected, virtual]

### 9.12.2 Member Function Documentation

#### 9.12.2.1 virtual OGPS\_Boolean OpenGPS::BinaryPointVectorReaderContext::Close () [pure virtual]

Implemented in [OpenGPS::BinaryLSBPointVectorReaderContext](#), and [OpenGPS::BinaryMSBPointVectorReaderContext](#).

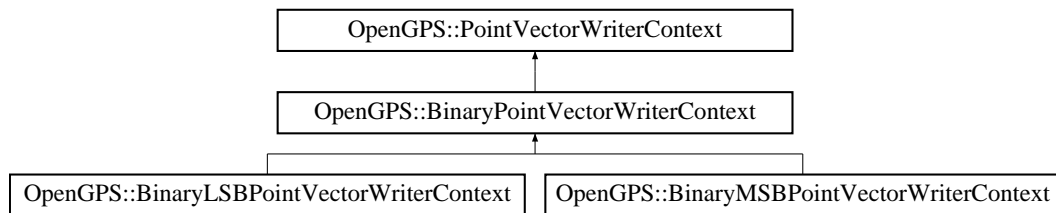
The documentation for this class was generated from the following files:

- [S:/openGPS/ISO5436\\_XML/trunk/src/ISO5436\\_2\\_XML/cxx/binary\\_point\\_vector\\_reader\\_context.hxx](#)
- [S:/openGPS/ISO5436\\_XML/trunk/src/ISO5436\\_2\\_XML/cxx/binary\\_lsb\\_point\\_vector\\_reader\\_context.cxx](#)

## 9.13 OpenGPS::BinaryPointVectorWriterContext Class Reference

```
#include <binary_point_vector_writer_context.hxx>
```

Inheritance diagram for OpenGPS::BinaryPointVectorWriterContext::



### Public Member Functions

- [BinaryPointVectorWriterContext](#) (zipFile handle)
- virtual OGPS\_Boolean [Close](#) ()
- virtual OGPS\_Boolean [MoveNext](#) ()
- virtual OGPS\_Boolean [Skip](#) ()

### Protected Member Functions

- std::ostream \* [GetStream](#) () const
- OGPS\_Boolean [HasStream](#) () const
- OGPS\_Boolean [IsGood](#) () const
- virtual [~BinaryPointVectorWriterContext](#) ()

### Private Attributes

- [ZipStreamBuffer](#) \* [m\\_Buffer](#)
- [ZipOutputStream](#) \* [m\\_Stream](#)

### 9.13.1 Constructor & Destructor Documentation

**9.13.1.1** `BinaryPointVectorWriterContext::BinaryPointVectorWriterContext (zipFile handle)`

**9.13.1.2** `BinaryPointVectorWriterContext::~~BinaryPointVectorWriterContext ()` [protected, virtual]

### 9.13.2 Member Function Documentation

**9.13.2.1** `OGPS_Boolean BinaryPointVectorWriterContext::Close ()` [virtual]

**9.13.2.2** `std::ostream * BinaryPointVectorWriterContext::GetStream ()` const [protected]

**9.13.2.3** `OGPS_Boolean BinaryPointVectorWriterContext::HasStream ()` const [protected]

**9.13.2.4** `OGPS_Boolean BinaryPointVectorWriterContext::IsGood ()` const [protected, virtual]

Implements [OpenGPS::PointVectorWriterContext](#).

**9.13.2.5** `OGPS_Boolean BinaryPointVectorWriterContext::MoveNext ()` [virtual]

Implements [OpenGPS::PointVectorWriterContext](#).

**9.13.2.6** `OGPS_Boolean BinaryPointVectorWriterContext::Skip ()` [virtual]

Implements [OpenGPS::PointVectorWriterContext](#).

### 9.13.3 Member Data Documentation

**9.13.3.1** `ZipStreamBuffer* OpenGPS::BinaryPointVectorWriterContext::m_Buffer` [private]

**9.13.3.2** `ZipOutputStream* OpenGPS::BinaryPointVectorWriterContext::m_Stream` [private]

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

- `S:/openGPS/ISO5436_XML/trunk/src/ISO5436_2_XML/cxx/binary_point_vector_writer_context.hxx`

- [S:/openGPS/ISO5436\\_XML/trunk/src/ISO5436\\_2\\_XML/cxx/binary\\_point\\_vector\\_writer\\_context.cxx](#)

## 9.14 xsd::DataLinkType Class Reference

```
#include <iso5436_2_xsd.hxx>
```

### 9.14.1 Detailed Description

Class corresponding to the DataLinkType schema type.

Defines a Link to a binary data file and a binary file containing the information about valid points.

#### MD5ChecksumPointData

Accessor and modifier functions for the MD5ChecksumPointData required element.

An MD5Checksum of the point data file like calculated by the unix command "md5sum". It consists of 32 hexadecimal digits. The binary representation is a 128 bit number.

- `typedef ::xsd::cxx::tree::traits< MD5ChecksumPointData\_type, wchar_t > MD5ChecksumPointData\_traits`  
*Element traits type.*
- `typedef ::xml_schema::hex_binary MD5ChecksumPointData\_type`  
*Element type.*
- `void MD5ChecksumPointData (::std::auto_ptr< MD5ChecksumPointData\_type > p)`  
*Set the element value without copying.*
- `void MD5ChecksumPointData (const MD5ChecksumPointData\_type &x)`  
*Set the element value.*
- `MD5ChecksumPointData\_type & MD5ChecksumPointData ()`  
*Return a read-write reference to the element.*
- `const MD5ChecksumPointData\_type & MD5ChecksumPointData () const`  
*Return a read-only (constant) reference to the element.*

#### MD5ChecksumValidPoints

Accessor and modifier functions for the MD5ChecksumValidPoints optional element.



An MD5Checksum of the valid points file like calculated by the unix command "md5sum". It consists of 32 hexadecimal digits. The binary representation is a 128 bit number.

- typedef ::xsd::cxx::tree::optional< [MD5ChecksumValidPoints\\_type](#) > [MD5ChecksumValidPoints\\_optional](#)  
*Element optional container type.*
- typedef ::xsd::cxx::tree::traits< [MD5ChecksumValidPoints\\_type](#), wchar\_t > [MD5ChecksumValidPoints\\_traits](#)  
*Element traits type.*
- typedef ::xml\_schema::hex\_binary [MD5ChecksumValidPoints\\_type](#)  
*Element type.*
- void [MD5ChecksumValidPoints](#) (::std::auto\_ptr< [MD5ChecksumValidPoints\\_type](#) > p)  
*Set the element value without copying.*
- void [MD5ChecksumValidPoints](#) (const [MD5ChecksumValidPoints\\_optional](#) &x)  
*Set the element value.*
- void [MD5ChecksumValidPoints](#) (const [MD5ChecksumValidPoints\\_type](#) &x)  
*Set the element value.*
- [MD5ChecksumValidPoints\\_optional](#) & [MD5ChecksumValidPoints](#) ()  
*Return a read-write reference to the element container.*
- const [MD5ChecksumValidPoints\\_optional](#) & [MD5ChecksumValidPoints](#) () const  
*Return a read-only (constant) reference to the element container.*

## PointDataLink

Accessor and modifier functions for the PointDataLink required element.

Relative filename in unix notation to a binary file with point data. Data can be specified directly in the xml file or with a link be stored in an external binary file. The Binary file has the same organisation as the DataList and has the datatypes specified in the axis description.

- typedef ::xsd::cxx::tree::traits< [PointDataLink\\_type](#), wchar\_t > [PointDataLink\\_traits](#)  
*Element traits type.*

- typedef ::xml\_schema::string PointDataLink\_type  
*Element type.*
- void PointDataLink (::std::auto\_ptr< PointDataLink\_type > p)  
*Set the element value without copying.*
- void PointDataLink (const PointDataLink\_type &x)  
*Set the element value.*
- PointDataLink\_type & PointDataLink ()  
*Return a read-write reference to the element.*
- const PointDataLink\_type & PointDataLink () const  
*Return a read-only (constant) reference to the element.*

### ValidPointsLink

Accessor and modifier functions for the ValidPointsLink optional element.

Relative filename in unix notation to a binary file that contains a packed array of bools. Each element that is true corresponds to a valid data point in the binary point data file.

If this tag does not exist, all points are valid except for floating point numbers of the special value "NaN" (Not a Number).

- typedef ::xsd::cxx::tree::optional< ValidPointsLink\_type > ValidPointsLink\_optional  
*Element optional container type.*
- typedef ::xsd::cxx::tree::traits< ValidPointsLink\_type, wchar\_t > ValidPointsLink\_traits  
*Element traits type.*
- typedef ::xml\_schema::string ValidPointsLink\_type  
*Element type.*
- void ValidPointsLink (::std::auto\_ptr< ValidPointsLink\_type > p)  
*Set the element value without copying.*
- void ValidPointsLink (const ValidPointsLink\_optional &x)  
*Set the element value.*
- void ValidPointsLink (const ValidPointsLink\_type &x)  
*Set the element value.*
- ValidPointsLink\_optional & ValidPointsLink ()

*Return a read-write reference to the element container.*

- const [ValidPointsLink\\_optional](#) & [ValidPointsLink](#) () const  
*Return a read-only (constant) reference to the element container.*

### Constructors

- virtual [DataLinkType](#) \* [\\_clone](#) (::xml\_schema::flags f=0,::xml\_schema::type \*c=0) const  
*Copy the object polymorphically.*
- [DataLinkType](#) (const [DataLinkType](#) &x,::xml\_schema::flags f=0,::xml\_schema::type \*c=0)  
*Copy constructor.*
- [DataLinkType](#) (const ::xercesc::DOMElement &e,::xml\_schema::flags f=0,::xml\_schema::type \*c=0)  
*Construct an instance from a DOM element.*
- [DataLinkType](#) (const [PointDataLink\\_type](#) &, const [MD5ChecksumPointData\\_type](#) &)  
*Construct an instance from the ultimate base and initializers for required elements and attributes.*

### Protected Member Functions

- void [parse](#) (::xsd::cxx::xml::dom::parser< wchar\_t > &,::xml\_schema::flags)

### Private Attributes

- ::xsd::cxx::tree::one< [MD5ChecksumPointData\\_type](#) > [MD5ChecksumPointData\\_](#)
- [MD5ChecksumValidPoints\\_optional](#) [MD5ChecksumValidPoints\\_](#)
- ::xsd::cxx::tree::one< [PointDataLink\\_type](#) > [PointDataLink\\_](#)
- [ValidPointsLink\\_optional](#) [ValidPointsLink\\_](#)

## 9.14.2 Member Typedef Documentation

**9.14.2.1** typedef ::xsd::cxx::tree::traits< [MD5ChecksumPointData\\_type](#), wchar\_t > [xsd::DataLinkType::MD5ChecksumPointData\\_traits](#)

Element traits type.

**9.14.2.2** typedef [::xml\\_schema::hex\\_binary](#) [xsd::DataLinkType::MD5ChecksumPointData\\_type](#)

Element type.

**9.14.2.3** typedef [::xsd::cxx::tree::optional](#)< [MD5ChecksumValidPoints\\_type](#) > [xsd::DataLinkType::MD5ChecksumValidPoints\\_optional](#)

Element optional container type.

**9.14.2.4** typedef [::xsd::cxx::tree::traits](#)< [MD5ChecksumValidPoints\\_type](#), [wchar\\_t](#) > [xsd::DataLinkType::MD5ChecksumValidPoints\\_traits](#)

Element traits type.

**9.14.2.5** typedef [::xml\\_schema::hex\\_binary](#) [xsd::DataLinkType::MD5ChecksumValidPoints\\_type](#)

Element type.

**9.14.2.6** typedef [::xsd::cxx::tree::traits](#)< [PointDataLink\\_type](#), [wchar\\_t](#) > [xsd::DataLinkType::PointDataLink\\_traits](#)

Element traits type.

**9.14.2.7** typedef [::xml\\_schema::string](#) [xsd::DataLinkType::PointDataLink\\_type](#)

Element type.

**9.14.2.8** typedef [::xsd::cxx::tree::optional](#)< [ValidPointsLink\\_type](#) > [xsd::DataLinkType::ValidPointsLink\\_optional](#)

Element optional container type.

**9.14.2.9** typedef [::xsd::cxx::tree::traits](#)< [ValidPointsLink\\_type](#), [wchar\\_t](#) > [xsd::DataLinkType::ValidPointsLink\\_traits](#)

Element traits type.

**9.14.2.10** typedef [::xml\\_schema::string](#) [xsd::DataLinkType::ValidPointsLink\\_type](#)

Element type.

### 9.14.3 Constructor & Destructor Documentation

**9.14.3.1** [xsd::DataLinkType::DataLinkType](#) ([const](#) [PointDataLink\\_type](#) &, [const](#) [MD5ChecksumPointData\\_type](#) &)

Construct an instance from the ultimate base and initializers for required elements and attributes.

**9.14.3.2 xsd::DataLinkType::DataLinkType (const ::xercesc::DOMElement & *e*, ::xml\_schema::flags *f* = 0, ::xml\_schema::type \* *c* = 0)**

Construct an instance from a DOM element.

**Parameters:**

- e* A DOM element to extract the data from.
- f* Flags to construct the new instance with.
- c* A pointer to the object that will contain the new instance.

**9.14.3.3 xsd::DataLinkType::DataLinkType (const DataLinkType & *x*, ::xml\_schema::flags *f* = 0, ::xml\_schema::type \* *c* = 0)**

Copy constructor.

**Parameters:**

- x* An instance to make a copy of.
- f* Flags to construct the copy with.
- c* A pointer to the object that will contain the copy.

For polymorphic object models use the `_clone` function instead.

#### 9.14.4 Member Function Documentation

**9.14.4.1 DataLinkType \* xsd::DataLinkType::\_clone (::xml\_schema::flags *f* = 0, ::xml\_schema::type \* *c* = 0) const** [virtual]

Copy the object polymorphically.

**Parameters:**

- f* Flags to construct the copy with.
- c* A pointer to the object that will contain the copy.

**Returns:**

A pointer to the dynamically allocated copy.

This function ensures that the dynamic type of an instance is used for copying and should be used for polymorphic object models instead of the copy constructor.

#### 9.14.4.2 void xsd::DataLinkType::MD5ChecksumPointData (::std::auto\_ptr< MD5ChecksumPointData\_type > *p*)

Set the element value without copying.

##### Parameters:

*p* A new value to use.

This function will try to use the passed value directly instead of making a copy.

#### 9.14.4.3 void xsd::DataLinkType::MD5ChecksumPointData (const MD5ChecksumPointData\_type & *x*)

Set the element value.

##### Parameters:

*x* A new value to set.

This function makes a copy of its argument and sets it as the new value of the element.

#### 9.14.4.4 MD5ChecksumPointData\_type& xsd::DataLinkType::MD5ChecksumPointData ()

Return a read-write reference to the element.

##### Returns:

A reference to the element.

#### 9.14.4.5 const MD5ChecksumPointData\_type& xsd::DataLinkType::MD5ChecksumPointData () const

Return a read-only (constant) reference to the element.

##### Returns:

A constant reference to the element.

#### 9.14.4.6 void xsd::DataLinkType::MD5ChecksumValidPoints (::std::auto\_ptr< MD5ChecksumValidPoints\_type > *p*)

Set the element value without copying.

##### Parameters:

*p* A new value to use.

This function will try to use the passed value directly instead of making a copy.

**9.14.4.7** void xsd::DataLinkType::MD5ChecksumValidPoints (const [MD5ChecksumValidPoints\\_optional](#) & *x*)

Set the element value.

**Parameters:**

*x* An optional container with the new value to set.

If the value is present in *x* then this function makes a copy of this value and sets it as the new value of the element. Otherwise the element container is set the 'not present' state.

**9.14.4.8** void xsd::DataLinkType::MD5ChecksumValidPoints (const [MD5ChecksumValidPoints\\_type](#) & *x*)

Set the element value.

**Parameters:**

*x* A new value to set.

This function makes a copy of its argument and sets it as the new value of the element.

**9.14.4.9** [MD5ChecksumValidPoints\\_optional](#)& xsd::DataLinkType::MD5ChecksumValidPoints ()

Return a read-write reference to the element container.

**Returns:**

A reference to the optional container.

**9.14.4.10** const [MD5ChecksumValidPoints\\_optional](#)& xsd::DataLinkType::MD5ChecksumValidPoints () const

Return a read-only (constant) reference to the element container.

**Returns:**

A constant reference to the optional container.

**9.14.4.11** void xsd::DataLinkType::parse (::xsd::cxx::xml::dom::parser< wchar\_t > &, [::xml\\_schema::flags](#)) [[protected](#)]

**9.14.4.12 void xsd::DataLinkType::PointDataLink (::std::auto\_ptr< [PointDataLink\\_type](#) > *p*)**

Set the element value without copying.

**Parameters:**

*p* A new value to use.

This function will try to use the passed value directly instead of making a copy.

**9.14.4.13 void xsd::DataLinkType::PointDataLink (const [PointDataLink\\_type](#) & *x*)**

Set the element value.

**Parameters:**

*x* A new value to set.

This function makes a copy of its argument and sets it as the new value of the element.

**9.14.4.14 [PointDataLink\\_type](#)& xsd::DataLinkType::PointDataLink ()**

Return a read-write reference to the element.

**Returns:**

A reference to the element.

**9.14.4.15 const [PointDataLink\\_type](#)& xsd::DataLinkType::PointDataLink () const**

Return a read-only (constant) reference to the element.

**Returns:**

A constant reference to the element.

**9.14.4.16 void xsd::DataLinkType::ValidPointsLink (::std::auto\_ptr< [ValidPointsLink\\_type](#) > *p*)**

Set the element value without copying.

**Parameters:**

*p* A new value to use.

This function will try to use the passed value directly instead of making a copy.



#### 9.14.4.17 void xsd::DataLinkType::ValidPointsLink (const ValidPointsLink\_optional & x)

Set the element value.

##### Parameters:

*x* An optional container with the new value to set.

If the value is present in *x* then this function makes a copy of this value and sets it as the new value of the element. Otherwise the element container is set the 'not present' state.

#### 9.14.4.18 void xsd::DataLinkType::ValidPointsLink (const ValidPointsLink\_type & x)

Set the element value.

##### Parameters:

*x* A new value to set.

This function makes a copy of its argument and sets it as the new value of the element.

#### 9.14.4.19 ValidPointsLink\_optional& xsd::DataLinkType::ValidPointsLink ()

Return a read-write reference to the element container.

##### Returns:

A reference to the optional container.

#### 9.14.4.20 const ValidPointsLink\_optional& xsd::DataLinkType::ValidPointsLink () const

Return a read-only (constant) reference to the element container.

##### Returns:

A constant reference to the optional container.

### 9.14.5 Member Data Documentation

#### 9.14.5.1 ::xsd::cxx::tree::one< MD5ChecksumPointData\_type > xsd::DataLinkType::MD5ChecksumPointData\_ [private]

#### 9.14.5.2 MD5ChecksumValidPoints\_optional xsd::DataLinkType::MD5ChecksumValidPoints\_ [private]

**9.14.5.3** `::xsd::cxx::tree::one< PointDataLink\_type > xsd::DataLinkType::PointDataLink\_ [private]`

**9.14.5.4** `ValidPointsLink\_optional xsd::DataLinkType::ValidPointsLink\_ [private]`

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

- S:/openGPS/ISO5436\_XML/trunk/src/ISO5436\_2\_XML/iso5436\_2\_xsd.hxx
- S:/openGPS/ISO5436\_XML/trunk/src/ISO5436\_2\_XML/iso5436\_2\_xsd.cxx

## 9.15 xsd::DataListType Class Reference

```
#include <iso5436_2_xsd.hxx>
```

### 9.15.1 Detailed Description

Class corresponding to the DataListType schema type.

The datalist contains the point coordinates in ASCII. A list can by definition not contain invalid points, because it does not define a topological neighbourhood. A list is always an unsorted list of 3D-points.

#### Datum

Accessor and modifier functions for the Datum sequence element.

Datum contains a ";" separated list of X,Y,Z floating point or integer coordinates. An empty Datum tag defines an invalid data point.

- `typedef Datum_sequence::const_iterator Datum\_const\_iterator`  
*Element constant iterator type.*
- `typedef Datum_sequence::iterator Datum\_iterator`  
*Element iterator type.*
- `typedef ::xsd::cxx::tree::sequence< Datum\_type > Datum\_sequence`  
*Element sequence container type.*
- `typedef ::xsd::cxx::tree::traits< Datum\_type, wchar_t > Datum\_traits`  
*Element traits type.*
- `typedef ::xsd::Datum Datum\_type`  
*Element type.*
- `void Datum (const Datum\_sequence &s)`  
*Copy elements from a given sequence.*

- [Datum\\_sequence](#) & [Datum](#) ()  
*Return a read-write reference to the element sequence.*
- const [Datum\\_sequence](#) & [Datum](#) () const  
*Return a read-only (constant) reference to the element sequence.*

### Constructors

- virtual [DataListType](#) \* [\\_clone](#) (::xml\_schema::flags f=0,::xml\_schema::type \*c=0) const  
*Copy the object polymorphically.*
- [DataListType](#) (const [DataListType](#) &x,::xml\_schema::flags f=0,::xml\_schema::type \*c=0)  
*Copy constructor.*
- [DataListType](#) (const ::xercesc::DOMElement &e,::xml\_schema::flags f=0,::xml\_schema::type \*c=0)  
*Construct an instance from a DOM element.*
- [DataListType](#) ()  
*Construct an instance from the ultimate base and initializers for required elements and attributes.*

### Protected Member Functions

- void [parse](#) (::xsd::cxx::xml::dom::parser< wchar\_t > &,::xml\_schema::flags)

### Private Attributes

- [Datum\\_sequence](#) [Datum\\_](#)

## 9.15.2 Member Typedef Documentation

### 9.15.2.1 typedef Datum\_sequence::const\_iterator xsd::DataListType::Datum\_const\_iterator

Element constant iterator type.

### 9.15.2.2 typedef Datum\_sequence::iterator xsd::DataListType::Datum\_iterator

Element iterator type.

### 9.15.2.3 typedef ::xsd::cxx::tree::sequence< Datum\_type > xsd::DataListType::Datum\_sequence

Element sequence container type.

### 9.15.2.4 typedef ::xsd::cxx::tree::traits< Datum\_type, wchar\_t > xsd::DataListType::Datum\_traits

Element traits type.

### 9.15.2.5 typedef ::xsd::Datum xsd::DataListType::Datum\_type

Element type.

## 9.15.3 Constructor & Destructor Documentation

### 9.15.3.1 xsd::DataListType::DataListType ()

Construct an instance from the ultimate base and initializers for required elements and attributes.

### 9.15.3.2 xsd::DataListType::DataListType (const ::xercesc::DOMElement & *e*, ::xml\_schema::flags *f* = 0, ::xml\_schema::type \* *c* = 0)

Construct an instance from a DOM element.

#### Parameters:

- e* A DOM element to extract the data from.
- f* Flags to construct the new instance with.
- c* A pointer to the object that will contain the new instance.

### 9.15.3.3 xsd::DataListType::DataListType (const DataListType & *x*, ::xml\_schema::flags *f* = 0, ::xml\_schema::type \* *c* = 0)

Copy constructor.

#### Parameters:

- x* An instance to make a copy of.
- f* Flags to construct the copy with.
- c* A pointer to the object that will contain the copy.

For polymorphic object models use the `_clone` function instead.

### 9.15.4 Member Function Documentation

**9.15.4.1** `DataListType * xsd::DataListType::_clone (::xml_schema::flags f = 0, ::xml_schema::type * c = 0) const` [virtual]

Copy the object polymorphically.

**Parameters:**

- f* Flags to construct the copy with.
- c* A pointer to the object that will contain the copy.

**Returns:**

A pointer to the dynamically allocated copy.

This function ensures that the dynamic type of an instance is used for copying and should be used for polymorphic object models instead of the copy constructor.

**9.15.4.2** `void xsd::DataListType::Datum (const Datum_sequence & s)`

Copy elements from a given sequence.

**Parameters:**

- s* A sequence to copy elements from.

For each element in *s* this function makes a copy and adds it to the sequence. Note that this operation completely changes the sequence and all old elements will be lost.

**9.15.4.3** `Datum_sequence& xsd::DataListType::Datum ()`

Return a read-write reference to the element sequence.

**Returns:**

A reference to the sequence container.

**9.15.4.4** `const Datum_sequence& xsd::DataListType::Datum () const`

Return a read-only (constant) reference to the element sequence.

**Returns:**

A constant reference to the sequence container.

**9.15.4.5** `void xsd::DataListType::parse (::xsd::cxx::xml::dom::parser< wchar_t > &, ::xml_schema::flags) [protected]`

### 9.15.5 Member Data Documentation

#### 9.15.5.1 Datum\_sequence xsd::DataListType::Datum\_ [private]

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

- S:/openGPS/ISO5436\_XML/trunk/src/ISO5436\_2\_XML/iso5436\_2\_xsd.hxx
- S:/openGPS/ISO5436\_XML/trunk/src/ISO5436\_2\_XML/iso5436\_2\_xsd.cxx

## 9.16 OpenGPS::DataPointImpl Class Reference

```
#include <data_point_impl.hxx>
```

### Public Member Functions

- [DataPointImpl](#) ()
- virtual double [Get](#) () const
- virtual OGPS\_Boolean [Get](#) (double \*const value) const
- virtual OGPS\_Boolean [Get](#) (float \*const value) const
- virtual OGPS\_Boolean [Get](#) (int \*const value) const
- virtual OGPS\_Boolean [Get](#) (short \*const value) const
- virtual OGPS\_DataPointType [GetType](#) () const
- virtual OGPS\_Boolean [IsValid](#) () const
- virtual OGPS\_Boolean [Set](#) (const DataPoint &src)
- virtual OGPS\_Boolean [Set](#) (const double value)
- virtual OGPS\_Boolean [Set](#) (const float value)
- virtual OGPS\_Boolean [Set](#) (const int value)
- virtual OGPS\_Boolean [Set](#) (const short value)
- virtual OGPS\_Boolean [SetNull](#) ()
- virtual [~DataPointImpl](#) ()

### Protected Member Functions

- virtual void [Reset](#) ()

### Private Types

- typedef [OpenGPS::DataPointImpl::\\_OGPS\\_DATA\\_POINT\\_VALUE](#) [OGPS\\_DataPointValue](#)

### Private Attributes

- OGPS\_DataPointType [m\\_Type](#)
- [OGPS\\_DataPointValue](#) [m\\_Value](#)

## Classes

- union [\\_OGPS\\_DATA\\_POINT\\_VALUE](#)

### 9.16.1 Member Typedef Documentation

**9.16.1.1** typedef union [OpenGPS::DataPointImpl::\\_OGPS\\_DATA\\_POINT\\_VALUE](#) [OpenGPS::DataPointImpl::OGPS\\_DataPointValue](#) [private]

### 9.16.2 Constructor & Destructor Documentation

**9.16.2.1** [DataPointImpl::DataPointImpl \(\)](#)

**9.16.2.2** [DataPointImpl::~DataPointImpl \(\)](#) [virtual]

### 9.16.3 Member Function Documentation

**9.16.3.1** double [DataPointImpl::Get \(\) const](#) [virtual]

**9.16.3.2** OGPS\_Boolean [DataPointImpl::Get \(double \\*const value\) const](#) [virtual]

**9.16.3.3** OGPS\_Boolean [DataPointImpl::Get \(float \\*const value\) const](#) [virtual]

**9.16.3.4** OGPS\_Boolean [DataPointImpl::Get \(int \\*const value\) const](#) [virtual]

**9.16.3.5** OGPS\_Boolean [DataPointImpl::Get \(short \\*const value\) const](#) [virtual]

**9.16.3.6** OGPS\_DataPointType [DataPointImpl::GetType \(\) const](#) [virtual]

**9.16.3.7** OGPS\_Boolean [DataPointImpl::IsValid \(\) const](#) [virtual]

**9.16.3.8** void [DataPointImpl::Reset \(\)](#) [protected, virtual]

**9.16.3.9** OGPS\_Boolean [DataPointImpl::Set \(const DataPoint & src\)](#) [virtual]

**9.16.3.10 OGPS\_Boolean DataPointImpl::Set (const double value)** [virtual]

**9.16.3.11 OGPS\_Boolean DataPointImpl::Set (const float value)** [virtual]

**9.16.3.12 OGPS\_Boolean DataPointImpl::Set (const int value)** [virtual]

**9.16.3.13 OGPS\_Boolean DataPointImpl::Set (const short value)** [virtual]

**9.16.3.14 OGPS\_Boolean DataPointImpl::SetNull ()** [virtual]

#### **9.16.4 Member Data Documentation**

**9.16.4.1 OGPS\_DataPointType** [OpenGPS::DataPointImpl::m\\_Type](#)  
[private]

**9.16.4.2 OGPS\_DataPointValue** [OpenGPS::DataPointImpl::m\\_Value](#)  
[private]

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

- [S:/openGPS/ISO5436\\_XML/trunk/src/ISO5436\\_2\\_XML/cxx/data\\_point\\_impl.hxx](#)
- [S:/openGPS/ISO5436\\_XML/trunk/src/ISO5436\\_2\\_XML/cxx/data\\_point\\_impl.cxx](#)

### **9.17 OpenGPS::DataPointImpl::\_OGPS\_DATA\_POINT\_VALUE Union Reference**

#### **Public Attributes**

- double [doubleValue](#)
- float [floatValue](#)
- short [int16Value](#)
- int [int32Value](#)

#### **9.17.1 Member Data Documentation**

**9.17.1.1 double** [OpenGPS::DataPointImpl::\\_OGPS\\_DATA\\_POINT\\_VALUE::doubleValue](#)



**9.17.1.2 float** [OpenGPS::DataPointImpl::\\_OGPS\\_DATA\\_POINT\\_VALUE::floatValue](#)

**9.17.1.3 short** [OpenGPS::DataPointImpl::\\_OGPS\\_DATA\\_POINT\\_VALUE::int16Value](#)

**9.17.1.4 int** [OpenGPS::DataPointImpl::\\_OGPS\\_DATA\\_POINT\\_VALUE::int32Value](#)

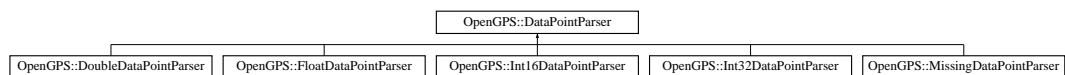
The documentation for this union was generated from the following file:

- [S:/openGPS/ISO5436\\_XML/trunk/src/ISO5436\\_2\\_XML/cxx/data\\_point\\_impl.hxx](#)

## 9.18 OpenGPS::DataPointParser Class Reference

```
#include <data_point_parser.hxx>
```

Inheritance diagram for OpenGPS::DataPointParser::



### Public Member Functions

- virtual OGPS\_Boolean [Read](#) ([PointVectorReaderContext](#) &context, DataPoint &value)=0
- virtual OGPS\_Boolean [Write](#) ([PointVectorWriterContext](#) &context, const DataPoint &value)=0
- virtual [~DataPointParser](#) ()

### Protected Member Functions

- [DataPointParser](#) ()

### 9.18.1 Constructor & Destructor Documentation

**9.18.1.1 DataPointParser::~~DataPointParser ()** [virtual]

**9.18.1.2 DataPointParser::DataPointParser ()** [protected]

## 9.18.2 Member Function Documentation

**9.18.2.1 virtual OGPS\_Boolean OpenGPS::DataPointParser::Read (Point-VectorReaderContext & context, DataPoint & value) [pure virtual]**

Implemented in [OpenGPS::DoubleDataPointParser](#), [OpenGPS::FloatDataPointParser](#), [OpenGPS::Int16DataPointParser](#), [OpenGPS::Int32DataPointParser](#), and [OpenGPS::MissingDataPointParser](#).

**9.18.2.2 virtual OGPS\_Boolean OpenGPS::DataPointParser::Write (Point-VectorWriterContext & context, const DataPoint & value) [pure virtual]**

Implemented in [OpenGPS::DoubleDataPointParser](#), [OpenGPS::FloatDataPointParser](#), [OpenGPS::Int16DataPointParser](#), [OpenGPS::Int32DataPointParser](#), and [OpenGPS::MissingDataPointParser](#).

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

- [S:/openGPS/ISO5436\\_XML/trunk/src/ISO5436\\_2\\_XML/cxx/data\\_point\\_parser.hxx](#)
- [S:/openGPS/ISO5436\\_XML/trunk/src/ISO5436\\_2\\_XML/cxx/int16\\_data\\_point\\_parser.cxx](#)

## 9.19 xsd::DataType Class Reference

```
#include <iso5436_2_xsd.hxx>
```

### 9.19.1 Detailed Description

Enumeration class corresponding to the DataType schema type.

#### Public Types

- [D](#)
- [F](#)
- [I](#)
- [L](#)
- enum [value](#) { [I](#), [L](#), [F](#), [D](#) }

*Underlying enum type.*

#### Public Member Functions

- virtual [DataType](#) \* [\\_clone](#) (::xml\_schema::flags f=0,::xml\_schema::type \*c=0) const

*Copy the object polymorphically.*

- `DataType (const DataType &x, ::xml_schema::flags f=0, ::xml_schema::type *c=0)`

*Copy constructor.*

- `DataType (const ::std::wstring &s, const ::xercesc::DOMElement *e, ::xml_schema::flags f=0, ::xml_schema::type *c=0)`

*Construct an instance from a string fragment.*

- `DataType (const ::xercesc::DOMAttr &a, ::xml_schema::flags f=0, ::xml_schema::type *c=0)`

*Construct an instance from a DOM attribute.*

- `DataType (const ::xercesc::DOMElement &e, ::xml_schema::flags f=0, ::xml_schema::type *c=0)`

*Construct an instance from a DOM element.*

- `DataType (const ::xml_schema::token &v)`

*Construct an instance from the base value.*

- `DataType (value v)`

*Construct an instance from the underlying enum value.*

- `virtual operator value () const`

*Implicit conversion operator to the underlying enum value.*

- `DataType & operator= (value v)`

*Assign the underlying enum value.*

### Static Public Attributes

- `static const value _xsd_DataType_indexes_ [4]`
- `static const wchar_t *const _xsd_DataType_literals_ [4]`

### Protected Member Functions

- `value _xsd_DataType_convert () const`

## 9.19.2 Member Enumeration Documentation

### 9.19.2.1 enum xsd::DataType::value

Underlying enum type.

**Enumerator:**

*I*  
*L*  
*F*  
*D*

**9.19.3 Constructor & Destructor Documentation****9.19.3.1 xsd::DataType::DataType (value *v*)**

Construct an instance from the underlying enum value.

**Parameters:**

*v* A enum value.

**9.19.3.2 xsd::DataType::DataType (const ::xml\_schema::token & *v*)**

Construct an instance from the base value.

**Parameters:**

*v* A base value.

**9.19.3.3 xsd::DataType::DataType (const ::xercesc::DOMElement & *e*, ::xml\_schema::flags *f* = 0, ::xml\_schema::type \* *c* = 0)**

Construct an instance from a DOM element.

**Parameters:**

*e* A DOM element to extract the data from.

*f* Flags to construct the new instance with.

*c* A pointer to the object that will contain the new instance.

**9.19.3.4 xsd::DataType::DataType (const ::xercesc::DOMAttr & *a*, ::xml\_schema::flags *f* = 0, ::xml\_schema::type \* *c* = 0)**

Construct an instance from a DOM attribute.

**Parameters:**

*a* A DOM attribute to extract the data from.

*f* Flags to construct the new instance with.

*c* A pointer to the object that will contain the new instance.

**9.19.3.5** `xsd::DataType::DataType (const ::std::wstring & s, const ::xercesc::DOMElement * e, ::xml_schema::flags f = 0, ::xml_schema::type * c = 0)`

Construct an instance from a string fragment.

**Parameters:**

- s* A string fragment to extract the data from.
- e* A DOM element containing the string fragment.
- f* Flags to construct the new instance with.
- c* A pointer to the object that will contain the new instance.

**9.19.3.6** `xsd::DataType::DataType (const DataType & x, ::xml_schema::flags f = 0, ::xml_schema::type * c = 0)`

Copy constructor.

**Parameters:**

- x* An instance to make a copy of.
- f* Flags to construct the copy with.
- c* A pointer to the object that will contain the copy.

For polymorphic object models use the `_clone` function instead.

## 9.19.4 Member Function Documentation

**9.19.4.1** `DataType * xsd::DataType::_clone (::xml_schema::flags f = 0, ::xml_schema::type * c = 0) const` [virtual]

Copy the object polymorphically.

**Parameters:**

- f* Flags to construct the copy with.
- c* A pointer to the object that will contain the copy.

**Returns:**

A pointer to the dynamically allocated copy.

This function ensures that the dynamic type of an instance is used for copying and should be used for polymorphic object models instead of the copy constructor.

**9.19.4.2** `DataType::value xsd::DataType::_xsd_DataType_convert () const` [protected]

**9.19.4.3** `virtual xsd::DataType::operator value () const [inline, virtual]`

Implicit conversion operator to the underlying enum value.

**Returns:**

A enum value.

**9.19.4.4** `DataType& xsd::DataType::operator= (value v)`

Assign the underlying enum value.

**Parameters:**

*v* A enum value.

**Returns:**

A refernce to the instance.

## 9.19.5 Member Data Documentation

**9.19.5.1** `const value xsd::DataType::_xsd_DataType_indexes_[4] [static]`

**9.19.5.2** `const wchar_t* const xsd::DataType::_xsd_DataType_literals_[4] [static]`

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

- [S:/openGPS/ISO5436\\_XML/trunk/src/ISO5436\\_2\\_XML/iso5436\\_2\\_xsd.hxx](#)
- [S:/openGPS/ISO5436\\_XML/trunk/src/ISO5436\\_2\\_XML/iso5436\\_2\\_xsd.cxx](#)

## 9.20 xsd::Datum Class Reference

```
#include <iso5436_2_xsd.hxx>
```

### 9.20.1 Detailed Description

Class corresponding to the Datum schema type.

#### Constructors

- `virtual Datum * _clone (::xml_schema::flags f=0,::xml_schema::type *c=0) const`  
*Copy the object polymorphically.*

- **Datum** (const **Datum** &x, ::xml\_schema::flags f=0, ::xml\_schema::type \*c=0)  
*Copy constructor.*
- **Datum** (const ::std::wstring &s, const ::xercesc::DOMElement \*e, ::xml\_schema::flags f=0, ::xml\_schema::type \*c=0)  
*Construct an instance from a string fragment.*
- **Datum** (const ::xercesc::DOMAttr &a, ::xml\_schema::flags f=0, ::xml\_schema::type \*c=0)  
*Construct an instance from a DOM attribute.*
- **Datum** (const ::xercesc::DOMElement &e, ::xml\_schema::flags f=0, ::xml\_schema::type \*c=0)  
*Construct an instance from a DOM element.*
- **Datum** (const ::xml\_schema::token &)  
*Construct an instance from the ultimate base and initializers for required elements and attributes.*
- **Datum** ()  
*Construct an instance from initializers for required elements and attributes.*

## 9.20.2 Constructor & Destructor Documentation

### 9.20.2.1 xsd::Datum::Datum ()

Construct an instance from initializers for required elements and attributes.

### 9.20.2.2 xsd::Datum::Datum (const ::xml\_schema::token &)

Construct an instance from the ultimate base and initializers for required elements and attributes.

### 9.20.2.3 xsd::Datum::Datum (const ::xercesc::DOMElement & e, ::xml\_schema::flags f = 0, ::xml\_schema::type \* c = 0)

Construct an instance from a DOM element.

#### Parameters:

- e* A DOM element to extract the data from.
- f* Flags to construct the new instance with.
- c* A pointer to the object that will contain the new instance.

**9.20.2.4** `xsd::Datum::Datum (const ::xercesc::DOMAttr & a, ::xml_schema::flags f = 0, ::xml_schema::type * c = 0)`

Construct an instance from a DOM attribute.

**Parameters:**

- a* A DOM attribute to extract the data from.
- f* Flags to construct the new instance with.
- c* A pointer to the object that will contain the new instance.

**9.20.2.5** `xsd::Datum::Datum (const ::std::wstring & s, const ::xercesc::DOMElement * e, ::xml_schema::flags f = 0, ::xml_schema::type * c = 0)`

Construct an instance from a string fragment.

**Parameters:**

- s* A string fragment to extract the data from.
- e* A DOM element containing the string fragment.
- f* Flags to construct the new instance with.
- c* A pointer to the object that will contain the new instance.

**9.20.2.6** `xsd::Datum::Datum (const Datum & x, ::xml_schema::flags f = 0, ::xml_schema::type * c = 0)`

Copy constructor.

**Parameters:**

- x* An instance to make a copy of.
- f* Flags to construct the copy with.
- c* A pointer to the object that will contain the copy.

For polymorphic object models use the `_clone` function instead.

### 9.20.3 Member Function Documentation

**9.20.3.1** `Datum * xsd::Datum::_clone (::xml_schema::flags f = 0, ::xml_schema::type * c = 0) const [virtual]`

Copy the object polymorphically.

**Parameters:**

- f* Flags to construct the copy with.



*c* A pointer to the object that will contain the copy.

#### Returns:

A pointer to the dynamically allocated copy.

This function ensures that the dynamic type of an instance is used for copying and should be used for polymorphic object models instead of the copy constructor.

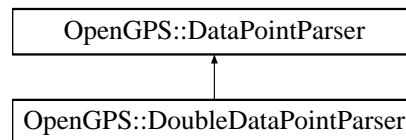
The documentation for this class was generated from the following files:

- S:/openGPS/ISO5436\_XML/trunk/src/ISO5436\_2\_XML/[iso5436\\_2\\_xsd.hxx](#)
- S:/openGPS/ISO5436\_XML/trunk/src/ISO5436\_2\_XML/[iso5436\\_2\\_xsd.cxx](#)

## 9.21 OpenGPS::DoubleDataPointParser Class Reference

```
#include <double_data_point_parser.hxx>
```

Inheritance diagram for OpenGPS::DoubleDataPointParser::



### Public Member Functions

- [DoubleDataPointParser](#) ()
- virtual OGPS\_Boolean [Read](#) ([PointVectorReaderContext](#) &context, DataPoint &value)
- virtual OGPS\_Boolean [Write](#) ([PointVectorWriterContext](#) &context, const DataPoint &value)
- virtual [~DoubleDataPointParser](#) ()

### 9.21.1 Constructor & Destructor Documentation

#### 9.21.1.1 DoubleDataPointParser::DoubleDataPointParser ()

#### 9.21.1.2 DoubleDataPointParser::~~DoubleDataPointParser () [virtual]

### 9.21.2 Member Function Documentation

#### 9.21.2.1 OGPS\_Boolean DoubleDataPointParser::Read ([PointVectorReaderContext](#) &context, DataPoint &value) [virtual]

Implements [OpenGPS::DataPointParser](#).

### 9.21.2.2 OGPS\_Boolean DoubleDataPointParser::Write ([PointVectorWriter-Context](#) & *context*, const [DataPoint](#) & *value*) [virtual]

Implements [OpenGPS::DataPointParser](#).

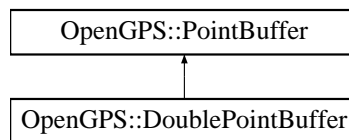
The documentation for this class was generated from the following files:

- S:/openGPS/ISO5436\_XML/trunk/src/ISO5436\_2\_XML/cxx/[double\\_data\\_point\\_parser.hxx](#)
- S:/openGPS/ISO5436\_XML/trunk/src/ISO5436\_2\_XML/cxx/[double\\_data\\_point\\_parser.cxx](#)

## 9.22 OpenGPS::DoublePointBuffer Class Reference

```
#include <double_point_buffer.hxx>
```

Inheritance diagram for OpenGPS::DoublePointBuffer::



### Public Member Functions

- virtual OGPS\_Boolean [Allocate](#) (const unsigned long size)
- [DoublePointBuffer](#) ()
- virtual OGPS\_Boolean [Get](#) (const unsigned long index, double &value) const
- virtual OGPS\_DataPointType [GetType](#) () const
- virtual OGPS\_Boolean [Set](#) (const unsigned long index, const double value)
- virtual OGPS\_Boolean [SetNull](#) (const unsigned long index)
- virtual [~DoublePointBuffer](#) ()

### Private Attributes

- double \* [m\\_Buffer](#)

### 9.22.1 Constructor & Destructor Documentation

#### 9.22.1.1 DoublePointBuffer::DoublePointBuffer ()

#### 9.22.1.2 DoublePointBuffer::~~DoublePointBuffer () [virtual]

### 9.22.2 Member Function Documentation

**9.22.2.1 OGPS\_Boolean DoublePointBuffer::Allocate** (const unsigned long *size*)  
[virtual]

Implements [OpenGPS::PointBuffer](#).

**9.22.2.2 OGPS\_Boolean DoublePointBuffer::Get** (const unsigned long *index*,  
double & *value*) const [virtual]

Reimplemented from [OpenGPS::PointBuffer](#).

**9.22.2.3 OGPS\_DataPointType DoublePointBuffer::GetType** () const  
[virtual]

Implements [OpenGPS::PointBuffer](#).

**9.22.2.4 OGPS\_Boolean DoublePointBuffer::Set** (const unsigned long *index*,  
const double *value*) [virtual]

Reimplemented from [OpenGPS::PointBuffer](#).

**9.22.2.5 OGPS\_Boolean DoublePointBuffer::SetNull** (const unsigned long *index*) [virtual]

Implements [OpenGPS::PointBuffer](#).

### 9.22.3 Member Data Documentation

**9.22.3.1 double\* [OpenGPS::DoublePointBuffer::m\\_Buffer](#)** [private]

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

- S:/openGPS/ISO5436\_XML/trunk/src/ISO5436\_2\_XML/cxx/[double\\_point\\_buffer.hxx](#)
- S:/openGPS/ISO5436\_XML/trunk/src/ISO5436\_2\_XML/cxx/[double\\_point\\_buffer.cxx](#)

## 9.23 OpenGPS::Environment Class Reference

```
#include <environment.hxx>
```

### Public Member Functions

- void [ByteSwap](#) (const OpenGPS::UnsignedBytePtr src, double \*const value) const
- OpenGPS::UnsignedBytePtr [ByteSwap](#) (const double \*value, OpenGPS::UnsignedBytePtr dst) const

- void [ByteSwap](#) (const OpenGPS::UnsignedBytePtr src, float \*const value) const
- OpenGPS::UnsignedBytePtr [ByteSwap](#) (const float \*value, OpenGPS::UnsignedBytePtr dst) const
- void [ByteSwap](#) (const OpenGPS::UnsignedBytePtr src, int \*const value) const
- OpenGPS::UnsignedBytePtr [ByteSwap](#) (const int \*value, OpenGPS::UnsignedBytePtr dst) const
- void [ByteSwap](#) (const OpenGPS::UnsignedBytePtr src, short \*const value) const
- OpenGPS::UnsignedBytePtr [ByteSwap](#) (const short \*value, OpenGPS::UnsignedBytePtr dst) const
- void [ByteSwap16](#) (const OpenGPS::UnsignedBytePtr src, short \*const value) const
- OpenGPS::UnsignedBytePtr [ByteSwap16](#) (const short \*value, OpenGPS::UnsignedBytePtr dst) const
- void [ByteSwap32](#) (const OpenGPS::UnsignedBytePtr src, float \*const value) const
- OpenGPS::UnsignedBytePtr [ByteSwap32](#) (const float \*value, OpenGPS::UnsignedBytePtr dst) const
- void [ByteSwap32](#) (const OpenGPS::UnsignedBytePtr src, int \*const value) const
- OpenGPS::UnsignedBytePtr [ByteSwap32](#) (const int \*value, OpenGPS::UnsignedBytePtr dst) const
- void [ByteSwap64](#) (const OpenGPS::UnsignedBytePtr src, double \*const value) const
- OpenGPS::UnsignedBytePtr [ByteSwap64](#) (const double \*value, OpenGPS::UnsignedBytePtr dst) const
- virtual OpenGPS::String [ConcatPathes](#) (const OpenGPS::String &path1, const OpenGPS::String &path2) const =0
- virtual OGPS\_Boolean [CreateDir](#) (const OpenGPS::String &path) const =0
- virtual OGPS\_Character [GetAltDirectorySeparator](#) () const =0
- virtual OGPS\_Character [GetDirectorySeparator](#) () const =0
- virtual OpenGPS::String [GetPathName](#) (const OpenGPS::String &path) const =0
- virtual OpenGPS::String [GetTempDir](#) () const =0
- virtual OpenGPS::String [GetUniqueName](#) () const =0
- virtual OGPS\_Boolean [IsLittleEndian](#) () const
- virtual OGPS\_Boolean [PathExists](#) (const OpenGPS::String &file) const =0
- virtual OGPS\_Boolean [RemoveDir](#) (const OpenGPS::String &path) const =0
- virtual OGPS\_Boolean [RemoveFile](#) (const OpenGPS::String &file) const =0
- virtual OGPS\_Boolean [RenameFile](#) (const String &src, const String &dst) const =0

#### Static Public Member Functions

- static const [Environment](#) \* [GetInstance](#) ()
- static void [Reset](#) ()

#### Protected Member Functions

- [Environment](#) ()
- virtual [~Environment](#) ()

### Static Protected Member Functions

- static [Environment](#) \* [CreateInstance](#) ()

### Static Private Attributes

- static [Environment](#) \* [m\\_Instance](#) = NULL

## 9.23.1 Constructor & Destructor Documentation

**9.23.1.1** [Environment::Environment](#) () [protected]

**9.23.1.2** [Environment::~~Environment](#) () [protected, virtual]

## 9.23.2 Member Function Documentation

**9.23.2.1** void [Environment::ByteSwap](#) (const [OpenGPS::UnsignedBytePtr](#) *src*, double \*const *value*) const

**9.23.2.2** [OpenGPS::UnsignedBytePtr](#) [Environment::ByteSwap](#) (const double \* *value*, [OpenGPS::UnsignedBytePtr](#) *dst*) const

**9.23.2.3** void [Environment::ByteSwap](#) (const [OpenGPS::UnsignedBytePtr](#) *src*, float \*const *value*) const

**9.23.2.4** [OpenGPS::UnsignedBytePtr](#) [Environment::ByteSwap](#) (const float \* *value*, [OpenGPS::UnsignedBytePtr](#) *dst*) const

**9.23.2.5** void [Environment::ByteSwap](#) (const [OpenGPS::UnsignedBytePtr](#) *src*, int \*const *value*) const

**9.23.2.6** [OpenGPS::UnsignedBytePtr](#) [Environment::ByteSwap](#) (const int \* *value*, [OpenGPS::UnsignedBytePtr](#) *dst*) const

**9.23.2.7** void [Environment::ByteSwap](#) (const [OpenGPS::UnsignedBytePtr](#) *src*, short \*const *value*) const

**9.23.2.8** [OpenGPS::UnsignedBytePtr](#) [Environment::ByteSwap](#) (const short \* *value*, [OpenGPS::UnsignedBytePtr](#) *dst*) const

**9.23.2.9** void [Environment::ByteSwap16](#) (const [OpenGPS::UnsignedBytePtr](#) *src*, short \*const *value*) const

**9.23.2.10** OpenGPS::UnsignedBytePtr Environment::ByteSwap16 (const short \* *value*, OpenGPS::UnsignedBytePtr *dst*) const

**9.23.2.11** void Environment::ByteSwap32 (const OpenGPS::UnsignedBytePtr *src*, float \*const *value*) const

**9.23.2.12** OpenGPS::UnsignedBytePtr Environment::ByteSwap32 (const float \* *value*, OpenGPS::UnsignedBytePtr *dst*) const

**9.23.2.13** void Environment::ByteSwap32 (const OpenGPS::UnsignedBytePtr *src*, int \*const *value*) const

**9.23.2.14** OpenGPS::UnsignedBytePtr Environment::ByteSwap32 (const int \* *value*, OpenGPS::UnsignedBytePtr *dst*) const

**9.23.2.15** void Environment::ByteSwap64 (const OpenGPS::UnsignedBytePtr *src*, double \*const *value*) const

**9.23.2.16** OpenGPS::UnsignedBytePtr Environment::ByteSwap64 (const double \* *value*, OpenGPS::UnsignedBytePtr *dst*) const

**9.23.2.17** virtual OpenGPS::String OpenGPS::Environment::ConcatPathes (const OpenGPS::String & *path1*, const OpenGPS::String & *path2*) const  
[pure virtual]

**9.23.2.18** virtual OGPS\_Boolean OpenGPS::Environment::CreateDir (const OpenGPS::String & *path*) const [pure virtual]

**9.23.2.19** static Environment\* OpenGPS::Environment::CreateInstance ()  
[static, protected]

**9.23.2.20** virtual OGPS\_Character OpenGPS::Environment::GetAltDirectory-Separator () const [pure virtual]

**9.23.2.21** virtual OGPS\_Character OpenGPS::Environment::GetDirectory-Separator () const [pure virtual]

**9.23.2.22** const Environment\* Environment::GetInstance () [static]

**9.23.2.23** virtual OpenGPS::String OpenGPS::Environment::GetPathName (const OpenGPS::String & *path*) const [pure virtual]

**9.23.2.24** `virtual OpenGPS::String OpenGPS::Environment::GetTempDir () const [pure virtual]`

**9.23.2.25** `virtual OpenGPS::String OpenGPS::Environment::GetUniqueName () const [pure virtual]`

**9.23.2.26** `OGPS_Boolean Environment::IsLittleEndian () const [virtual]`

**9.23.2.27** `virtual OGPS_Boolean OpenGPS::Environment::PathExists (const OpenGPS::String &file) const [pure virtual]`

**9.23.2.28** `virtual OGPS_Boolean OpenGPS::Environment::RemoveDir (const OpenGPS::String &path) const [pure virtual]`

**9.23.2.29** `virtual OGPS_Boolean OpenGPS::Environment::RemoveFile (const OpenGPS::String &file) const [pure virtual]`

**9.23.2.30** `virtual OGPS_Boolean OpenGPS::Environment::RenameFile (const String &src, const String &dst) const [pure virtual]`

**9.23.2.31** `void Environment::Reset () [static]`

### 9.23.3 Member Data Documentation

**9.23.3.1** `Environment * Environment::m_Instance = NULL [static, private]`

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

- S:/openGPS/ISO5436\_XML/trunk/src/ISO5436\_2\_XML/cxx/[environment.hxx](#)
- S:/openGPS/ISO5436\_XML/trunk/src/ISO5436\_2\_XML/cxx/[environment.cxx](#)

## 9.24 xsd::FeatureType Class Reference

```
#include <iso5436_2_xsd.hxx>
```

### 9.24.1 Detailed Description

Class corresponding to the FeatureType schema type.

## Constructors

- virtual `FeatureType * _clone (::xml_schema::flags f=0, ::xml_schema::type *c=0) const`  
*Copy the object polymorphically.*
- `FeatureType (const FeatureType &x, ::xml_schema::flags f=0, ::xml_schema::type *c=0)`  
*Copy constructor.*
- `FeatureType (const ::std::wstring &s, const ::xercesc::DOMElement *e, ::xml_schema::flags f=0, ::xml_schema::type *c=0)`  
*Construct an instance from a string fragment.*
- `FeatureType (const ::xercesc::DOMAttr &a, ::xml_schema::flags f=0, ::xml_schema::type *c=0)`  
*Construct an instance from a DOM attribute.*
- `FeatureType (const ::xercesc::DOMElement &e, ::xml_schema::flags f=0, ::xml_schema::type *c=0)`  
*Construct an instance from a DOM element.*
- `FeatureType (const ::xml_schema::token &)`  
*Construct an instance from the ultimate base and initializers for required elements and attributes.*
- `FeatureType ()`  
*Construct an instance from initializers for required elements and attributes.*

## 9.24.2 Constructor & Destructor Documentation

### 9.24.2.1 xsd::FeatureType::FeatureType ()

Construct an instance from initializers for required elements and attributes.

### 9.24.2.2 xsd::FeatureType::FeatureType (const ::xml\_schema::token &)

Construct an instance from the ultimate base and initializers for required elements and attributes.

### 9.24.2.3 xsd::FeatureType::FeatureType (const ::xercesc::DOMElement & e, ::xml\_schema::flags f = 0, ::xml\_schema::type \* c = 0)

Construct an instance from a DOM element.

## Parameters:

- e* A DOM element to extract the data from.



*f* Flags to construct the new instance with.  
*c* A pointer to the object that will contain the new instance.

**9.24.2.4 xsd::FeatureType::FeatureType (const ::xercesc::DOMAttr & *a*, ::xml\_schema::flags *f* = 0, ::xml\_schema::type \* *c* = 0)**

Construct an instance from a DOM attribute.

**Parameters:**

*a* A DOM attribute to extract the data from.  
*f* Flags to construct the new instance with.  
*c* A pointer to the object that will contain the new instance.

**9.24.2.5 xsd::FeatureType::FeatureType (const ::std::wstring & *s*, const ::xercesc::DOMElement \* *e*, ::xml\_schema::flags *f* = 0, ::xml\_schema::type \* *c* = 0)**

Construct an instance from a string fragment.

**Parameters:**

*s* A string fragment to extract the data from.  
*e* A DOM element containing the string fragment.  
*f* Flags to construct the new instance with.  
*c* A pointer to the object that will contain the new instance.

**9.24.2.6 xsd::FeatureType::FeatureType (const FeatureType & *x*, ::xml\_schema::flags *f* = 0, ::xml\_schema::type \* *c* = 0)**

Copy constructor.

**Parameters:**

*x* An instance to make a copy of.  
*f* Flags to construct the copy with.  
*c* A pointer to the object that will contain the copy.

For polymorphic object models use the `_clone` function instead.

### 9.24.3 Member Function Documentation

**9.24.3.1 FeatureType \* xsd::FeatureType::\_clone (::xml\_schema::flags *f* = 0, ::xml\_schema::type \* *c* = 0) const** [virtual]

Copy the object polymorphically.

**Parameters:**

- f* Flags to construct the copy with.
- c* A pointer to the object that will contain the copy.

**Returns:**

A pointer to the dynamically allocated copy.

This function ensures that the dynamic type of an instance is used for copying and should be used for polymorphic object models instead of the copy constructor.

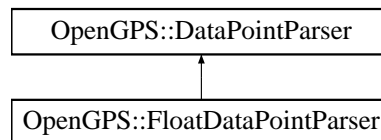
The documentation for this class was generated from the following files:

- [S:/openGPS/ISO5436\\_XML/trunk/src/ISO5436\\_2\\_XML/iso5436\\_2\\_xsd.hxx](#)
- [S:/openGPS/ISO5436\\_XML/trunk/src/ISO5436\\_2\\_XML/iso5436\\_2\\_xsd.cxx](#)

**9.25 OpenGPS::FloatDataPointParser Class Reference**

```
#include <float_data_point_parser.hxx>
```

Inheritance diagram for OpenGPS::FloatDataPointParser::

**Public Member Functions**

- [FloatDataPointParser](#) ()
- virtual OGPS\_Boolean [Read](#) ([PointVectorReaderContext](#) &context, DataPoint &value)
- virtual OGPS\_Boolean [Write](#) ([PointVectorWriterContext](#) &context, const DataPoint &value)
- virtual [~FloatDataPointParser](#) ()

**9.25.1 Constructor & Destructor Documentation****9.25.1.1 FloatDataPointParser::FloatDataPointParser ()****9.25.1.2 FloatDataPointParser::~~FloatDataPointParser ()** [virtual]

### 9.25.2 Member Function Documentation

#### 9.25.2.1 OGPS\_Boolean FloatDataPointParser::Read ([PointVectorReader-Context](#) & *context*, [DataPoint](#) & *value*) [virtual]

Implements [OpenGPS::DataPointParser](#).

#### 9.25.2.2 OGPS\_Boolean FloatDataPointParser::Write ([PointVectorWriter-Context](#) & *context*, const [DataPoint](#) & *value*) [virtual]

Implements [OpenGPS::DataPointParser](#).

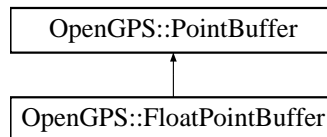
The documentation for this class was generated from the following files:

- [S:/openGPS/ISO5436\\_XML/trunk/src/ISO5436\\_2\\_XML/cxx/float\\_data\\_point\\_parser.hxx](#)
- [S:/openGPS/ISO5436\\_XML/trunk/src/ISO5436\\_2\\_XML/cxx/float\\_data\\_point\\_parser.cxx](#)

## 9.26 OpenGPS::FloatPointBuffer Class Reference

```
#include <float_point_buffer.hxx>
```

Inheritance diagram for OpenGPS::FloatPointBuffer::



### Public Member Functions

- virtual OGPS\_Boolean [Allocate](#) (const unsigned long size)
- [FloatPointBuffer](#) ()
- virtual OGPS\_Boolean [Get](#) (const unsigned long index, float &value) const
- virtual OGPS\_DataPointType [GetType](#) () const
- virtual OGPS\_Boolean [Set](#) (const unsigned long index, const float value)
- virtual OGPS\_Boolean [SetNull](#) (const unsigned long index)
- virtual [~FloatPointBuffer](#) ()

### Private Attributes

- float \* [m\\_Buffer](#)

### 9.26.1 Constructor & Destructor Documentation

#### 9.26.1.1 FloatPointBuffer::FloatPointBuffer ()

**9.26.1.2 FloatPointBuffer::~FloatPointBuffer ()** [virtual]**9.26.2 Member Function Documentation****9.26.2.1 OGPS\_Boolean FloatPointBuffer::Allocate (const unsigned long *size*)** [virtual]

Implements [OpenGPS::PointBuffer](#).

**9.26.2.2 OGPS\_Boolean FloatPointBuffer::Get (const unsigned long *index*, float & *value*) const** [virtual]

Reimplemented from [OpenGPS::PointBuffer](#).

**9.26.2.3 OGPS\_DataPointType FloatPointBuffer::GetType () const** [virtual]

Implements [OpenGPS::PointBuffer](#).

**9.26.2.4 OGPS\_Boolean FloatPointBuffer::Set (const unsigned long *index*, const float *value*)** [virtual]

Reimplemented from [OpenGPS::PointBuffer](#).

**9.26.2.5 OGPS\_Boolean FloatPointBuffer::SetNull (const unsigned long *index*)** [virtual]

Implements [OpenGPS::PointBuffer](#).

**9.26.3 Member Data Documentation****9.26.3.1 float\* [OpenGPS::FloatPointBuffer::m\\_Buffer](#)** [private]

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

- [S:/openGPS/ISO5436\\_XML/trunk/src/ISO5436\\_2\\_XML/cxx/float\\_point\\_buffer.hxx](#)
- [S:/openGPS/ISO5436\\_XML/trunk/src/ISO5436\\_2\\_XML/cxx/float\\_point\\_buffer.cxx](#)

**9.27 xsd::InstrumentType Class Reference**

```
#include <iso5436_2_xsd.hxx>
```

**9.27.1 Detailed Description**

Class corresponding to the InstrumentType schema type.

## Manufacturer

Accessor and modifier functions for the Manufacturer required element.

Name of the equipment manufacturer

- `typedef ::xsd::cxx::tree::traits< Manufacturer\_type, wchar_t > Manufacturer\_traits`  
*Element traits type.*
- `typedef ::xml_schema::token Manufacturer\_type`  
*Element type.*
- `void Manufacturer (::std::auto_ptr< Manufacturer\_type > p)`  
*Set the element value without copying.*
- `void Manufacturer (const Manufacturer\_type &x)`  
*Set the element value.*
- `Manufacturer\_type & Manufacturer ()`  
*Return a read-write reference to the element.*
- `const Manufacturer\_type & Manufacturer () const`  
*Return a read-only (constant) reference to the element.*

## Model

Accessor and modifier functions for the Model required element.

Name of the machine model used for the measurement

- `typedef ::xsd::cxx::tree::traits< Model\_type, wchar_t > Model\_traits`  
*Element traits type.*
- `typedef ::xml_schema::token Model\_type`  
*Element type.*
- `void Model (::std::auto_ptr< Model\_type > p)`  
*Set the element value without copying.*
- `void Model (const Model\_type &x)`  
*Set the element value.*
- `Model\_type & Model ()`  
*Return a read-write reference to the element.*

- const [Model\\_type](#) & [Model](#) () const  
*Return a read-only (constant) reference to the element.*

## Serial

Accessor and modifier functions for the Serial required element.

Serial number of the machine.

- typedef ::xsd::cxx::tree::traits< [Serial\\_type](#), wchar\_t > [Serial\\_traits](#)  
*Element traits type.*
- typedef ::xml\_schema::token [Serial\\_type](#)  
*Element type.*
- void [Serial](#) (::std::auto\_ptr< [Serial\\_type](#) > p)  
*Set the element value without copying.*
- void [Serial](#) (const [Serial\\_type](#) &x)  
*Set the element value.*
- [Serial\\_type](#) & [Serial](#) ()  
*Return a read-write reference to the element.*
- const [Serial\\_type](#) & [Serial](#) () const  
*Return a read-only (constant) reference to the element.*

## Version

Accessor and modifier functions for the Version required element.

Software and hardware version strings used to create this file.

- typedef ::xsd::cxx::tree::traits< [Version\\_type](#), wchar\_t > [Version\\_traits](#)  
*Element traits type.*
- typedef ::xml\_schema::token [Version\\_type](#)  
*Element type.*
- void [Version](#) (::std::auto\_ptr< [Version\\_type](#) > p)  
*Set the element value without copying.*
- void [Version](#) (const [Version\\_type](#) &x)  
*Set the element value.*

- [Version\\_type](#) & [Version](#) ()  
*Return a read-write reference to the element.*
- const [Version\\_type](#) & [Version](#) () const  
*Return a read-only (constant) reference to the element.*

### Constructors

- virtual [InstrumentType](#) \* [\\_clone](#) (::xml\_schema::flags f=0,::xml\_schema::type \*c=0) const  
*Copy the object polymorphically.*
- [InstrumentType](#) (const [InstrumentType](#) &x,::xml\_schema::flags f=0,::xml\_schema::type \*c=0)  
*Copy constructor.*
- [InstrumentType](#) (const ::xercesc::DOMElement &e,::xml\_schema::flags f=0,::xml\_schema::type \*c=0)  
*Construct an instance from a DOM element.*
- [InstrumentType](#) (const [Manufacturer\\_type](#) &, const [Model\\_type](#) &, const [Serial\\_type](#) &, const [Version\\_type](#) &)  
*Construct an instance from the ultimate base and initializers for required elements and attributes.*

### Protected Member Functions

- void [parse](#) (::xsd::cxx::xml::dom::parser< wchar\_t > &,::xml\_schema::flags)

### Private Attributes

- ::xsd::cxx::tree::one< [Manufacturer\\_type](#) > [Manufacturer\\_](#)
- ::xsd::cxx::tree::one< [Model\\_type](#) > [Model\\_](#)
- ::xsd::cxx::tree::one< [Serial\\_type](#) > [Serial\\_](#)
- ::xsd::cxx::tree::one< [Version\\_type](#) > [Version\\_](#)

## 9.27.2 Member Typedef Documentation

### 9.27.2.1 typedef ::xsd::cxx::tree::traits< [Manufacturer\\_type](#), wchar\_t > [xsd::InstrumentType::Manufacturer\\_traits](#)

Element traits type.

### 9.27.2.2 typedef ::xml\_schema::token xsd::InstrumentType::Manufacturer\_type

Element type.

### 9.27.2.3 typedef ::xsd::cxx::tree::traits< Model\_type, wchar\_t > xsd::InstrumentType::Model\_traits

Element traits type.

### 9.27.2.4 typedef ::xml\_schema::token xsd::InstrumentType::Model\_type

Element type.

### 9.27.2.5 typedef ::xsd::cxx::tree::traits< Serial\_type, wchar\_t > xsd::InstrumentType::Serial\_traits

Element traits type.

### 9.27.2.6 typedef ::xml\_schema::token xsd::InstrumentType::Serial\_type

Element type.

### 9.27.2.7 typedef ::xsd::cxx::tree::traits< Version\_type, wchar\_t > xsd::InstrumentType::Version\_traits

Element traits type.

### 9.27.2.8 typedef ::xml\_schema::token xsd::InstrumentType::Version\_type

Element type.

## 9.27.3 Constructor & Destructor Documentation

### 9.27.3.1 xsd::InstrumentType::InstrumentType (const Manufacturer\_type &, const Model\_type &, const Serial\_type &, const Version\_type &)

Construct an instance from the ultimate base and initializers for required elements and attributes.

### 9.27.3.2 xsd::InstrumentType::InstrumentType (const ::xercesc::DOMElement & *e*, ::xml\_schema::flags *f* = 0, ::xml\_schema::type \* *c* = 0)

Construct an instance from a DOM element.

#### Parameters:

*e* A DOM element to extract the data from.

*f* Flags to construct the new instance with.



*c* A pointer to the object that will contain the new instance.

**9.27.3.3 xsd::InstrumentType::InstrumentType (const InstrumentType & *x*, ::xml\_schema::flags *f* = 0, ::xml\_schema::type \* *c* = 0)**

Copy constructor.

**Parameters:**

- x* An instance to make a copy of.
- f* Flags to construct the copy with.
- c* A pointer to the object that will contain the copy.

For polymorphic object models use the `_clone` function instead.

**9.27.4 Member Function Documentation**

**9.27.4.1 InstrumentType \* xsd::InstrumentType::\_clone (::xml\_schema::flags *f* = 0, ::xml\_schema::type \* *c* = 0) const** [virtual]

Copy the object polymorphically.

**Parameters:**

- f* Flags to construct the copy with.
- c* A pointer to the object that will contain the copy.

**Returns:**

A pointer to the dynamically allocated copy.

This function ensures that the dynamic type of an instance is used for copying and should be used for polymorphic object models instead of the copy constructor.

**9.27.4.2 void xsd::InstrumentType::Manufacturer (::std::auto\_ptr< Manufacturer\_type > *p*)**

Set the element value without copying.

**Parameters:**

- p* A new value to use.

This function will try to use the passed value directly instead of making a copy.

#### 9.27.4.3 void xsd::InstrumentType::Manufacturer (const [Manufacturer\\_type](#) & *x*)

Set the element value.

##### Parameters:

*x* A new value to set.

This function makes a copy of its argument and sets it as the new value of the element.

#### 9.27.4.4 [Manufacturer\\_type](#)& xsd::InstrumentType::Manufacturer ()

Return a read-write reference to the element.

##### Returns:

A reference to the element.

#### 9.27.4.5 const [Manufacturer\\_type](#)& xsd::InstrumentType::Manufacturer () const

Return a read-only (constant) reference to the element.

##### Returns:

A constant reference to the element.

#### 9.27.4.6 void xsd::InstrumentType::Model (::std::auto\_ptr< [Model\\_type](#) > *p*)

Set the element value without copying.

##### Parameters:

*p* A new value to use.

This function will try to use the passed value directly instead of making a copy.

#### 9.27.4.7 void xsd::InstrumentType::Model (const [Model\\_type](#) & *x*)

Set the element value.

##### Parameters:

*x* A new value to set.

This function makes a copy of its argument and sets it as the new value of the element.

#### 9.27.4.8 [Model\\_type](#)& xsd::InstrumentType::Model ()

Return a read-write reference to the element.

**Returns:**

A reference to the element.

#### 9.27.4.9 const [Model\\_type](#)& xsd::InstrumentType::Model () const

Return a read-only (constant) reference to the element.

**Returns:**

A constant reference to the element.

#### 9.27.4.10 void xsd::InstrumentType::parse (::xsd::cxx::xml::dom::parser< wchar\_t > &, [::xml\\_schema::flags](#)) [protected]

#### 9.27.4.11 void xsd::InstrumentType::Serial (::std::auto\_ptr< [Serial\\_type](#) > p)

Set the element value without copying.

**Parameters:**

*p* A new value to use.

This function will try to use the passed value directly instead of making a copy.

#### 9.27.4.12 void xsd::InstrumentType::Serial (const [Serial\\_type](#) & x)

Set the element value.

**Parameters:**

*x* A new value to set.

This function makes a copy of its argument and sets it as the new value of the element.

#### 9.27.4.13 [Serial\\_type](#)& xsd::InstrumentType::Serial ()

Return a read-write reference to the element.

**Returns:**

A reference to the element.

**9.27.4.14** `const Serial_type& xsd::InstrumentType::Serial () const`

Return a read-only (constant) reference to the element.

**Returns:**

A constant reference to the element.

**9.27.4.15** `void xsd::InstrumentType::Version (::std::auto_ptr< Version_type > p)`

Set the element value without copying.

**Parameters:**

*p* A new value to use.

This function will try to use the passed value directly instead of making a copy.

**9.27.4.16** `void xsd::InstrumentType::Version (const Version_type & x)`

Set the element value.

**Parameters:**

*x* A new value to set.

This function makes a copy of its argument and sets it as the new value of the element.

**9.27.4.17** `Version_type& xsd::InstrumentType::Version ()`

Return a read-write reference to the element.

**Returns:**

A reference to the element.

**9.27.4.18** `const Version_type& xsd::InstrumentType::Version () const`

Return a read-only (constant) reference to the element.

**Returns:**

A constant reference to the element.

**9.27.5 Member Data Documentation****9.27.5.1** `::xsd::cxx::tree::one< Manufacturer_type > xsd::InstrumentType::Manufacturer_ [private]`

**9.27.5.2** `::xsd::cxx::tree::one< Model\_type > xsd::InstrumentType::Model\_ [private]`

**9.27.5.3** `::xsd::cxx::tree::one< Serial\_type > xsd::InstrumentType::Serial\_ [private]`

**9.27.5.4** `::xsd::cxx::tree::one< Version\_type > xsd::InstrumentType::Version\_ [private]`

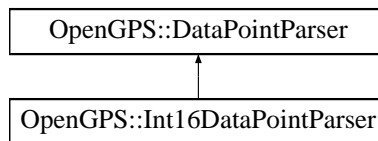
The documentation for this class was generated from the following files:

- [S:/openGPS/ISO5436\\_XML/trunk/src/ISO5436\\_2\\_XML/iso5436\\_2\\_xsd.hxx](#)
- [S:/openGPS/ISO5436\\_XML/trunk/src/ISO5436\\_2\\_XML/iso5436\\_2\\_xsd.cxx](#)

## 9.28 OpenGPS::Int16DataPointParser Class Reference

```
#include <int16_data_point_parser.hxx>
```

Inheritance diagram for OpenGPS::Int16DataPointParser::



### Public Member Functions

- [Int16DataPointParser](#) ()
- virtual OGPS\_Boolean [Read](#) ([PointVectorReaderContext](#) &context, DataPoint &value)
- virtual OGPS\_Boolean [Write](#) ([PointVectorWriterContext](#) &context, const DataPoint &value)
- virtual [~Int16DataPointParser](#) ()

### 9.28.1 Constructor & Destructor Documentation

**9.28.1.1** [Int16DataPointParser::Int16DataPointParser](#) ()

**9.28.1.2** [Int16DataPointParser::~~Int16DataPointParser](#) () `[virtual]`

### 9.28.2 Member Function Documentation

**9.28.2.1** OGPS\_Boolean [Int16DataPointParser::Read](#) ([PointVectorReaderContext](#) &context, DataPoint &value) `[virtual]`

Implements [OpenGPS::DataPointParser](#).

### 9.28.2.2 OGPS\_Boolean Int16DataPointParser::Write ([PointVectorWriter-Context](#) & *context*, const [DataPoint](#) & *value*) [virtual]

Implements [OpenGPS::DataPointParser](#).

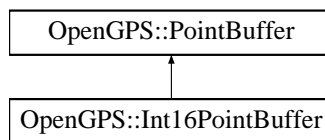
The documentation for this class was generated from the following files:

- [S:/openGPS/ISO5436\\_XML/trunk/src/ISO5436\\_2\\_XML/cxx/int16\\_data\\_point\\_parser.hxx](#)
- [S:/openGPS/ISO5436\\_XML/trunk/src/ISO5436\\_2\\_XML/cxx/int16\\_data\\_point\\_parser.cxx](#)

## 9.29 OpenGPS::Int16PointBuffer Class Reference

```
#include <int16_point_buffer.hxx>
```

Inheritance diagram for OpenGPS::Int16PointBuffer:



### Public Member Functions

- virtual OGPS\_Boolean [Allocate](#) (const unsigned long size)
- virtual OGPS\_Boolean [Get](#) (const unsigned long index, short &value) const
- virtual OGPS\_DataPointType [GetType](#) () const
- [Int16PointBuffer](#) ()
- virtual OGPS\_Boolean [Set](#) (const unsigned long index, const short value)
- virtual OGPS\_Boolean [SetNull](#) (const unsigned long index)
- virtual [~Int16PointBuffer](#) ()

### Private Attributes

- short \* [m\\_Buffer](#)

### 9.29.1 Constructor & Destructor Documentation

#### 9.29.1.1 Int16PointBuffer::Int16PointBuffer ()

#### 9.29.1.2 Int16PointBuffer::~~Int16PointBuffer () [virtual]

### 9.29.2 Member Function Documentation

**9.29.2.1** `OGPS_Boolean Int16PointBuffer::Allocate (const unsigned long size)`  
[virtual]

Implements [OpenGPS::PointBuffer](#).

**9.29.2.2** `OGPS_Boolean Int16PointBuffer::Get (const unsigned long index, short &value) const` [virtual]

Reimplemented from [OpenGPS::PointBuffer](#).

**9.29.2.3** `OGPS_DataPointType Int16PointBuffer::GetType () const`  
[virtual]

Implements [OpenGPS::PointBuffer](#).

**9.29.2.4** `OGPS_Boolean Int16PointBuffer::Set (const unsigned long index, const short value)` [virtual]

Reimplemented from [OpenGPS::PointBuffer](#).

**9.29.2.5** `OGPS_Boolean Int16PointBuffer::SetNull (const unsigned long index)`  
[virtual]

Implements [OpenGPS::PointBuffer](#).

### 9.29.3 Member Data Documentation

**9.29.3.1** `short* OpenGPS::Int16PointBuffer::m_Buffer` [private]

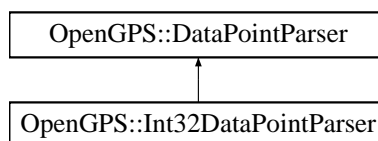
The documentation for this class was generated from the following files:

- [S:/openGPS/ISO5436\\_XML/trunk/src/ISO5436\\_2\\_XML/cxx/int16\\_point\\_buffer.hxx](#)
- [S:/openGPS/ISO5436\\_XML/trunk/src/ISO5436\\_2\\_XML/cxx/int16\\_point\\_buffer.cxx](#)

## 9.30 OpenGPS::Int32DataPointParser Class Reference

```
#include <int32_data_point_parser.hxx>
```

Inheritance diagram for OpenGPS::Int32DataPointParser::



**Public Member Functions**

- [Int32DataPointParser](#) ()
- virtual OGPS\_Boolean [Read](#) ([PointVectorReaderContext](#) &context, [DataPoint](#) &value)
- virtual OGPS\_Boolean [Write](#) ([PointVectorWriterContext](#) &context, const [DataPoint](#) &value)
- virtual [~Int32DataPointParser](#) ()

**9.30.1 Constructor & Destructor Documentation****9.30.1.1 Int32DataPointParser::Int32DataPointParser ()****9.30.1.2 Int32DataPointParser::~~Int32DataPointParser ()** [virtual]**9.30.2 Member Function Documentation****9.30.2.1 OGPS\_Boolean Int32DataPointParser::Read ([PointVectorReaderContext](#) &context, [DataPoint](#) &value)** [virtual]

Implements [OpenGPS::DataPointParser](#).

**9.30.2.2 OGPS\_Boolean Int32DataPointParser::Write ([PointVectorWriterContext](#) &context, const [DataPoint](#) &value)** [virtual]

Implements [OpenGPS::DataPointParser](#).

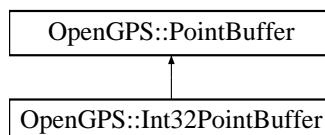
The documentation for this class was generated from the following files:

- S:/openGPS/ISO5436\_XML/trunk/src/ISO5436\_2\_XML/cxx/int32\_data\_point\_parser.hxx
- S:/openGPS/ISO5436\_XML/trunk/src/ISO5436\_2\_XML/cxx/int32\_data\_point\_parser.cxx

**9.31 OpenGPS::Int32PointBuffer Class Reference**

```
#include <int32_point_buffer.hxx>
```

Inheritance diagram for OpenGPS::Int32PointBuffer::





## Public Member Functions

- virtual OGPS\_Boolean [Allocate](#) (const unsigned long size)
- virtual OGPS\_Boolean [Get](#) (const unsigned long index, int &value) const
- virtual OGPS\_DataPointType [GetType](#) () const
- [Int32PointBuffer](#) ()
- virtual OGPS\_Boolean [Set](#) (const unsigned long index, const int value)
- virtual OGPS\_Boolean [SetNull](#) (const unsigned long index)
- virtual [~Int32PointBuffer](#) ()

## Private Attributes

- int \* [m\\_Buffer](#)

### 9.31.1 Constructor & Destructor Documentation

#### 9.31.1.1 Int32PointBuffer::Int32PointBuffer ()

#### 9.31.1.2 Int32PointBuffer::~~Int32PointBuffer () [virtual]

### 9.31.2 Member Function Documentation

#### 9.31.2.1 OGPS\_Boolean Int32PointBuffer::Allocate (const unsigned long size) [virtual]

Implements [OpenGPS::PointBuffer](#).

#### 9.31.2.2 OGPS\_Boolean Int32PointBuffer::Get (const unsigned long index, int & value) const [virtual]

Reimplemented from [OpenGPS::PointBuffer](#).

#### 9.31.2.3 OGPS\_DataPointType Int32PointBuffer::GetType () const [virtual]

Implements [OpenGPS::PointBuffer](#).

#### 9.31.2.4 OGPS\_Boolean Int32PointBuffer::Set (const unsigned long index, const int value) [virtual]

Reimplemented from [OpenGPS::PointBuffer](#).

#### 9.31.2.5 OGPS\_Boolean Int32PointBuffer::SetNull (const unsigned long index) [virtual]

Implements [OpenGPS::PointBuffer](#).

### 9.31.3 Member Data Documentation

#### 9.31.3.1 `int* OpenGPS::Int32PointBuffer::m_Buffer` [private]

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

- `S:/openGPS/ISO5436_XML/trunk/src/ISO5436_2_XML/cxx/int32_point_buffer.hxx`
- `S:/openGPS/ISO5436_XML/trunk/src/ISO5436_2_XML/cxx/int32_point_buffer.cxx`

## 9.32 OpenGPS::ISO5436\_2Container Class Reference

```
#include <iso5436_2_container.hxx>
```

### Public Member Functions

- virtual `OGPS_Boolean Close` ()
- virtual `OGPS_Boolean Create` (const `xsd::Record1Type` &record1, const `xsd::Record2Type` &record2, const `OGPS_Boolean` useBinaryData=TRUE)
- virtual `OGPS_Boolean Create` (const `xsd::Record1Type` &record1, const `xsd::Record2Type` &record2, const `xsd::MatrixDimensionType` &matrix, const `OGPS_Boolean` useBinaryData=TRUE)
- virtual `PointIteratorAutoPtr CreateNextPointIterator` ()
- virtual `PointIteratorAutoPtr CreatePrevPointIterator` ()
- virtual const `ISO5436_2TypeAutoPtr` & `GetDocument` ()
- virtual `OGPS_Boolean GetListCoord` (const unsigned long index, double \*x, double \*y, double \*z)
- virtual `OGPS_Boolean GetListPoint` (const unsigned long index, `PointVector` &vector)
- virtual `OGPS_Boolean GetMatrixCoord` (const unsigned long u, const unsigned long v, const unsigned long w, double \*x, double \*y, double \*z)
- virtual `OGPS_Boolean GetMatrixPoint` (const unsigned long u, const unsigned long v, const unsigned long w, `PointVector` &vector)
- unsigned long `GetMaxU` () const
- unsigned long `GetMaxV` () const
- unsigned long `GetMaxW` () const
- `OGPS_Boolean HasValidPointsLink` () const
- `OGPS_Boolean IsBinary` () const
- `OGPS_Boolean IsMatrix` () const
- virtual `OGPS_Boolean IsMatrixCoordValid` (unsigned long u, unsigned long v, unsigned long w)
- `ISO5436_2Container` (const `String` &file, const `String` &temp)
- virtual `OGPS_Boolean Open` (const `OGPS_Boolean` readOnly=TRUE)
- `ISO5436_2Container` & `operator=` (const `ISO5436_2Container` &src)
- virtual `OGPS_Boolean SetListPoint` (const unsigned long index, const `PointVector` &vector)

- virtual OGPS\_Boolean [SetMatrixPoint](#) (const unsigned long u, const unsigned long v, const unsigned long w, const PointVector \*vector)
- virtual OGPS\_Boolean [Write](#) ()
- virtual [~ISO5436\\_2Container](#) ()

### Protected Member Functions

- void [AddFile](#) (const String &filePath)
- OGPS\_Boolean [Compress](#) ()
- OGPS\_Boolean [CreateDocument](#) (const [xsd::Record1Type](#) \*record1, const [xsd::Record2Type](#) \*record2, const [xsd::MatrixDimensionType](#) \*matrix, const OGPS\_Boolean useBinaryData)
- OGPS\_Boolean [CreatePointBuffer](#) ()
- virtual OGPS\_Boolean [CreatePointVectorParser](#) ([PointVectorParserBuilder](#) &builder) const
- virtual [PointVectorParserBuilder](#) \* [CreatePointVectorParserBuilder](#) () const
- virtual [PointVectorReaderContext](#) \* [CreatePointVectorReaderContext](#) ()
- virtual [PointVectorWriterContext](#) \* [CreatePointVectorWriterContext](#) (zipFile handle) const
- virtual OGPS\_Boolean [CreateVectorBuffer](#) ([VectorBufferBuilder](#) &builder) const
- virtual [VectorBufferBuilder](#) \* [CreateVectorBufferBuilder](#) () const
- OGPS\_Boolean [Decompress](#) (const String &src, const String &dst) const
- OGPS\_Boolean [Decompress](#) ()
- OGPS\_Boolean [DecompressDataBin](#) ()
- OGPS\_Boolean [DecompressMain](#) () const
- OGPS\_DataPointType [GetAxisDataType](#) (const [xsd::AxisDescriptionType](#) &axis, const OGPS\_Boolean incremental) const
- String [GetChecksumArchiveName](#) () const
- String [GetChecksumFileName](#) () const
- String [GetContainerTempFilePath](#) () const
- const String & [GetFilePath](#) () const
- String [GetFullFilePath](#) () const
- String [GetMainArchiveName](#) () const
- String [GetMainFileName](#) () const
- unsigned long [GetPointCount](#) () const
- String [GetPointDataArchiveName](#) () const
- String [GetPointDataFileName](#) ()
- const String & [GetTempDir](#) () const
- String [GetValidPointsArchiveName](#) () const
- String [GetValidPointsFileName](#) ()
- OGPS\_DataPointType [GetXaxisDataType](#) () const
- OGPS\_DataPointType [GetYaxisDataType](#) () const
- OGPS\_DataPointType [GetZaxisDataType](#) () const
- OGPS\_Boolean [HasDocument](#) () const
- OGPS\_Boolean [HasVectorBuffer](#) () const
- OGPS\_Boolean [ReadDocument](#) ()

- OGPS\_Boolean [ReadXmlDocument](#) ()
- void [Reset](#) ()
- void [ResetXmlPointList](#) ()
- OGPS\_Boolean [SavePointBuffer](#) (zipFile handle)
- OGPS\_Boolean [SaveValidPointsLink](#) (zipFile handle)
- OGPS\_Boolean [SaveXmlDocument](#) (zipFile handle)

#### Private Member Functions

- OGPS\_Boolean [CreateTempDir](#) ()
- [VectorBuffer](#) \*const [GetVectorBuffer](#) () const
- OGPS\_Boolean [HasTempDir](#) () const
- OGPS\_Boolean [RemoveTempDir](#) ()

#### Private Attributes

- [ISO5436\\_2TypeAutoPtr](#) m\_Document
- String m\_FilePath
- OGPS\_Boolean m\_IsReadOnly
- String m\_PointDataFileName
- [PointVectorAutoPtr](#) m\_PointVector
- [PointVectorProxyContext](#) m\_ProxyContext
- String m\_TempBasePath
- String m\_TempPath
- String m\_ValidPointsFileName
- [VectorBufferBuilderAutoPtr](#) m\_VectorBufferBuilder

### 9.32.1 Constructor & Destructor Documentation

**9.32.1.1** `ISO5436_2Container::ISO5436_2Container (const String & file, const String & temp)`

**9.32.1.2** `ISO5436_2Container::~~ISO5436_2Container ()` [virtual]

### 9.32.2 Member Function Documentation

**9.32.2.1** `void ISO5436_2Container::AddFile (const String & filePath)`  
[protected]

**9.32.2.2** `OGPS_Boolean ISO5436_2Container::Close ()` [virtual]

**9.32.2.3** `OGPS_Boolean ISO5436_2Container::Compress ()` [protected]

**9.32.2.4** `OGPS_Boolean ISO5436_2Container::Create (const xsd::Record1Type & record1, const xsd::Record2Type & record2, const OGPS_Boolean useBinaryData = TRUE)` [virtual]

**9.32.2.5** `OGPS_Boolean ISO5436_2Container::Create (const xsd::Record1Type & record1, const xsd::Record2Type & record2, const xsd::MatrixDimensionType & matrix, const OGPS_Boolean useBinaryData = TRUE)` [virtual]

**9.32.2.6** `OGPS_Boolean ISO5436_2Container::CreateDocument (const xsd::Record1Type * record1, const xsd::Record2Type * record2, const xsd::MatrixDimensionType * matrix, const OGPS_Boolean useBinaryData)` [protected]

**9.32.2.7** `PointIteratorAutoPtr ISO5436_2Container::CreateNextPointIterator ()` [virtual]

**9.32.2.8** `OGPS_Boolean ISO5436_2Container::CreatePointBuffer ()` [protected]

**9.32.2.9** `OGPS_Boolean ISO5436_2Container::CreatePointVectorParser (PointVectorParserBuilder & builder) const` [protected, virtual]

**9.32.2.10** `PointVectorParserBuilder * ISO5436_2Container::CreatePointVectorParserBuilder () const` [protected, virtual]

**9.32.2.11** `PointVectorReaderContext * ISO5436_2Container::CreatePointVectorReaderContext ()` [protected, virtual]

**9.32.2.12** `PointVectorWriterContext * ISO5436_2Container::CreatePointVectorWriterContext (zipFile handle) const` [protected, virtual]

**9.32.2.13** `PointIteratorAutoPtr ISO5436_2Container::CreatePrevPointIterator ()` [virtual]

**9.32.2.14** `OGPS_Boolean ISO5436_2Container::CreateTempDir ()` [private]

**9.32.2.15** `OGPS_Boolean ISO5436_2Container::CreateVectorBuffer (VectorBufferBuilder & builder) const` [protected, virtual]

**9.32.2.16** `VectorBufferBuilder * ISO5436_2Container::CreateVectorBufferBuilder () const` [protected, virtual]

**9.32.2.17** `OGPS_Boolean OpenGPS::ISO5436_2Container::Decompress (const String & src, const String & dst) const` `[protected]`

**9.32.2.18** `OGPS_Boolean ISO5436_2Container::Decompress ()` `[protected]`

**9.32.2.19** `OGPS_Boolean ISO5436_2Container::DecompressDataBin ()` `[protected]`

**9.32.2.20** `OGPS_Boolean ISO5436_2Container::DecompressMain () const` `[protected]`

**9.32.2.21** `OGPS_DataPointType ISO5436_2Container::GetAxisDataType (const xsd::AxisDescriptionType & axis, const OGPS_Boolean incremental) const` `[protected]`

**9.32.2.22** `String ISO5436_2Container::GetChecksumArchiveName () const` `[protected]`

**9.32.2.23** `String ISO5436_2Container::GetChecksumFileName () const` `[protected]`

**9.32.2.24** `String ISO5436_2Container::GetContainerTempFilePath () const` `[protected]`

**9.32.2.25** `const ISO5436\_2TypeAutoPtr & ISO5436_2Container::GetDocument ()` `[virtual]`

**9.32.2.26** `const String & ISO5436_2Container::GetFilePath () const` `[protected]`

**9.32.2.27** `String ISO5436_2Container::GetFullFilePath () const` `[protected]`

**9.32.2.28** `OGPS_Boolean ISO5436_2Container::GetListCoord (const unsigned long index, double * x, double * y, double * z)` `[virtual]`

**9.32.2.29** `OGPS_Boolean ISO5436_2Container::GetListPoint (const unsigned long index, PointVector & vector)` `[virtual]`

**9.32.2.30** String ISO5436\_2Container::GetMainArchiveName () const  
[protected]

**9.32.2.31** String ISO5436\_2Container::GetMainFileName () const  
[protected]

**9.32.2.32** OGPS\_Boolean ISO5436\_2Container::GetMatrixCoord (const unsigned long *u*, const unsigned long *v*, const unsigned long *w*, double \* *x*, double \* *y*, double \* *z*) [virtual]

**9.32.2.33** OGPS\_Boolean ISO5436\_2Container::GetMatrixPoint (const unsigned long *u*, const unsigned long *v*, const unsigned long *w*, PointVector & *vector*) [virtual]

**9.32.2.34** unsigned long ISO5436\_2Container::GetMaxU () const

**9.32.2.35** unsigned long ISO5436\_2Container::GetMaxV () const

**9.32.2.36** unsigned long ISO5436\_2Container::GetMaxW () const

**9.32.2.37** unsigned long ISO5436\_2Container::GetPointCount () const  
[protected]

**9.32.2.38** String ISO5436\_2Container::GetPointDataArchiveName () const  
[protected]

**9.32.2.39** String ISO5436\_2Container::GetPointDataFileName ()  
[protected]

**9.32.2.40** const String & ISO5436\_2Container::GetTempDir () const  
[protected]

**9.32.2.41** String ISO5436\_2Container::GetValidPointsArchiveName () const  
[protected]

**9.32.2.42** String ISO5436\_2Container::GetValidPointsFileName ()  
[protected]

**9.32.2.43** VectorBuffer \*const ISO5436\_2Container::GetVectorBuffer () const  
[private]

**9.32.2.44** `OGPS_DataPointType ISO5436_2Container::GetXaxisDataType () const` [protected]

**9.32.2.45** `OGPS_DataPointType ISO5436_2Container::GetYaxisDataType () const` [protected]

**9.32.2.46** `OGPS_DataPointType ISO5436_2Container::GetZaxisDataType () const` [protected]

**9.32.2.47** `OGPS_Boolean ISO5436_2Container::HasDocument () const` [protected]

**9.32.2.48** `OGPS_Boolean ISO5436_2Container::HasTempDir () const` [private]

**9.32.2.49** `OGPS_Boolean ISO5436_2Container::HasValidPointsLink () const`

**9.32.2.50** `OGPS_Boolean ISO5436_2Container::HasVectorBuffer () const` [protected]

**9.32.2.51** `OGPS_Boolean ISO5436_2Container::IsBinary () const`

**9.32.2.52** `OGPS_Boolean ISO5436_2Container::IsMatrix () const`

**9.32.2.53** `OGPS_Boolean ISO5436_2Container::IsMatrixCoordValid (unsigned long u, unsigned long v, unsigned long w)` [virtual]

**9.32.2.54** `OGPS_Boolean ISO5436_2Container::Open (const OGPS_Boolean readOnly = TRUE)` [virtual]

**9.32.2.55** `ISO5436_2Container & ISO5436_2Container::operator= (const ISO5436_2Container & src)`

**9.32.2.56** `OGPS_Boolean ISO5436_2Container::ReadDocument ()` [protected]

**9.32.2.57** `OGPS_Boolean ISO5436_2Container::ReadXmlDocument ()` [protected]



**9.32.2.58** `OGPS_Boolean ISO5436_2Container::RemoveTempDir ()` [private]

**9.32.2.59** `void ISO5436_2Container::Reset ()` [protected]

**9.32.2.60** `void ISO5436_2Container::ResetXmlPointList ()` [protected]

**9.32.2.61** `OGPS_Boolean ISO5436_2Container::SavePointBuffer (zipFile handle)` [protected]

**9.32.2.62** `OGPS_Boolean ISO5436_2Container::SaveValidPointsLink (zipFile handle)` [protected]

**9.32.2.63** `OGPS_Boolean ISO5436_2Container::SaveXmlDocument (zipFile handle)` [protected]

**9.32.2.64** `OGPS_Boolean ISO5436_2Container::SetListPoint (const unsigned long index, const PointVector & vector)` [virtual]

**9.32.2.65** `OGPS_Boolean ISO5436_2Container::SetMatrixPoint (const unsigned long u, const unsigned long v, const unsigned long w, const PointVector * vector)` [virtual]

**9.32.2.66** `OGPS_Boolean ISO5436_2Container::Write ()` [virtual]

### 9.32.3 Member Data Documentation

**9.32.3.1** `ISO5436_2TypeAutoPtr OpenGPS::ISO5436_2Container::m_Document` [private]

**9.32.3.2** `String OpenGPS::ISO5436_2Container::m_FilePath` [private]

**9.32.3.3** `OGPS_Boolean OpenGPS::ISO5436_2Container::m_IsReadOnly` [private]

**9.32.3.4** `String OpenGPS::ISO5436_2Container::m_PointDataFileName` [private]

**9.32.3.5** `PointVectorAutoPtr OpenGPS::ISO5436_2Container::m_PointVector` [private]

**9.32.3.6 PointVectorProxyContext** [OpenGPS::ISO5436\\_2Container::m\\_ProxyContext](#) [private]

**9.32.3.7 String** [OpenGPS::ISO5436\\_2Container::m\\_TempBasePath](#) [private]

**9.32.3.8 String** [OpenGPS::ISO5436\\_2Container::m\\_TempPath](#) [private]

**9.32.3.9 String** [OpenGPS::ISO5436\\_2Container::m\\_ValidPointsFileName](#) [private]

**9.32.3.10 VectorBufferBuilderAutoPtr** [OpenGPS::ISO5436\\_2Container::m\\_VectorBufferBuilder](#) [private]

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

- [S:/openGPS/ISO5436\\_XML/trunk/src/ISO5436\\_2\\_XML/cxx/iso5436\\_2\\_container.hxx](#)
- [S:/openGPS/ISO5436\\_XML/trunk/src/ISO5436\\_2\\_XML/cxx/iso5436\\_2\\_container.cxx](#)

## 9.33 xsd::ISO5436\_2Type Class Reference

```
#include <iso5436_2_xsd.hxx>
```

### 9.33.1 Detailed Description

Class corresponding to the ISO5436\_2Type schema type.

This is the top tag of a data file

#### Record1

Accessor and modifier functions for the Record1 required element.

- `typedef ::xsd::cxx::tree::traits< Record1\_type, wchar_t > Record1\_traits`  
*Element traits type.*
- `typedef ::xsd::Record1Type Record1\_type`  
*Element type.*
- `void Record1 (::std::auto_ptr< Record1\_type > p)`  
*Set the element value without copying.*

- void [Record1](#) (const [Record1\\_type](#) &x)  
*Set the element value.*
- [Record1\\_type](#) & [Record1](#) ()  
*Return a read-write reference to the element.*
- const [Record1\\_type](#) & [Record1](#) () const  
*Return a read-only (constant) reference to the element.*

## Record2

Accessor and modifier functions for the Record2 optional element.

- typedef ::xsd::cxx::tree::optional< [Record2\\_type](#) > [Record2\\_optional](#)  
*Element optional container type.*
- typedef ::xsd::cxx::tree::traits< [Record2\\_type](#), wchar\_t > [Record2\\_traits](#)  
*Element traits type.*
- typedef ::xsd::Record2Type [Record2\\_type](#)  
*Element type.*
- void [Record2](#) (::std::auto\_ptr< [Record2\\_type](#) > p)  
*Set the element value without copying.*
- void [Record2](#) (const [Record2\\_optional](#) &x)  
*Set the element value.*
- void [Record2](#) (const [Record2\\_type](#) &x)  
*Set the element value.*
- [Record2\\_optional](#) & [Record2](#) ()  
*Return a read-write reference to the element container.*
- const [Record2\\_optional](#) & [Record2](#) () const  
*Return a read-only (constant) reference to the element container.*

## Record3

Accessor and modifier functions for the Record3 required element.

- typedef ::xsd::cxx::tree::traits< [Record3\\_type](#), wchar\_t > [Record3\\_traits](#)  
*Element traits type.*

- typedef [xsd::Record3Type](#) [Record3\\_type](#)  
*Element type.*
- void [Record3](#) (::std::auto\_ptr< [Record3\\_type](#) > p)  
*Set the element value without copying.*
- void [Record3](#) (const [Record3\\_type](#) &x)  
*Set the element value.*
- [Record3\\_type](#) & [Record3](#) ()  
*Return a read-write reference to the element.*
- const [Record3\\_type](#) & [Record3](#) () const  
*Return a read-only (constant) reference to the element.*

## Record4

Accessor and modifier functions for the Record4 required element.

- typedef [xsd::cxx::tree::traits](#)< [Record4\\_type](#), [wchar\\_t](#) > [Record4\\_traits](#)  
*Element traits type.*
- typedef [xsd::Record4Type](#) [Record4\\_type](#)  
*Element type.*
- void [Record4](#) (::std::auto\_ptr< [Record4\\_type](#) > p)  
*Set the element value without copying.*
- void [Record4](#) (const [Record4\\_type](#) &x)  
*Set the element value.*
- [Record4\\_type](#) & [Record4](#) ()  
*Return a read-write reference to the element.*
- const [Record4\\_type](#) & [Record4](#) () const  
*Return a read-only (constant) reference to the element.*

## Constructors

- virtual [ISO5436\\_2Type](#) \* [\\_clone](#) (::xml\_schema::flags f=0,::xml\_schema::type \*c=0) const  
*Copy the object polymorphically.*

- [ISO5436\\_2Type](#) (const [ISO5436\\_2Type](#) &x,::xml\_schema::flags f=0,::xml\_schema::type \*c=0)  
*Copy constructor.*
- [ISO5436\\_2Type](#) (const ::xercesc::DOMElement &e,::xml\_schema::flags f=0,::xml\_schema::type \*c=0)  
*Construct an instance from a DOM element.*
- [ISO5436\\_2Type](#) (const [Record1\\_type](#) &, const [Record3\\_type](#) &, const [Record4\\_type](#) &)  
*Construct an instance from the ultimate base and initializers for required elements and attributes.*

### Protected Member Functions

- void [parse](#) (::xsd::cxx::xml::dom::parser< wchar\_t > &,::xml\_schema::flags)

### Private Attributes

- ::xsd::cxx::tree::one< [Record1\\_type](#) > [Record1\\_](#)
- [Record2\\_optional](#) [Record2\\_](#)
- ::xsd::cxx::tree::one< [Record3\\_type](#) > [Record3\\_](#)
- ::xsd::cxx::tree::one< [Record4\\_type](#) > [Record4\\_](#)

## 9.33.2 Member Typedef Documentation

**9.33.2.1** typedef ::xsd::cxx::tree::traits< [Record1\\_type](#), wchar\_t > [xsd::ISO5436\\_2Type::Record1\\_traits](#)

Element traits type.

**9.33.2.2** typedef ::xsd::Record1Type [xsd::ISO5436\\_2Type::Record1\\_type](#)

Element type.

**9.33.2.3** typedef ::xsd::cxx::tree::optional< [Record2\\_type](#) > [xsd::ISO5436\\_2Type::Record2\\_optional](#)

Element optional container type.

**9.33.2.4** typedef ::xsd::cxx::tree::traits< [Record2\\_type](#), wchar\_t > [xsd::ISO5436\\_2Type::Record2\\_traits](#)

Element traits type.

**9.33.2.5 typedef ::xsd::Record2Type xsd::ISO5436\_2Type::Record2\_type**

Element type.

**9.33.2.6 typedef ::xsd::cxx::tree::traits< Record3\_type, wchar\_t > xsd::ISO5436\_2Type::Record3\_traits**

Element traits type.

**9.33.2.7 typedef ::xsd::Record3Type xsd::ISO5436\_2Type::Record3\_type**

Element type.

**9.33.2.8 typedef ::xsd::cxx::tree::traits< Record4\_type, wchar\_t > xsd::ISO5436\_2Type::Record4\_traits**

Element traits type.

**9.33.2.9 typedef ::xsd::Record4Type xsd::ISO5436\_2Type::Record4\_type**

Element type.

**9.33.3 Constructor & Destructor Documentation****9.33.3.1 xsd::ISO5436\_2Type::ISO5436\_2Type (const Record1\_type &, const Record3\_type &, const Record4\_type &)**

Construct an instance from the ultimate base and initializers for required elements and attributes.

**9.33.3.2 xsd::ISO5436\_2Type::ISO5436\_2Type (const ::xercesc::DOMElement & *e*, ::xml\_schema::flags *f* = 0, ::xml\_schema::type \* *c* = 0)**

Construct an instance from a DOM element.

**Parameters:**

- e* A DOM element to extract the data from.
- f* Flags to construct the new instance with.
- c* A pointer to the object that will contain the new instance.

**9.33.3.3 xsd::ISO5436\_2Type::ISO5436\_2Type (const ISO5436\_2Type & *x*, ::xml\_schema::flags *f* = 0, ::xml\_schema::type \* *c* = 0)**

Copy constructor.

**Parameters:**

- x* An instance to make a copy of.

*f* Flags to construct the copy with.

*c* A pointer to the object that will contain the copy.

For polymorphic object models use the `_clone` function instead.

### 9.33.4 Member Function Documentation

**9.33.4.1** `ISO5436_2Type * xsd::ISO5436_2Type::_clone (::xml_schema::flags f = 0, ::xml_schema::type * c = 0) const` [virtual]

Copy the object polymorphically.

#### Parameters:

*f* Flags to construct the copy with.

*c* A pointer to the object that will contain the copy.

#### Returns:

A pointer to the dynamically allocated copy.

This function ensures that the dynamic type of an instance is used for copying and should be used for polymorphic object models instead of the copy constructor.

**9.33.4.2** `void xsd::ISO5436_2Type::parse (::xsd::cxx::xml::dom::parser< wchar_t > &, ::xml_schema::flags)` [protected]

**9.33.4.3** `void xsd::ISO5436_2Type::Record1 (::std::auto_ptr< Record1_type > p)`

Set the element value without copying.

#### Parameters:

*p* A new value to use.

This function will try to use the passed value directly instead of making a copy.

**9.33.4.4** `void xsd::ISO5436_2Type::Record1 (const Record1_type & x)`

Set the element value.

#### Parameters:

*x* A new value to set.

This function makes a copy of its argument and sets it as the new value of the element.

#### 9.33.4.5 [ISO5436\\_2Type::Record1\\_type](#) & xsd::ISO5436\_2Type::Record1 ()

Return a read-write reference to the element.

##### Returns:

A reference to the element.

#### 9.33.4.6 `const ISO5436\_2Type::Record1\_type & xsd::ISO5436_2Type::Record1 () const`

Return a read-only (constant) reference to the element.

##### Returns:

A constant reference to the element.

#### 9.33.4.7 `void xsd::ISO5436_2Type::Record2 (::std::auto_ptr< Record2\_type > p)`

Set the element value without copying.

##### Parameters:

*p* A new value to use.

This function will try to use the passed value directly instead of making a copy.

#### 9.33.4.8 `void xsd::ISO5436_2Type::Record2 (const Record2\_optional & x)`

Set the element value.

##### Parameters:

*x* An optional container with the new value to set.

If the value is present in *x* then this function makes a copy of this value and sets it as the new value of the element. Otherwise the element container is set the 'not present' state.

#### 9.33.4.9 `void xsd::ISO5436_2Type::Record2 (const Record2\_type & x)`

Set the element value.

##### Parameters:

*x* A new value to set.

This function makes a copy of its argument and sets it as the new value of the element.



#### 9.33.4.10 [ISO5436\\_2Type::Record2\\_optional](#) & xsd::ISO5436\_2Type::Record2 ()

Return a read-write reference to the element container.

##### Returns:

A reference to the optional container.

#### 9.33.4.11 const [ISO5436\\_2Type::Record2\\_optional](#) & xsd::ISO5436\_2Type::Record2 () const

Return a read-only (constant) reference to the element container.

##### Returns:

A constant reference to the optional container.

#### 9.33.4.12 void xsd::ISO5436\_2Type::Record3 (::std::auto\_ptr< [Record3\\_type](#) > *p*)

Set the element value without copying.

##### Parameters:

*p* A new value to use.

This function will try to use the passed value directly instead of making a copy.

#### 9.33.4.13 void xsd::ISO5436\_2Type::Record3 (const [Record3\\_type](#) & *x*)

Set the element value.

##### Parameters:

*x* A new value to set.

This function makes a copy of its argument and sets it as the new value of the element.

#### 9.33.4.14 [ISO5436\\_2Type::Record3\\_type](#) & xsd::ISO5436\_2Type::Record3 ()

Return a read-write reference to the element.

##### Returns:

A reference to the element.

**9.33.4.15** `const ISO5436_2Type::Record3_type & xsd::ISO5436_2Type::Record3 () const`

Return a read-only (constant) reference to the element.

**Returns:**

A constant reference to the element.

**9.33.4.16** `void xsd::ISO5436_2Type::Record4 (::std::auto_ptr< Record4_type > p)`

Set the element value without copying.

**Parameters:**

*p* A new value to use.

This function will try to use the passed value directly instead of making a copy.

**9.33.4.17** `void xsd::ISO5436_2Type::Record4 (const Record4_type & x)`

Set the element value.

**Parameters:**

*x* A new value to set.

This function makes a copy of its argument and sets it as the new value of the element.

**9.33.4.18** `ISO5436_2Type::Record4_type & xsd::ISO5436_2Type::Record4 ()`

Return a read-write reference to the element.

**Returns:**

A reference to the element.

**9.33.4.19** `const ISO5436_2Type::Record4_type & xsd::ISO5436_2Type::Record4 () const`

Return a read-only (constant) reference to the element.

**Returns:**

A constant reference to the element.

### 9.33.5 Member Data Documentation

**9.33.5.1** `::xsd::cxx::tree::one< Record1\_type > xsd::ISO5436\_2Type::Record1\_ [private]`

**9.33.5.2** `Record2\_optional xsd::ISO5436\_2Type::Record2\_ [private]`

**9.33.5.3** `::xsd::cxx::tree::one< Record3\_type > xsd::ISO5436\_2Type::Record3\_ [private]`

**9.33.5.4** `::xsd::cxx::tree::one< Record4\_type > xsd::ISO5436\_2Type::Record4\_ [private]`

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

- [S:/openGPS/ISO5436\\_XML/trunk/src/ISO5436\\_2\\_XML/iso5436\\_2\\_xsd.hxx](#)
- [S:/openGPS/ISO5436\\_XML/trunk/src/ISO5436\\_2\\_XML/iso5436\\_2\\_xsd.cxx](#)

## 9.34 xsd::MatrixDimensionType Class Reference

```
#include <iso5436_2_xsd.hxx>
```

### 9.34.1 Detailed Description

Class corresponding to the MatrixDimensionType schema type.

Defines the size of the 3 dimensions of the data matrix.

#### SizeX

Accessor and modifier functions for the SizeX required element.

Define the size of the first dimension of the data matrix

- `typedef ::xsd::cxx::tree::traits< SizeX\_type, wchar_t > SizeX\_traits`  
*Element traits type.*
- `typedef ::xml_schema::unsigned_long SizeX\_type`  
*Element type.*
- `void SizeX (const SizeX\_type &x)`  
*Set the element value.*
- `SizeX\_type & SizeX ()`  
*Return a read-write reference to the element.*

- const [SizeX\\_type](#) & [SizeX](#) () const  
*Return a read-only (constant) reference to the element.*

### SizeY

Accessor and modifier functions for the SizeY required element.

Define the size of the second dimension of the data matrix

- typedef ::xsd::cxx::tree::traits< [SizeY\\_type](#), wchar\_t > [SizeY\\_traits](#)  
*Element traits type.*
- typedef ::xml\_schema::unsigned\_long [SizeY\\_type](#)  
*Element type.*
- void [SizeY](#) (const [SizeY\\_type](#) &x)  
*Set the element value.*
- [SizeY\\_type](#) & [SizeY](#) ()  
*Return a read-write reference to the element.*
- const [SizeY\\_type](#) & [SizeY](#) () const  
*Return a read-only (constant) reference to the element.*

### SizeZ

Accessor and modifier functions for the SizeZ required element.

Define the size of the third dimension of the data matrix

- typedef ::xsd::cxx::tree::traits< [SizeZ\\_type](#), wchar\_t > [SizeZ\\_traits](#)  
*Element traits type.*
- typedef ::xml\_schema::unsigned\_long [SizeZ\\_type](#)  
*Element type.*
- void [SizeZ](#) (const [SizeZ\\_type](#) &x)  
*Set the element value.*
- [SizeZ\\_type](#) & [SizeZ](#) ()  
*Return a read-write reference to the element.*
- const [SizeZ\\_type](#) & [SizeZ](#) () const  
*Return a read-only (constant) reference to the element.*

### Constructors

- virtual [MatrixDimensionType](#) \* [\\_clone](#) (::xml\_schema::flags f=0,::xml\_schema::type \*c=0) const  
*Copy the object polymorphically.*
- [MatrixDimensionType](#) (const [MatrixDimensionType](#) &x,::xml\_schema::flags f=0,::xml\_schema::type \*c=0)  
*Copy constructor.*
- [MatrixDimensionType](#) (const ::xercesc::DOMElement &e,::xml\_schema::flags f=0,::xml\_schema::type \*c=0)  
*Construct an instance from a DOM element.*
- [MatrixDimensionType](#) (const [SizeX\\_type](#) &, const [SizeY\\_type](#) &, const [SizeZ\\_type](#) &)  
*Construct an instance from the ultimate base and initializers for required elements and attributes.*

### Protected Member Functions

- void [parse](#) (::xsd::cxx::xml::dom::parser< wchar\_t > &,::xml\_schema::flags)

### Private Attributes

- ::xsd::cxx::tree::one< [SizeX\\_type](#) > [SizeX\\_](#)
- ::xsd::cxx::tree::one< [SizeY\\_type](#) > [SizeY\\_](#)
- ::xsd::cxx::tree::one< [SizeZ\\_type](#) > [SizeZ\\_](#)

#### 9.34.2 Member Typedef Documentation

##### 9.34.2.1 typedef ::xsd::cxx::tree::traits< [SizeX\\_type](#), wchar\_t > [xsd::MatrixDimensionType::SizeX\\_traits](#)

Element traits type.

##### 9.34.2.2 typedef ::xml\_schema::unsigned\_long [xsd::MatrixDimensionType::SizeX\\_type](#)

Element type.

##### 9.34.2.3 typedef ::xsd::cxx::tree::traits< [SizeY\\_type](#), wchar\_t > [xsd::MatrixDimensionType::SizeY\\_traits](#)

Element traits type.

**9.34.2.4** typedef [::xml\\_schema::unsigned\\_long](#) [xsd::MatrixDimensionType::SizeY\\_type](#)

Element type.

**9.34.2.5** typedef [::xsd::cxx::tree::traits< \[SizeZ\\\_type\]\(#\), wchar\\_t >](#) [xsd::MatrixDimensionType::SizeZ\\_traits](#)

Element traits type.

**9.34.2.6** typedef [::xml\\_schema::unsigned\\_long](#) [xsd::MatrixDimensionType::SizeZ\\_type](#)

Element type.

### 9.34.3 Constructor & Destructor Documentation

**9.34.3.1** [xsd::MatrixDimensionType::MatrixDimensionType](#) (const [SizeX\\_type](#) &, const [SizeY\\_type](#) &, const [SizeZ\\_type](#) &)

Construct an instance from the ultimate base and initializers for required elements and attributes.

**9.34.3.2** [xsd::MatrixDimensionType::MatrixDimensionType](#) (const [::xercesc::DOMElement](#) & *e*, [::xml\\_schema::flags](#) *f* = 0, [::xml\\_schema::type](#) \* *c* = 0)

Construct an instance from a DOM element.

#### Parameters:

- e* A DOM element to extract the data from.
- f* Flags to construct the new instance with.
- c* A pointer to the object that will contain the new instance.

**9.34.3.3** [xsd::MatrixDimensionType::MatrixDimensionType](#) (const [MatrixDimensionType](#) & *x*, [::xml\\_schema::flags](#) *f* = 0, [::xml\\_schema::type](#) \* *c* = 0)

Copy constructor.

#### Parameters:

- x* An instance to make a copy of.
- f* Flags to construct the copy with.
- c* A pointer to the object that will contain the copy.

For polymorphic object models use the `_clone` function instead.

### 9.34.4 Member Function Documentation

**9.34.4.1** `MatrixDimensionType * xsd::MatrixDimensionType::_clone (::xml_schema::flags f = 0, ::xml_schema::type * c = 0) const` [virtual]

Copy the object polymorphically.

**Parameters:**

*f* Flags to construct the copy with.

*c* A pointer to the object that will contain the copy.

**Returns:**

A pointer to the dynamically allocated copy.

This function ensures that the dynamic type of an instance is used for copying and should be used for polymorphic object models instead of the copy constructor.

**9.34.4.2** `void xsd::MatrixDimensionType::parse (::xsd::cxx::xml::dom::parser< wchar_t > &, ::xml_schema::flags) [protected]`

**9.34.4.3** `void xsd::MatrixDimensionType::SizeX (const SizeX_type & x)`

Set the element value.

**Parameters:**

*x* A new value to set.

This function makes a copy of its argument and sets it as the new value of the element.

**9.34.4.4** `SizeX_type& xsd::MatrixDimensionType::SizeX ()`

Return a read-write reference to the element.

**Returns:**

A reference to the element.

**9.34.4.5** `const SizeX_type& xsd::MatrixDimensionType::SizeX () const`

Return a read-only (constant) reference to the element.

**Returns:**

A constant reference to the element.

**9.34.4.6 void xsd::MatrixDimensionType::SizeY (const SizeY\_type & x)**

Set the element value.

**Parameters:**

*x* A new value to set.

This function makes a copy of its argument and sets it as the new value of the element.

**9.34.4.7 SizeY\_type& xsd::MatrixDimensionType::SizeY ()**

Return a read-write reference to the element.

**Returns:**

A reference to the element.

**9.34.4.8 const SizeY\_type& xsd::MatrixDimensionType::SizeY () const**

Return a read-only (constant) reference to the element.

**Returns:**

A constant reference to the element.

**9.34.4.9 void xsd::MatrixDimensionType::SizeZ (const SizeZ\_type & x)**

Set the element value.

**Parameters:**

*x* A new value to set.

This function makes a copy of its argument and sets it as the new value of the element.

**9.34.4.10 SizeZ\_type& xsd::MatrixDimensionType::SizeZ ()**

Return a read-write reference to the element.

**Returns:**

A reference to the element.

**9.34.4.11 const SizeZ\_type& xsd::MatrixDimensionType::SizeZ () const**

Return a read-only (constant) reference to the element.

**Returns:**

A constant reference to the element.



### 9.34.5 Member Data Documentation

**9.34.5.1** `::xsd::cxx::tree::one< SizeX\_type > xsd::MatrixDimensionType::SizeX_` [private]

**9.34.5.2** `::xsd::cxx::tree::one< SizeY\_type > xsd::MatrixDimensionType::SizeY_` [private]

**9.34.5.3** `::xsd::cxx::tree::one< SizeZ\_type > xsd::MatrixDimensionType::SizeZ_` [private]

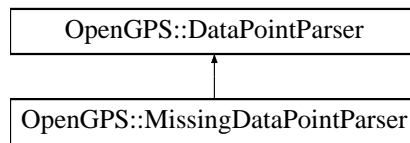
The documentation for this class was generated from the following files:

- `S:/openGPS/ISO5436_XML/trunk/src/ISO5436_2_XML/iso5436\_2\_xsd.hxx`
- `S:/openGPS/ISO5436_XML/trunk/src/ISO5436_2_XML/iso5436\_2\_xsd.cxx`

## 9.35 OpenGPS::MissingDataPointParser Class Reference

```
#include <missing_data_point_parser.hxx>
```

Inheritance diagram for OpenGPS::MissingDataPointParser::



### Public Member Functions

- [MissingDataPointParser](#) ()
- virtual OGPS\_Boolean [Read](#) ([PointVectorReaderContext](#) &context, DataPoint &value)
- virtual OGPS\_Boolean [Write](#) ([PointVectorWriterContext](#) &context, const DataPoint &value)
- virtual [~MissingDataPointParser](#) ()

### 9.35.1 Constructor & Destructor Documentation

**9.35.1.1** `MissingDataPointParser::MissingDataPointParser ()`

**9.35.1.2** `MissingDataPointParser::~~MissingDataPointParser ()` [virtual]

### 9.35.2 Member Function Documentation

#### 9.35.2.1 OGPS\_Boolean MissingDataPointParser::Read ([PointVectorReader-Context](#) & *context*, [DataPoint](#) & *value*) [virtual]

Implements [OpenGPS::DataPointParser](#).

#### 9.35.2.2 OGPS\_Boolean MissingDataPointParser::Write ([PointVectorWriter-Context](#) & *context*, const [DataPoint](#) & *value*) [virtual]

Implements [OpenGPS::DataPointParser](#).

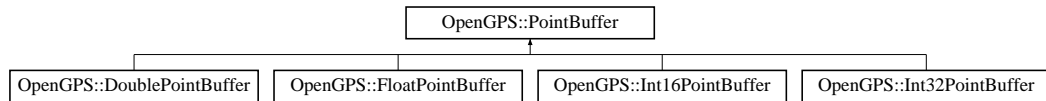
The documentation for this class was generated from the following files:

- S:/openGPS/ISO5436\_XML/trunk/src/ISO5436\_2\_XML/cxx/[missing\\_data\\_point\\_parser.hxx](#)
- S:/openGPS/ISO5436\_XML/trunk/src/ISO5436\_2\_XML/cxx/[missing\\_data\\_point\\_parser.cxx](#)

## 9.36 OpenGPS::PointBuffer Class Reference

```
#include <point_buffer.hxx>
```

Inheritance diagram for OpenGPS::PointBuffer::



### Public Member Functions

- virtual OGPS\_Boolean [Allocate](#) (const unsigned long size)=0
- virtual OGPS\_Boolean [Get](#) (const unsigned long index, double &value) const
- virtual OGPS\_Boolean [Get](#) (const unsigned long index, float &value) const
- virtual OGPS\_Boolean [Get](#) (const unsigned long index, int &value) const
- virtual OGPS\_Boolean [Get](#) (const unsigned long index, short &value) const
- virtual unsigned long [GetSize](#) () const
- virtual OGPS\_DataPointType [GetType](#) () const =0
- virtual OGPS\_Boolean [Set](#) (const unsigned long index, const double value)
- virtual OGPS\_Boolean [Set](#) (const unsigned long index, const float value)
- virtual OGPS\_Boolean [Set](#) (const unsigned long index, const int value)
- virtual OGPS\_Boolean [Set](#) (const unsigned long index, const short value)
- virtual OGPS\_Boolean [SetNull](#) (const unsigned long index)=0
- virtual [~PointBuffer](#) ()

### Protected Member Functions

- OpenGPS::UnsignedBytePtr [Allocate](#) (const unsigned long size, const size\_t typeSize)
- void [Free](#) (OpenGPS::UnsignedBytePtr \*value)
- [PointBuffer](#) ()

### Private Attributes

- unsigned long [m\\_Size](#)

#### 9.36.1 Constructor & Destructor Documentation

**9.36.1.1** [PointBuffer::PointBuffer](#) () [protected]

**9.36.1.2** [PointBuffer::~~PointBuffer](#) () [virtual]

#### 9.36.2 Member Function Documentation

**9.36.2.1** OpenGPS::UnsignedBytePtr [PointBuffer::Allocate](#) (const unsigned long size, const size\_t typeSize) [protected]

**9.36.2.2** virtual OGPS\_Boolean [OpenGPS::PointBuffer::Allocate](#) (const unsigned long size) [pure virtual]

Implemented in [OpenGPS::DoublePointBuffer](#), [OpenGPS::FloatPointBuffer](#), [OpenGPS::Int16PointBuffer](#), and [OpenGPS::Int32PointBuffer](#).

**9.36.2.3** void [PointBuffer::Free](#) (OpenGPS::UnsignedBytePtr \* value) [protected]

**9.36.2.4** OGPS\_Boolean [PointBuffer::Get](#) (const unsigned long index, double & value) const [virtual]

Reimplemented in [OpenGPS::DoublePointBuffer](#).

**9.36.2.5** OGPS\_Boolean [PointBuffer::Get](#) (const unsigned long index, float & value) const [virtual]

Reimplemented in [OpenGPS::FloatPointBuffer](#).

**9.36.2.6** OGPS\_Boolean [PointBuffer::Get](#) (const unsigned long index, int & value) const [virtual]

Reimplemented in [OpenGPS::Int32PointBuffer](#).

**9.36.2.7 OGPS\_Boolean PointBuffer::Get (const unsigned long *index*, short & *value*) const** [virtual]

Reimplemented in [OpenGPS::Int16PointBuffer](#).

**9.36.2.8 unsigned long PointBuffer::GetSize () const** [virtual]

**9.36.2.9 virtual OGPS\_DataPointType OpenGPS::PointBuffer::GetType () const** [pure virtual]

Implemented in [OpenGPS::DoublePointBuffer](#), [OpenGPS::FloatPointBuffer](#), [OpenGPS::Int16PointBuffer](#), and [OpenGPS::Int32PointBuffer](#).

**9.36.2.10 OGPS\_Boolean PointBuffer::Set (const unsigned long *index*, const double *value*)** [virtual]

Reimplemented in [OpenGPS::DoublePointBuffer](#).

**9.36.2.11 OGPS\_Boolean PointBuffer::Set (const unsigned long *index*, const float *value*)** [virtual]

Reimplemented in [OpenGPS::FloatPointBuffer](#).

**9.36.2.12 OGPS\_Boolean PointBuffer::Set (const unsigned long *index*, const int *value*)** [virtual]

Reimplemented in [OpenGPS::Int32PointBuffer](#).

**9.36.2.13 OGPS\_Boolean PointBuffer::Set (const unsigned long *index*, const short *value*)** [virtual]

Reimplemented in [OpenGPS::Int16PointBuffer](#).

**9.36.2.14 virtual OGPS\_Boolean OpenGPS::PointBuffer::SetNull (const unsigned long *index*)** [pure virtual]

Implemented in [OpenGPS::DoublePointBuffer](#), [OpenGPS::FloatPointBuffer](#), [OpenGPS::Int16PointBuffer](#), and [OpenGPS::Int32PointBuffer](#).

### 9.36.3 Member Data Documentation

**9.36.3.1 unsigned long [OpenGPS::PointBuffer::m\\_Size](#)** [private]

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

- [S:/openGPS/ISO5436\\_XML/trunk/src/ISO5436\\_2\\_XML/cxx/point\\_buffer.hxx](#)
- [S:/openGPS/ISO5436\\_XML/trunk/src/ISO5436\\_2\\_XML/cxx/point\\_buffer.cxx](#)

## 9.37 OpenGPS::PointIteratorImpl Class Reference

```
#include <point_iterator_impl.hxx>
```

### Public Member Functions

- virtual OGPS\_Boolean [CreateNext](#) ()
- virtual OGPS\_Boolean [GetCurrent](#) (PointVector &vector)
- virtual OGPS\_Boolean [GetPosition](#) (unsigned long \*const u, unsigned long \*const v, unsigned long \*const w) const
- virtual OGPS\_Boolean [GetPosition](#) (unsigned long \*const index) const
- virtual OGPS\_Boolean [HasNext](#) () const
- virtual OGPS\_Boolean [HasPrev](#) () const
- virtual OGPS\_Boolean [MoveNext](#) ()
- virtual OGPS\_Boolean [MovePrev](#) ()
- [PointIteratorImpl](#) & [operator=](#) (const [PointIteratorImpl](#) &src)
- [PointIteratorImpl](#) ([ISO5436\\_2Container](#) \*const handle, const OGPS\_Boolean isForward, const OGPS\_Boolean isMatrix)
- virtual void [ResetNext](#) ()
- virtual void [ResetPrev](#) ()
- virtual OGPS\_Boolean [SetCurrent](#) (const PointVector \*vector)
- virtual [~PointIteratorImpl](#) ()

### Private Attributes

- PointVector \* [m\\_Buffer](#)
- [ISO5436\\_2Container](#) \*const [m\\_Handle](#)
- OGPS\_Boolean [m\\_IsForward](#)
- OGPS\_Boolean [m\\_IsMatrix](#)
- OGPS\_Boolean [m\\_IsReset](#)
- unsigned long [m\\_U](#)
- unsigned long [m\\_V](#)
- unsigned long [m\\_W](#)

### 9.37.1 Constructor & Destructor Documentation

**9.37.1.1** [PointIteratorImpl::PointIteratorImpl](#) ([ISO5436\\_2Container](#) \*const handle, const OGPS\_Boolean isForward, const OGPS\_Boolean isMatrix)

**9.37.1.2** [PointIteratorImpl::~~PointIteratorImpl](#) () [virtual]

### 9.37.2 Member Function Documentation

**9.37.2.1** OGPS\_Boolean [PointIteratorImpl::CreateNext](#) () [virtual]

**9.37.2.2** `OGPS_Boolean PointIteratorImpl::GetCurrent (PointVector & vector)`  
[virtual]

**9.37.2.3** `OGPS_Boolean PointIteratorImpl::GetPosition (unsigned long *const u, unsigned long *const v, unsigned long *const w) const` [virtual]

**9.37.2.4** `OGPS_Boolean PointIteratorImpl::GetPosition (unsigned long *const index) const` [virtual]

**9.37.2.5** `OGPS_Boolean PointIteratorImpl::HasNext () const` [virtual]

**9.37.2.6** `OGPS_Boolean PointIteratorImpl::HasPrev () const` [virtual]

**9.37.2.7** `OGPS_Boolean PointIteratorImpl::MoveNext ()` [virtual]

**9.37.2.8** `OGPS_Boolean PointIteratorImpl::MovePrev ()` [virtual]

**9.37.2.9** `PointIteratorImpl & PointIteratorImpl::operator= (const PointIteratorImpl & src)`

**9.37.2.10** `void PointIteratorImpl::ResetNext ()` [virtual]

**9.37.2.11** `void PointIteratorImpl::ResetPrev ()` [virtual]

**9.37.2.12** `OGPS_Boolean PointIteratorImpl::SetCurrent (const PointVector * vector)` [virtual]

### 9.37.3 Member Data Documentation

**9.37.3.1** `PointVector* OpenGPS::PointIteratorImpl::m_Buffer` [private]

**9.37.3.2** `ISO5436_2Container* const OpenGPS::PointIteratorImpl::m_Handle`  
[private]

**9.37.3.3** `OGPS_Boolean OpenGPS::PointIteratorImpl::m_IsForward`  
[private]

**9.37.3.4** `OGPS_Boolean OpenGPS::PointIteratorImpl::m_IsMatrix`  
[private]

9.37.3.5 OGPS\_Boolean [OpenGPS::PointIteratorImpl::m\\_IsReset](#)  
[private]

9.37.3.6 unsigned long [OpenGPS::PointIteratorImpl::m\\_U](#) [private]

9.37.3.7 unsigned long [OpenGPS::PointIteratorImpl::m\\_V](#) [private]

9.37.3.8 unsigned long [OpenGPS::PointIteratorImpl::m\\_W](#) [private]

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

- [S:/openGPS/ISO5436\\_XML/trunk/src/ISO5436\\_2\\_XML/cxx/point\\_iterator\\_impl.hxx](#)
- [S:/openGPS/ISO5436\\_XML/trunk/src/ISO5436\\_2\\_XML/cxx/point\\_iterator\\_impl.cxx](#)

## 9.38 OpenGPS::PointVectorInputBinaryFileStream Class Reference

```
#include <point_vector_iostream.hxx>
```

### Public Member Functions

- [PointVectorInputBinaryFileStream](#) (const String &filePath)
- [~PointVectorInputBinaryFileStream](#) ()

### Private Types

- typedef std::basic\_ifstream< OpenGPS::UnsignedByte > [BaseType](#)

### 9.38.1 Member Typedef Documentation

9.38.1.1 typedef std::basic\_ifstream<OpenGPS::UnsignedByte> [OpenGPS::PointVectorInputBinaryFileStream::BaseType](#) [private]

### 9.38.2 Constructor & Destructor Documentation

9.38.2.1 [PointVectorInputBinaryFileStream::PointVectorInputBinaryFileStream](#) (const String &filePath)

### 9.38.2.2 PointVectorInputBinaryFileStream::~~PointVectorInputBinaryFileStream ()

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

- S:/openGPS/ISO5436\_XML/trunk/src/ISO5436\_2\_XML/cxx/[point\\_vector\\_istream.hxx](#)
- S:/openGPS/ISO5436\_XML/trunk/src/ISO5436\_2\_XML/cxx/[point\\_vector\\_istream.cxx](#)

## 9.39 OpenGPS::PointVectorInputStringStream Class Reference

```
#include <point_vector_istream.hxx>
```

### Public Member Functions

- [PointVectorInputStringStream](#) (const OpenGPS::String &s)
- [PointVectorInputStringStream](#) ()
- [~PointVectorInputStringStream](#) ()

### Private Types

- typedef std::basic\_istream< OGPS\_Character > [BaseType](#)

### 9.39.1 Member Typedef Documentation

**9.39.1.1** typedef std::basic\_istream<OGPS\_Character> [OpenGPS::PointVectorInputStringStream::BaseType](#) [private]

### 9.39.2 Constructor & Destructor Documentation

#### 9.39.2.1 PointVectorInputStringStream::PointVectorInputStringStream ()

#### 9.39.2.2 PointVectorInputStringStream::PointVectorInputStringStream (const OpenGPS::String &s)

#### 9.39.2.3 PointVectorInputStringStream::~~PointVectorInputStringStream ()

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

- S:/openGPS/ISO5436\_XML/trunk/src/ISO5436\_2\_XML/cxx/[point\\_vector\\_istream.hxx](#)
- S:/openGPS/ISO5436\_XML/trunk/src/ISO5436\_2\_XML/cxx/[point\\_vector\\_istream.cxx](#)



## 9.40 OpenGPS::PointVectorInvariantLocale Class Reference

```
#include <point_vector_iostream.hxx>
```

### Public Member Functions

- [PointVectorInvariantLocale \(\)](#)
- [~PointVectorInvariantLocale \(\)](#)

### Static Public Member Functions

- static const [PointVectorInvariantLocale](#) & [GetInstance \(\)](#)

### Private Types

- typedef std::locale [BaseType](#)

### Static Private Attributes

- static [PointVectorInvariantLocale](#) [m\\_Instance](#)

#### 9.40.1 Member Typedef Documentation

**9.40.1.1** typedef std::locale [OpenGPS::PointVectorInvariantLocale::BaseType](#)  
[private]

#### 9.40.2 Constructor & Destructor Documentation

**9.40.2.1** [PointVectorInvariantLocale::PointVectorInvariantLocale \(\)](#)

**9.40.2.2** [PointVectorInvariantLocale::~~PointVectorInvariantLocale \(\)](#)

#### 9.40.3 Member Function Documentation

**9.40.3.1** const [PointVectorInvariantLocale](#) & [PointVectorInvariantLocale::GetInstance \(\)](#) [static]

#### 9.40.4 Member Data Documentation

**9.40.4.1** [PointVectorInvariantLocale](#) [PointVectorInvariantLocale::m\\_Instance](#)  
[static, private]

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

- [S:/openGPS/ISO5436\\_XML/trunk/src/ISO5436\\_2\\_XML/cxx/point\\_vector\\_ostream.hxx](#)
- [S:/openGPS/ISO5436\\_XML/trunk/src/ISO5436\\_2\\_XML/cxx/point\\_vector\\_ostream.cxx](#)

## 9.41 OpenGPS::PointVectorOutputBinaryFileStream Class Reference

```
#include <point_vector_ostream.hxx>
```

### Public Member Functions

- [PointVectorOutputBinaryFileStream](#) (const String &filePath)
- [~PointVectorOutputBinaryFileStream](#) ()

### Private Types

- typedef std::basic\_ofstream< OpenGPS::UnsignedByte > [BaseType](#)

#### 9.41.1 Member Typedef Documentation

**9.41.1.1** typedef std::basic\_ofstream<OpenGPS::UnsignedByte> [OpenGPS::PointVectorOutputBinaryFileStream::BaseType](#) [private]

#### 9.41.2 Constructor & Destructor Documentation

**9.41.2.1** PointVectorOutputBinaryFileStream::PointVectorOutputBinaryFileStream (const String & *filePath*)

**9.41.2.2** PointVectorOutputBinaryFileStream::~~PointVectorOutputBinaryFileStream ()

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

- [S:/openGPS/ISO5436\\_XML/trunk/src/ISO5436\\_2\\_XML/cxx/point\\_vector\\_ostream.hxx](#)
- [S:/openGPS/ISO5436\\_XML/trunk/src/ISO5436\\_2\\_XML/cxx/point\\_vector\\_ostream.cxx](#)

## 9.42 OpenGPS::PointVectorOutputStringStream Class Reference

```
#include <point_vector_ostream.hxx>
```

**Public Member Functions**

- [PointVectorOutputStream \(\)](#)
- [~PointVectorOutputStream \(\)](#)

**Private Types**

- typedef std::basic\_ostringstream< OGPS\_Character > [BaseType](#)

**9.42.1 Member Typedef Documentation**

**9.42.1.1** typedef std::basic\_ostringstream<OGPS\_Character> [OpenGPS::PointVectorOutputStream::BaseType](#) [private]

**9.42.2 Constructor & Destructor Documentation**

**9.42.2.1** [PointVectorOutputStream::PointVectorOutputStream \(\)](#)

**9.42.2.2** [PointVectorOutputStream::~~PointVectorOutputStream \(\)](#)

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

- S:/openGPS/ISO5436\_XML/trunk/src/ISO5436\_2\_XML/cxx/[point\\_vector\\_ostream.hxx](#)
- S:/openGPS/ISO5436\_XML/trunk/src/ISO5436\_2\_XML/cxx/[point\\_vector\\_ostream.cxx](#)

**9.43 OpenGPS::PointVectorParser Class Reference**

```
#include <point_vector_parser.hxx>
```

**Public Member Functions**

- virtual [DataPointParser](#) \* [CreateDataPointParser](#) (const OGPS\_DataPointType dataType) const
- [PointVectorParser \(\)](#)
- virtual OGPS\_Boolean [Read](#) ([PointVectorReaderContext](#) &context, [PointVectorBase](#) &value)
- void [SetX](#) ([DataPointParser](#) \*value)
- void [SetY](#) ([DataPointParser](#) \*value)
- void [SetZ](#) ([DataPointParser](#) \*value)
- virtual OGPS\_Boolean [Write](#) ([PointVectorWriterContext](#) &context, const [PointVectorBase](#) &value)
- virtual [~PointVectorParser \(\)](#)

### Private Attributes

- [DataPointParser](#) \* [m\\_X](#)
- [DataPointParser](#) \* [m\\_Y](#)
- [DataPointParser](#) \* [m\\_Z](#)

### 9.43.1 Constructor & Destructor Documentation

#### 9.43.1.1 [PointVectorParser::PointVectorParser \(\)](#)

#### 9.43.1.2 [PointVectorParser::~~PointVectorParser \(\)](#) [virtual]

### 9.43.2 Member Function Documentation

#### 9.43.2.1 [DataPointParser](#) \* [PointVectorParser::CreateDataPointParser](#) (const [OGPS\\_DataPointType](#) *dataType*) const [virtual]

#### 9.43.2.2 [OGPS\\_Boolean](#) [PointVectorParser::Read](#) ([PointVectorReaderContext](#) & *context*, [PointVectorBase](#) & *value*) [virtual]

#### 9.43.2.3 void [PointVectorParser::SetX](#) ([DataPointParser](#) \* *value*)

#### 9.43.2.4 void [PointVectorParser::SetY](#) ([DataPointParser](#) \* *value*)

#### 9.43.2.5 void [PointVectorParser::SetZ](#) ([DataPointParser](#) \* *value*)

#### 9.43.2.6 [OGPS\\_Boolean](#) [PointVectorParser::Write](#) ([PointVectorWriterContext](#) & *context*, const [PointVectorBase](#) & *value*) [virtual]

### 9.43.3 Member Data Documentation

#### 9.43.3.1 [DataPointParser](#)\* [OpenGPS::PointVectorParser::m\\_X](#) [private]

#### 9.43.3.2 [DataPointParser](#)\* [OpenGPS::PointVectorParser::m\\_Y](#) [private]

#### 9.43.3.3 [DataPointParser](#)\* [OpenGPS::PointVectorParser::m\\_Z](#) [private]

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

- [S:/openGPS/ISO5436\\_XML/trunk/src/ISO5436\\_2\\_XML/cxx/point\\_vector\\_parser.hxx](#)
- [S:/openGPS/ISO5436\\_XML/trunk/src/ISO5436\\_2\\_XML/cxx/point\\_vector\\_parser.cxx](#)

## 9.44 OpenGPS::PointVectorParserBuilder Class Reference

```
#include <point_vector_parser_builder.hxx>
```

### Public Member Functions

- virtual OGPS\_Boolean [BuildParser](#) ()
- virtual OGPS\_Boolean [BuildX](#) (const OGPS\_DataPointType dataType)
- virtual OGPS\_Boolean [BuildY](#) (const OGPS\_DataPointType dataType)
- virtual OGPS\_Boolean [BuildZ](#) (const OGPS\_DataPointType dataType)
- virtual [PointVectorParser](#) \* [GetParser](#) ()
- [PointVectorParserBuilder](#) ()
- virtual [~PointVectorParserBuilder](#) ()

### Private Attributes

- [PointVectorParser](#) \* [m\\_Parser](#)

### 9.44.1 Constructor & Destructor Documentation

#### 9.44.1.1 PointVectorParserBuilder::PointVectorParserBuilder ()

**9.44.1.2 PointVectorParserBuilder::~~PointVectorParserBuilder** ()  
[virtual]

### 9.44.2 Member Function Documentation

**9.44.2.1 OGPS\_Boolean PointVectorParserBuilder::BuildParser** ()  
[virtual]

**9.44.2.2 OGPS\_Boolean PointVectorParserBuilder::BuildX (const OGPS\_DataPointType dataType)** [virtual]

**9.44.2.3 OGPS\_Boolean PointVectorParserBuilder::BuildY (const OGPS\_DataPointType dataType)** [virtual]

**9.44.2.4 OGPS\_Boolean PointVectorParserBuilder::BuildZ (const OGPS\_DataPointType dataType)** [virtual]

**9.44.2.5 PointVectorParser** \* **PointVectorParserBuilder::GetParser** ()  
[virtual]

### 9.44.3 Member Data Documentation

**9.44.3.1 PointVectorParser\*** **OpenGPS::PointVectorParserBuilder::m\_Parser**  
[private]

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

- S:/openGPS/ISO5436\_XML/trunk/src/ISO5436\_2\_XML/cxx/[point\\_vector\\_parser\\_builder.hxx](#)
- S:/openGPS/ISO5436\_XML/trunk/src/ISO5436\_2\_XML/cxx/[point\\_vector\\_parser\\_builder.cxx](#)

## 9.45 OpenGPS::PointVectorProxy Class Reference

```
#include <point_vector_proxy.hxx>
```

### Public Member Functions

- virtual OGPS\_Boolean [Get](#) (PointVectorBase &value) const
- virtual DataPoint \*const [GetX](#) ()
- virtual const DataPoint \* [GetX](#) () const
- virtual DataPoint \*const [GetY](#) ()
- virtual const DataPoint \* [GetY](#) () const
- virtual DataPoint \*const [GetZ](#) ()
- virtual const DataPoint \* [GetZ](#) () const
- [PointVectorProxy](#) (const [PointVectorProxyContext](#) \*context, const [VectorBuffer](#) \*buffer)
- virtual OGPS\_Boolean [Set](#) (const PointVectorBase &value)
- virtual OGPS\_Boolean [SetNull](#) ()
- virtual [~PointVectorProxy](#) ()

### Private Attributes

- const [VectorBuffer](#) \* [m\\_Buffer](#)
- const [PointVectorProxyContext](#) \* [m\\_Context](#)
- [DataPointProxyContext](#) \* [m\\_U](#)
- [DataPointProxyContext](#) \* [m\\_V](#)
- [DataPointProxyContext](#) \* [m\\_W](#)
- DataPoint \* [m\\_X](#)
- DataPoint \* [m\\_Y](#)
- DataPoint \* [m\\_Z](#)

## Classes

- class [DataPointProxy](#)
- class [DataPointProxyContext](#)
- class [UDataPointProxyContext](#)
- class [VDataPointProxyContext](#)
- class [WDataPointProxyContext](#)

### 9.45.1 Constructor & Destructor Documentation

**9.45.1.1** [PointVectorProxy::PointVectorProxy](#) (const [PointVectorProxyContext](#) \* *context*, const [VectorBuffer](#) \* *buffer*)

**9.45.1.2** [PointVectorProxy::~~PointVectorProxy](#) () [virtual]

### 9.45.2 Member Function Documentation

**9.45.2.1** [OGPS\\_Boolean](#) [PointVectorProxy::Get](#) ([PointVectorBase](#) & *value*) const [virtual]

**9.45.2.2** [DataPoint](#) \*const [PointVectorProxy::GetX](#) () [virtual]

**9.45.2.3** const [DataPoint](#) \* [PointVectorProxy::GetX](#) () const [virtual]

**9.45.2.4** [DataPoint](#) \*const [PointVectorProxy::GetY](#) () [virtual]

**9.45.2.5** const [DataPoint](#) \* [PointVectorProxy::GetY](#) () const [virtual]

**9.45.2.6** [DataPoint](#) \*const [PointVectorProxy::GetZ](#) () [virtual]

**9.45.2.7** const [DataPoint](#) \* [PointVectorProxy::GetZ](#) () const [virtual]

**9.45.2.8** [OGPS\\_Boolean](#) [PointVectorProxy::Set](#) (const [PointVectorBase](#) & *value*) [virtual]

**9.45.2.9** [OGPS\\_Boolean](#) [PointVectorProxy::SetNull](#) () [virtual]

### 9.45.3 Member Data Documentation

**9.45.3.1** const [VectorBuffer](#)\* [OpenGPS::PointVectorProxy::m\\_Buffer](#) [private]

**9.45.3.2** `const PointVectorProxyContext* OpenGPS::PointVectorProxy::m_Context` [private]

**9.45.3.3** `DataPointProxyContext* OpenGPS::PointVectorProxy::m_U` [private]

**9.45.3.4** `DataPointProxyContext* OpenGPS::PointVectorProxy::m_V` [private]

**9.45.3.5** `DataPointProxyContext* OpenGPS::PointVectorProxy::m_W` [private]

**9.45.3.6** `DataPoint* OpenGPS::PointVectorProxy::m_X` [private]

**9.45.3.7** `DataPoint* OpenGPS::PointVectorProxy::m_Y` [private]

**9.45.3.8** `DataPoint* OpenGPS::PointVectorProxy::m_Z` [private]

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

- `S:/openGPS/ISO5436_XML/trunk/src/ISO5436_2_XML/cxx/point_vector_proxy.hxx`
- `S:/openGPS/ISO5436_XML/trunk/src/ISO5436_2_XML/cxx/point_vector_proxy.cxx`

## 9.46 OpenGPS::PointVectorProxy::DataPointProxy Class Reference

### Public Member Functions

- `DataPointProxy` (const `DataPointProxyContext` \*context)
- virtual double `Get` () const
- virtual OGPS\_Boolean `Get` (double \*const value) const
- virtual OGPS\_Boolean `Get` (float \*const value) const
- virtual OGPS\_Boolean `Get` (int \*const value) const
- virtual OGPS\_Boolean `Get` (short \*const value) const
- virtual OGPS\_DataPointType `GetType` () const
- virtual OGPS\_Boolean `IsValid` () const
- virtual OGPS\_Boolean `Set` (const `DataPoint` &src)
- virtual OGPS\_Boolean `Set` (const double value)
- virtual OGPS\_Boolean `Set` (const float value)
- virtual OGPS\_Boolean `Set` (const int value)
- virtual OGPS\_Boolean `Set` (const short value)
- virtual OGPS\_Boolean `SetNull` ()
- virtual `~DataPointProxy` ()



### Protected Member Functions

- virtual void [Reset](#) ()

### Private Attributes

- const [DataPointProxyContext](#) \* [m\\_Context](#)

### 9.46.1 Constructor & Destructor Documentation

**9.46.1.1** [PointVectorProxy::DataPointProxy::DataPointProxy](#) (const [DataPointProxyContext](#) \* *context*)

**9.46.1.2** [PointVectorProxy::DataPointProxy::~~DataPointProxy](#) ()  
[virtual]

### 9.46.2 Member Function Documentation

**9.46.2.1** [double PointVectorProxy::DataPointProxy::Get](#) () const [virtual]

**9.46.2.2** [OGPS\\_Boolean PointVectorProxy::DataPointProxy::Get](#) (double \*const *value*) const [virtual]

**9.46.2.3** [OGPS\\_Boolean PointVectorProxy::DataPointProxy::Get](#) (float \*const *value*) const [virtual]

**9.46.2.4** [OGPS\\_Boolean PointVectorProxy::DataPointProxy::Get](#) (int \*const *value*) const [virtual]

**9.46.2.5** [OGPS\\_Boolean PointVectorProxy::DataPointProxy::Get](#) (short \*const *value*) const [virtual]

**9.46.2.6** [OGPS\\_DataPointType PointVectorProxy::DataPointProxy::GetType](#) () const [virtual]

**9.46.2.7** [OGPS\\_Boolean PointVectorProxy::DataPointProxy::IsValid](#) () const [virtual]

**9.46.2.8** [void PointVectorProxy::DataPointProxy::Reset](#) () [protected, virtual]

## 9.47 OpenGPS::PointVectorProxy::DataPointProxyContext Class Reference 69

**9.46.2.9 OGPS\_Boolean PointVectorProxy::DataPointProxy::Set (const DataPoint & src) [virtual]**

**9.46.2.10 OGPS\_Boolean PointVectorProxy::DataPointProxy::Set (const double value) [virtual]**

**9.46.2.11 OGPS\_Boolean PointVectorProxy::DataPointProxy::Set (const float value) [virtual]**

**9.46.2.12 OGPS\_Boolean PointVectorProxy::DataPointProxy::Set (const int value) [virtual]**

**9.46.2.13 OGPS\_Boolean PointVectorProxy::DataPointProxy::Set (const short value) [virtual]**

**9.46.2.14 OGPS\_Boolean PointVectorProxy::DataPointProxy::SetNull () [virtual]**

### 9.46.3 Member Data Documentation

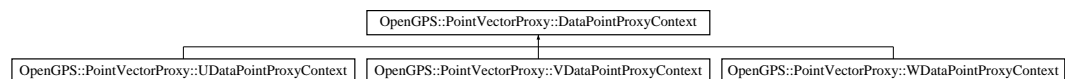
**9.46.3.1 const DataPointProxyContext\* OpenGPS::PointVectorProxy::DataPointProxy::m\_Context [private]**

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

- S:/openGPS/ISO5436\_XML/trunk/src/ISO5436\_2\_XML/cxx/[point\\_vector\\_proxy.hxx](#)
- S:/openGPS/ISO5436\_XML/trunk/src/ISO5436\_2\_XML/cxx/[point\\_vector\\_proxy.cxx](#)

## 9.47 OpenGPS::PointVectorProxy::DataPointProxyContext Class Reference

Inheritance diagram for OpenGPS::PointVectorProxy::DataPointProxyContext::



### Public Member Functions

- virtual [PointBuffer](#) \*const [GetBuffer](#) () const =0
- virtual unsigned long [GetIndex](#) () const =0

## 9.47 OpenGPS::PointVectorProxy::DataPointProxyContext Class Reference 70

- virtual OGPS\_Boolean [IsValid](#) () const
- virtual [~DataPointProxyContext](#) ()

### Protected Member Functions

- [DataPointProxyContext](#) ([PointVectorProxy](#) \*vector)

### Protected Attributes

- [PointVectorProxy](#) \* [m\\_Vector](#)

### 9.47.1 Constructor & Destructor Documentation

**9.47.1.1** [PointVectorProxy::DataPointProxyContext::~~DataPointProxyContext](#) () [virtual]

**9.47.1.2** [PointVectorProxy::DataPointProxyContext::DataPointProxyContext](#) ([PointVectorProxy](#) \* *vector*) [protected]

### 9.47.2 Member Function Documentation

**9.47.2.1** virtual [PointBuffer](#)\* const [OpenGPS::PointVectorProxy::DataPointProxyContext::GetBuffer](#) () const [pure virtual]

Implemented in [OpenGPS::PointVectorProxy::UDataPointProxyContext](#), [OpenGPS::PointVectorProxy::VDataPointProxyContext](#), and [OpenGPS::PointVectorProxy::WDataPointProxyContext](#).

**9.47.2.2** virtual unsigned long [OpenGPS::PointVectorProxy::DataPointProxyContext::GetIndex](#) () const [pure virtual]

Implemented in [OpenGPS::PointVectorProxy::UDataPointProxyContext](#), [OpenGPS::PointVectorProxy::VDataPointProxyContext](#), and [OpenGPS::PointVectorProxy::WDataPointProxyContext](#).

**9.47.2.3** OGPS\_Boolean [PointVectorProxy::DataPointProxyContext::IsValid](#) () const [virtual]

### 9.47.3 Member Data Documentation

**9.47.3.1** [PointVectorProxy](#)\* [OpenGPS::PointVectorProxy::DataPointProxyContext::m\\_Vector](#) [protected]

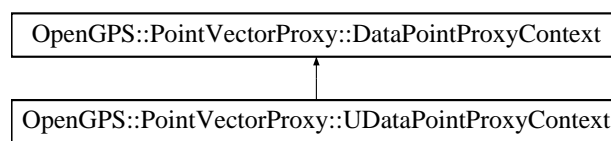
The documentation for this class was generated from the following files:

## 9.48 OpenGPS::PointVectorProxy::UDataPointProxyContext Class Reference 171

- [S:/openGPS/ISO5436\\_XML/trunk/src/ISO5436\\_2\\_XML/cxx/point\\_vector\\_proxy.hxx](#)
- [S:/openGPS/ISO5436\\_XML/trunk/src/ISO5436\\_2\\_XML/cxx/point\\_vector\\_proxy.cxx](#)

### 9.48 OpenGPS::PointVectorProxy::UDataPointProxyContext Class Reference

Inheritance diagram for OpenGPS::PointVectorProxy::UDataPointProxyContext::



#### Public Member Functions

- virtual [PointBuffer](#) \*const [GetBuffer](#) () const
- virtual unsigned long [GetIndex](#) () const
- [UDataPointProxyContext](#) ([PointVectorProxy](#) \*vector)
- virtual [~UDataPointProxyContext](#) ()

#### 9.48.1 Constructor & Destructor Documentation

**9.48.1.1** [PointVectorProxy::UDataPointProxyContext::UDataPointProxyContext](#) ([PointVectorProxy](#) \* vector)

**9.48.1.2** [PointVectorProxy::UDataPointProxyContext::~~UDataPointProxyContext](#) () [virtual]

#### 9.48.2 Member Function Documentation

**9.48.2.1** [PointBuffer](#) \*const [PointVectorProxy::UDataPointProxyContext::GetBuffer](#) () const [virtual]

Implements [OpenGPS::PointVectorProxy::DataPointProxyContext](#).

**9.48.2.2** unsigned long [PointVectorProxy::UDataPointProxyContext::GetIndex](#) () const [virtual]

Implements [OpenGPS::PointVectorProxy::DataPointProxyContext](#).

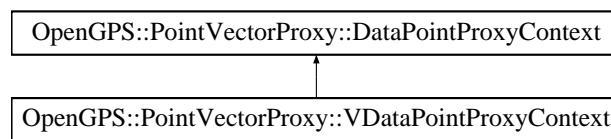
The documentation for this class was generated from the following files:

## 9.49 OpenGPS::PointVectorProxy::VDataPointProxyContext Class Reference 172

- [S:/openGPS/ISO5436\\_XML/trunk/src/ISO5436\\_2\\_XML/cxx/point\\_vector\\_proxy.hxx](#)
- [S:/openGPS/ISO5436\\_XML/trunk/src/ISO5436\\_2\\_XML/cxx/point\\_vector\\_proxy.cxx](#)

### 9.49 OpenGPS::PointVectorProxy::VDataPointProxyContext Class Reference

Inheritance diagram for OpenGPS::PointVectorProxy::VDataPointProxyContext::



#### Public Member Functions

- virtual [PointBuffer](#) \*const [GetBuffer](#) () const
- virtual unsigned long [GetIndex](#) () const
- [VDataPointProxyContext](#) ([PointVectorProxy](#) \*vector)
- virtual [~VDataPointProxyContext](#) ()

#### 9.49.1 Constructor & Destructor Documentation

**9.49.1.1** [PointVectorProxy::VDataPointProxyContext::VDataPointProxyContext](#) ([PointVectorProxy](#) \* vector)

**9.49.1.2** [PointVectorProxy::VDataPointProxyContext::~~VDataPointProxyContext](#) () [virtual]

#### 9.49.2 Member Function Documentation

**9.49.2.1** [PointBuffer](#) \*const [PointVectorProxy::VDataPointProxyContext::GetBuffer](#) () const [virtual]

Implements [OpenGPS::PointVectorProxy::DataPointProxyContext](#).

**9.49.2.2** unsigned long [PointVectorProxy::VDataPointProxyContext::GetIndex](#) () const [virtual]

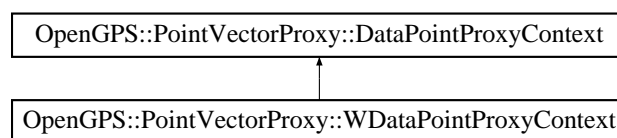
Implements [OpenGPS::PointVectorProxy::DataPointProxyContext](#).

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

- [S:/openGPS/ISO5436\\_XML/trunk/src/ISO5436\\_2\\_XML/cxx/point\\_vector\\_proxy.hxx](#)
- [S:/openGPS/ISO5436\\_XML/trunk/src/ISO5436\\_2\\_XML/cxx/point\\_vector\\_proxy.cxx](#)

## 9.50 OpenGPS::PointVectorProxy::WDataPointProxyContext Class Reference

Inheritance diagram for OpenGPS::PointVectorProxy::WDataPointProxyContext::



### Public Member Functions

- virtual [PointBuffer](#) \*const [GetBuffer](#) () const
- virtual unsigned long [GetIndex](#) () const
- [WDataPointProxyContext](#) ([PointVectorProxy](#) \*vector)
- virtual [~WDataPointProxyContext](#) ()

### 9.50.1 Constructor & Destructor Documentation

**9.50.1.1** [PointVectorProxy::WDataPointProxyContext::WDataPointProxyContext](#) ([PointVectorProxy](#) \* vector)

**9.50.1.2** [PointVectorProxy::WDataPointProxyContext::~~WDataPointProxyContext](#) () [virtual]

### 9.50.2 Member Function Documentation

**9.50.2.1** [PointBuffer](#) \*const [PointVectorProxy::WDataPointProxyContext::GetBuffer](#) () const [virtual]

Implements [OpenGPS::PointVectorProxy::DataPointProxyContext](#).

**9.50.2.2** unsigned long [PointVectorProxy::WDataPointProxyContext::GetIndex](#) () const [virtual]

Implements [OpenGPS::PointVectorProxy::DataPointProxyContext](#).

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

- [S:/openGPS/ISO5436\\_XML/trunk/src/ISO5436\\_2\\_XML/cxx/point\\_vector\\_proxy.hxx](#)
- [S:/openGPS/ISO5436\\_XML/trunk/src/ISO5436\\_2\\_XML/cxx/point\\_vector\\_proxy.cxx](#)

## 9.51 OpenGPS::PointVectorProxyContext Class Reference

```
#include <point_vector_proxy_context.hxx>
```

### Public Member Functions

- virtual unsigned long [GetU](#) () const
- virtual unsigned long [GetV](#) () const
- virtual unsigned long [GetW](#) () const
- [PointVectorProxyContext](#) ()
- virtual void [SetU](#) (const unsigned long value)
- virtual void [SetV](#) (const unsigned long value)
- virtual void [SetW](#) (const unsigned long value)
- virtual [~PointVectorProxyContext](#) ()

### Private Attributes

- unsigned long [m\\_U](#)
- unsigned long [m\\_V](#)
- unsigned long [m\\_W](#)

### 9.51.1 Constructor & Destructor Documentation

#### 9.51.1.1 [PointVectorProxyContext::PointVectorProxyContext](#) ()

**9.51.1.2 [PointVectorProxyContext::~~PointVectorProxyContext](#)** ()  
[virtual]

### 9.51.2 Member Function Documentation

**9.51.2.1 unsigned long [PointVectorProxyContext::GetU](#) () const** [virtual]

**9.51.2.2 unsigned long [PointVectorProxyContext::GetV](#) () const** [virtual]

**9.51.2.3 unsigned long [PointVectorProxyContext::GetW](#) () const** [virtual]

**9.51.2.4** void PointVectorProxyContext::SetU (const unsigned long value)  
[virtual]

**9.51.2.5** void PointVectorProxyContext::SetV (const unsigned long value)  
[virtual]

**9.51.2.6** void PointVectorProxyContext::SetW (const unsigned long value)  
[virtual]

### 9.51.3 Member Data Documentation

**9.51.3.1** unsigned long OpenGPS::PointVectorProxyContext::m\_U  
[private]

**9.51.3.2** unsigned long OpenGPS::PointVectorProxyContext::m\_V  
[private]

**9.51.3.3** unsigned long OpenGPS::PointVectorProxyContext::m\_W  
[private]

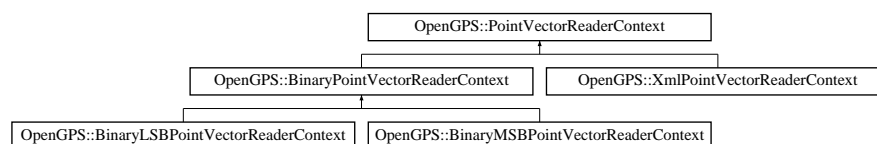
The documentation for this class was generated from the following files:

- S:/openGPS/ISO5436\_XML/trunk/src/ISO5436\_2\_XML/cxx/[point\\_vector\\_proxy\\_context.hxx](#)
- S:/openGPS/ISO5436\_XML/trunk/src/ISO5436\_2\_XML/cxx/[point\\_vector\\_proxy\\_context.cxx](#)

## 9.52 OpenGPS::PointVectorReaderContext Class Reference

```
#include <point_vector_reader_context.hxx>
```

Inheritance diagram for OpenGPS::PointVectorReaderContext::



### Public Member Functions

- virtual OGPS\_Boolean [IsValid](#) () const =0
- virtual OGPS\_Boolean [MoveNext](#) ()=0
- virtual OGPS\_Boolean [Read](#) (double \*value)=0
- virtual OGPS\_Boolean [Read](#) (float \*value)=0



- virtual OGPS\_Boolean [Read](#) (int \*value)=0
- virtual OGPS\_Boolean [Read](#) (short \*value)=0
- virtual OGPS\_Boolean [Skip](#) ()=0
- virtual [~PointVectorReaderContext](#) ()

### Protected Member Functions

- [PointVectorReaderContext](#) ()

### 9.52.1 Constructor & Destructor Documentation

**9.52.1.1** [PointVectorReaderContext::~~PointVectorReaderContext](#) ()  
[virtual]

**9.52.1.2** [PointVectorReaderContext::PointVectorReaderContext](#) ()  
[protected]

### 9.52.2 Member Function Documentation

**9.52.2.1** virtual OGPS\_Boolean [OpenGPS::PointVectorReaderContext::IsValid](#) () const [pure virtual]

Implemented in [OpenGPS::BinaryLSBPointVectorReaderContext](#), [OpenGPS::Binary-MSBPointVectorReaderContext](#), and [OpenGPS::XmlPointVectorReaderContext](#).

**9.52.2.2** virtual OGPS\_Boolean [OpenGPS::PointVectorReaderContext::Move-Next](#) () [pure virtual]

Implemented in [OpenGPS::BinaryLSBPointVectorReaderContext](#), [OpenGPS::Binary-MSBPointVectorReaderContext](#), and [OpenGPS::XmlPointVectorReaderContext](#).

**9.52.2.3** virtual OGPS\_Boolean [OpenGPS::PointVectorReaderContext::Read](#) (double \* *value*) [pure virtual]

Implemented in [OpenGPS::BinaryLSBPointVectorReaderContext](#), [OpenGPS::Binary-MSBPointVectorReaderContext](#), and [OpenGPS::XmlPointVectorReaderContext](#).

**9.52.2.4** virtual OGPS\_Boolean [OpenGPS::PointVectorReaderContext::Read](#) (float \* *value*) [pure virtual]

Implemented in [OpenGPS::BinaryLSBPointVectorReaderContext](#), [OpenGPS::Binary-MSBPointVectorReaderContext](#), and [OpenGPS::XmlPointVectorReaderContext](#).

**9.52.2.5 virtual OGPS\_Boolean OpenGPS::PointVectorReaderContext::Read (int \* *value*)** [pure virtual]

Implemented in [OpenGPS::BinaryLSBPointVectorReaderContext](#), [OpenGPS::Binary-MSBPointVectorReaderContext](#), and [OpenGPS::XmlPointVectorReaderContext](#).

**9.52.2.6 virtual OGPS\_Boolean OpenGPS::PointVectorReaderContext::Read (short \* *value*)** [pure virtual]

Implemented in [OpenGPS::BinaryLSBPointVectorReaderContext](#), [OpenGPS::Binary-MSBPointVectorReaderContext](#), and [OpenGPS::XmlPointVectorReaderContext](#).

**9.52.2.7 virtual OGPS\_Boolean OpenGPS::PointVectorReaderContext::Skip ()** [pure virtual]

Implemented in [OpenGPS::BinaryLSBPointVectorReaderContext](#), [OpenGPS::Binary-MSBPointVectorReaderContext](#), and [OpenGPS::XmlPointVectorReaderContext](#).

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

- [S:/openGPS/ISO5436\\_XML/trunk/src/ISO5436\\_2\\_XML/cxx/point\\_vector\\_reader\\_context.hxx](#)
- [S:/openGPS/ISO5436\\_XML/trunk/src/ISO5436\\_2\\_XML/cxx/xml\\_point\\_vector\\_reader\\_context.cxx](#)

## 9.53 OpenGPS::PointVectorWhitespaceFacet Class Reference

```
#include <point_vector_iostream.hxx>
```

### Public Member Functions

- [PointVectorWhitespaceFacet](#) (size\_t refs=0)
- [~PointVectorWhitespaceFacet](#) ()

### Protected Member Functions

- virtual bool [do\\_is](#) (mask msk, OGPS\_Character ch) const

### Private Types

- typedef std::ctype< OGPS\_Character > [BaseType](#)

### 9.53.1 Member Typedef Documentation

**9.53.1.1 typedef std::ctype<OGPS\_Character> [OpenGPS::PointVector-WhitespaceFacet::BaseType](#)** [private]

### 9.53.2 Constructor & Destructor Documentation

**9.53.2.1 PointVectorWhitespaceFacet::PointVectorWhitespaceFacet (size\_t refs = 0)**

**9.53.2.2 PointVectorWhitespaceFacet::~~PointVectorWhitespaceFacet ()**

### 9.53.3 Member Function Documentation

**9.53.3.1 bool PointVectorWhitespaceFacet::do\_is (mask *msk*, OGPS\_Character *ch*) const** [protected, virtual]

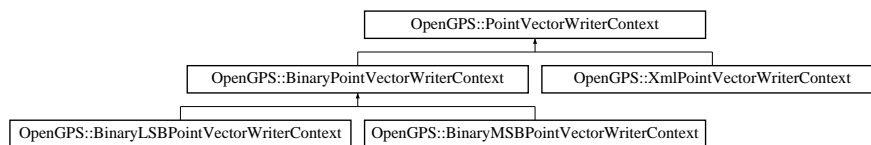
The documentation for this class was generated from the following files:

- S:/openGPS/ISO5436\_XML/trunk/src/ISO5436\_2\_XML/cxx/[point\\_vector\\_istream.hxx](#)
- S:/openGPS/ISO5436\_XML/trunk/src/ISO5436\_2\_XML/cxx/[point\\_vector\\_istream.cxx](#)

## 9.54 OpenGPS::PointVectorWriterContext Class Reference

```
#include <point_vector_writer_context.hxx>
```

Inheritance diagram for OpenGPS::PointVectorWriterContext::



### Public Member Functions

- virtual OGPS\_Boolean [MoveNext](#) ()=0
- virtual OGPS\_Boolean [Skip](#) ()=0
- virtual OGPS\_Boolean [Write](#) (const double \*value)=0
- virtual OGPS\_Boolean [Write](#) (const float \*value)=0
- virtual OGPS\_Boolean [Write](#) (const int \*value)=0
- virtual OGPS\_Boolean [Write](#) (const short \*value)=0
- virtual [~PointVectorWriterContext](#) ()

### Protected Member Functions

- virtual OGPS\_Boolean [IsGood](#) () const =0
- [PointVectorWriterContext](#) ()

### 9.54.1 Constructor & Destructor Documentation

**9.54.1.1** `PointVectorWriterContext::~~PointVectorWriterContext` ()  
[virtual]

**9.54.1.2** `PointVectorWriterContext::PointVectorWriterContext` ()  
[protected]

### 9.54.2 Member Function Documentation

**9.54.2.1** `virtual OGPS_Boolean OpenGPS::PointVectorWriterContext::IsGood`  
`() const` [protected, pure virtual]

Implemented in [OpenGPS::BinaryPointVectorWriterContext](#), and [OpenGPS::Xml-PointVectorWriterContext](#).

**9.54.2.2** `virtual OGPS_Boolean OpenGPS::PointVectorWriterContext::Move-`  
`Next()` [pure virtual]

Implemented in [OpenGPS::BinaryPointVectorWriterContext](#), and [OpenGPS::Xml-PointVectorWriterContext](#).

**9.54.2.3** `virtual OGPS_Boolean OpenGPS::PointVectorWriterContext::Skip` ()  
[pure virtual]

Implemented in [OpenGPS::BinaryPointVectorWriterContext](#), and [OpenGPS::Xml-PointVectorWriterContext](#).

**9.54.2.4** `virtual OGPS_Boolean OpenGPS::PointVectorWriterContext::Write`  
`(const double * value)` [pure virtual]

Implemented in [OpenGPS::BinaryLSBPointVectorWriterContext](#), [OpenGPS::Binary-MSBPointVectorWriterContext](#), and [OpenGPS::XmlPointVectorWriterContext](#).

**9.54.2.5** `virtual OGPS_Boolean OpenGPS::PointVectorWriterContext::Write`  
`(const float * value)` [pure virtual]

Implemented in [OpenGPS::BinaryLSBPointVectorWriterContext](#), [OpenGPS::Binary-MSBPointVectorWriterContext](#), and [OpenGPS::XmlPointVectorWriterContext](#).

**9.54.2.6** `virtual OGPS_Boolean OpenGPS::PointVectorWriterContext::Write`  
`(const int * value)` [pure virtual]

Implemented in [OpenGPS::BinaryLSBPointVectorWriterContext](#), [OpenGPS::Binary-MSBPointVectorWriterContext](#), and [OpenGPS::XmlPointVectorWriterContext](#).

### 9.54.2.7 virtual OGPS\_Boolean OpenGPS::PointVectorWriterContext::Write (const short \* *value*) [pure virtual]

Implemented in [OpenGPS::BinaryLSBPointVectorWriterContext](#), [OpenGPS::BinaryMSBPointVectorWriterContext](#), and [OpenGPS::XmlPointVectorWriterContext](#).

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

- S:/openGPS/ISO5436\_XML/trunk/src/ISO5436\_2\_XML/cxx/[point\\_vector\\_writer\\_context.hxx](#)
- S:/openGPS/ISO5436\_XML/trunk/src/ISO5436\_2\_XML/cxx/[xml\\_point\\_vector\\_writer\\_context.cxx](#)

## 9.55 xsd::ProbingSystemType Class Reference

```
#include <iso5436_2_xsd.hxx>
```

### 9.55.1 Detailed Description

Class corresponding to the ProbingSystemType schema type.

#### Identification

Accessor and modifier functions for the Identification required element.

Vendor specific identification of probe tip, lens, etc...

- typedef ::xsd::cxx::tree::traits< [Identification\\_type](#), wchar\_t > [Identification\\_traits](#)  
*Element traits type.*
- typedef ::xml\_schema::token [Identification\\_type](#)  
*Element type.*
- void [Identification](#) (::std::auto\_ptr< [Identification\\_type](#) > p)  
*Set the element value without copying.*
- void [Identification](#) (const [Identification\\_type](#) &x)  
*Set the element value.*
- [Identification\\_type](#) & [Identification](#) ()  
*Return a read-write reference to the element.*
- const [Identification\\_type](#) & [Identification](#) () const  
*Return a read-only (constant) reference to the element.*

## Type

Accessor and modifier functions for the Type required element.

one of "NonContacting" or "Contacting"

- `typedef ::xsd::cxx::tree::traits< Type_type, wchar_t > Type_traits`  
*Element traits type.*
- `typedef ::xsd::Type Type_type`  
*Element type.*
- `void Type (::std::auto_ptr< Type_type > p)`  
*Set the element value without copying.*
- `void Type (const Type_type &x)`  
*Set the element value.*
- `Type_type & Type ()`  
*Return a read-write reference to the element.*
- `const Type_type & Type () const`  
*Return a read-only (constant) reference to the element.*

## Constructors

- `virtual ProbingSystemType * _clone (::xml_schema::flags f=0, ::xml_schema::type *c=0) const`  
*Copy the object polymorphically.*
- `ProbingSystemType (const ProbingSystemType &x, ::xml_schema::flags f=0, ::xml_schema::type *c=0)`  
*Copy constructor.*
- `ProbingSystemType (const ::xercesc::DOMElement &e, ::xml_schema::flags f=0, ::xml_schema::type *c=0)`  
*Construct an instance from a DOM element.*
- `ProbingSystemType (const Type_type &, const Identification_type &)`  
*Construct an instance from the ultimate base and initializers for required elements and attributes.*

## Protected Member Functions

- `void parse (::xsd::cxx::xml::dom::parser< wchar_t > &, ::xml_schema::flags)`

**Private Attributes**

- `::xsd::cxx::tree::one< Identification\_type > Identification\_`
- `::xsd::cxx::tree::one< Type\_type > Type\_`

**9.55.2 Member Typedef Documentation**

**9.55.2.1** `typedef ::xsd::cxx::tree::traits< Identification\_type, wchar_t > xsd::ProbingSystemType::Identification\_traits`

Element traits type.

**9.55.2.2** `typedef ::xml\_schema::token xsd::ProbingSystemType::Identification\_type`

Element type.

**9.55.2.3** `typedef ::xsd::cxx::tree::traits< Type\_type, wchar_t > xsd::ProbingSystemType::Type\_traits`

Element traits type.

**9.55.2.4** `typedef ::xsd::Type xsd::ProbingSystemType::Type\_type`

Element type.

**9.55.3 Constructor & Destructor Documentation**

**9.55.3.1** `xsd::ProbingSystemType::ProbingSystemType (const Type\_type &, const Identification\_type &)`

Construct an instance from the ultimate base and initializers for required elements and attributes.

**9.55.3.2** `xsd::ProbingSystemType::ProbingSystemType (const ::xercesc::DOMElement & e, ::xml\_schema::flags f = 0, ::xml\_schema::type * c = 0)`

Construct an instance from a DOM element.

**Parameters:**

- e* A DOM element to extract the data from.
- f* Flags to construct the new instance with.
- c* A pointer to the object that will contain the new instance.

### 9.55.3.3 xsd::ProbingSystemType::ProbingSystemType (const ProbingSystemType & *x*, ::xml\_schema::flags *f* = 0, ::xml\_schema::type \* *c* = 0)

Copy constructor.

#### Parameters:

- x* An instance to make a copy of.
- f* Flags to construct the copy with.
- c* A pointer to the object that will contain the copy.

For polymorphic object models use the `_clone` function instead.

## 9.55.4 Member Function Documentation

### 9.55.4.1 ProbingSystemType \* xsd::ProbingSystemType::\_clone (::xml\_schema::flags *f* = 0, ::xml\_schema::type \* *c* = 0) const [virtual]

Copy the object polymorphically.

#### Parameters:

- f* Flags to construct the copy with.
- c* A pointer to the object that will contain the copy.

#### Returns:

A pointer to the dynamically allocated copy.

This function ensures that the dynamic type of an instance is used for copying and should be used for polymorphic object models instead of the copy constructor.

### 9.55.4.2 void xsd::ProbingSystemType::Identification (::std::auto\_ptr< Identification\_type > *p*)

Set the element value without copying.

#### Parameters:

- p* A new value to use.

This function will try to use the passed value directly instead of making a copy.

### 9.55.4.3 void xsd::ProbingSystemType::Identification (const Identification\_type & *x*)

Set the element value.

#### Parameters:

- x* A new value to set.

This function makes a copy of its argument and sets it as the new value of the element.



**9.55.4.4 Identification\_type& xsd::ProbingSystemType::Identification ()**

Return a read-write reference to the element.

**Returns:**

A reference to the element.

**9.55.4.5 const Identification\_type& xsd::ProbingSystemType::Identification () const**

Return a read-only (constant) reference to the element.

**Returns:**

A constant reference to the element.

**9.55.4.6 void xsd::ProbingSystemType::parse (::xsd::cxx::xml::dom::parser< wchar\_t > &, ::xml\_schema::flags) [protected]****9.55.4.7 void xsd::ProbingSystemType::Type (::std::auto\_ptr< Type\_type > p)**

Set the element value without copying.

**Parameters:**

*p* A new value to use.

This function will try to use the passed value directly instead of making a copy.

**9.55.4.8 void xsd::ProbingSystemType::Type (const Type\_type & x)**

Set the element value.

**Parameters:**

*x* A new value to set.

This function makes a copy of its argument and sets it as the new value of the element.

**9.55.4.9 Type\_type& xsd::ProbingSystemType::Type ()**

Return a read-write reference to the element.

**Returns:**

A reference to the element.

**9.55.4.10** const [Type\\_type](#)& xsd::ProbingSystemType::Type () const

Return a read-only (constant) reference to the element.

**Returns:**

A constant reference to the element.

**9.55.5 Member Data Documentation****9.55.5.1** ::xsd::cxx::tree::one< [Identification\\_type](#) > [xsd::ProbingSystemType::Identification\\_](#) [private]**9.55.5.2** ::xsd::cxx::tree::one< [Type\\_type](#) > [xsd::ProbingSystemType::Type\\_](#) [private]

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

- S:/openGPS/ISO5436\_XML/trunk/src/ISO5436\_2\_XML/[iso5436\\_2\\_xsd.hxx](#)
- S:/openGPS/ISO5436\_XML/trunk/src/ISO5436\_2\_XML/[iso5436\\_2\\_xsd.cxx](#)

**9.56 xsd::Record1Type Class Reference**

```
#include <iso5436_2_xsd.hxx>
```

**9.56.1 Detailed Description**

Class corresponding to the Record1Type schema type.

Record1 contains the axis description

**Axes**

Accessor and modifier functions for the Axes required element.

Axis description

- typedef ::xsd::cxx::tree::traits< [Axes\\_type](#), wchar\_t > [Axes\\_traits](#)  
*Element traits type.*
- typedef ::xsd::AxesType [Axes\\_type](#)  
*Element type.*
- void [Axes](#) (::std::auto\_ptr< [Axes\\_type](#) > p)  
*Set the element value without copying.*
- void [Axes](#) (const [Axes\\_type](#) &x)

*Set the element value.*

- [Axes\\_type](#) & [Axes](#) ()

*Return a read-write reference to the element.*

- const [Axes\\_type](#) & [Axes](#) () const

*Return a read-only (constant) reference to the element.*

## FeatureType

Accessor and modifier functions for the FeatureType required element.

"SUR" for surface type feature, "PRF" for profile type feature. Profile features are allways defined as a matrix of size (N,1,M) with N beeing the number of points in the profile and M the number of layers in z-direction.

- typedef ::xsd::cxx::tree::traits< [FeatureType\\_type](#), wchar\_t > [FeatureType\\_traits](#)

*Element traits type.*

- typedef ::xsd::FeatureType [FeatureType\\_type](#)

*Element type.*

- void [FeatureType](#) (::std::auto\_ptr< [FeatureType\\_type](#) > p)

*Set the element value without copying.*

- void [FeatureType](#) (const [FeatureType\\_type](#) &x)

*Set the element value.*

- [FeatureType\\_type](#) & [FeatureType](#) ()

*Return a read-write reference to the element.*

- const [FeatureType\\_type](#) & [FeatureType](#) () const

*Return a read-only (constant) reference to the element.*

## Revision

Accessor and modifier functions for the Revision required element.

Revision of file format. Currently: ISO5436 - 2000

- typedef ::xsd::cxx::tree::traits< [Revision\\_type](#), wchar\_t > [Revision\\_traits](#)

*Element traits type.*

- typedef ::xml\_schema::token [Revision\\_type](#)

*Element type.*

- void [Revision](#) (::std::auto\_ptr< [Revision\\_type](#) > p)  
*Set the element value without copying.*
- void [Revision](#) (const [Revision\\_type](#) &x)  
*Set the element value.*
- [Revision\\_type](#) & [Revision](#) ()  
*Return a read-write reference to the element.*
- const [Revision\\_type](#) & [Revision](#) () const  
*Return a read-only (constant) reference to the element.*

### Constructors

- virtual [Record1Type](#) \* [\\_clone](#) (::xml\_schema::flags f=0,::xml\_schema::type \*c=0) const  
*Copy the object polymorphically.*
- [Record1Type](#) (const [Record1Type](#) &x,::xml\_schema::flags f=0,::xml\_schema::type \*c=0)  
*Copy constructor.*
- [Record1Type](#) (const ::xercesc::DOMElement &e,::xml\_schema::flags f=0,::xml\_schema::type \*c=0)  
*Construct an instance from a DOM element.*
- [Record1Type](#) (const [Revision\\_type](#) &, const [FeatureType\\_type](#) &, const [Axes\\_type](#) &)  
*Construct an instance from the ultimate base and initializers for required elements and attributes.*

### Protected Member Functions

- void [parse](#) (::xsd::cxx::xml::dom::parser< wchar\_t > &,::xml\_schema::flags)

### Private Attributes

- ::xsd::cxx::tree::one< [Axes\\_type](#) > [Axes\\_](#)
- ::xsd::cxx::tree::one< [FeatureType\\_type](#) > [FeatureType\\_](#)
- ::xsd::cxx::tree::one< [Revision\\_type](#) > [Revision\\_](#)

## 9.56.2 Member Typedef Documentation

**9.56.2.1** `typedef ::xsd::cxx::tree::traits< Axes\_type, wchar_t > xsd::Record1Type::Axes\_traits`

Element traits type.

**9.56.2.2** `typedef ::xsd::AxesType xsd::Record1Type::Axes\_type`

Element type.

**9.56.2.3** `typedef ::xsd::cxx::tree::traits< FeatureType\_type, wchar_t > xsd::Record1Type::FeatureType\_traits`

Element traits type.

**9.56.2.4** `typedef ::xsd::FeatureType xsd::Record1Type::FeatureType\_type`

Element type.

**9.56.2.5** `typedef ::xsd::cxx::tree::traits< Revision\_type, wchar_t > xsd::Record1Type::Revision\_traits`

Element traits type.

**9.56.2.6** `typedef ::xml_schema::token xsd::Record1Type::Revision\_type`

Element type.

## 9.56.3 Constructor & Destructor Documentation

**9.56.3.1** `xsd::Record1Type::Record1Type (const Revision\_type &, const FeatureType\_type &, const Axes\_type &)`

Construct an instance from the ultimate base and initializers for required elements and attributes.

**9.56.3.2** `xsd::Record1Type::Record1Type (const ::xercesc::DOMElement & e, ::xml\_schema::flags f = 0, ::xml\_schema::type * c = 0)`

Construct an instance from a DOM element.

### Parameters:

- e* A DOM element to extract the data from.
- f* Flags to construct the new instance with.
- c* A pointer to the object that will contain the new instance.

### 9.56.3.3 xsd::Record1Type::Record1Type (const Record1Type & *x*, ::xml\_schema::flags *f* = 0, ::xml\_schema::type \* *c* = 0)

Copy constructor.

#### Parameters:

- x* An instance to make a copy of.
- f* Flags to construct the copy with.
- c* A pointer to the object that will contain the copy.

For polymorphic object models use the `_clone` function instead.

## 9.56.4 Member Function Documentation

### 9.56.4.1 Record1Type \* xsd::Record1Type::\_clone (::xml\_schema::flags *f* = 0, ::xml\_schema::type \* *c* = 0) const [virtual]

Copy the object polymorphically.

#### Parameters:

- f* Flags to construct the copy with.
- c* A pointer to the object that will contain the copy.

#### Returns:

A pointer to the dynamically allocated copy.

This function ensures that the dynamic type of an instance is used for copying and should be used for polymorphic object models instead of the copy constructor.

### 9.56.4.2 void xsd::Record1Type::Axes (::std::auto\_ptr< Axes\_type > *p*)

Set the element value without copying.

#### Parameters:

- p* A new value to use.

This function will try to use the passed value directly instead of making a copy.

### 9.56.4.3 void xsd::Record1Type::Axes (const Axes\_type & *x*)

Set the element value.

#### Parameters:

- x* A new value to set.

This function makes a copy of its argument and sets it as the new value of the element.

#### 9.56.4.4 [Record1Type::Axes\\_type](#) & xsd::Record1Type::Axes ()

Return a read-write reference to the element.

##### Returns:

A reference to the element.

#### 9.56.4.5 `const Record1Type::Axes\_type & xsd::Record1Type::Axes () const`

Return a read-only (constant) reference to the element.

##### Returns:

A constant reference to the element.

#### 9.56.4.6 `void xsd::Record1Type::FeatureType (::std::auto_ptr< FeatureType\_type > p)`

Set the element value without copying.

##### Parameters:

*p* A new value to use.

This function will try to use the passed value directly instead of making a copy.

#### 9.56.4.7 `void xsd::Record1Type::FeatureType (const FeatureType\_type & x)`

Set the element value.

##### Parameters:

*x* A new value to set.

This function makes a copy of its argument and sets it as the new value of the element.

#### 9.56.4.8 [Record1Type::FeatureType\\_type](#) & xsd::Record1Type::FeatureType ()

Return a read-write reference to the element.

##### Returns:

A reference to the element.

#### 9.56.4.9 const [Record1Type::FeatureType\\_type](#) & xsd::Record1Type::FeatureType () const

Return a read-only (constant) reference to the element.

##### Returns:

A constant reference to the element.

#### 9.56.4.10 void xsd::Record1Type::parse (::xsd::cxx::xml::dom::parser< wchar\_t > &, ::[xml\\_schema::flags](#)) [protected]

#### 9.56.4.11 void xsd::Record1Type::Revision (::std::auto\_ptr< [Revision\\_type](#) > p)

Set the element value without copying.

##### Parameters:

*p* A new value to use.

This function will try to use the passed value directly instead of making a copy.

#### 9.56.4.12 void xsd::Record1Type::Revision (const [Revision\\_type](#) & x)

Set the element value.

##### Parameters:

*x* A new value to set.

This function makes a copy of its argument and sets it as the new value of the element.

#### 9.56.4.13 [Record1Type::Revision\\_type](#) & xsd::Record1Type::Revision ()

Return a read-write reference to the element.

##### Returns:

A reference to the element.

#### 9.56.4.14 const [Record1Type::Revision\\_type](#) & xsd::Record1Type::Revision () const

Return a read-only (constant) reference to the element.

##### Returns:

A constant reference to the element.



### 9.56.5 Member Data Documentation

**9.56.5.1** `::xsd::cxx::tree::one< Axes_type > xsd::Record1Type::Axes_-`  
[private]

**9.56.5.2** `::xsd::cxx::tree::one< FeatureType_type >`  
`xsd::Record1Type::FeatureType_` [private]

**9.56.5.3** `::xsd::cxx::tree::one< Revision_type > xsd::Record1Type::Revision_-`  
[private]

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

- S:/openGPS/ISO5436\_XML/trunk/src/ISO5436\_2\_XML/iso5436\_2\_xsd.hxx
- S:/openGPS/ISO5436\_XML/trunk/src/ISO5436\_2\_XML/iso5436\_2\_xsd.cxx

## 9.57 xsd::Record2Type Class Reference

```
#include <iso5436_2_xsd.hxx>
```

### 9.57.1 Detailed Description

Class corresponding to the Record2Type schema type.

Record2 is optional and contains the metadata of the data set.

#### CalibrationDate

Accessor and modifier functions for the CalibrationDate required element.

Date of currently used calibration

- `typedef ::xsd::cxx::tree::traits< CalibrationDate_type, wchar_t > CalibrationDate_traits`  
*Element traits type.*
- `typedef ::xml_schema::date_time CalibrationDate_type`  
*Element type.*
- `void CalibrationDate (::std::auto_ptr< CalibrationDate_type > p)`  
*Set the element value without copying.*
- `void CalibrationDate (const CalibrationDate_type &x)`  
*Set the element value.*
- `CalibrationDate_type & CalibrationDate ()`

*Return a read-write reference to the element.*

- const [CalibrationDate\\_type](#) & [CalibrationDate](#) () const

*Return a read-only (constant) reference to the element.*

## Comment

Accessor and modifier functions for the Comment optional element.

User comment to this data set

- typedef ::xsd::cxx::tree::optional< [Comment\\_type](#) > [Comment\\_optional](#)  
*Element optional container type.*
- typedef ::xsd::cxx::tree::traits< [Comment\\_type](#), wchar\_t > [Comment\\_traits](#)  
*Element traits type.*
- typedef ::xml\_schema::string [Comment\\_type](#)  
*Element type.*
- void [Comment](#) (::std::auto\_ptr< [Comment\\_type](#) > p)  
*Set the element value without copying.*
- void [Comment](#) (const [Comment\\_optional](#) &x)  
*Set the element value.*
- void [Comment](#) (const [Comment\\_type](#) &x)  
*Set the element value.*
- [Comment\\_optional](#) & [Comment](#) ()  
*Return a read-write reference to the element container.*
- const [Comment\\_optional](#) & [Comment](#) () const  
*Return a read-only (constant) reference to the element container.*

## Creator

Accessor and modifier functions for the Creator optional element.

Optional name of the creator of the file: Name of the measuring person.

- typedef ::xsd::cxx::tree::optional< [Creator\\_type](#) > [Creator\\_optional](#)  
*Element optional container type.*
- typedef ::xsd::cxx::tree::traits< [Creator\\_type](#), wchar\_t > [Creator\\_traits](#)

*Element traits type.*

- typedef [::xml\\_schema::token](#) [Creator\\_type](#)  
*Element type.*
- void [Creator](#) (::std::auto\_ptr< [Creator\\_type](#) > p)  
*Set the element value without copying.*
- void [Creator](#) (const [Creator\\_optional](#) &x)  
*Set the element value.*
- void [Creator](#) (const [Creator\\_type](#) &x)  
*Set the element value.*
- [Creator\\_optional](#) & [Creator](#) ()  
*Return a read-write reference to the element container.*
- const [Creator\\_optional](#) & [Creator](#) () const  
*Return a read-only (constant) reference to the element container.*

## Date

Accessor and modifier functions for the Date required element.

Date and time of file creation.

- typedef [::xsd::cxx::tree::traits](#)< [Date\\_type](#), [wchar\\_t](#) > [Date\\_traits](#)  
*Element traits type.*
- typedef [::xml\\_schema::date\\_time](#) [Date\\_type](#)  
*Element type.*
- void [Date](#) (::std::auto\_ptr< [Date\\_type](#) > p)  
*Set the element value without copying.*
- void [Date](#) (const [Date\\_type](#) &x)  
*Set the element value.*
- [Date\\_type](#) & [Date](#) ()  
*Return a read-write reference to the element.*
- const [Date\\_type](#) & [Date](#) () const  
*Return a read-only (constant) reference to the element.*

## Instrument

Accessor and modifier functions for the Instrument required element.

- typedef ::xsd::cxx::tree::traits< [Instrument\\_type](#), wchar\_t > [Instrument\\_traits](#)  
*Element traits type.*
- typedef ::xsd::InstrumentType [Instrument\\_type](#)  
*Element type.*
- void [Instrument](#) (::std::auto\_ptr< [Instrument\\_type](#) > p)  
*Set the element value without copying.*
- void [Instrument](#) (const [Instrument\\_type](#) &x)  
*Set the element value.*
- [Instrument\\_type](#) & [Instrument](#) ()  
*Return a read-write reference to the element.*
- const [Instrument\\_type](#) & [Instrument](#) () const  
*Return a read-only (constant) reference to the element.*

## ProbingSystem

Accessor and modifier functions for the ProbingSystem required element.

- typedef ::xsd::cxx::tree::traits< [ProbingSystem\\_type](#), wchar\_t > [ProbingSystem\\_traits](#)  
*Element traits type.*
- typedef ::xsd::ProbingSystemType [ProbingSystem\\_type](#)  
*Element type.*
- void [ProbingSystem](#) (::std::auto\_ptr< [ProbingSystem\\_type](#) > p)  
*Set the element value without copying.*
- void [ProbingSystem](#) (const [ProbingSystem\\_type](#) &x)  
*Set the element value.*
- [ProbingSystem\\_type](#) & [ProbingSystem](#) ()  
*Return a read-write reference to the element.*
- const [ProbingSystem\\_type](#) & [ProbingSystem](#) () const  
*Return a read-only (constant) reference to the element.*

## Constructors

- virtual [Record2Type](#) \* [\\_clone](#) (::xml\_schema::flags f=0,::xml\_schema::type \*c=0) const  
*Copy the object polymorphically.*
- [Record2Type](#) (const [Record2Type](#) &x,::xml\_schema::flags f=0,::xml\_schema::type \*c=0)  
*Copy constructor.*
- [Record2Type](#) (const ::xercesc::DOMElement &e,::xml\_schema::flags f=0,::xml\_schema::type \*c=0)  
*Construct an instance from a DOM element.*
- [Record2Type](#) (const [Date\\_type](#) &, const [Instrument\\_type](#) &, const [CalibrationDate\\_type](#) &, const [ProbingSystem\\_type](#) &)  
*Construct an instance from the ultimate base and initializers for required elements and attributes.*

## Protected Member Functions

- void [parse](#) (::xsd::cxx::xml::dom::parser< wchar\_t > &,::xml\_schema::flags)

## Private Attributes

- ::xsd::cxx::tree::one< [CalibrationDate\\_type](#) > [CalibrationDate\\_](#)
- [Comment\\_optional](#) [Comment\\_](#)
- [Creator\\_optional](#) [Creator\\_](#)
- ::xsd::cxx::tree::one< [Date\\_type](#) > [Date\\_](#)
- ::xsd::cxx::tree::one< [Instrument\\_type](#) > [Instrument\\_](#)
- ::xsd::cxx::tree::one< [ProbingSystem\\_type](#) > [ProbingSystem\\_](#)

### 9.57.2 Member Typedef Documentation

**9.57.2.1** typedef ::xsd::cxx::tree::traits< [CalibrationDate\\_type](#), wchar\_t > [xsd::Record2Type::CalibrationDate\\_traits](#)

Element traits type.

**9.57.2.2** typedef ::xml\_schema::date\_time [xsd::Record2Type::CalibrationDate\\_type](#)

Element type.

**9.57.2.3** typedef ::xsd::cxx::tree::optional< [Comment\\_type](#) >  
[xsd::Record2Type::Comment\\_optional](#)

Element optional container type.

**9.57.2.4** typedef ::xsd::cxx::tree::traits< [Comment\\_type](#), wchar\_t >  
[xsd::Record2Type::Comment\\_traits](#)

Element traits type.

**9.57.2.5** typedef ::xml\_schema::string xsd::Record2Type::Comment\_type

Element type.

**9.57.2.6** typedef ::xsd::cxx::tree::optional< [Creator\\_type](#) >  
[xsd::Record2Type::Creator\\_optional](#)

Element optional container type.

**9.57.2.7** typedef ::xsd::cxx::tree::traits< [Creator\\_type](#), wchar\_t >  
[xsd::Record2Type::Creator\\_traits](#)

Element traits type.

**9.57.2.8** typedef ::xml\_schema::token xsd::Record2Type::Creator\_type

Element type.

**9.57.2.9** typedef ::xsd::cxx::tree::traits< [Date\\_type](#), wchar\_t >  
[xsd::Record2Type::Date\\_traits](#)

Element traits type.

**9.57.2.10** typedef ::xml\_schema::date\_time xsd::Record2Type::Date\_type

Element type.

**9.57.2.11** typedef ::xsd::cxx::tree::traits< [Instrument\\_type](#), wchar\_t >  
[xsd::Record2Type::Instrument\\_traits](#)

Element traits type.

**9.57.2.12** typedef ::xsd::InstrumentType xsd::Record2Type::Instrument\_type

Element type.

**9.57.2.13** `typedef ::xsd::cxx::tree::traits< ProbingSystem\_type, wchar_t > xsd::Record2Type::ProbingSystem\_traits`

Element traits type.

**9.57.2.14** `typedef ::xsd::ProbingSystemType xsd::Record2Type::ProbingSystem\_type`

Element type.

### 9.57.3 Constructor & Destructor Documentation

**9.57.3.1** `xsd::Record2Type::Record2Type (const Date\_type &, const Instrument\_type &, const CalibrationDate\_type &, const ProbingSystem\_type &)`

Construct an instance from the ultimate base and initializers for required elements and attributes.

**9.57.3.2** `xsd::Record2Type::Record2Type (const ::xercesc::DOMElement & e, ::xml\_schema::flags f = 0, ::xml\_schema::type * c = 0)`

Construct an instance from a DOM element.

#### Parameters:

- e* A DOM element to extract the data from.
- f* Flags to construct the new instance with.
- c* A pointer to the object that will contain the new instance.

**9.57.3.3** `xsd::Record2Type::Record2Type (const Record2Type & x, ::xml\_schema::flags f = 0, ::xml\_schema::type * c = 0)`

Copy constructor.

#### Parameters:

- x* An instance to make a copy of.
- f* Flags to construct the copy with.
- c* A pointer to the object that will contain the copy.

For polymorphic object models use the `_clone` function instead.

### 9.57.4 Member Function Documentation

**9.57.4.1** `Record2Type * xsd::Record2Type::\_clone (::xml\_schema::flags f = 0, ::xml\_schema::type * c = 0) const virtual`

Copy the object polymorphically.

**Parameters:**

- f* Flags to construct the copy with.
- c* A pointer to the object that will contain the copy.

**Returns:**

A pointer to the dynamically allocated copy.

This function ensures that the dynamic type of an instance is used for copying and should be used for polymorphic object models instead of the copy constructor.

**9.57.4.2** `void xsd::Record2Type::CalibrationDate (::std::auto_ptr< CalibrationDate_type > p)`

Set the element value without copying.

**Parameters:**

- p* A new value to use.

This function will try to use the passed value directly instead of making a copy.

**9.57.4.3** `void xsd::Record2Type::CalibrationDate (const CalibrationDate_type & x)`

Set the element value.

**Parameters:**

- x* A new value to set.

This function makes a copy of its argument and sets it as the new value of the element.

**9.57.4.4** `Record2Type::CalibrationDate_type & xsd::Record2Type::CalibrationDate ()`

Return a read-write reference to the element.

**Returns:**

A reference to the element.

**9.57.4.5** `const Record2Type::CalibrationDate_type & xsd::Record2Type::CalibrationDate () const`

Return a read-only (constant) reference to the element.

**Returns:**

A constant reference to the element.



**9.57.4.6 void xsd::Record2Type::Comment (::std::auto\_ptr< [Comment\\_type](#) > *p*)**

Set the element value without copying.

**Parameters:**

*p* A new value to use.

This function will try to use the passed value directly instead of making a copy.

**9.57.4.7 void xsd::Record2Type::Comment (const [Comment\\_optional](#) & *x*)**

Set the element value.

**Parameters:**

*x* An optional container with the new value to set.

If the value is present in *x* then this function makes a copy of this value and sets it as the new value of the element. Otherwise the element container is set the 'not present' state.

**9.57.4.8 void xsd::Record2Type::Comment (const [Comment\\_type](#) & *x*)**

Set the element value.

**Parameters:**

*x* A new value to set.

This function makes a copy of its argument and sets it as the new value of the element.

**9.57.4.9 [Record2Type::Comment\\_optional](#) & xsd::Record2Type::Comment ()**

Return a read-write reference to the element container.

**Returns:**

A reference to the optional container.

**9.57.4.10 const [Record2Type::Comment\\_optional](#) & xsd::Record2Type::Comment () const**

Return a read-only (constant) reference to the element container.

**Returns:**

A constant reference to the optional container.

**9.57.4.11 void xsd::Record2Type::Creator (::std::auto\_ptr< Creator\_type > p)**

Set the element value without copying.

**Parameters:**

*p* A new value to use.

This function will try to use the passed value directly instead of making a copy.

**9.57.4.12 void xsd::Record2Type::Creator (const Creator\_optional & x)**

Set the element value.

**Parameters:**

*x* An optional container with the new value to set.

If the value is present in *x* then this function makes a copy of this value and sets it as the new value of the element. Otherwise the element container is set the 'not present' state.

**9.57.4.13 void xsd::Record2Type::Creator (const Creator\_type & x)**

Set the element value.

**Parameters:**

*x* A new value to set.

This function makes a copy of its argument and sets it as the new value of the element.

**9.57.4.14 Record2Type::Creator\_optional & xsd::Record2Type::Creator ()**

Return a read-write reference to the element container.

**Returns:**

A reference to the optional container.

**9.57.4.15 const Record2Type::Creator\_optional & xsd::Record2Type::Creator () const**

Return a read-only (constant) reference to the element container.

**Returns:**

A constant reference to the optional container.

**9.57.4.16 void xsd::Record2Type::Date (::std::auto\_ptr< [Date\\_type](#) > *p*)**

Set the element value without copying.

**Parameters:**

*p* A new value to use.

This function will try to use the passed value directly instead of making a copy.

**9.57.4.17 void xsd::Record2Type::Date (const [Date\\_type](#) & *x*)**

Set the element value.

**Parameters:**

*x* A new value to set.

This function makes a copy of its argument and sets it as the new value of the element.

**9.57.4.18 [Record2Type::Date\\_type](#) & xsd::Record2Type::Date ()**

Return a read-write reference to the element.

**Returns:**

A reference to the element.

**9.57.4.19 const [Record2Type::Date\\_type](#) & xsd::Record2Type::Date () const**

Return a read-only (constant) reference to the element.

**Returns:**

A constant reference to the element.

**9.57.4.20 void xsd::Record2Type::Instrument (::std::auto\_ptr< [Instrument\\_type](#) > *p*)**

Set the element value without copying.

**Parameters:**

*p* A new value to use.

This function will try to use the passed value directly instead of making a copy.

**9.57.4.21 void xsd::Record2Type::Instrument (const Instrument\_type & x)**

Set the element value.

**Parameters:**

*x* A new value to set.

This function makes a copy of its argument and sets it as the new value of the element.

**9.57.4.22 Record2Type::Instrument\_type & xsd::Record2Type::Instrument ()**

Return a read-write reference to the element.

**Returns:**

A reference to the element.

**9.57.4.23 const Record2Type::Instrument\_type & xsd::Record2Type::Instrument () const**

Return a read-only (constant) reference to the element.

**Returns:**

A constant reference to the element.

**9.57.4.24 void xsd::Record2Type::parse (::xsd::cxx::xml::dom::parser< wchar\_t > &, ::xml\_schema::flags) [protected]****9.57.4.25 void xsd::Record2Type::ProbingSystem (::std::auto\_ptr< Probing-System\_type > p)**

Set the element value without copying.

**Parameters:**

*p* A new value to use.

This function will try to use the passed value directly instead of making a copy.

**9.57.4.26 void xsd::Record2Type::ProbingSystem (const ProbingSystem\_type & x)**

Set the element value.

**Parameters:**

*x* A new value to set.

This function makes a copy of its argument and sets it as the new value of the element.

#### 9.57.4.27 [Record2Type::ProbingSystem\\_type](#) & xsd::Record2Type::ProbingSystem ()

Return a read-write reference to the element.

**Returns:**

A reference to the element.

#### 9.57.4.28 [const](#) [Record2Type::ProbingSystem\\_type](#) & xsd::Record2Type::ProbingSystem () [const](#)

Return a read-only (constant) reference to the element.

**Returns:**

A constant reference to the element.

### 9.57.5 Member Data Documentation

#### 9.57.5.1 [::xsd::cxx::tree::one< CalibrationDate\\_type >](#) [xsd::Record2Type::CalibrationDate\\_](#) [private]

#### 9.57.5.2 [Comment\\_optional](#) [xsd::Record2Type::Comment\\_](#) [private]

#### 9.57.5.3 [Creator\\_optional](#) [xsd::Record2Type::Creator\\_](#) [private]

#### 9.57.5.4 [::xsd::cxx::tree::one< Date\\_type >](#) [xsd::Record2Type::Date\\_](#) [private]

#### 9.57.5.5 [::xsd::cxx::tree::one< Instrument\\_type >](#) [xsd::Record2Type::Instrument\\_](#) [private]

#### 9.57.5.6 [::xsd::cxx::tree::one< ProbingSystem\\_type >](#) [xsd::Record2Type::ProbingSystem\\_](#) [private]

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

- S:/openGPS/ISO5436\_XML/trunk/src/ISO5436\_2\_XML/iso5436\_2\_xsd.hxx
- S:/openGPS/ISO5436\_XML/trunk/src/ISO5436\_2\_XML/iso5436\_2\_xsd.cxx

## 9.58 xsd::Record3Type Class Reference

```
#include <iso5436_2_xsd.hxx>
```

### 9.58.1 Detailed Description

Class corresponding to the Record3Type schema type.

Record 3 contains the measured data.

#### DataLink

Accessor and modifier functions for the DataLink optional element.

Link specification to an external binary data file.

- typedef ::xsd::cxx::tree::optional< [DataLink\\_type](#) > [DataLink\\_optional](#)  
*Element optional container type.*
- typedef ::xsd::cxx::tree::traits< [DataLink\\_type](#), wchar\_t > [DataLink\\_traits](#)  
*Element traits type.*
- typedef ::xsd::DataLinkType [DataLink\\_type](#)  
*Element type.*
- void [DataLink](#) (::std::auto\_ptr< [DataLink\\_type](#) > p)  
*Set the element value without copying.*
- void [DataLink](#) (const [DataLink\\_optional](#) &x)  
*Set the element value.*
- void [DataLink](#) (const [DataLink\\_type](#) &x)  
*Set the element value.*
- [DataLink\\_optional](#) & [DataLink](#) ()  
*Return a read-write reference to the element container.*
- const [DataLink\\_optional](#) & [DataLink](#) () const  
*Return a read-only (constant) reference to the element container.*

#### DataList

Accessor and modifier functions for the DataList optional element.

Data list is ordered like specified in DataOrder: Z-Index is empty (only one sample per pixel) X is fastest index, Y is slower, Z is slowest: (x1,y1),(x2,y1),(x3,y1),(x4,y1),(x1,y2)...

- typedef ::xsd::cxx::tree::optional< [DataList\\_type](#) > [DataList\\_optional](#)  
*Element optional container type.*
- typedef ::xsd::cxx::tree::traits< [DataList\\_type](#), wchar\_t > [DataList\\_traits](#)  
*Element traits type.*
- typedef ::xsd::DataListType [DataList\\_type](#)  
*Element type.*
- void [DataList](#) (::std::auto\_ptr< [DataList\\_type](#) > p)  
*Set the element value without copying.*
- void [DataList](#) (const [DataList\\_optional](#) &x)  
*Set the element value.*
- void [DataList](#) (const [DataList\\_type](#) &x)  
*Set the element value.*
- [DataList\\_optional](#) & [DataList](#) ()  
*Return a read-write reference to the element container.*
- const [DataList\\_optional](#) & [DataList](#) () const  
*Return a read-only (constant) reference to the element container.*

### ListDimension

Accessor and modifier functions for the ListDimension optional element.

A list does specify an unordered data set like a point cloud which does not contain topologic information.

- typedef ::xsd::cxx::tree::optional< [ListDimension\\_type](#) > [ListDimension\\_optional](#)  
*Element optional container type.*
- typedef ::xsd::cxx::tree::traits< [ListDimension\\_type](#), wchar\_t > [ListDimension\\_traits](#)  
*Element traits type.*
- typedef ::xml\_schema::unsigned\_long [ListDimension\\_type](#)  
*Element type.*
- void [ListDimension](#) (const [ListDimension\\_optional](#) &x)  
*Set the element value.*
- void [ListDimension](#) (const [ListDimension\\_type](#) &x)

*Set the element value.*

- [ListDimension\\_optional](#) & [ListDimension](#) ()

*Return a read-write reference to the element container.*

- const [ListDimension\\_optional](#) & [ListDimension](#) () const

*Return a read-only (constant) reference to the element container.*

## MatrixDimension

Accessor and modifier functions for the MatrixDimension optional element.

- typedef `::xsd::cxx::tree::optional< MatrixDimension\_type > MatrixDimension\_optional`

*Element optional container type.*

- typedef `::xsd::cxx::tree::traits< MatrixDimension\_type, wchar_t > MatrixDimension\_traits`

*Element traits type.*

- typedef `::xsd::MatrixDimensionType MatrixDimension\_type`

*Element type.*

- void [MatrixDimension](#) (::std::auto\_ptr< [MatrixDimension\\_type](#) > p)

*Set the element value without copying.*

- void [MatrixDimension](#) (const [MatrixDimension\\_optional](#) &x)

*Set the element value.*

- void [MatrixDimension](#) (const [MatrixDimension\\_type](#) &x)

*Set the element value.*

- [MatrixDimension\\_optional](#) & [MatrixDimension](#) ()

*Return a read-write reference to the element container.*

- const [MatrixDimension\\_optional](#) & [MatrixDimension](#) () const

*Return a read-only (constant) reference to the element container.*

## Constructors

- virtual [Record3Type](#) \* [\\_clone](#) (::xml\_schema::flags f=0,::xml\_schema::type \*c=0) const

*Copy the object polymorphically.*



- [Record3Type](#) (const [Record3Type](#) &x,::xml\_schema::flags f=0,::xml\_schema::type \*c=0)  
*Copy constructor.*
- [Record3Type](#) (const ::xercesc::DOMElement &e,::xml\_schema::flags f=0,::xml\_schema::type \*c=0)  
*Construct an instance from a DOM element.*
- [Record3Type](#) ()  
*Construct an instance from the ultimate base and initializers for required elements and attributes.*

### Protected Member Functions

- void [parse](#) (::xsd::cxx::xml::dom::parser< wchar\_t > &,::xml\_schema::flags)

### Private Attributes

- [DataLink\\_optional](#) [DataLink\\_](#)
- [DataList\\_optional](#) [DataList\\_](#)
- [ListDimension\\_optional](#) [ListDimension\\_](#)
- [MatrixDimension\\_optional](#) [MatrixDimension\\_](#)

## 9.58.2 Member Typedef Documentation

**9.58.2.1** typedef ::xsd::cxx::tree::optional< [DataLink\\_type](#) >  
[xsd::Record3Type::DataLink\\_optional](#)

Element optional container type.

**9.58.2.2** typedef ::xsd::cxx::tree::traits< [DataLink\\_type](#), wchar\_t >  
[xsd::Record3Type::DataLink\\_traits](#)

Element traits type.

**9.58.2.3** typedef ::xsd::DataLinkType [xsd::Record3Type::DataLink\\_type](#)

Element type.

**9.58.2.4** typedef ::xsd::cxx::tree::optional< [DataList\\_type](#) >  
[xsd::Record3Type::DataList\\_optional](#)

Element optional container type.

**9.58.2.5** typedef ::xsd::cxx::tree::traits< [DataList\\_type](#), wchar\_t >  
[xsd::Record3Type::DataList\\_traits](#)

Element traits type.

**9.58.2.6** typedef ::xsd::DataListType xsd::Record3Type::DataList\_type

Element type.

**9.58.2.7** typedef ::xsd::cxx::tree::optional< [ListDimension\\_type](#) >  
[xsd::Record3Type::ListDimension\\_optional](#)

Element optional container type.

**9.58.2.8** typedef ::xsd::cxx::tree::traits< [ListDimension\\_type](#), wchar\_t >  
[xsd::Record3Type::ListDimension\\_traits](#)

Element traits type.

**9.58.2.9** typedef ::xml\_schema::unsigned\_long xsd::Record3Type::List-  
[Dimension\\_type](#)

Element type.

**9.58.2.10** typedef ::xsd::cxx::tree::optional< [MatrixDimension\\_type](#) >  
[xsd::Record3Type::MatrixDimension\\_optional](#)

Element optional container type.

**9.58.2.11** typedef ::xsd::cxx::tree::traits< [MatrixDimension\\_type](#), wchar\_t >  
[xsd::Record3Type::MatrixDimension\\_traits](#)

Element traits type.

**9.58.2.12** typedef ::xsd::MatrixDimensionType xsd::Record3Type::Matrix-  
[Dimension\\_type](#)

Element type.

### 9.58.3 Constructor & Destructor Documentation

#### 9.58.3.1 xsd::Record3Type::Record3Type ()

Construct an instance from the ultimate base and initializers for required elements and attributes.

### 9.58.3.2 xsd::Record3Type::Record3Type (const ::xercesc::DOMElement & *e*, ::xml\_schema::flags *f* = 0, ::xml\_schema::type \* *c* = 0)

Construct an instance from a DOM element.

#### Parameters:

- e* A DOM element to extract the data from.
- f* Flags to construct the new instance with.
- c* A pointer to the object that will contain the new instance.

### 9.58.3.3 xsd::Record3Type::Record3Type (const Record3Type & *x*, ::xml\_schema::flags *f* = 0, ::xml\_schema::type \* *c* = 0)

Copy constructor.

#### Parameters:

- x* An instance to make a copy of.
- f* Flags to construct the copy with.
- c* A pointer to the object that will contain the copy.

For polymorphic object models use the `_clone` function instead.

## 9.58.4 Member Function Documentation

### 9.58.4.1 Record3Type \* xsd::Record3Type::\_clone (::xml\_schema::flags *f* = 0, ::xml\_schema::type \* *c* = 0) const [virtual]

Copy the object polymorphically.

#### Parameters:

- f* Flags to construct the copy with.
- c* A pointer to the object that will contain the copy.

#### Returns:

A pointer to the dynamically allocated copy.

This function ensures that the dynamic type of an instance is used for copying and should be used for polymorphic object models instead of the copy constructor.

### 9.58.4.2 void xsd::Record3Type::DataLink (::std::auto\_ptr< DataLink\_type > *p*)

Set the element value without copying.

#### Parameters:

- p* A new value to use.

This function will try to use the passed value directly instead of making a copy.

**9.58.4.3 void xsd::Record3Type::DataLink (const DataLink\_optional & x)**

Set the element value.

**Parameters:**

*x* An optional container with the new value to set.

If the value is present in *x* then this function makes a copy of this value and sets it as the new value of the element. Otherwise the element container is set the 'not present' state.

**9.58.4.4 void xsd::Record3Type::DataLink (const DataLink\_type & x)**

Set the element value.

**Parameters:**

*x* A new value to set.

This function makes a copy of its argument and sets it as the new value of the element.

**9.58.4.5 Record3Type::DataLink\_optional & xsd::Record3Type::DataLink ()**

Return a read-write reference to the element container.

**Returns:**

A reference to the optional container.

**9.58.4.6 const Record3Type::DataLink\_optional & xsd::Record3Type::DataLink () const**

Return a read-only (constant) reference to the element container.

**Returns:**

A constant reference to the optional container.

**9.58.4.7 void xsd::Record3Type::DataList (::std::auto\_ptr< DataList\_type > p)**

Set the element value without copying.

**Parameters:**

*p* A new value to use.

This function will try to use the passed value directly instead of making a copy.

**9.58.4.8 void xsd::Record3Type::DataList (const DataList\_optional & x)**

Set the element value.

**Parameters:**

*x* An optional container with the new value to set.

If the value is present in *x* then this function makes a copy of this value and sets it as the new value of the element. Otherwise the element container is set the 'not present' state.

**9.58.4.9 void xsd::Record3Type::DataList (const DataList\_type & x)**

Set the element value.

**Parameters:**

*x* A new value to set.

This function makes a copy of its argument and sets it as the new value of the element.

**9.58.4.10 Record3Type::DataList\_optional & xsd::Record3Type::DataList ()**

Return a read-write reference to the element container.

**Returns:**

A reference to the optional container.

**9.58.4.11 const Record3Type::DataList\_optional & xsd::Record3Type::DataList () const**

Return a read-only (constant) reference to the element container.

**Returns:**

A constant reference to the optional container.

**9.58.4.12 void xsd::Record3Type::ListDimension (const ListDimension\_optional & x)**

Set the element value.

**Parameters:**

*x* An optional container with the new value to set.

If the value is present in *x* then this function makes a copy of this value and sets it as the new value of the element. Otherwise the element container is set the 'not present' state.

**9.58.4.13 void xsd::Record3Type::ListDimension (const ListDimension\_type & x)**

Set the element value.

**Parameters:**

*x* A new value to set.

This function makes a copy of its argument and sets it as the new value of the element.

**9.58.4.14 Record3Type::ListDimension\_optional & xsd::Record3Type::ListDimension ()**

Return a read-write reference to the element container.

**Returns:**

A reference to the optional container.

**9.58.4.15 const Record3Type::ListDimension\_optional & xsd::Record3Type::ListDimension () const**

Return a read-only (constant) reference to the element container.

**Returns:**

A constant reference to the optional container.

**9.58.4.16 void xsd::Record3Type::MatrixDimension (::std::auto\_ptr< MatrixDimension\_type > p)**

Set the element value without copying.

**Parameters:**

*p* A new value to use.

This function will try to use the passed value directly instead of making a copy.

**9.58.4.17 void xsd::Record3Type::MatrixDimension (const MatrixDimension\_optional & x)**

Set the element value.

**Parameters:**

*x* An optional container with the new value to set.

If the value is present in *x* then this function makes a copy of this value and sets it as the new value of the element. Otherwise the element container is set the 'not present' state.

#### 9.58.4.18 void xsd::Record3Type::MatrixDimension (const [MatrixDimension\\_type](#) & x)

Set the element value.

##### Parameters:

*x* A new value to set.

This function makes a copy of its argument and sets it as the new value of the element.

#### 9.58.4.19 [Record3Type::MatrixDimension\\_optional](#) & xsd::Record3Type::MatrixDimension ()

Return a read-write reference to the element container.

##### Returns:

A reference to the optional container.

#### 9.58.4.20 const [Record3Type::MatrixDimension\\_optional](#) & xsd::Record3Type::MatrixDimension () const

Return a read-only (constant) reference to the element container.

##### Returns:

A constant reference to the optional container.

#### 9.58.4.21 void xsd::Record3Type::parse (::xsd::cxx::xml::dom::parser< wchar\_t > &, [::xml\\_schema::flags](#)) [protected]

### 9.58.5 Member Data Documentation

#### 9.58.5.1 [DataLink\\_optional](#) xsd::Record3Type::DataLink\_ [private]

#### 9.58.5.2 [DataList\\_optional](#) xsd::Record3Type::DataList\_ [private]

#### 9.58.5.3 [ListDimension\\_optional](#) xsd::Record3Type::ListDimension\_ [private]

#### 9.58.5.4 [MatrixDimension\\_optional](#) xsd::Record3Type::MatrixDimension\_ [private]

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

- S:/openGPS/ISO5436\_XML/trunk/src/ISO5436\_2\_XML/[iso5436\\_2\\_xsd.hxx](#)
- S:/openGPS/ISO5436\_XML/trunk/src/ISO5436\_2\_XML/[iso5436\\_2\\_xsd.cxx](#)

## 9.59 xsd::Record4Type Class Reference

```
#include <iso5436_2_xsd.hxx>
```

### 9.59.1 Detailed Description

Class corresponding to the Record4Type schema type.

Record4 contains only the checksum of the xml file.

#### ChecksumFile

Accessor and modifier functions for the ChecksumFile required element.

An URI pointing to an external ascii file containing an MD5 digest with a 32 byte hexadecimal MD5Checksum of the whole XML-file and its filename as produced by the unix command "md5sum". The checksum can be calculated by the unix command "md5sum main.xml >md5checksum.hex" and checked by the command "md5sum -c md5checksum.hex". Default name of the checksum file is "md5checksum.hex".

- typedef ::xsd::cxx::tree::traits< [ChecksumFile\\_type](#), wchar\_t > [ChecksumFile\\_traits](#)  
*Element traits type.*
- typedef ::xml\_schema::string [ChecksumFile\\_type](#)  
*Element type.*
- void [ChecksumFile](#) (::std::auto\_ptr< [ChecksumFile\\_type](#) > p)  
*Set the element value without copying.*
- void [ChecksumFile](#) (const [ChecksumFile\\_type](#) &x)  
*Set the element value.*
- [ChecksumFile\\_type](#) & [ChecksumFile](#) ()  
*Return a read-write reference to the element.*
- const [ChecksumFile\\_type](#) & [ChecksumFile](#) () const  
*Return a read-only (constant) reference to the element.*

#### Constructors

- virtual [Record4Type](#) \* [\\_clone](#) (::xml\_schema::flags f=0,::xml\_schema::type \*c=0) const  
*Copy the object polymorphically.*



- `Record4Type` (const `Record4Type` &*x*, ::xml\_schema::flags *f*=0, ::xml\_schema::type \**c*=0)  
*Copy constructor.*
- `Record4Type` (const ::xercesc::DOMElement &*e*, ::xml\_schema::flags *f*=0, ::xml\_schema::type \**c*=0)  
*Construct an instance from a DOM element.*
- `Record4Type` (const `ChecksumFile_type` &)  
*Construct an instance from the ultimate base and initializers for required elements and attributes.*

### Protected Member Functions

- void `parse` (::xsd::cxx::xml::dom::parser< wchar\_t > &, ::xml\_schema::flags)

### Private Attributes

- ::xsd::cxx::tree::one< `ChecksumFile_type` > `ChecksumFile_`

## 9.59.2 Member Typedef Documentation

**9.59.2.1** typedef ::xsd::cxx::tree::traits< `ChecksumFile_type`, wchar\_t > `xsd::Record4Type::ChecksumFile_traits`

Element traits type.

**9.59.2.2** typedef ::xml\_schema::string `xsd::Record4Type::ChecksumFile_type`

Element type.

## 9.59.3 Constructor & Destructor Documentation

**9.59.3.1** `xsd::Record4Type::Record4Type` (const `ChecksumFile_type` &)

Construct an instance from the ultimate base and initializers for required elements and attributes.

**9.59.3.2** `xsd::Record4Type::Record4Type` (const ::xercesc::DOMElement & *e*, ::xml\_schema::flags *f*=0, ::xml\_schema::type \**c*=0)

Construct an instance from a DOM element.

### Parameters:

- e* A DOM element to extract the data from.

- f* Flags to construct the new instance with.
- c* A pointer to the object that will contain the new instance.

### 9.59.3.3 xsd::Record4Type::Record4Type (const Record4Type & *x*, ::xml\_schema::flags *f* = 0, ::xml\_schema::type \* *c* = 0)

Copy constructor.

#### Parameters:

- x* An instance to make a copy of.
- f* Flags to construct the copy with.
- c* A pointer to the object that will contain the copy.

For polymorphic object models use the `_clone` function instead.

## 9.59.4 Member Function Documentation

### 9.59.4.1 Record4Type \* xsd::Record4Type::\_clone (::xml\_schema::flags *f* = 0, ::xml\_schema::type \* *c* = 0) const [virtual]

Copy the object polymorphically.

#### Parameters:

- f* Flags to construct the copy with.
- c* A pointer to the object that will contain the copy.

#### Returns:

A pointer to the dynamically allocated copy.

This function ensures that the dynamic type of an instance is used for copying and should be used for polymorphic object models instead of the copy constructor.

### 9.59.4.2 void xsd::Record4Type::ChecksumFile (::std::auto\_ptr< Checksum-File\_type > *p*)

Set the element value without copying.

#### Parameters:

- p* A new value to use.

This function will try to use the passed value directly instead of making a copy.

**9.59.4.3 void xsd::Record4Type::ChecksumFile (const [ChecksumFile\\_type](#) & x)**

Set the element value.

**Parameters:**

*x* A new value to set.

This function makes a copy of its argument and sets it as the new value of the element.

**9.59.4.4 [Record4Type::ChecksumFile\\_type](#) & xsd::Record4Type::ChecksumFile ()**

Return a read-write reference to the element.

**Returns:**

A reference to the element.

**9.59.4.5 const [Record4Type::ChecksumFile\\_type](#) & xsd::Record4Type::ChecksumFile () const**

Return a read-only (constant) reference to the element.

**Returns:**

A constant reference to the element.

**9.59.4.6 void xsd::Record4Type::parse (::xsd::cxx::xml::dom::parser< wchar\_t > &, [::xml\\_schema::flags](#)) [protected]****9.59.5 Member Data Documentation****9.59.5.1 ::xsd::cxx::tree::one< [ChecksumFile\\_type](#) > xsd::Record4Type::ChecksumFile\_ [private]**

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

- S:/openGPS/ISO5436\_XML/trunk/src/ISO5436\_2\_XML/[iso5436\\_2\\_xsd.hxx](#)
- S:/openGPS/ISO5436\_XML/trunk/src/ISO5436\_2\_XML/[iso5436\\_2\\_xsd.cxx](#)

**9.60 xsd::RotationMatrixElementType Class Reference**

```
#include <iso5436_2_xsd.hxx>
```

### 9.60.1 Detailed Description

Class corresponding to the RotationMatrixElementType schema type.

An element of a pure rotation matrix is limited to a value range of [-1..1].

#### Constructors

- virtual [RotationMatrixElementType](#) \* [\\_clone](#) (::xml\_schema::flags f=0, ::xml\_schema::type \*c=0) const  
*Copy the object polymorphically.*
- [RotationMatrixElementType](#) (const [RotationMatrixElementType](#) &x, ::xml\_schema::flags f=0, ::xml\_schema::type \*c=0)  
*Copy constructor.*
- [RotationMatrixElementType](#) (const ::std::wstring &s, const ::xercesc::DOMElement \*e, ::xml\_schema::flags f=0, ::xml\_schema::type \*c=0)  
*Construct an instance from a string fragment.*
- [RotationMatrixElementType](#) (const ::xercesc::DOMAttr &a, ::xml\_schema::flags f=0, ::xml\_schema::type \*c=0)  
*Construct an instance from a DOM attribute.*
- [RotationMatrixElementType](#) (const ::xercesc::DOMElement &e, ::xml\_schema::flags f=0, ::xml\_schema::type \*c=0)  
*Construct an instance from a DOM element.*
- [RotationMatrixElementType](#) (const ::xml\_schema::double\_ &)  
*Construct an instance from the ultimate base and initializers for required elements and attributes.*

### 9.60.2 Constructor & Destructor Documentation

#### 9.60.2.1 xsd::RotationMatrixElementType::RotationMatrixElementType (const ::xml\_schema::double\_ &)

Construct an instance from the ultimate base and initializers for required elements and attributes.

#### 9.60.2.2 xsd::RotationMatrixElementType::RotationMatrixElementType (const ::xercesc::DOMElement & e, ::xml\_schema::flags f = 0, ::xml\_schema::type \* c = 0)

Construct an instance from a DOM element.

**Parameters:**

- e* A DOM element to extract the data from.
- f* Flags to construct the new instance with.
- c* A pointer to the object that will contain the new instance.

**9.60.2.3 xsd::RotationMatrixElementType::RotationMatrixElementType**  
 (const ::xercesc::DOMAttr & *a*, ::xml\_schema::flags *f* = 0, ::xml\_schema::type \* *c* = 0)

Construct an instance from a DOM attribute.

**Parameters:**

- a* A DOM attribute to extract the data from.
- f* Flags to construct the new instance with.
- c* A pointer to the object that will contain the new instance.

**9.60.2.4 xsd::RotationMatrixElementType::RotationMatrixElementType**  
 (const ::std::wstring & *s*, const ::xercesc::DOMElement \* *e*, ::xml\_schema::flags *f* = 0, ::xml\_schema::type \* *c* = 0)

Construct an instance from a string fragment.

**Parameters:**

- s* A string fragment to extract the data from.
- e* A DOM element containing the string fragment.
- f* Flags to construct the new instance with.
- c* A pointer to the object that will contain the new instance.

**9.60.2.5 xsd::RotationMatrixElementType::RotationMatrixElementType**  
 (const RotationMatrixElementType & *x*, ::xml\_schema::flags *f* = 0, ::xml\_schema::type \* *c* = 0)

Copy constructor.

**Parameters:**

- x* An instance to make a copy of.
- f* Flags to construct the copy with.
- c* A pointer to the object that will contain the copy.

For polymorphic object models use the `_clone` function instead.

### 9.60.3 Member Function Documentation

**9.60.3.1 RotationMatrixElementType \* xsd::RotationMatrixElementType::\_clone (::xml\_schema::flags *f* = 0, ::xml\_schema::type \* *c* = 0) const** [virtual]

Copy the object polymorphically.

#### Parameters:

- f* Flags to construct the copy with.
- c* A pointer to the object that will contain the copy.

#### Returns:

A pointer to the dynamically allocated copy.

This function ensures that the dynamic type of an instance is used for copying and should be used for polymorphic object models instead of the copy constructor.

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

- [S:/openGPS/ISO5436\\_XML/trunk/src/ISO5436\\_2\\_XML/iso5436\\_2\\_xsd.hxx](#)
- [S:/openGPS/ISO5436\\_XML/trunk/src/ISO5436\\_2\\_XML/iso5436\\_2\\_xsd.cxx](#)

## 9.61 xsd::RotationType Class Reference

```
#include <iso5436_2_xsd.hxx>
```

### 9.61.1 Detailed Description

Class corresponding to the RotationType schema type.

The optional transformation contains a 3D rotation matrix *R* with 3 by 3 elements that is used to rotate the data points in its final orientation. The full transformation consists of a rotation and a following translation that is taken from the [AxisDescriptionType.Offset](#) elements:  $Q = R \cdot P + T$  With *Q* being the final point, *P* the coordinate as specified in Record3, *R* the 3 by 3 rotation matrix and *T* the 3-element offset vector. The \* denotes a matrix product. The formula for the x coordinate is:  $Q_x = r11 \cdot P_x + r12 \cdot P_y + r13 \cdot P_z + T_x$ . The formula for the y coordinate is:  $Q_y = r21 \cdot P_x + r22 \cdot P_y + r23 \cdot P_z + T_y$ . The formula for the z coordinate is:  $Q_z = r31 \cdot P_x + r32 \cdot P_y + r33 \cdot P_z + T_z$ .

#### r11

Accessor and modifier functions for the r11 required element.

- `typedef ::xsd::cxx::tree::traits< r11\_type, wchar_t > r11\_traits`  
*Element traits type.*

- typedef ::xsd::RotationMatrixElementType r11\_type  
*Element type.*
- void r11 (::std::auto\_ptr< r11\_type > p)  
*Set the element value without copying.*
- void r11 (const r11\_type &x)  
*Set the element value.*
- r11\_type & r11 ()  
*Return a read-write reference to the element.*
- const r11\_type & r11 () const  
*Return a read-only (constant) reference to the element.*

## r12

Accessor and modifier functions for the r12 required element.

- typedef ::xsd::cxx::tree::traits< r12\_type, wchar\_t > r12\_traits  
*Element traits type.*
- typedef ::xsd::RotationMatrixElementType r12\_type  
*Element type.*
- void r12 (::std::auto\_ptr< r12\_type > p)  
*Set the element value without copying.*
- void r12 (const r12\_type &x)  
*Set the element value.*
- r12\_type & r12 ()  
*Return a read-write reference to the element.*
- const r12\_type & r12 () const  
*Return a read-only (constant) reference to the element.*

## r13

Accessor and modifier functions for the r13 required element.

- typedef ::xsd::cxx::tree::traits< r13\_type, wchar\_t > r13\_traits  
*Element traits type.*

- typedef [xsd::RotationMatrixElementType r13\\_type](#)  
*Element type.*
- void [r13](#) (::std::auto\_ptr< [r13\\_type](#) > p)  
*Set the element value without copying.*
- void [r13](#) (const [r13\\_type](#) &x)  
*Set the element value.*
- [r13\\_type](#) & [r13](#) ()  
*Return a read-write reference to the element.*
- const [r13\\_type](#) & [r13](#) () const  
*Return a read-only (constant) reference to the element.*

**r21**

Accessor and modifier functions for the r21 required element.

- typedef [xsd::cxx::tree::traits< r21\\_type, wchar\\_t > r21\\_traits](#)  
*Element traits type.*
- typedef [xsd::RotationMatrixElementType r21\\_type](#)  
*Element type.*
- void [r21](#) (::std::auto\_ptr< [r21\\_type](#) > p)  
*Set the element value without copying.*
- void [r21](#) (const [r21\\_type](#) &x)  
*Set the element value.*
- [r21\\_type](#) & [r21](#) ()  
*Return a read-write reference to the element.*
- const [r21\\_type](#) & [r21](#) () const  
*Return a read-only (constant) reference to the element.*

**r22**

Accessor and modifier functions for the r22 required element.

- typedef [xsd::cxx::tree::traits< r22\\_type, wchar\\_t > r22\\_traits](#)



*Element traits type.*

- typedef ::xsd::RotationMatrixElementType r22\_type  
*Element type.*
- void r22 (::std::auto\_ptr< r22\_type > p)  
*Set the element value without copying.*
- void r22 (const r22\_type &x)  
*Set the element value.*
- r22\_type & r22 ()  
*Return a read-write reference to the element.*
- const r22\_type & r22 () const  
*Return a read-only (constant) reference to the element.*

## r23

Accessor and modifier functions for the r23 required element.

- typedef ::xsd::cxx::tree::traits< r23\_type, wchar\_t > r23\_traits  
*Element traits type.*
- typedef ::xsd::RotationMatrixElementType r23\_type  
*Element type.*
- void r23 (::std::auto\_ptr< r23\_type > p)  
*Set the element value without copying.*
- void r23 (const r23\_type &x)  
*Set the element value.*
- r23\_type & r23 ()  
*Return a read-write reference to the element.*
- const r23\_type & r23 () const  
*Return a read-only (constant) reference to the element.*

## r31

Accessor and modifier functions for the r31 required element.

- typedef ::xsd::cxx::tree::traits< r31\_type, wchar\_t > r31\_traits

*Element traits type.*

- typedef ::xsd::RotationMatrixElementType r31\_type  
*Element type.*
- void r31 (::std::auto\_ptr< r31\_type > p)  
*Set the element value without copying.*
- void r31 (const r31\_type &x)  
*Set the element value.*
- r31\_type & r31 ()  
*Return a read-write reference to the element.*
- const r31\_type & r31 () const  
*Return a read-only (constant) reference to the element.*

## r32

Accessor and modifier functions for the r32 required element.

- typedef ::xsd::cxx::tree::traits< r32\_type, wchar\_t > r32\_traits  
*Element traits type.*
- typedef ::xsd::RotationMatrixElementType r32\_type  
*Element type.*
- void r32 (::std::auto\_ptr< r32\_type > p)  
*Set the element value without copying.*
- void r32 (const r32\_type &x)  
*Set the element value.*
- r32\_type & r32 ()  
*Return a read-write reference to the element.*
- const r32\_type & r32 () const  
*Return a read-only (constant) reference to the element.*

## r33

Accessor and modifier functions for the r33 required element.

- typedef ::xsd::cxx::tree::traits< r33\_type, wchar\_t > r33\_traits

*Element traits type.*

- typedef `::xsd::RotationMatrixElementType r33_type`  
*Element type.*
- void `r33 (::std::auto_ptr< r33_type > p)`  
*Set the element value without copying.*
- void `r33 (const r33_type &x)`  
*Set the element value.*
- `r33_type & r33 ()`  
*Return a read-write reference to the element.*
- `const r33_type & r33 () const`  
*Return a read-only (constant) reference to the element.*

### Constructors

- virtual `RotationType * _clone (::xml_schema::flags f=0, ::xml_schema::type *c=0) const`  
*Copy the object polymorphically.*
- `RotationType (const RotationType &x, ::xml_schema::flags f=0, ::xml_schema::type *c=0)`  
*Copy constructor.*
- `RotationType (const ::xercesc::DOMElement &e, ::xml_schema::flags f=0, ::xml_schema::type *c=0)`  
*Construct an instance from a DOM element.*
- `RotationType (const r11_type &, const r12_type &, const r13_type &, const r21_type &, const r22_type &, const r23_type &, const r31_type &, const r32_type &, const r33_type &)`  
*Construct an instance from the ultimate base and initializers for required elements and attributes.*

### Protected Member Functions

- void `parse (::xsd::cxx::xml::dom::parser< wchar_t > &, ::xml_schema::flags)`

**Private Attributes**

- `::xsd::cxx::tree::one< r11_type > r11_`
- `::xsd::cxx::tree::one< r12_type > r12_`
- `::xsd::cxx::tree::one< r13_type > r13_`
- `::xsd::cxx::tree::one< r21_type > r21_`
- `::xsd::cxx::tree::one< r22_type > r22_`
- `::xsd::cxx::tree::one< r23_type > r23_`
- `::xsd::cxx::tree::one< r31_type > r31_`
- `::xsd::cxx::tree::one< r32_type > r32_`
- `::xsd::cxx::tree::one< r33_type > r33_`

**9.61.2 Member Typedef Documentation****9.61.2.1 typedef ::xsd::cxx::tree::traits< r11\_type, wchar\_t > xsd::RotationType::r11\_traits**

Element traits type.

**9.61.2.2 typedef ::xsd::RotationMatrixElementType xsd::RotationType::r11\_type**

Element type.

**9.61.2.3 typedef ::xsd::cxx::tree::traits< r12\_type, wchar\_t > xsd::RotationType::r12\_traits**

Element traits type.

**9.61.2.4 typedef ::xsd::RotationMatrixElementType xsd::RotationType::r12\_type**

Element type.

**9.61.2.5 typedef ::xsd::cxx::tree::traits< r13\_type, wchar\_t > xsd::RotationType::r13\_traits**

Element traits type.

**9.61.2.6 typedef ::xsd::RotationMatrixElementType xsd::RotationType::r13\_type**

Element type.

**9.61.2.7 typedef ::xsd::cxx::tree::traits< r21\_type, wchar\_t > xsd::RotationType::r21\_traits**

Element traits type.

**9.61.2.8** `typedef ::xsd::RotationMatrixElementType xsd::RotationType::r21_type`

Element type.

**9.61.2.9** `typedef ::xsd::cxx::tree::traits< r22_type, wchar_t > xsd::RotationType::r22_traits`

Element traits type.

**9.61.2.10** `typedef ::xsd::RotationMatrixElementType xsd::RotationType::r22_type`

Element type.

**9.61.2.11** `typedef ::xsd::cxx::tree::traits< r23_type, wchar_t > xsd::RotationType::r23_traits`

Element traits type.

**9.61.2.12** `typedef ::xsd::RotationMatrixElementType xsd::RotationType::r23_type`

Element type.

**9.61.2.13** `typedef ::xsd::cxx::tree::traits< r31_type, wchar_t > xsd::RotationType::r31_traits`

Element traits type.

**9.61.2.14** `typedef ::xsd::RotationMatrixElementType xsd::RotationType::r31_type`

Element type.

**9.61.2.15** `typedef ::xsd::cxx::tree::traits< r32_type, wchar_t > xsd::RotationType::r32_traits`

Element traits type.

**9.61.2.16** `typedef ::xsd::RotationMatrixElementType xsd::RotationType::r32_type`

Element type.

**9.61.2.17** `typedef ::xsd::cxx::tree::traits< r33_type, wchar_t > xsd::RotationType::r33_traits`

Element traits type.

### 9.61.2.18 typedef ::xsd::RotationMatrixElementType xsd::RotationType::r33\_type

Element type.

## 9.61.3 Constructor & Destructor Documentation

### 9.61.3.1 xsd::RotationType::RotationType (const r11\_type &, const r12\_type &, const r13\_type &, const r21\_type &, const r22\_type &, const r23\_type &, const r31\_type &, const r32\_type &, const r33\_type &)

Construct an instance from the ultimate base and initializers for required elements and attributes.

### 9.61.3.2 xsd::RotationType::RotationType (const ::xercesc::DOMElement & *e*, ::xml\_schema::flags *f* = 0, ::xml\_schema::type \* *c* = 0)

Construct an instance from a DOM element.

#### Parameters:

- e* A DOM element to extract the data from.
- f* Flags to construct the new instance with.
- c* A pointer to the object that will contain the new instance.

### 9.61.3.3 xsd::RotationType::RotationType (const RotationType & *x*, ::xml\_schema::flags *f* = 0, ::xml\_schema::type \* *c* = 0)

Copy constructor.

#### Parameters:

- x* An instance to make a copy of.
- f* Flags to construct the copy with.
- c* A pointer to the object that will contain the copy.

For polymorphic object models use the `_clone` function instead.

## 9.61.4 Member Function Documentation

### 9.61.4.1 RotationType \* xsd::RotationType::\_clone (::xml\_schema::flags *f* = 0, ::xml\_schema::type \* *c* = 0) const [virtual]

Copy the object polymorphically.

#### Parameters:

- f* Flags to construct the copy with.

*c* A pointer to the object that will contain the copy.

**Returns:**

A pointer to the dynamically allocated copy.

This function ensures that the dynamic type of an instance is used for copying and should be used for polymorphic object models instead of the copy constructor.

**9.61.4.2** void xsd::RotationType::parse (::xsd::cxx::xml::dom::parser<wchar\_t> &, ::xml\_schema::flags) [protected]

**9.61.4.3** void xsd::RotationType::r11 (::std::auto\_ptr< r11\_type > p)

Set the element value without copying.

**Parameters:**

*p* A new value to use.

This function will try to use the passed value directly instead of making a copy.

**9.61.4.4** void xsd::RotationType::r11 (const r11\_type & x)

Set the element value.

**Parameters:**

*x* A new value to set.

This function makes a copy of its argument and sets it as the new value of the element.

**9.61.4.5** r11\_type& xsd::RotationType::r11 ()

Return a read-write reference to the element.

**Returns:**

A reference to the element.

**9.61.4.6** const r11\_type& xsd::RotationType::r11 () const

Return a read-only (constant) reference to the element.

**Returns:**

A constant reference to the element.

**9.61.4.7 void xsd::RotationType::r12 (::std::auto\_ptr< r12\_type > p)**

Set the element value without copying.

**Parameters:**

*p* A new value to use.

This function will try to use the passed value directly instead of making a copy.

**9.61.4.8 void xsd::RotationType::r12 (const r12\_type & x)**

Set the element value.

**Parameters:**

*x* A new value to set.

This function makes a copy of its argument and sets it as the new value of the element.

**9.61.4.9 r12\_type& xsd::RotationType::r12 ()**

Return a read-write reference to the element.

**Returns:**

A reference to the element.

**9.61.4.10 const r12\_type& xsd::RotationType::r12 () const**

Return a read-only (constant) reference to the element.

**Returns:**

A constant reference to the element.

**9.61.4.11 void xsd::RotationType::r13 (::std::auto\_ptr< r13\_type > p)**

Set the element value without copying.

**Parameters:**

*p* A new value to use.

This function will try to use the passed value directly instead of making a copy.



**9.61.4.12 void xsd::RotationType::r13 (const r13\_type & x)**

Set the element value.

**Parameters:**

*x* A new value to set.

This function makes a copy of its argument and sets it as the new value of the element.

**9.61.4.13 r13\_type& xsd::RotationType::r13 ()**

Return a read-write reference to the element.

**Returns:**

A reference to the element.

**9.61.4.14 const r13\_type& xsd::RotationType::r13 () const**

Return a read-only (constant) reference to the element.

**Returns:**

A constant reference to the element.

**9.61.4.15 void xsd::RotationType::r21 (::std::auto\_ptr< r21\_type > p)**

Set the element value without copying.

**Parameters:**

*p* A new value to use.

This function will try to use the passed value directly instead of making a copy.

**9.61.4.16 void xsd::RotationType::r21 (const r21\_type & x)**

Set the element value.

**Parameters:**

*x* A new value to set.

This function makes a copy of its argument and sets it as the new value of the element.

**9.61.4.17** `r21_type& xsd::RotationType::r21 ()`

Return a read-write reference to the element.

**Returns:**

A reference to the element.

**9.61.4.18** `const r21_type& xsd::RotationType::r21 () const`

Return a read-only (constant) reference to the element.

**Returns:**

A constant reference to the element.

**9.61.4.19** `void xsd::RotationType::r22 (::std::auto_ptr< r22_type > p)`

Set the element value without copying.

**Parameters:**

*p* A new value to use.

This function will try to use the passed value directly instead of making a copy.

**9.61.4.20** `void xsd::RotationType::r22 (const r22_type & x)`

Set the element value.

**Parameters:**

*x* A new value to set.

This function makes a copy of its argument and sets it as the new value of the element.

**9.61.4.21** `r22_type& xsd::RotationType::r22 ()`

Return a read-write reference to the element.

**Returns:**

A reference to the element.

**9.61.4.22** `const r22_type& xsd::RotationType::r22 () const`

Return a read-only (constant) reference to the element.

**Returns:**

A constant reference to the element.

**9.61.4.23 void xsd::RotationType::r23 (::std::auto\_ptr< r23\_type > p)**

Set the element value without copying.

**Parameters:**

*p* A new value to use.

This function will try to use the passed value directly instead of making a copy.

**9.61.4.24 void xsd::RotationType::r23 (const r23\_type & x)**

Set the element value.

**Parameters:**

*x* A new value to set.

This function makes a copy of its argument and sets it as the new value of the element.

**9.61.4.25 r23\_type& xsd::RotationType::r23 ()**

Return a read-write reference to the element.

**Returns:**

A reference to the element.

**9.61.4.26 const r23\_type& xsd::RotationType::r23 () const**

Return a read-only (constant) reference to the element.

**Returns:**

A constant reference to the element.

**9.61.4.27 void xsd::RotationType::r31 (::std::auto\_ptr< r31\_type > p)**

Set the element value without copying.

**Parameters:**

*p* A new value to use.

This function will try to use the passed value directly instead of making a copy.

**9.61.4.28 void xsd::RotationType::r31 (const r31\_type & x)**

Set the element value.

**Parameters:**

*x* A new value to set.

This function makes a copy of its argument and sets it as the new value of the element.

**9.61.4.29 r31\_type& xsd::RotationType::r31 ()**

Return a read-write reference to the element.

**Returns:**

A reference to the element.

**9.61.4.30 const r31\_type& xsd::RotationType::r31 () const**

Return a read-only (constant) reference to the element.

**Returns:**

A constant reference to the element.

**9.61.4.31 void xsd::RotationType::r32 (::std::auto\_ptr< r32\_type > p)**

Set the element value without copying.

**Parameters:**

*p* A new value to use.

This function will try to use the passed value directly instead of making a copy.

**9.61.4.32 void xsd::RotationType::r32 (const r32\_type & x)**

Set the element value.

**Parameters:**

*x* A new value to set.

This function makes a copy of its argument and sets it as the new value of the element.

**9.61.4.33** `r32_type& xsd::RotationType::r32 ()`

Return a read-write reference to the element.

**Returns:**

A reference to the element.

**9.61.4.34** `const r32_type& xsd::RotationType::r32 () const`

Return a read-only (constant) reference to the element.

**Returns:**

A constant reference to the element.

**9.61.4.35** `void xsd::RotationType::r33 (::std::auto_ptr< r33_type > p)`

Set the element value without copying.

**Parameters:**

*p* A new value to use.

This function will try to use the passed value directly instead of making a copy.

**9.61.4.36** `void xsd::RotationType::r33 (const r33_type & x)`

Set the element value.

**Parameters:**

*x* A new value to set.

This function makes a copy of its argument and sets it as the new value of the element.

**9.61.4.37** `r33_type& xsd::RotationType::r33 ()`

Return a read-write reference to the element.

**Returns:**

A reference to the element.

**9.61.4.38** `const r33_type& xsd::RotationType::r33 () const`

Return a read-only (constant) reference to the element.

**Returns:**

A constant reference to the element.

### 9.61.5 Member Data Documentation

**9.61.5.1** `::xsd::cxx::tree::one< r11\_type > xsd::RotationType::r11\_`  
[private]

**9.61.5.2** `::xsd::cxx::tree::one< r12\_type > xsd::RotationType::r12\_`  
[private]

**9.61.5.3** `::xsd::cxx::tree::one< r13\_type > xsd::RotationType::r13\_`  
[private]

**9.61.5.4** `::xsd::cxx::tree::one< r21\_type > xsd::RotationType::r21\_`  
[private]

**9.61.5.5** `::xsd::cxx::tree::one< r22\_type > xsd::RotationType::r22\_`  
[private]

**9.61.5.6** `::xsd::cxx::tree::one< r23\_type > xsd::RotationType::r23\_`  
[private]

**9.61.5.7** `::xsd::cxx::tree::one< r31\_type > xsd::RotationType::r31\_`  
[private]

**9.61.5.8** `::xsd::cxx::tree::one< r32\_type > xsd::RotationType::r32\_`  
[private]

**9.61.5.9** `::xsd::cxx::tree::one< r33\_type > xsd::RotationType::r33\_`  
[private]

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

- [S:/openGPS/ISO5436\\_XML/trunk/src/ISO5436\\_2\\_XML/iso5436\\_2\\_xsd.hxx](#)
- [S:/openGPS/ISO5436\\_XML/trunk/src/ISO5436\\_2\\_XML/iso5436\\_2\\_xsd.cxx](#)

## 9.62 xsd::Type Class Reference

```
#include <iso5436_2_xsd.hxx>
```

### 9.62.1 Detailed Description

Enumeration class corresponding to the Type schema type.

## Public Types

- [Contacting](#)
- [NonContacting](#)
- enum [value](#) { [Contacting](#), [NonContacting](#) }

*Underlying enum type.*

## Public Member Functions

- virtual [Type](#) \* [\\_clone](#) (::xml\_schema::flags f=0, ::xml\_schema::type \*c=0) const  
*Copy the object polymorphically.*
- virtual [operator value](#) () const  
*Implicit conversion operator to the underlying enum value.*
- [Type](#) & [operator=](#) (value v)  
*Assign the underlying enum value.*
- [Type](#) (const [Type](#) &x, ::xml\_schema::flags f=0, ::xml\_schema::type \*c=0)  
*Copy constructor.*
- [Type](#) (const ::std::wstring &s, const ::xercesc::DOMElement \*e, ::xml\_schema::flags f=0, ::xml\_schema::type \*c=0)  
*Construct an instance from a string fragment.*
- [Type](#) (const ::xercesc::DOMAttr &a, ::xml\_schema::flags f=0, ::xml\_schema::type \*c=0)  
*Construct an instance from a DOM attribute.*
- [Type](#) (const ::xercesc::DOMElement &e, ::xml\_schema::flags f=0, ::xml\_schema::type \*c=0)  
*Construct an instance from a DOM element.*
- [Type](#) (const ::xml\_schema::token &v)  
*Construct an instance from the base value.*
- [Type](#) (value v)  
*Construct an instance from the underlying enum value.*

## Static Public Attributes

- static const [value](#) [\\_xsd\\_Type\\_indexes\\_](#) [2]
- static const wchar\_t \*const [\\_xsd\\_Type\\_literals\\_](#) [2]

### Protected Member Functions

- [value \\_xsd\\_Type\\_convert](#) () const

### 9.62.2 Member Enumeration Documentation

#### 9.62.2.1 enum [xsd::Type::value](#)

Underlying enum type.

#### Enumerator:

*Contacting*

*NonContacting*

### 9.62.3 Constructor & Destructor Documentation

#### 9.62.3.1 xsd::Type::Type (value *v*)

Construct an instance from the underlying enum value.

#### Parameters:

*v* A enum value.

#### 9.62.3.2 xsd::Type::Type (const ::xml\_schema::token & *v*)

Construct an instance from the base value.

#### Parameters:

*v* A base value.

#### 9.62.3.3 xsd::Type::Type (const ::xercesc::DOMElement & *e*, ::xml\_schema::flags *f* = 0, ::xml\_schema::type \* *c* = 0)

Construct an instance from a DOM element.

#### Parameters:

*e* A DOM element to extract the data from.

*f* Flags to construct the new instance with.

*c* A pointer to the object that will contain the new instance.



**9.62.3.4** `xsd::Type::Type (const ::xercesc::DOMAttr & a, ::xml_schema::flags f = 0, ::xml_schema::type * c = 0)`

Construct an instance from a DOM attribute.

**Parameters:**

- a* A DOM attribute to extract the data from.
- f* Flags to construct the new instance with.
- c* A pointer to the object that will contain the new instance.

**9.62.3.5** `xsd::Type::Type (const ::std::wstring & s, const ::xercesc::DOMElement * e, ::xml_schema::flags f = 0, ::xml_schema::type * c = 0)`

Construct an instance from a string fragment.

**Parameters:**

- s* A string fragment to extract the data from.
- e* A DOM element containing the string fragment.
- f* Flags to construct the new instance with.
- c* A pointer to the object that will contain the new instance.

**9.62.3.6** `xsd::Type::Type (const Type & x, ::xml_schema::flags f = 0, ::xml_schema::type * c = 0)`

Copy constructor.

**Parameters:**

- x* An instance to make a copy of.
- f* Flags to construct the copy with.
- c* A pointer to the object that will contain the copy.

For polymorphic object models use the `_clone` function instead.

## 9.62.4 Member Function Documentation

**9.62.4.1** `Type * xsd::Type::_clone (::xml_schema::flags f = 0, ::xml_schema::type * c = 0) const [virtual]`

Copy the object polymorphically.

**Parameters:**

- f* Flags to construct the copy with.

*c* A pointer to the object that will contain the copy.

#### Returns:

A pointer to the dynamically allocated copy.

This function ensures that the dynamic type of an instance is used for copying and should be used for polymorphic object models instead of the copy constructor.

#### 9.62.4.2 `Type::value xsd::Type::_xsd_Type_convert () const` [protected]

#### 9.62.4.3 `virtual xsd::Type::operator value () const` [inline, virtual]

Implicit conversion operator to the underlying enum value.

#### Returns:

A enum value.

#### 9.62.4.4 `Type& xsd::Type::operator= (value v)`

Assign the underlying enum value.

#### Parameters:

*v* A enum value.

#### Returns:

A reference to the instance.

### 9.62.5 Member Data Documentation

#### 9.62.5.1 `const value xsd::Type::_xsd_Type_indexes [2]` [static]

#### 9.62.5.2 `const wchar_t* const xsd::Type::_xsd_Type_literals [2]` [static]

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

- S:/openGPS/ISO5436\_XML/trunk/src/ISO5436\_2\_XML/[iso5436\\_2\\_xsd.hxx](#)
- S:/openGPS/ISO5436\_XML/trunk/src/ISO5436\_2\_XML/[iso5436\\_2\\_xsd.cxx](#)

## 9.63 OpenGPS::ValidBuffer Class Reference

```
#include <valid_buffer.hxx>
```

### Public Member Functions

- virtual OGPS\_Boolean [Allocate](#) (const unsigned int size)
- virtual OGPS\_Boolean [IsAllocated](#) () const
- virtual OGPS\_Boolean [IsValid](#) (const unsigned int index) const
- virtual OGPS\_Boolean [Read](#) (std::basic\_istream< OpenGPS::UnsignedByte > &stream, const unsigned int pointCount)
- virtual OGPS\_Boolean [SetValid](#) (const unsigned int index, const OGPS\_Boolean value)
- [ValidBuffer](#) ()
- virtual OGPS\_Boolean [Write](#) (std::ostream &stream)
- [~ValidBuffer](#) ()

### Protected Member Functions

- virtual OGPS\_Boolean [AllocateRaw](#) (const unsigned int rawSize)
- virtual void [Reset](#) ()

### Private Attributes

- unsigned long [m\\_RawSize](#)
- unsigned long [m\\_Size](#)
- OpenGPS::UnsignedBytePtr [m\\_Valid](#)

## 9.63.1 Constructor & Destructor Documentation

### 9.63.1.1 ValidBuffer::ValidBuffer ()

### 9.63.1.2 ValidBuffer::~~ValidBuffer ()

## 9.63.2 Member Function Documentation

### 9.63.2.1 OGPS\_Boolean ValidBuffer::Allocate (const unsigned int *size*) [virtual]

### 9.63.2.2 OGPS\_Boolean ValidBuffer::AllocateRaw (const unsigned int *rawSize*) [protected, virtual]

### 9.63.2.3 OGPS\_Boolean ValidBuffer::IsAllocated () const [virtual]

### 9.63.2.4 OGPS\_Boolean ValidBuffer::IsValid (const unsigned int *index*) const [virtual]

**9.63.2.5** `OGPS_Boolean ValidBuffer::Read (std::basic_istream< OpenGPS::UnsignedByte > & stream, const unsigned int pointCount)` [virtual]

**9.63.2.6** `void ValidBuffer::Reset ()` [protected, virtual]

**9.63.2.7** `OGPS_Boolean ValidBuffer::SetValid (const unsigned int index, const OGPS_Boolean value)` [virtual]

**9.63.2.8** `OGPS_Boolean ValidBuffer::Write (std::ostream & stream)` [virtual]

### 9.63.3 Member Data Documentation

**9.63.3.1** `unsigned long OpenGPS::ValidBuffer::m_RawSize` [private]

**9.63.3.2** `unsigned long OpenGPS::ValidBuffer::m_Size` [private]

**9.63.3.3** `OpenGPS::UnsignedBytePtr OpenGPS::ValidBuffer::m_Valid` [private]

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

- S:/openGPS/ISO5436\_XML/trunk/src/ISO5436\_2\_XML/cxx/valid\_buffer.hxx
- S:/openGPS/ISO5436\_XML/trunk/src/ISO5436\_2\_XML/cxx/valid\_buffer.cxx

## 9.64 OpenGPS::VectorBuffer Class Reference

```
#include <vector_buffer.hxx>
```

### Public Member Functions

- virtual `PointVectorAutoPtr GetPointVectorProxy (const PointVectorProxy-Context &context)` const
- virtual `ValidBuffer *const GetValid ()` const
- virtual `PointBuffer *const GetX ()` const
- virtual `PointBuffer *const GetY ()` const
- virtual `PointBuffer *const GetZ ()` const
- virtual void `SetValid (ValidBuffer *value)`
- virtual void `SetX (PointBuffer *value)`
- virtual void `SetY (PointBuffer *value)`
- virtual void `SetZ (PointBuffer *value)`
- `VectorBuffer ()`
- `~VectorBuffer ()`

### Private Attributes

- [ValidBuffer](#) \* [m\\_Valid](#)
- [PointBuffer](#) \* [m\\_X](#)
- [PointBuffer](#) \* [m\\_Y](#)
- [PointBuffer](#) \* [m\\_Z](#)

### 9.64.1 Constructor & Destructor Documentation

#### 9.64.1.1 [VectorBuffer::VectorBuffer](#) ()

#### 9.64.1.2 [VectorBuffer::~~VectorBuffer](#) ()

### 9.64.2 Member Function Documentation

#### 9.64.2.1 [PointVectorAutoPtr](#) [VectorBuffer::GetPointVectorProxy](#) (const [PointVectorProxyContext](#) & *context*) const [virtual]

#### 9.64.2.2 [ValidBuffer](#) \*const [VectorBuffer::GetValid](#) () const [virtual]

#### 9.64.2.3 [PointBuffer](#) \*const [VectorBuffer::GetX](#) () const [virtual]

#### 9.64.2.4 [PointBuffer](#) \*const [VectorBuffer::GetY](#) () const [virtual]

#### 9.64.2.5 [PointBuffer](#) \*const [VectorBuffer::GetZ](#) () const [virtual]

#### 9.64.2.6 void [VectorBuffer::SetValid](#) ([ValidBuffer](#) \* *value*) [virtual]

#### 9.64.2.7 void [VectorBuffer::SetX](#) ([PointBuffer](#) \* *value*) [virtual]

#### 9.64.2.8 void [VectorBuffer::SetY](#) ([PointBuffer](#) \* *value*) [virtual]

#### 9.64.2.9 void [VectorBuffer::SetZ](#) ([PointBuffer](#) \* *value*) [virtual]

### 9.64.3 Member Data Documentation

#### 9.64.3.1 [ValidBuffer](#)\* [OpenGPS::VectorBuffer::m\\_Valid](#) [private]

#### 9.64.3.2 [PointBuffer](#)\* [OpenGPS::VectorBuffer::m\\_X](#) [private]

9.64.3.3 [PointBuffer\\*](#) [OpenGPS::VectorBuffer::m\\_Y](#) [private]

9.64.3.4 [PointBuffer\\*](#) [OpenGPS::VectorBuffer::m\\_Z](#) [private]

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

- S:/openGPS/ISO5436\_XML/trunk/src/ISO5436\_2\_XML/cxx/[vector\\_buffer.hxx](#)
- S:/openGPS/ISO5436\_XML/trunk/src/ISO5436\_2\_XML/cxx/[vector\\_buffer.cxx](#)

## 9.65 OpenGPS::VectorBufferBuilder Class Reference

```
#include <vector_buffer_builder.hxx>
```

### Public Member Functions

- virtual OGPS\_Boolean [BuildBuffer](#) ()
- virtual OGPS\_Boolean [BuildValid](#) (const unsigned long size)
- virtual OGPS\_Boolean [BuildX](#) (const OGPS\_DataPointType dataType, const unsigned long size)
- virtual OGPS\_Boolean [BuildY](#) (const OGPS\_DataPointType dataType, const unsigned long size)
- virtual OGPS\_Boolean [BuildZ](#) (const OGPS\_DataPointType dataType, const unsigned long size)
- virtual [VectorBuffer](#) \*const [GetBuffer](#) () const
- [VectorBufferBuilder](#) ()
- [~VectorBufferBuilder](#) ()

### Private Member Functions

- [PointBuffer](#) \* [CreatePointBuffer](#) (const OGPS\_DataPointType dataType, const unsigned long size, OGPS\_Boolean \*const retval) const

### Private Attributes

- [VectorBuffer](#) \* [m\\_Buffer](#)

## 9.65.1 Constructor & Destructor Documentation

9.65.1.1 [VectorBufferBuilder::VectorBufferBuilder](#) ()

9.65.1.2 [VectorBufferBuilder::~~VectorBufferBuilder](#) ()

### 9.65.2 Member Function Documentation

**9.65.2.1** `OGPS_Boolean VectorBufferBuilder::BuildBuffer ()` [virtual]

**9.65.2.2** `OGPS_Boolean VectorBufferBuilder::BuildValid (const unsigned long size)` [virtual]

**9.65.2.3** `OGPS_Boolean VectorBufferBuilder::BuildX (const OGPS_DataPointType dataType, const unsigned long size)` [virtual]

**9.65.2.4** `OGPS_Boolean VectorBufferBuilder::BuildY (const OGPS_DataPointType dataType, const unsigned long size)` [virtual]

**9.65.2.5** `OGPS_Boolean VectorBufferBuilder::BuildZ (const OGPS_DataPointType dataType, const unsigned long size)` [virtual]

**9.65.2.6** `PointBuffer * VectorBufferBuilder::CreatePointBuffer (const OGPS_DataPointType dataType, const unsigned long size, OGPS_Boolean *const retval) const` [private]

**9.65.2.7** `VectorBuffer *const VectorBufferBuilder::GetBuffer () const` [virtual]

### 9.65.3 Member Data Documentation

**9.65.3.1** `VectorBuffer*` [OpenGPS::VectorBufferBuilder::m\\_Buffer](#) [private]

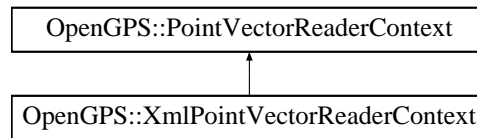
The documentation for this class was generated from the following files:

- [S:/openGPS/ISO5436\\_XML/trunk/src/ISO5436\\_2\\_XML/cxx/vector\\_buffer\\_builder.hxx](#)
- [S:/openGPS/ISO5436\\_XML/trunk/src/ISO5436\\_2\\_XML/cxx/vector\\_buffer\\_builder.cxx](#)

## 9.66 OpenGPS::XmlPointVectorReaderContext Class Reference

```
#include <xml_point_vector_reader_context.hxx>
```

Inheritance diagram for OpenGPS::XmlPointVectorReaderContext::



## Public Types

- typedef [xsd::DataListType::Datum\\_sequence](#) [StringList](#)

## Public Member Functions

- virtual OGPS\_Boolean [IsValid](#) () const
- virtual OGPS\_Boolean [MoveNext](#) ()
- virtual OGPS\_Boolean [Read](#) (double \*value)
- virtual OGPS\_Boolean [Read](#) (float \*value)
- virtual OGPS\_Boolean [Read](#) (int \*value)
- virtual OGPS\_Boolean [Read](#) (short \*value)
- virtual OGPS\_Boolean [Skip](#) ()
- [XmlPointVectorReaderContext](#) (const [StringList](#) \*pointVectorList)
- virtual [~XmlPointVectorReaderContext](#) ()

## Protected Member Functions

- virtual OGPS\_Boolean [IsGood](#) () const

## Private Member Functions

- void [Reset](#) ()
- void [Set](#) (const String &buf)

## Private Attributes

- unsigned long [m\\_Next](#)
- const [StringList](#) \* [m\\_PointVectorList](#)
- [PointVectorInputStringStream](#) \* [m\\_Stream](#)

### 9.66.1 Member Typedef Documentation

#### 9.66.1.1 typedef [xsd::DataListType::Datum\\_sequence](#) [OpenGPS::XmlPointVectorReaderContext::StringList](#)



## 9.66.2 Constructor & Destructor Documentation

**9.66.2.1** `XmlPointVectorReaderContext::XmlPointVectorReaderContext (const StringList * pointVectorList)`

**9.66.2.2** `XmlPointVectorReaderContext::~~XmlPointVectorReaderContext ()`  
[virtual]

## 9.66.3 Member Function Documentation

**9.66.3.1** `OGPS_Boolean XmlPointVectorReaderContext::IsGood () const`  
[protected, virtual]

**9.66.3.2** `OGPS_Boolean XmlPointVectorReaderContext::IsValid () const`  
[virtual]

Implements [OpenGPS::PointVectorReaderContext](#).

**9.66.3.3** `OGPS_Boolean XmlPointVectorReaderContext::MoveNext ()`  
[virtual]

Implements [OpenGPS::PointVectorReaderContext](#).

**9.66.3.4** `OGPS_Boolean XmlPointVectorReaderContext::Read (double * value)`  
[virtual]

Implements [OpenGPS::PointVectorReaderContext](#).

**9.66.3.5** `OGPS_Boolean XmlPointVectorReaderContext::Read (float * value)`  
[virtual]

Implements [OpenGPS::PointVectorReaderContext](#).

**9.66.3.6** `OGPS_Boolean XmlPointVectorReaderContext::Read (int * value)`  
[virtual]

Implements [OpenGPS::PointVectorReaderContext](#).

**9.66.3.7** `OGPS_Boolean XmlPointVectorReaderContext::Read (short * value)`  
[virtual]

Implements [OpenGPS::PointVectorReaderContext](#).

**9.66.3.8** `void XmlPointVectorReaderContext::Reset ()` [private]

**9.66.3.9** void XmlPointVectorReaderContext::Set (const String & buf)  
[private]

**9.66.3.10** OGPS\_Boolean XmlPointVectorReaderContext::Skip ()  
[virtual]

Implements [OpenGPS::PointVectorReaderContext](#).

## 9.66.4 Member Data Documentation

**9.66.4.1** unsigned long OpenGPS::XmlPointVectorReaderContext::m\_Next  
[private]

**9.66.4.2** const StringList\* OpenGPS::XmlPointVectorReaderContext::m\_  
PointVectorList [private]

**9.66.4.3** PointVectorInputStream\* OpenGPS::XmlPointVectorReader-  
Context::m\_Stream [private]

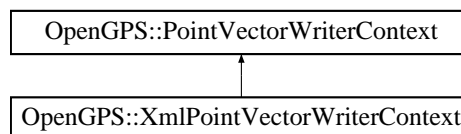
The documentation for this class was generated from the following files:

- S:/openGPS/ISO5436\_XML/trunk/src/ISO5436\_2\_XML/cxx/xml\_point\_  
vector\_reader\_context.hxx
- S:/openGPS/ISO5436\_XML/trunk/src/ISO5436\_2\_XML/cxx/xml\_point\_  
vector\_reader\_context.cxx

## 9.67 OpenGPS::XmlPointVectorWriterContext Class Reference

```
#include <xml_point_vector_writer_context.hxx>
```

Inheritance diagram for OpenGPS::XmlPointVectorWriterContext::



### Public Types

- typedef [xsd::DataListType::Datum\\_sequence](#) StringList

### Public Member Functions

- virtual OGPS\_Boolean [MoveNext](#) ()

- virtual OGPS\_Boolean [Skip](#) ()
- virtual OGPS\_Boolean [Write](#) (const double \*value)
- virtual OGPS\_Boolean [Write](#) (const float \*value)
- virtual OGPS\_Boolean [Write](#) (const int \*value)
- virtual OGPS\_Boolean [Write](#) (const short \*value)
- [XmlPointVectorWriterContext](#) ([StringList](#) \*pointVectorList)
- virtual [~XmlPointVectorWriterContext](#) ()

### Protected Member Functions

- virtual void [AppendSeparator](#) ()
- virtual OGPS\_Boolean [IsGood](#) () const

### Private Member Functions

- String [Get](#) () const
- void [Reset](#) ()

### Private Attributes

- OGPS\_Boolean [m\\_NeedsSeparator](#)
- [StringList](#) \* [m\\_PointVectorList](#)
- [PointVectorOutputStream](#) \* [m\\_Stream](#)

## 9.67.1 Member Typedef Documentation

**9.67.1.1** typedef [xsd::DataListType::Datum\\_sequence](#) [OpenGPS::XmlPointVectorWriterContext::StringList](#)

## 9.67.2 Constructor & Destructor Documentation

**9.67.2.1** [XmlPointVectorWriterContext::XmlPointVectorWriterContext](#) ([StringList](#) \*pointVectorList)

**9.67.2.2** [XmlPointVectorWriterContext::~~XmlPointVectorWriterContext](#) ()  
[virtual]

## 9.67.3 Member Function Documentation

**9.67.3.1** void [XmlPointVectorWriterContext::AppendSeparator](#) ()  
[protected, virtual]

**9.67.3.2** String [XmlPointVectorWriterContext::Get](#) () const [private]

**9.67.3.3** `OGPS_Boolean XmlPointVectorWriterContext::IsGood () const`  
[protected, virtual]

Implements [OpenGPS::PointVectorWriterContext](#).

**9.67.3.4** `OGPS_Boolean XmlPointVectorWriterContext::MoveNext ()`  
[virtual]

Implements [OpenGPS::PointVectorWriterContext](#).

**9.67.3.5** `void XmlPointVectorWriterContext::Reset ()` [private]

**9.67.3.6** `OGPS_Boolean XmlPointVectorWriterContext::Skip ()` [virtual]

Implements [OpenGPS::PointVectorWriterContext](#).

**9.67.3.7** `OGPS_Boolean XmlPointVectorWriterContext::Write (const double *  
value)` [virtual]

Implements [OpenGPS::PointVectorWriterContext](#).

**9.67.3.8** `OGPS_Boolean XmlPointVectorWriterContext::Write (const float *  
value)` [virtual]

Implements [OpenGPS::PointVectorWriterContext](#).

**9.67.3.9** `OGPS_Boolean XmlPointVectorWriterContext::Write (const int *  
value)` [virtual]

Implements [OpenGPS::PointVectorWriterContext](#).

**9.67.3.10** `OGPS_Boolean XmlPointVectorWriterContext::Write (const short *  
value)` [virtual]

Implements [OpenGPS::PointVectorWriterContext](#).

## 9.67.4 Member Data Documentation

**9.67.4.1** `OGPS_Boolean OpenGPS::XmlPointVectorWriterContext::m_Needs-  
Separator` [private]

**9.67.4.2** `StringList* OpenGPS::XmlPointVectorWriterContext::m_Point-  
VectorList` [private]

### 9.67.4.3 PointVectorOutputStream\* OpenGPS::XmlPointVectorWriter-Context::m\_Stream [private]

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

- S:/openGPS/ISO5436\_XML/trunk/src/ISO5436\_2\_XML/cxx/xml\_point\_vector\_writer\_context.hxx
- S:/openGPS/ISO5436\_XML/trunk/src/ISO5436\_2\_XML/cxx/xml\_point\_vector\_writer\_context.cxx

## 9.68 OpenGPS::ZipOutputStream Class Reference

```
#include <zip_stream_buffer.hxx>
```

### Public Member Functions

- [ZipOutputStream](#) ([ZipStreamBuffer](#) &buffer)
- [~ZipOutputStream](#) ()

### Private Types

- typedef std::ostream [BaseType](#)

### 9.68.1 Member Typedef Documentation

**9.68.1.1** typedef std::ostream [OpenGPS::ZipOutputStream::BaseType](#) [private]

### 9.68.2 Constructor & Destructor Documentation

**9.68.2.1** [ZipOutputStream::ZipOutputStream](#) ([ZipStreamBuffer](#) & *buffer*)

**9.68.2.2** [ZipOutputStream::~~ZipOutputStream](#) ()

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

- S:/openGPS/ISO5436\_XML/trunk/src/ISO5436\_2\_XML/cxx/zip\_stream\_buffer.hxx
- S:/openGPS/ISO5436\_XML/trunk/src/ISO5436\_2\_XML/cxx/zip\_stream\_buffer.cxx

## 9.69 OpenGPS::ZipStreamBuffer Class Reference

```
#include <zip_stream_buffer.hxx>
```

### Public Member Functions

- [ZipStreamBuffer](#) (zipFile handle)
- [~ZipStreamBuffer](#) ()

### Protected Member Functions

- virtual std::streamsize [xspn](#) (const char \*s, std::streamsize n)

### Private Types

- typedef std::streambuf [BaseType](#)

### Private Attributes

- zipFile [m\\_Handle](#)

#### 9.69.1 Member Typedef Documentation

**9.69.1.1** typedef std::streambuf [OpenGPS::ZipStreamBuffer::BaseType](#)  
[private]

#### 9.69.2 Constructor & Destructor Documentation

**9.69.2.1** ZipStreamBuffer::ZipStreamBuffer (zipFile *handle*)

**9.69.2.2** ZipStreamBuffer::~~ZipStreamBuffer ()

#### 9.69.3 Member Function Documentation

**9.69.3.1** std::streamsize ZipStreamBuffer::xspn (const char \* s, std::streamsize n) [protected, virtual]

#### 9.69.4 Member Data Documentation

**9.69.4.1** zipFile [OpenGPS::ZipStreamBuffer::m\\_Handle](#) [private]

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

- S:/openGPS/ISO5436\_XML/trunk/src/ISO5436\_2\_XML/cxx/[zip\\_stream\\_buffer.hxx](#)
- S:/openGPS/ISO5436\_XML/trunk/src/ISO5436\_2\_XML/cxx/[zip\\_stream\\_buffer.cxx](#)

## 10 openGPS ISO 5436-2 XML File Documentation

### 10.1 S:/openGPS/ISO5436\_XML/trunk/src/Doxygen.cpp File Reference

#### 10.1.1 Detailed Description

Title page of documentation, no source code.

#### Namespaces

- namespace [std](#)

### 10.2 S:/openGPS/ISO5436\_XML/trunk/src/ISO5436\_2\_XML/c/data\_point.cxx File Reference

```
#include <opengps/data_point.h>
#include "data_point.hxx"
#include "../cxx/data_point_impl.hxx"
```

#### Functions

- `OGPS_DataPointType` [ogps\\_GetDataType](#) (const [OGPS\\_DataPointPtr](#) *dataPoint*)
- `double` [ogps\\_GetDouble](#) (const [OGPS\\_DataPointPtr](#) *dataPoint*)
- `float` [ogps\\_GetFloat](#) (const [OGPS\\_DataPointPtr](#) *dataPoint*)
- `short` [ogps\\_GetInt16](#) (const [OGPS\\_DataPointPtr](#) *dataPoint*)
- `int` [ogps\\_GetInt32](#) (const [OGPS\\_DataPointPtr](#) *dataPoint*)
- `void` [ogps\\_SetDouble](#) ([OGPS\\_DataPointPtr](#) *const dataPoint*, const double value)
- `void` [ogps\\_SetFloat](#) ([OGPS\\_DataPointPtr](#) *const dataPoint*, const float value)
- `void` [ogps\\_SetInt16](#) ([OGPS\\_DataPointPtr](#) *const dataPoint*, const short value)
- `void` [ogps\\_SetInt32](#) ([OGPS\\_DataPointPtr](#) *const dataPoint*, const int value)

#### 10.2.1 Function Documentation

**10.2.1.1** `OGPS_DataPointType` [ogps\\_GetDataType](#) (const [OGPS\\_DataPointPtr](#) *dataPoint*)

**10.2.1.2** `double` [ogps\\_GetDouble](#) (const [OGPS\\_DataPointPtr](#) *dataPoint*)

**10.2.1.3** `float` [ogps\\_GetFloat](#) (const [OGPS\\_DataPointPtr](#) *dataPoint*)

**10.2.1.4** `short` [ogps\\_GetInt16](#) (const [OGPS\\_DataPointPtr](#) *dataPoint*)

10.2.1.5 int ogps\_GetInt32 (const [OGPS\\_DataPointPtr](#) *dataPoint*)

10.2.1.6 void ogps\_SetDouble ([OGPS\\_DataPointPtr](#) const *dataPoint*, const double *value*)

10.2.1.7 void ogps\_SetFloat ([OGPS\\_DataPointPtr](#) const *dataPoint*, const float *value*)

10.2.1.8 void ogps\_SetInt16 ([OGPS\\_DataPointPtr](#) const *dataPoint*, const short *value*)

10.2.1.9 void ogps\_SetInt32 ([OGPS\\_DataPointPtr](#) const *dataPoint*, const int *value*)

## 10.3 S:/openGPS/ISO5436\_XML/trunk/src/ISO5436\_2\_XML/c/data\_point.hxx File Reference

### Namespaces

- namespace [OpenGPS](#)

### Classes

- struct [\\_OGPS\\_DATA\\_POINT](#)

### Typedefs

- typedef [\\_OGPS\\_DATA\\_POINT](#) [OGPS\\_DataPoint](#)
- typedef [\\_OGPS\\_DATA\\_POINT](#) \* [OGPS\\_DataPointPtr](#)

#### 10.3.1 Typedef Documentation

10.3.1.1 typedef struct [\\_OGPS\\_DATA\\_POINT](#) [OGPS\\_DataPoint](#)

10.3.1.2 typedef struct [\\_OGPS\\_DATA\\_POINT](#) \* [OGPS\\_DataPointPtr](#)

## 10.4 S:/openGPS/ISO5436\_XML/trunk/src/ISO5436\_2\_XML/c/iso5436\_2\_handle.cxx File Reference

```
#include <opengps/iso5436_2_handle.h>
```

```
#include <opengps/iso5436_2_handle.hxx>
```



## 10.4

S:/openGPS/ISO5436\_XML/trunk/src/ISO5436\_2\_XML/c/iso5436\_2\_handle.cxx  
File Reference 256

```
#include "iso5436_2_handle.hxx"
#include "point_iterator.hxx"
#include "point_vector.hxx"
#include "../cxx/iso5436_2_container.hxx"
#include "../cxx/stdafx.hxx"
```

### Functions

- OGPS\_Boolean [ogps\\_CloseISO5436\\_2](#) (OGPS\_ISO5436\_2Handle \*handle)
- [OGPS\\_ISO5436\\_2Handle ogps\\_CreateListISO5436\\_2](#) (const OGPS\_Character \*file, const OGPS\_Character \*temp, const [xsd::Record1Type](#) &record1, const [xsd::Record2Type](#) &record2, const OGPS\_Boolean useBinaryData)
- [OGPS\\_ISO5436\\_2Handle ogps\\_CreateMatrixISO5436\\_2](#) (const OGPS\_Character \*file, const OGPS\_Character \*temp, const [xsd::Record1Type](#) &record1, const [xsd::Record2Type](#) &record2, const [xsd::MatrixDimensionType](#) &matrix, const OGPS\_Boolean useBinaryData)
- [OGPS\\_PointIteratorPtr ogps\\_CreateNextPointIterator](#) (const OGPS\_ISO5436\_2Handle handle)
- [OGPS\\_PointIteratorPtr ogps\\_CreatePrevPointIterator](#) (const OGPS\_ISO5436\_2Handle handle)
- [xsd::ISO5436\\_2Type \\* ogps\\_GetDocument](#) (const OGPS\_ISO5436\_2Handle handle)
- OGPS\_Boolean [ogps\\_GetListCoord](#) (const [OGPS\\_ISO5436\\_2Handle](#) handle, const unsigned long index, double \*x, double \*y, double \*z)
- OGPS\_Boolean [ogps\\_GetListPoint](#) (const [OGPS\\_ISO5436\\_2Handle](#) handle, const unsigned long index, [OGPS\\_PointVectorPtr](#) const vector)
- OGPS\_Boolean [ogps\\_GetMatrixCoord](#) (const [OGPS\\_ISO5436\\_2Handle](#) handle, const unsigned long u, const unsigned long v, const unsigned long w, double \*x, double \*y, double \*z)
- OGPS\_Boolean [ogps\\_GetMatrixPoint](#) (const [OGPS\\_ISO5436\\_2Handle](#) handle, const unsigned long u, const unsigned long v, const unsigned long w, [OGPS\\_PointVectorPtr](#) const vector)
- OGPS\_Boolean [ogps\\_IsMatrixCoordValid](#) (const [OGPS\\_ISO5436\\_2Handle](#) handle, const unsigned long u, const unsigned long v, const unsigned long w)
- [OGPS\\_ISO5436\\_2Handle ogps\\_OpenISO5436\\_2](#) (const OGPS\_Character \*file, const OGPS\_Character \*temp, const OGPS\_Boolean readOnly)
- OGPS\_Boolean [ogps\\_SetListPoint](#) (const [OGPS\\_ISO5436\\_2Handle](#) handle, const unsigned long index, const [OGPS\\_PointVectorPtr](#) vector)
- OGPS\_Boolean [ogps\\_SetMatrixPoint](#) (const [OGPS\\_ISO5436\\_2Handle](#) handle, const unsigned long u, const unsigned long v, const unsigned long w, const [OGPS\\_PointVectorPtr](#) vector)
- OGPS\_Boolean [ogps\\_WriteISO5436\\_2](#) (const [OGPS\\_ISO5436\\_2Handle](#) handle)

### 10.4.1 Function Documentation

10.4.1.1 **OGPS\_Boolean** **ogps\_CloseISO5436\_2** (**OGPS\_ISO5436\_2Handle** \* *handle*)

10.4.1.2 **OGPS\_ISO5436\_2Handle** **ogps\_CreateListISO5436\_2** (const **OGPS\_**Character \* *file*, const **OGPS\_**Character \* *temp*, const **xsd::Record1Type** & *record1*, const **xsd::Record2Type** & *record2*, const **OGPS\_Boolean** *useBinaryData*)

10.4.1.3 **OGPS\_ISO5436\_2Handle** **ogps\_CreateMatrixISO5436\_2** (const **OGPS\_**Character \* *file*, const **OGPS\_**Character \* *temp*, const **xsd::Record1Type** & *record1*, const **xsd::Record2Type** & *record2*, const **xsd::MatrixDimensionType** & *matrix*, const **OGPS\_Boolean** *useBinaryData*)

10.4.1.4 **OGPS\_PointIteratorPtr** **ogps\_CreateNextPointIterator** (const **OGPS\_**ISO5436\_2Handle *handle*)

10.4.1.5 **OGPS\_PointIteratorPtr** **ogps\_CreatePrevPointIterator** (const **OGPS\_**ISO5436\_2Handle *handle*)

10.4.1.6 **xsd::ISO5436\_2Type\*** **ogps\_GetDocument** (const **OGPS\_ISO5436\_**2Handle *handle*)

10.4.1.7 **OGPS\_Boolean** **ogps\_GetListCoord** (const **OGPS\_ISO5436\_2Handle** *handle*, const unsigned long *index*, double \* *x*, double \* *y*, double \* *z*)

10.4.1.8 **OGPS\_Boolean** **ogps\_GetListPoint** (const **OGPS\_ISO5436\_2Handle** *handle*, const unsigned long *index*, **OGPS\_PointVectorPtr** const *vector*)

10.4.1.9 **OGPS\_Boolean** **ogps\_GetMatrixCoord** (const **OGPS\_ISO5436\_2Handle** *handle*, const unsigned long *u*, const unsigned long *v*, const unsigned long *w*, double \* *x*, double \* *y*, double \* *z*)

10.4.1.10 **OGPS\_Boolean** **ogps\_GetMatrixPoint** (const **OGPS\_ISO5436\_2Handle** *handle*, const unsigned long *u*, const unsigned long *v*, const unsigned long *w*, **OGPS\_PointVectorPtr** const *vector*)

10.4.1.11 **OGPS\_Boolean** **ogps\_IsMatrixCoordValid** (const **OGPS\_ISO5436\_2Handle** *handle*, const unsigned long *u*, const unsigned long *v*, const unsigned long *w*)

## 10.5

S:/openGPS/ISO5436\_XML/trunk/src/ISO5436\_2\_XML/c/iso5436\_2\_handle.hxx  
File Reference 258

10.4.1.12 [OGPS\\_ISO5436\\_2Handle](#) ogps\_OpenISO5436\_2 (const [OGPS\\_](#)  
[Character](#) \**file*, const [OGPS\\_](#)[Character](#) \**temp*, const [OGPS\\_](#)[Boolean](#) *readOnly*)

10.4.1.13 [OGPS\\_](#)[Boolean](#) ogps\_SetListPoint (const [OGPS\\_ISO5436\\_2Handle](#)  
*handle*, const unsigned long *index*, const [OGPS\\_](#)[PointVectorPtr](#) *vector*)

10.4.1.14 [OGPS\\_](#)[Boolean](#) ogps\_SetMatrixPoint (const [OGPS\\_ISO5436\\_2Handle](#) *handle*, const unsigned long *u*, const unsigned long *v*, const unsigned long *w*, const [OGPS\\_](#)[PointVectorPtr](#) *vector*)

10.4.1.15 [OGPS\\_](#)[Boolean](#) ogps\_WriteISO5436\_2 (const [OGPS\\_ISO5436\\_2Handle](#) *handle*)

## 10.5 S:/openGPS/ISO5436\_XML/trunk/src/ISO5436\_2\_XML/c/iso5436\_2\_handle.hxx File Reference

### Namespaces

- namespace [OpenGPS](#)

### Classes

- struct [\\_OGPS\\_ISO5436\\_2\\_HANDLE](#)

### Typedefs

- typedef [\\_OGPS\\_ISO5436\\_2\\_HANDLE](#) [OGPS\\_ISO5436\\_2](#)
- typedef [\\_OGPS\\_ISO5436\\_2\\_HANDLE](#) \* [OGPS\\_ISO5436\\_2Handle](#)

### 10.5.1 Typedef Documentation

10.5.1.1 typedef struct [\\_OGPS\\_ISO5436\\_2\\_HANDLE](#) [OGPS\\_ISO5436\\_2](#)

10.5.1.2 typedef struct [\\_OGPS\\_ISO5436\\_2\\_HANDLE](#) \* [OGPS\\_ISO5436\\_2Handle](#)

## 10.6 S:/openGPS/ISO5436\_XML/trunk/src/ISO5436\_2\_XML/c/point\_iterator.cxx File Reference

```
#include <opengps/point_iterator.h>
#include "../cxx/point_iterator_impl.hxx"
```

## 10.6

S:/openGPS/ISO5436\_XML/trunk/src/ISO5436\_2\_XML/c/point\_iterator.cxx

File Reference

259

```
#include "point_iterator.hxx"
```

```
#include "point_vector.hxx"
```

### Functions

- OGPS\_Boolean [ogps\\_CreateNextPoint](#) (OGPS\_PointIteratorPtr const iterator)
- void [ogps\\_FreePointIterator](#) (OGPS\_PointIteratorPtr \*iterator)
- OGPS\_Boolean [ogps\\_GetCurrentPoint](#) (const OGPS\_PointIteratorPtr iterator, OGPS\_PointVectorPtr const vector)
- OGPS\_Boolean [ogps\\_GetListPosition](#) (const OGPS\_PointIteratorPtr iterator, unsigned long \*index)
- OGPS\_Boolean [ogps\\_GetMatrixPosition](#) (const OGPS\_PointIteratorPtr iterator, unsigned long \*u, unsigned long \*v, unsigned long \*w)
- OGPS\_Boolean [ogps\\_HasNextPoint](#) (const OGPS\_PointIteratorPtr iterator)
- OGPS\_Boolean [ogps\\_HasPrevPoint](#) (const OGPS\_PointIteratorPtr iterator)
- OGPS\_Boolean [ogps\\_MoveNextPoint](#) (OGPS\_PointIteratorPtr const iterator)
- OGPS\_Boolean [ogps\\_MovePrevPoint](#) (OGPS\_PointIteratorPtr const iterator)
- void [ogps\\_ResetNextPointIterator](#) (OGPS\_PointIteratorPtr const iterator)
- void [ogps\\_ResetPrevPointIterator](#) (OGPS\_PointIteratorPtr const iterator)
- OGPS\_Boolean [ogps\\_SetCurrentPoint](#) (const OGPS\_PointIteratorPtr iterator, const OGPS\_PointVectorPtr vector)

### 10.6.1 Function Documentation

**10.6.1.1** OGPS\_Boolean [ogps\\_CreateNextPoint](#) ([OGPS\\_PointIteratorPtr](#) const *iterator*)

**10.6.1.2** void [ogps\\_FreePointIterator](#) ([OGPS\\_PointIteratorPtr](#) \* *iterator*)

**10.6.1.3** OGPS\_Boolean [ogps\\_GetCurrentPoint](#) (const [OGPS\\_PointIteratorPtr](#) *iterator*, [OGPS\\_PointVectorPtr](#) const *vector*)

**10.6.1.4** OGPS\_Boolean [ogps\\_GetListPosition](#) (const [OGPS\\_PointIteratorPtr](#) *iterator*, unsigned long \* *index*)

**10.6.1.5** OGPS\_Boolean [ogps\\_GetMatrixPosition](#) (const [OGPS\\_PointIteratorPtr](#) *iterator*, unsigned long \* *u*, unsigned long \* *v*, unsigned long \* *w*)

**10.6.1.6** OGPS\_Boolean [ogps\\_HasNextPoint](#) (const [OGPS\\_PointIteratorPtr](#) *iterator*)

**10.6.1.7** OGPS\_Boolean [ogps\\_HasPrevPoint](#) (const [OGPS\\_PointIteratorPtr](#) *iterator*)

## 10.7

S:/openGPS/ISO5436\_XML/trunk/src/ISO5436\_2\_XML/c/point\_iterator.hxx

File Reference

260

**10.6.1.8** OGPS\_Boolean ogps\_MoveNextPoint ([OGPS\\_PointIteratorPtr](#) const *iterator*)

**10.6.1.9** OGPS\_Boolean ogps\_MovePrevPoint ([OGPS\\_PointIteratorPtr](#) const *iterator*)

**10.6.1.10** void ogps\_ResetNextPointIterator ([OGPS\\_PointIteratorPtr](#) const *iterator*)

**10.6.1.11** void ogps\_ResetPrevPointIterator ([OGPS\\_PointIteratorPtr](#) const *iterator*)

**10.6.1.12** OGPS\_Boolean ogps\_SetCurrentPoint (const [OGPS\\_PointIteratorPtr](#) *iterator*, const [OGPS\\_PointVectorPtr](#) *vector*)

## 10.7 S:/openGPS/ISO5436\_XML/trunk/src/ISO5436\_2\_XML/c/point\_iterator.hxx File Reference

```
#include "../cxx/auto_ptr_types.hxx"
```

### Classes

- struct [\\_OGPS\\_POINT\\_ITERATOR](#)

### Typedefs

- typedef [\\_OGPS\\_POINT\\_ITERATOR](#) [OGPS\\_PointIterator](#)
- typedef [\\_OGPS\\_POINT\\_ITERATOR](#) \* [OGPS\\_PointIteratorPtr](#)

### 10.7.1 Typedef Documentation

**10.7.1.1** typedef struct [\\_OGPS\\_POINT\\_ITERATOR](#) [OGPS\\_PointIterator](#)

**10.7.1.2** typedef struct [\\_OGPS\\_POINT\\_ITERATOR](#) \* [OGPS\\_PointIteratorPtr](#)

## 10.8 S:/openGPS/ISO5436\_XML/trunk/src/ISO5436\_2\_XML/c/point\_vector.cxx File Reference

```
#include <opengps/point_vector.h>
```

```
#include <opengps/point_vector.hxx>
```

```
#include "data_point.hxx"
#include "point_vector.hxx"
```

## Functions

- [OGPS\\_PointVectorPtr](#) [ogps\\_CreatePointVector](#) (void)
- void [ogps\\_FreePointVector](#) ([OGPS\\_PointVectorPtr](#) \*vector)
- double [ogps\\_GetDoubleX](#) (const [OGPS\\_PointVectorPtr](#) vector)
- double [ogps\\_GetDoubleY](#) (const [OGPS\\_PointVectorPtr](#) vector)
- double [ogps\\_GetDoubleZ](#) (const [OGPS\\_PointVectorPtr](#) vector)
- float [ogps\\_GetFloatX](#) (const [OGPS\\_PointVectorPtr](#) vector)
- float [ogps\\_GetFloatY](#) (const [OGPS\\_PointVectorPtr](#) vector)
- float [ogps\\_GetFloatZ](#) (const [OGPS\\_PointVectorPtr](#) vector)
- short [ogps\\_GetInt16X](#) (const [OGPS\\_PointVectorPtr](#) vector)
- short [ogps\\_GetInt16Y](#) (const [OGPS\\_PointVectorPtr](#) vector)
- short [ogps\\_GetInt16Z](#) (const [OGPS\\_PointVectorPtr](#) vector)
- int [ogps\\_GetInt32X](#) (const [OGPS\\_PointVectorPtr](#) vector)
- int [ogps\\_GetInt32Y](#) (const [OGPS\\_PointVectorPtr](#) vector)
- int [ogps\\_GetInt32Z](#) (const [OGPS\\_PointVectorPtr](#) vector)
- [OGPS\\_DataPointPtr](#) const [ogps\\_GetX](#) ([OGPS\\_PointVectorPtr](#) const vector)
- void [ogps\\_GetXYZ](#) (const [OGPS\\_PointVectorPtr](#) vector, double \*x, double \*y, double \*z)
- [OGPS\\_DataPointPtr](#) const [ogps\\_GetY](#) ([OGPS\\_PointVectorPtr](#) const vector)
- [OGPS\\_DataPointPtr](#) const [ogps\\_GetZ](#) ([OGPS\\_PointVectorPtr](#) const vector)
- [OGPS\\_Boolean](#) [ogps\\_IsValidPoint](#) (const [OGPS\\_PointVectorPtr](#) vector)
- void [ogps\\_SetDoubleX](#) ([OGPS\\_PointVectorPtr](#) const vector, const double value)
- void [ogps\\_SetDoubleY](#) ([OGPS\\_PointVectorPtr](#) const vector, const double value)
- void [ogps\\_SetDoubleZ](#) ([OGPS\\_PointVectorPtr](#) const vector, const double value)
- void [ogps\\_SetFloatX](#) ([OGPS\\_PointVectorPtr](#) const vector, const float value)
- void [ogps\\_SetFloatY](#) ([OGPS\\_PointVectorPtr](#) const vector, const float value)
- void [ogps\\_SetFloatZ](#) ([OGPS\\_PointVectorPtr](#) const vector, const float value)
- void [ogps\\_SetInt16X](#) ([OGPS\\_PointVectorPtr](#) const vector, const short value)
- void [ogps\\_SetInt16Y](#) ([OGPS\\_PointVectorPtr](#) const vector, const short value)
- void [ogps\\_SetInt16Z](#) ([OGPS\\_PointVectorPtr](#) const vector, const short value)
- void [ogps\\_SetInt32X](#) ([OGPS\\_PointVectorPtr](#) const vector, const int value)
- void [ogps\\_SetInt32Y](#) ([OGPS\\_PointVectorPtr](#) const vector, const int value)
- void [ogps\\_SetInt32Z](#) ([OGPS\\_PointVectorPtr](#) const vector, const int value)

### 10.8.1 Function Documentation

#### 10.8.1.1 [OGPS\\_PointVectorPtr](#) [ogps\\_CreatePointVector](#) (void)

#### 10.8.1.2 void [ogps\\_FreePointVector](#) ([OGPS\\_PointVectorPtr](#) \* vector)

#### 10.8.1.3 double [ogps\\_GetDoubleX](#) (const [OGPS\\_PointVectorPtr](#) vector)

## 10.8

S:/openGPS/ISO5436\_XML/trunk/src/ISO5436\_2\_XML/c/point\_vector.cxx File  
Reference 262

---

10.8.1.4 double ogps\_GetDoubleY (const **OGPS\_PointVectorPtr** *vector*)

10.8.1.5 double ogps\_GetDoubleZ (const **OGPS\_PointVectorPtr** *vector*)

10.8.1.6 float ogps\_GetFloatX (const **OGPS\_PointVectorPtr** *vector*)

10.8.1.7 float ogps\_GetFloatY (const **OGPS\_PointVectorPtr** *vector*)

10.8.1.8 float ogps\_GetFloatZ (const **OGPS\_PointVectorPtr** *vector*)

10.8.1.9 short ogps\_GetInt16X (const **OGPS\_PointVectorPtr** *vector*)

10.8.1.10 short ogps\_GetInt16Y (const **OGPS\_PointVectorPtr** *vector*)

10.8.1.11 short ogps\_GetInt16Z (const **OGPS\_PointVectorPtr** *vector*)

10.8.1.12 int ogps\_GetInt32X (const **OGPS\_PointVectorPtr** *vector*)

10.8.1.13 int ogps\_GetInt32Y (const **OGPS\_PointVectorPtr** *vector*)

10.8.1.14 int ogps\_GetInt32Z (const **OGPS\_PointVectorPtr** *vector*)

10.8.1.15 **OGPS\_DataPointPtr** const ogps\_GetX (**OGPS\_PointVectorPtr** const *vector*)

10.8.1.16 void ogps\_GetXYZ (const **OGPS\_PointVectorPtr** *vector*, double \* *x*, double \* *y*, double \* *z*)

10.8.1.17 **OGPS\_DataPointPtr** const ogps\_GetY (**OGPS\_PointVectorPtr** const *vector*)

10.8.1.18 **OGPS\_DataPointPtr** const ogps\_GetZ (**OGPS\_PointVectorPtr** const *vector*)

10.8.1.19 **OGPS\_Boolean** ogps\_IsValidPoint (const **OGPS\_PointVectorPtr** *vector*)

10.8.1.20 void ogps\_SetDoubleX (**OGPS\_PointVectorPtr** const *vector*, const double *value*)

## 10.9

S:/openGPS/ISO5436\_XML/trunk/src/ISO5436\_2\_XML/cxx/point\_vector.cxx

File Reference

263

10.8.1.21 void ogps\_SetDoubleY ([OGPS\\_PointVectorPtr](#) const *vector*, const double *value*)

10.8.1.22 void ogps\_SetDoubleZ ([OGPS\\_PointVectorPtr](#) const *vector*, const double *value*)

10.8.1.23 void ogps\_SetFloatX ([OGPS\\_PointVectorPtr](#) const *vector*, const float *value*)

10.8.1.24 void ogps\_SetFloatY ([OGPS\\_PointVectorPtr](#) const *vector*, const float *value*)

10.8.1.25 void ogps\_SetFloatZ ([OGPS\\_PointVectorPtr](#) const *vector*, const float *value*)

10.8.1.26 void ogps\_SetInt16X ([OGPS\\_PointVectorPtr](#) const *vector*, const short *value*)

10.8.1.27 void ogps\_SetInt16Y ([OGPS\\_PointVectorPtr](#) const *vector*, const short *value*)

10.8.1.28 void ogps\_SetInt16Z ([OGPS\\_PointVectorPtr](#) const *vector*, const short *value*)

10.8.1.29 void ogps\_SetInt32X ([OGPS\\_PointVectorPtr](#) const *vector*, const int *value*)

10.8.1.30 void ogps\_SetInt32Y ([OGPS\\_PointVectorPtr](#) const *vector*, const int *value*)

10.8.1.31 void ogps\_SetInt32Z ([OGPS\\_PointVectorPtr](#) const *vector*, const int *value*)

## 10.9 S:/openGPS/ISO5436\_XML/trunk/src/ISO5436\_2\_XML/cxx/point\_vector.cxx File Reference

```
#include "data_point_impl.hxx"
#include <opengps/point_vector.hxx>
#include "stdafx.hxx"
```



## 10.10

S:/openGPS/ISO5436\_XML/trunk/src/ISO5436\_2\_XML/c/point\_vector.hxx File Reference 264

### 10.10 S:/openGPS/ISO5436\_XML/trunk/src/ISO5436\_2\_XML/c/point\_vector.hxx File Reference

```
#include <opengps/point_vector.hxx>
```

#### Classes

- struct [\\_OGPS\\_POINT\\_VECTOR](#)

#### Typedefs

- typedef [\\_OGPS\\_POINT\\_VECTOR](#) [OGPS\\_PointVector](#)
- typedef [\\_OGPS\\_POINT\\_VECTOR](#) \* [OGPS\\_PointVectorPtr](#)

#### 10.10.1 Typedef Documentation

10.10.1.1 typedef struct [\\_OGPS\\_POINT\\_VECTOR](#) [OGPS\\_PointVector](#)

10.10.1.2 typedef struct [\\_OGPS\\_POINT\\_VECTOR](#) \* [OGPS\\_PointVectorPtr](#)

### 10.11 S:/openGPS/ISO5436\_XML/trunk/src/ISO5436\_2\_XML/cxx/auto\_ptr\_types.hxx File Reference

```
#include <memory>
```

#### Namespaces

- namespace [OpenGPS](#)
- namespace [xsd](#)

#### Typedefs

- typedef std::auto\_ptr< [xsd::ISO5436\\_2Type](#) > [OpenGPS::ISO5436\\_2Type-AutoPtr](#)
- typedef std::auto\_ptr< [PointIterator](#) > [OpenGPS::PointIteratorAutoPtr](#)
- typedef std::auto\_ptr< [PointVectorBase](#) > [OpenGPS::PointVectorAutoPtr](#)
- typedef std::auto\_ptr< [PointVectorParserBuilder](#) > [OpenGPS::PointVector-ParserBuilderAutoPtr](#)
- typedef std::auto\_ptr< [VectorBuffer](#) > [OpenGPS::VectorBufferAutoPtr](#)
- typedef std::auto\_ptr< [VectorBufferBuilder](#) > [OpenGPS::VectorBufferBuilder-AutoPtr](#)

**10.12** S:/openGPS/ISO5436\_XML/trunk/src/ISO5436\_2\_XML/cxx/binary\_-  
lsb\_point\_vector\_reader\_context.cxx File

Reference

265

**10.12** S:/openGPS/ISO5436\_XML/trunk/src/ISO5436\_2\_-  
XML/cxx/binary\_lsb\_point\_vector\_reader\_context.cxx  
File Reference

```
#include "binary_lsb_point_vector_reader_context.hxx"  
#include "point_vector_iostream.hxx"  
#include "stdafx.hxx"
```

**10.13** S:/openGPS/ISO5436\_XML/trunk/src/ISO5436\_2\_-  
XML/cxx/binary\_lsb\_point\_vector\_reader\_context.hxx  
File Reference

```
#include "binary_point_vector_reader_context.hxx"
```

#### Namespaces

- namespace [OpenGPS](#)

#### Classes

- class [OpenGPS::BinaryLSBPointVectorReaderContext](#)

**10.14** S:/openGPS/ISO5436\_XML/trunk/src/ISO5436\_2\_-  
XML/cxx/binary\_lsb\_point\_vector\_writer\_context.cxx  
File Reference

```
#include "binary_lsb_point_vector_writer_context.hxx"  
#include "point_vector_iostream.hxx"  
#include "stdafx.hxx"
```

**10.15** S:/openGPS/ISO5436\_XML/trunk/src/ISO5436\_2\_-  
XML/cxx/binary\_lsb\_point\_vector\_writer\_context.hxx  
File Reference

```
#include "binary_point_vector_writer_context.hxx"
```

#### Namespaces

- namespace [OpenGPS](#)

**Classes**

- class [OpenGPS::BinaryLSBPointVectorWriterContext](#)

**10.16 S:/openGPS/ISO5436\_XML/trunk/src/ISO5436\_2\_-  
XML/cxx/binary\_msb\_point\_vector\_reader\_context.cxx  
File Reference**

```
#include "binary_msb_point_vector_reader_context.hxx"
#include "binary_lsb_point_vector_reader_context.hxx"
#include "point_vector_iostream.hxx"
#include "environment.hxx"
#include "stdafx.hxx"
```

**10.17 S:/openGPS/ISO5436\_XML/trunk/src/ISO5436\_2\_-  
XML/cxx/binary\_msb\_point\_vector\_reader\_context.hxx  
File Reference**

```
#include "binary_point_vector_reader_context.hxx"
```

**Namespaces**

- namespace [OpenGPS](#)

**Classes**

- class [OpenGPS::BinaryMSBPointVectorReaderContext](#)

**10.18 S:/openGPS/ISO5436\_XML/trunk/src/ISO5436\_2\_-  
XML/cxx/binary\_msb\_point\_vector\_writer\_context.cxx  
File Reference**

```
#include "binary_msb_point_vector_writer_context.hxx"
#include "point_vector_iostream.hxx"
#include "environment.hxx"
#include "stdafx.hxx"
```

**10.19** S:/openGPS/ISO5436\_XML/trunk/src/ISO5436\_2\_XML/cxx/binary\_-  
msb\_point\_vector\_writer\_context.hxx File

Reference

267

**10.19** S:/openGPS/ISO5436\_XML/trunk/src/ISO5436\_2\_-  
XML/cxx/binary\_msb\_point\_vector\_writer\_context.hxx  
File Reference

```
#include "binary_point_vector_writer_context.hxx"
```

#### Namespaces

- namespace [OpenGPS](#)

#### Classes

- class [OpenGPS::BinaryMSBPointVectorWriterContext](#)

**10.20** S:/openGPS/ISO5436\_XML/trunk/src/ISO5436\_2\_-  
XML/cxx/binary\_point\_vector\_reader\_context.hxx File  
Reference

```
#include "point_vector_reader_context.hxx"
```

#### Namespaces

- namespace [OpenGPS](#)

#### Classes

- class [OpenGPS::BinaryPointVectorReaderContext](#)

**10.21** S:/openGPS/ISO5436\_XML/trunk/src/ISO5436\_2\_-  
XML/cxx/binary\_point\_vector\_writer\_context.cxx File  
Reference

```
#include "binary_point_vector_writer_context.hxx"
```

```
#include "stdafx.hxx"
```

**10.22** S:/openGPS/ISO5436\_XML/trunk/src/ISO5436\_2\_-  
XML/cxx/binary\_point\_vector\_writer\_context.hxx File  
Reference

```
#include "point_vector_writer_context.hxx"
```

```
#include "zip_stream_buffer.hxx"
```

**10.23 S:/openGPS/ISO5436\_XML/trunk/src/ISO5436\_2\_XML/cxx/data\_point\_impl.cxx File Reference**

268

**Namespaces**

- namespace [OpenGPS](#)

**Classes**

- class [OpenGPS::BinaryPointVectorWriterContext](#)

**10.23 S:/openGPS/ISO5436\_XML/trunk/src/ISO5436\_2\_XML/cxx/data\_point\_impl.cxx File Reference**

```
#include "data_point_impl.hxx"
#include "stdafx.hxx"
```

**10.24 S:/openGPS/ISO5436\_XML/trunk/src/ISO5436\_2\_XML/cxx/data\_point\_impl.hxx File Reference**

```
#include <opengps/data_point.hxx>
```

**Namespaces**

- namespace [OpenGPS](#)

**Classes**

- class [OpenGPS::DataPointImpl](#)
- union [OpenGPS::DataPointImpl::\\_OGPS\\_DATA\\_POINT\\_VALUE](#)

**10.25 S:/openGPS/ISO5436\_XML/trunk/src/ISO5436\_2\_XML/cxx/data\_point\_parser.hxx File Reference**

```
#include <opengps/opengps.hxx>
```

**Namespaces**

- namespace [OpenGPS](#)

**Classes**

- class [OpenGPS::DataPointParser](#)

**10.26 S:/openGPS/ISO5436\_XML/trunk/src/ISO5436\_2\_XML/cxx/double\_data\_point\_parser.cxx File Reference**

```
#include "double_data_point_parser.hxx"
#include "point_vector_reader_context.hxx"
#include "point_vector_writer_context.hxx"
#include <opengps/data_point.hxx>
#include "stdafx.hxx"
```

**10.27 S:/openGPS/ISO5436\_XML/trunk/src/ISO5436\_2\_XML/cxx/double\_data\_point\_parser.hxx File Reference**

```
#include "data_point_parser.hxx"
```

**Namespaces**

- namespace [OpenGPS](#)

**Classes**

- class [OpenGPS::DoubleDataPointParser](#)

**10.28 S:/openGPS/ISO5436\_XML/trunk/src/ISO5436\_2\_XML/cxx/double\_point\_buffer.cxx File Reference**

```
#include "double_point_buffer.hxx"
#include "stdafx.hxx"
```

**10.29 S:/openGPS/ISO5436\_XML/trunk/src/ISO5436\_2\_XML/cxx/double\_point\_buffer.hxx File Reference**

```
#include "point_buffer.hxx"
```

**Namespaces**

- namespace [OpenGPS](#)

**Classes**

- class [OpenGPS::DoublePointBuffer](#)

## 10.30

S:/openGPS/ISO5436\_XML/trunk/src/ISO5436\_2\_XML/cxx/environment.cxx

File Reference

270

### 10.30 S:/openGPS/ISO5436\_XML/trunk/src/ISO5436\_2\_XML/cxx/environment.cxx File Reference

```
#include "environment.hxx"
```

```
#include "stdafx.hxx"
```

#### Defines

- `#define _OGPS_DOUBLE_SIZE 8`
- `#define _OGPS_FLOAT_SIZE 4`
- `#define _OGPS_INT_SIZE 4`
- `#define _OGPS_SHORT_SIZE 2`

#### 10.30.1 Define Documentation

10.30.1.1 `#define _OGPS_DOUBLE_SIZE 8`

10.30.1.2 `#define _OGPS_FLOAT_SIZE 4`

10.30.1.3 `#define _OGPS_INT_SIZE 4`

10.30.1.4 `#define _OGPS_SHORT_SIZE 2`

### 10.31 S:/openGPS/ISO5436\_XML/trunk/src/ISO5436\_2\_XML/cxx/environment.hxx File Reference

```
#include <opengps/opengps.hxx>
```

#### Namespaces

- namespace [OpenGPS](#)

#### Classes

- class [OpenGPS::Environment](#)

### 10.32 S:/openGPS/ISO5436\_XML/trunk/src/ISO5436\_2\_XML/cxx/float\_data\_point\_parser.cxx File Reference

```
#include "float_data_point_parser.hxx"
```

```
#include "point_vector_reader_context.hxx"
```

### 10.33 S:/openGPS/ISO5436\_XML/trunk/src/ISO5436\_2\_XML/cxx/float\_data\_point\_parser.hxx File Reference 271

```
#include "point_vector_writer_context.hxx"
#include <opengps/data_point.hxx>
#include "stdafx.hxx"
```

### 10.33 S:/openGPS/ISO5436\_XML/trunk/src/ISO5436\_2\_XML/cxx/float\_data\_point\_parser.hxx File Reference

```
#include "data_point_parser.hxx"
```

#### Namespaces

- namespace [OpenGPS](#)

#### Classes

- class [OpenGPS::FloatDataPointParser](#)

### 10.34 S:/openGPS/ISO5436\_XML/trunk/src/ISO5436\_2\_XML/cxx/float\_point\_buffer.cxx File Reference

```
#include "float_point_buffer.hxx"
#include "stdafx.hxx"
```

### 10.35 S:/openGPS/ISO5436\_XML/trunk/src/ISO5436\_2\_XML/cxx/float\_point\_buffer.hxx File Reference

```
#include "point_buffer.hxx"
```

#### Namespaces

- namespace [OpenGPS](#)

#### Classes

- class [OpenGPS::FloatPointBuffer](#)

### 10.36 S:/openGPS/ISO5436\_XML/trunk/src/ISO5436\_2\_XML/cxx/int16\_data\_point\_parser.cxx File Reference

```
#include "int16_data_point_parser.hxx"
#include "point_vector_reader_context.hxx"
```



**10.37 S:/openGPS/ISO5436\_XML/trunk/src/ISO5436\_2\_XML/cxx/int16\_data\_point\_parser.hxx File Reference**

272

```
#include "point_vector_writer_context.hxx"
#include <opengps/data_point.hxx>
#include "stdafx.hxx"
```

**10.37 S:/openGPS/ISO5436\_XML/trunk/src/ISO5436\_2\_XML/cxx/int16\_data\_point\_parser.hxx File Reference**

```
#include "data_point_parser.hxx"
```

**Namespaces**

- namespace [OpenGPS](#)

**Classes**

- class [OpenGPS::Int16DataPointParser](#)

**10.38 S:/openGPS/ISO5436\_XML/trunk/src/ISO5436\_2\_XML/cxx/int16\_point\_buffer.cxx File Reference**

```
#include "int16_point_buffer.hxx"
#include "stdafx.hxx"
```

**10.39 S:/openGPS/ISO5436\_XML/trunk/src/ISO5436\_2\_XML/cxx/int16\_point\_buffer.hxx File Reference**

```
#include "point_buffer.hxx"
```

**Namespaces**

- namespace [OpenGPS](#)

**Classes**

- class [OpenGPS::Int16PointBuffer](#)

**10.40 S:/openGPS/ISO5436\_XML/trunk/src/ISO5436\_2\_XML/cxx/int32\_data\_point\_parser.cxx File Reference**

```
#include "int32_data_point_parser.hxx"
#include "point_vector_reader_context.hxx"
```

**10.41 S:/openGPS/ISO5436\_XML/trunk/src/ISO5436\_2\_XML/cxx/int32\_data\_point\_parser.hxx File Reference**

273

```
#include "point_vector_writer_context.hxx"
#include <opengps/data_point.hxx>
#include "stdafx.hxx"
```

**10.41 S:/openGPS/ISO5436\_XML/trunk/src/ISO5436\_2\_XML/cxx/int32\_data\_point\_parser.hxx File Reference**

```
#include "data_point_parser.hxx"
```

**Namespaces**

- namespace [OpenGPS](#)

**Classes**

- class [OpenGPS::Int32DataPointParser](#)

**10.42 S:/openGPS/ISO5436\_XML/trunk/src/ISO5436\_2\_XML/cxx/int32\_point\_buffer.cxx File Reference**

```
#include "int32_point_buffer.hxx"
#include "stdafx.hxx"
```

**10.43 S:/openGPS/ISO5436\_XML/trunk/src/ISO5436\_2\_XML/cxx/int32\_point\_buffer.hxx File Reference**

```
#include "point_buffer.hxx"
```

**Namespaces**

- namespace [OpenGPS](#)

**Classes**

- class [OpenGPS::Int32PointBuffer](#)

**10.44 S:/openGPS/ISO5436\_XML/trunk/src/ISO5436\_2\_XML/cxx/iso5436\_2.cxx File Reference**

```
#include <opengps/iso5436_2.hxx>
#include "iso5436_2_container.hxx"
```

```
#include "stdafx.hxx"
```

#### **10.45 S:/openGPS/ISO5436\_XML/trunk/src/ISO5436\_2\_- XML/cxx/iso5436\_2\_container.cxx File Reference**

```
#include "iso5436_2_container.hxx"  
#include "point_iterator_impl.hxx"  
#include <opengps/iso5436_2_xsd.hxx>  
#include <opengps/point_vector.hxx>  
#include "point_vector_parser_builder.hxx"  
#include "point_vector_parser.hxx"  
#include "xml_point_vector_reader_context.hxx"  
#include "xml_point_vector_writer_context.hxx"  
#include "binary_lsb_point_vector_reader_context.hxx"  
#include "binary_msb_point_vector_reader_context.hxx"  
#include "binary_lsb_point_vector_writer_context.hxx"  
#include "binary_msb_point_vector_writer_context.hxx"  
#include "vector_buffer_builder.hxx"  
#include "vector_buffer.hxx"  
#include "point_vector_proxy_context.hxx"  
#include "environment.hxx"  
#include "point_vector_istream.hxx"  
#include "zip_stream_buffer.hxx"  
#include <iostream>  
#include <fstream>  
#include <unzip.h>  
#include <zip.h>  
#include "stdafx.hxx"
```

#### **Defines**

- #define [MAX\\_ZIP\\_CHUNK](#) (256\*1024)

#### **10.45.1 Define Documentation**

##### **10.45.1.1 #define MAX\_ZIP\_CHUNK (256\*1024)**

**10.46 S:/openGPS/ISO5436\_XML/trunk/src/ISO5436\_2\_-  
XML/cxx/iso5436\_2\_container.hxx File Reference**

```
#include <opengps/iso5436_2.hxx>
#include <opengps/data_point_type.h>
#include "point_vector_proxy_context.hxx"
#include <zip.h>
#include "auto_ptr_types.hxx"
```

**Namespaces**

- namespace [OpenGPS](#)
- namespace [xsd](#)

**Classes**

- class [OpenGPS::ISO5436\\_2Container](#)

**10.47 S:/openGPS/ISO5436\_XML/trunk/src/ISO5436\_2\_-  
XML/cxx/missing\_data\_point\_parser.cxx File Reference**

```
#include "missing_data_point_parser.hxx"
#include "point_vector_reader_context.hxx"
#include "point_vector_writer_context.hxx"
#include <opengps/data_point.hxx>
#include "stdafx.hxx"
```

**10.48 S:/openGPS/ISO5436\_XML/trunk/src/ISO5436\_2\_-  
XML/cxx/missing\_data\_point\_parser.hxx File Reference**

```
#include "data_point_parser.hxx"
```

**Namespaces**

- namespace [OpenGPS](#)

**Classes**

- class [OpenGPS::MissingDataPointParser](#)

## 10.49

S:/openGPS/ISO5436\_XML/trunk/src/ISO5436\_2\_XML/cxx/point\_buffer.cxx

File Reference

276

### 10.49 S:/openGPS/ISO5436\_XML/trunk/src/ISO5436\_2\_XML/cxx/point\_buffer.cxx File Reference

```
#include "point_buffer.hxx"
#include "stdafx.hxx"
#include <stdlib.h>
```

### 10.50 S:/openGPS/ISO5436\_XML/trunk/src/ISO5436\_2\_XML/cxx/point\_buffer.hxx File Reference

```
#include <opengps/opengps.hxx>
#include <opengps/data_point_type.h>
```

#### Namespaces

- namespace [OpenGPS](#)

#### Classes

- class [OpenGPS::PointBuffer](#)

### 10.51 S:/openGPS/ISO5436\_XML/trunk/src/ISO5436\_2\_XML/cxx/point\_iterator\_impl.cxx File Reference

```
#include "point_iterator_impl.hxx"
#include <opengps/point_vector.hxx>
#include "stdafx.hxx"
```

### 10.52 S:/openGPS/ISO5436\_XML/trunk/src/ISO5436\_2\_XML/cxx/point\_iterator\_impl.hxx File Reference

```
#include <opengps/point_iterator.hxx>
#include "iso5436_2_container.hxx"
```

#### Namespaces

- namespace [OpenGPS](#)

#### Classes

- class [OpenGPS::PointIteratorImpl](#)

**10.53 S:/openGPS/ISO5436\_XML/trunk/src/ISO5436\_2\_XML/cxx/point\_vector\_iostream.cxx File Reference**

```
#include "point_vector_iostream.hxx"
#include "stdafx.hxx"
```

**10.54 S:/openGPS/ISO5436\_XML/trunk/src/ISO5436\_2\_XML/cxx/point\_vector\_iostream.hxx File Reference**

```
#include <opengps/opengps.hxx>
#include <xlocale>
#include <sstream>
#include <fstream>
```

**Namespaces**

- namespace [OpenGPS](#)

**Classes**

- class [OpenGPS::PointVectorInputBinaryFileStream](#)
- class [OpenGPS::PointVectorInputStringStream](#)
- class [OpenGPS::PointVectorInvariantLocale](#)
- class [OpenGPS::PointVectorOutputBinaryFileStream](#)
- class [OpenGPS::PointVectorOutputStringStream](#)
- class [OpenGPS::PointVectorWhitespaceFacet](#)

**10.55 S:/openGPS/ISO5436\_XML/trunk/src/ISO5436\_2\_XML/cxx/point\_vector\_parser.cxx File Reference**

```
#include "point_vector_parser.hxx"
#include "int16_data_point_parser.hxx"
#include "int32_data_point_parser.hxx"
#include "float_data_point_parser.hxx"
#include "double_data_point_parser.hxx"
#include "missing_data_point_parser.hxx"
#include "point_vector_reader_context.hxx"
#include "point_vector_writer_context.hxx"
#include <opengps/point_vector_base.hxx>
#include "stdafx.hxx"
```

**10.56 S:/openGPS/ISO5436\_XML/trunk/src/ISO5436\_2\_XML/cxx/point\_vector\_parser.hxx File Reference**

278

**10.56 S:/openGPS/ISO5436\_XML/trunk/src/ISO5436\_2\_XML/cxx/point\_vector\_parser.hxx File Reference**

```
#include <opengps/opengps.hxx>
#include <opengps/data_point_type.h>
```

**Namespaces**

- namespace [OpenGPS](#)

**Classes**

- class [OpenGPS::PointVectorParser](#)

**10.57 S:/openGPS/ISO5436\_XML/trunk/src/ISO5436\_2\_XML/cxx/point\_vector\_parser\_builder.cxx File Reference**

```
#include "point_vector_parser_builder.hxx"
#include "point_vector_parser.hxx"
#include "stdafx.hxx"
```

**10.58 S:/openGPS/ISO5436\_XML/trunk/src/ISO5436\_2\_XML/cxx/point\_vector\_parser\_builder.hxx File Reference**

```
#include <opengps/opengps.hxx>
#include <opengps/data_point_type.h>
```

**Namespaces**

- namespace [OpenGPS](#)

**Classes**

- class [OpenGPS::PointVectorParserBuilder](#)

**10.59 S:/openGPS/ISO5436\_XML/trunk/src/ISO5436\_2\_XML/cxx/point\_vector\_proxy.cxx File Reference**

```
#include "point_vector_proxy.hxx"
#include "point_vector_proxy_context.hxx"
#include "vector_buffer.hxx"
```

**10.60 S:/openGPS/ISO5436\_XML/trunk/src/ISO5436\_2\_XML/cxx/point\_vector\_proxy.hxx File Reference**

279

```
#include "point_buffer.hxx"
#include "stdafx.hxx"
```

**10.60 S:/openGPS/ISO5436\_XML/trunk/src/ISO5436\_2\_XML/cxx/point\_vector\_proxy.hxx File Reference**

```
#include <opengps/point_vector_base.hxx>
#include <opengps/data_point.hxx>
```

**Namespaces**

- namespace [OpenGPS](#)

**Classes**

- class [OpenGPS::PointVectorProxy](#)
- class [OpenGPS::PointVectorProxy::DataPointProxy](#)
- class [OpenGPS::PointVectorProxy::DataPointProxyContext](#)
- class [OpenGPS::PointVectorProxy::UDataPointProxyContext](#)
- class [OpenGPS::PointVectorProxy::VDataPointProxyContext](#)
- class [OpenGPS::PointVectorProxy::WDataPointProxyContext](#)

**10.61 S:/openGPS/ISO5436\_XML/trunk/src/ISO5436\_2\_XML/cxx/point\_vector\_proxy\_context.cxx File Reference**

```
#include "point_vector_proxy_context.hxx"
#include "stdafx.hxx"
```

**10.62 S:/openGPS/ISO5436\_XML/trunk/src/ISO5436\_2\_XML/cxx/point\_vector\_proxy\_context.hxx File Reference**

**Namespaces**

- namespace [OpenGPS](#)

**Classes**

- class [OpenGPS::PointVectorProxyContext](#)

**10.63 S:/openGPS/ISO5436\_XML/trunk/src/ISO5436\_2\_XML/cxx/point\_vector\_reader\_context.hxx File Reference**

```
#include <opengps/opengps.hxx>
```



## 10.64 S:/openGPS/ISO5436\_XML/trunk/src/ISO5436\_2\_XML/cxx/point\_vector\_writer\_context.hxx File

### Reference

280

#### Namespaces

- namespace [OpenGPS](#)

#### Classes

- class [OpenGPS::PointVectorReaderContext](#)

## 10.64 S:/openGPS/ISO5436\_XML/trunk/src/ISO5436\_2\_XML/cxx/point\_vector\_writer\_context.hxx File Reference

```
#include <opengps/opengps.hxx>
```

#### Namespaces

- namespace [OpenGPS](#)

#### Classes

- class [OpenGPS::PointVectorWriterContext](#)

## 10.65 S:/openGPS/ISO5436\_XML/trunk/src/ISO5436\_2\_XML/cxx/stdafx.hxx File Reference

```
#include <tchar.h>
```

#### Defines

- [#define \\_OPENGPS\\_BINFORMAT\\_DOUBLE\\_SIZE 8](#)
- [#define \\_OPENGPS\\_BINFORMAT\\_FLOAT\\_SIZE 4](#)
- [#define \\_OPENGPS\\_BINFORMAT\\_INT16\\_SIZE 2](#)
- [#define \\_OPENGPS\\_BINFORMAT\\_INT32\\_SIZE 4](#)

### 10.65.1 Define Documentation

**10.65.1.1** [#define \\_OPENGPS\\_BINFORMAT\\_DOUBLE\\_SIZE 8](#)

**10.65.1.2** [#define \\_OPENGPS\\_BINFORMAT\\_FLOAT\\_SIZE 4](#)

**10.65.1.3** [#define \\_OPENGPS\\_BINFORMAT\\_INT16\\_SIZE 2](#)

**10.65.1.4** [#define \\_OPENGPS\\_BINFORMAT\\_INT32\\_SIZE 4](#)

10.66

S:/openGPS/ISO5436\_XML/trunk/src/ISO5436\_2\_XML/cxx/stream\_types.hxx

File Reference

281

## 10.66 S:/openGPS/ISO5436\_XML/trunk/src/ISO5436\_2\_XML/cxx/stream\_types.hxx File Reference

```
#include <sstream>
```

### Namespaces

- namespace [OpenGPS](#)

### Typedefs

- typedef std::ostringstream [OpenGPS::OutStringStream](#)

## 10.67 S:/openGPS/ISO5436\_XML/trunk/src/ISO5436\_2\_XML/cxx/string.cxx File Reference

```
#include <opengps/opengps.hxx>
```

```
#include "stdafx.hxx"
```

## 10.68 S:/openGPS/ISO5436\_XML/trunk/src/ISO5436\_2\_XML/cxx/valid\_buffer.cxx File Reference

```
#include "valid_buffer.hxx"
```

```
#include "stdafx.hxx"
```

## 10.69 S:/openGPS/ISO5436\_XML/trunk/src/ISO5436\_2\_XML/cxx/valid\_buffer.hxx File Reference

```
#include <opengps/opengps.hxx>
```

```
#include <iostream>
```

### Namespaces

- namespace [OpenGPS](#)

### Classes

- class [OpenGPS::ValidBuffer](#)

## 10.70

S:/openGPS/ISO5436\_XML/trunk/src/ISO5436\_2\_XML/cxx/vector\_buffer.cxx

File Reference

282

### 10.70 S:/openGPS/ISO5436\_XML/trunk/src/ISO5436\_2\_XML/cxx/vector\_buffer.cxx File Reference

```
#include "vector_buffer.hxx"
#include "point_vector_proxy.hxx"
#include "stdafx.hxx"
```

### 10.71 S:/openGPS/ISO5436\_XML/trunk/src/ISO5436\_2\_XML/cxx/vector\_buffer.hxx File Reference

```
#include <opengps/data_point_type.h>
#include <opengps/point_vector_base.hxx>
#include "valid_buffer.hxx"
#include "auto_ptr_types.hxx"
```

#### Namespaces

- namespace [OpenGPS](#)

#### Classes

- class [OpenGPS::VectorBuffer](#)

### 10.72 S:/openGPS/ISO5436\_XML/trunk/src/ISO5436\_2\_XML/cxx/vector\_buffer\_builder.cxx File Reference

```
#include "vector_buffer_builder.hxx"
#include "vector_buffer.hxx"
#include "int16_point_buffer.hxx"
#include "int32_point_buffer.hxx"
#include "float_point_buffer.hxx"
#include "double_point_buffer.hxx"
#include "stdafx.hxx"
```

### 10.73 S:/openGPS/ISO5436\_XML/trunk/src/ISO5436\_2\_XML/cxx/vector\_buffer\_builder.hxx File Reference

```
#include <opengps/opengps.hxx>
#include <opengps/data_point_type.h>
```

**10.74 S:/openGPS/ISO5436\_XML/trunk/src/ISO5436\_2\_XML/cxx/win32\_environment.cxx File**

**Reference**

283

**Namespaces**

- namespace [OpenGPS](#)

**Classes**

- class [OpenGPS::VectorBufferBuilder](#)

**10.74 S:/openGPS/ISO5436\_XML/trunk/src/ISO5436\_2\_XML/cxx/win32\_environment.cxx File Reference**

**10.75 S:/openGPS/ISO5436\_XML/trunk/src/ISO5436\_2\_XML/cxx/win32\_environment.hxx File Reference**

**10.76 S:/openGPS/ISO5436\_XML/trunk/src/ISO5436\_2\_XML/cxx/xml\_point\_vector\_reader\_context.cxx File Reference**

```
#include "xml_point_vector_reader_context.hxx"
#include "point_vector_iostream.hxx"
#include "stdafx.hxx"
```

**10.77 S:/openGPS/ISO5436\_XML/trunk/src/ISO5436\_2\_XML/cxx/xml\_point\_vector\_reader\_context.hxx File Reference**

```
#include "point_vector_reader_context.hxx"
#include <opengps/iso5436_2_xsd.hxx>
```

**Namespaces**

- namespace [OpenGPS](#)

**Classes**

- class [OpenGPS::XmlPointVectorReaderContext](#)

**10.78 S:/openGPS/ISO5436\_XML/trunk/src/ISO5436\_2\_XML/cxx/xml\_point\_vector\_writer\_context.cxx File Reference**

```
#include "xml_point_vector_writer_context.hxx"
```

**10.79 S:/openGPS/ISO5436\_XML/trunk/src/ISO5436\_2\_XML/cxx/xml\_point\_vector\_writer\_context.hxx File Reference**

284

```
#include "point_vector_iostream.hxx"
#include "stdafx.hxx"
```

**10.79 S:/openGPS/ISO5436\_XML/trunk/src/ISO5436\_2\_XML/cxx/xml\_point\_vector\_writer\_context.hxx File Reference**

```
#include "point_vector_writer_context.hxx"
#include <opengps/iso5436_2_xsd.hxx>
```

**Namespaces**

- namespace [OpenGPS](#)

**Classes**

- class [OpenGPS::XmlPointVectorWriterContext](#)

**10.80 S:/openGPS/ISO5436\_XML/trunk/src/ISO5436\_2\_XML/cxx/zip\_stream\_buffer.cxx File Reference**

```
#include "zip_stream_buffer.hxx"
#include "stdafx.hxx"
```

**10.81 S:/openGPS/ISO5436\_XML/trunk/src/ISO5436\_2\_XML/cxx/zip\_stream\_buffer.hxx File Reference**

```
#include <ostream>
#include <zip.h>
```

**Namespaces**

- namespace [OpenGPS](#)

**Classes**

- class [OpenGPS::ZipOutputStream](#)
- class [OpenGPS::ZipStreamBuffer](#)

## 10.82 S:/openGPS/ISO5436\_XML/trunk/src/ISO5436\_2\_XML/iso5436\_2\_xsd.cxx File Reference

```
#include <xsd/cxx/pre.hxx>
#include "iso5436_2_xsd.hxx"
#include <xsd/cxx/xml/dom/parsing-source.hxx>
#include <ostream>
#include <istream>
#include <xercesc/framework/Wrapper4InputSource.hpp>
#include <xsd/cxx/xml/sax/std-input-source.hxx>
#include <xsd/cxx/tree/error-handler.hxx>
#include <xsd/cxx/xml/dom/serialization-source.hxx>
#include <xsd/cxx/post.hxx>
```

### Namespaces

- namespace [xsd](#)

### Serialization functions for the ISO5436\_2 document root.

The only global element: The root node

- `::xsd::cxx::xml::dom::auto_ptr< ::xercesc::DOMDocument > xsd::ISO5436_2 (const ::xsd::ISO5436_2Type &x, const ::xml_schema::namespace_infomap &m, ::xml_schema::flags f=0)`  
*Serialize to a new Xerces-C++ DOM document.*
- `void xsd::ISO5436_2 (::xercesc::DOMDocument &d, const ::xsd::ISO5436_2Type &x, ::xml_schema::flags f=0)`  
*Serialize to an existing Xerces-C++ DOM document.*
- `void xsd::ISO5436_2 (::xercesc::XMLFormatTarget &ft, const ::xsd::ISO5436_2Type &x, const ::xml_schema::namespace_infomap &m, ::xercesc::DOMErrorHandler &eh, const ::std::wstring &e=L"UTF-8", ::xml_schema::flags f=0)`  
*Serialize to a Xerces-C++ XML format target with a Xerces-C++ DOM error handler.*
- `void xsd::ISO5436_2 (::xercesc::XMLFormatTarget &ft, const ::xsd::ISO5436_2Type &x, const ::xml_schema::namespace_infomap &m, ::xml_schema::error_handler &eh, const ::std::wstring &e=L"UTF-8", ::xml_schema::flags f=0)`  
*Serialize to a Xerces-C++ XML format target with an error handler.*

- void [xsd::ISO5436\\_2](#) (::xercesc::XMLFormatTarget &ft, const [::xsd::ISO5436\\_2Type](#) &x, const [::xml\\_schema::namespace\\_infomap](#) &m, const [::std::wstring](#) &e=L"UTF-8",[::xml\\_schema::flags](#) f=0)  
*Serialize to a Xerces-C++ XML format target.*
- void [xsd::ISO5436\\_2](#) (::std::ostream &os, const [::xsd::ISO5436\\_2Type](#) &x, const [::xml\\_schema::namespace\\_infomap](#) &m,[::xercesc::DOMErrorHandler](#) &eh, const [::std::wstring](#) &e=L"UTF-8",[::xml\\_schema::flags](#) f=0)  
*Serialize to a standard output stream with a Xerces-C++ DOM error handler.*
- void [xsd::ISO5436\\_2](#) (::std::ostream &os, const [::xsd::ISO5436\\_2Type](#) &x, const [::xml\\_schema::namespace\\_infomap](#) &m,[::xml\\_schema::error\\_handler](#) &eh, const [::std::wstring](#) &e=L"UTF-8",[::xml\\_schema::flags](#) f=0)  
*Serialize to a standard output stream with an error handler.*
- void [xsd::ISO5436\\_2](#) (::std::ostream &os, const [::xsd::ISO5436\\_2Type](#) &x, const [::xml\\_schema::namespace\\_infomap](#) &m, const [::std::wstring](#) &e=L"UTF-8",[::xml\\_schema::flags](#) f=0)  
*Serialize to a standard output stream.*

### Parsing functions for the ISO5436\_2 document root.

The only global element: The root node

- [::std::auto\\_ptr< ::xsd::ISO5436\\_2Type > xsd::ISO5436\\_2](#) (::xercesc::DOMDocument \*d,[::xml\\_schema::flags](#) f=0, const [::xml\\_schema::properties](#) &p=[::xml\\_schema::properties\(\)](#))  
*Parse a Xerces-C++ DOM document.*
- [::std::auto\\_ptr< ::xsd::ISO5436\\_2Type > xsd::ISO5436\\_2](#) (const [::xercesc::DOMDocument](#) &d,[::xml\\_schema::flags](#) f=0, const [::xml\\_schema::properties](#) &p=[::xml\\_schema::properties\(\)](#))  
*Parse a Xerces-C++ DOM document.*
- [::std::auto\\_ptr< ::xsd::ISO5436\\_2Type > xsd::ISO5436\\_2](#) (const [::xercesc::DOMInputSource](#) &is,[::xercesc::DOMErrorHandler](#) &eh,[::xml\\_schema::flags](#) f=0, const [::xml\\_schema::properties](#) &p=[::xml\\_schema::properties\(\)](#))  
*Parse a Xerces-C++ DOM input source with a Xerces-C++ DOM error handler.*
- [::std::auto\\_ptr< ::xsd::ISO5436\\_2Type > xsd::ISO5436\\_2](#) (const [::xercesc::DOMInputSource](#) &is,[::xml\\_schema::error\\_handler](#) &eh,[::xml\\_schema::flags](#) f=0, const [::xml\\_schema::properties](#) &p=[::xml\\_schema::properties\(\)](#))  
*Parse a Xerces-C++ DOM input source with an error handler.*

- `::std::auto_ptr< ::xsd::ISO5436_2Type > xsd::ISO5436_2 (const ::xercesc::DOMInputSource &is, ::xml_schema::flags f=0, const ::xml_schema::properties &p=::xml_schema::properties())`  
*Parse a Xerces-C++ DOM input source.*
- `::std::auto_ptr< ::xsd::ISO5436_2Type > xsd::ISO5436_2 (::std::istream &is, const ::std::wstring &id, ::xercesc::DOMErrorHandler &eh, ::xml_schema::flags f=0, const ::xml_schema::properties &p=::xml_schema::properties())`  
*Parse a standard input stream with a resource id and a Xerces-C++ DOM error handler.*
- `::std::auto_ptr< ::xsd::ISO5436_2Type > xsd::ISO5436_2 (::std::istream &is, const ::std::wstring &id, ::xml_schema::error_handler &eh, ::xml_schema::flags f=0, const ::xml_schema::properties &p=::xml_schema::properties())`  
*Parse a standard input stream with a resource id and an error handler.*
- `::std::auto_ptr< ::xsd::ISO5436_2Type > xsd::ISO5436_2 (::std::istream &is, const ::std::wstring &id, ::xml_schema::flags f=0, const ::xml_schema::properties &p=::xml_schema::properties())`  
*Parse a standard input stream with a resource id.*
- `::std::auto_ptr< ::xsd::ISO5436_2Type > xsd::ISO5436_2 (::std::istream &is, ::xercesc::DOMErrorHandler &eh, ::xml_schema::flags f=0, const ::xml_schema::properties &p=::xml_schema::properties())`  
*Parse a standard input stream with a Xerces-C++ DOM error handler.*
- `::std::auto_ptr< ::xsd::ISO5436_2Type > xsd::ISO5436_2 (::std::istream &is, ::xml_schema::error_handler &eh, ::xml_schema::flags f=0, const ::xml_schema::properties &p=::xml_schema::properties())`  
*Parse a standard input stream with an error handler.*
- `::std::auto_ptr< ::xsd::ISO5436_2Type > xsd::ISO5436_2 (::std::istream &is, ::xml_schema::flags f=0, const ::xml_schema::properties &p=::xml_schema::properties())`  
*Parse a standard input stream.*
- `::std::auto_ptr< ::xsd::ISO5436_2Type > xsd::ISO5436_2 (const ::std::wstring &uri, ::xercesc::DOMErrorHandler &eh, ::xml_schema::flags f=0, const ::xml_schema::properties &p=::xml_schema::properties())`  
*Parse a URI or a local file with a Xerces-C++ DOM error handler.*
- `::std::auto_ptr< ::xsd::ISO5436_2Type > xsd::ISO5436_2 (const ::std::wstring &uri, ::xml_schema::error_handler &eh, ::xml_schema::flags f=0, const ::xml_schema::properties &p=::xml_schema::properties())`  
*Parse a URI or a local file with an error handler.*



- `::std::auto_ptr< ::xsd::ISO5436_2Type > xsd::ISO5436_2 (const ::std::wstring &uri, ::xml_schema::flags f=0, const ::xml_schema::properties &p=::xml_schema::properties())`

*Parse a URI or a local file.*

## Functions

- `bool xsd::operator!= (const RotationType &x, const RotationType &y)`
- `bool xsd::operator!= (const MatrixDimensionType &x, const MatrixDimensionType &y)`
- `bool xsd::operator!= (const DataLinkType &x, const DataLinkType &y)`
- `bool xsd::operator!= (const DataListType &x, const DataListType &y)`
- `bool xsd::operator!= (const ProbingSystemType &x, const ProbingSystemType &y)`
- `bool xsd::operator!= (const InstrumentType &x, const InstrumentType &y)`
- `bool xsd::operator!= (const AxisDescriptionType &x, const AxisDescriptionType &y)`
- `bool xsd::operator!= (const AxesType &x, const AxesType &y)`
- `bool xsd::operator!= (const Record4Type &x, const Record4Type &y)`
- `bool xsd::operator!= (const Record3Type &x, const Record3Type &y)`
- `bool xsd::operator!= (const Record2Type &x, const Record2Type &y)`
- `bool xsd::operator!= (const ISO5436_2Type &x, const ISO5436_2Type &y)`
- `bool xsd::operator!= (const Record1Type &x, const Record1Type &y)`
- `void xsd::operator<< (::xsd::cxx::tree::list_stream< wchar_t > &l, const Datum &i)`
- `void xsd::operator<< (::xercesc::DOMAttr &a, const Datum &i)`
- `void xsd::operator<< (::xercesc::DOMElement &e, const Datum &i)`
- `void xsd::operator<< (::xsd::cxx::tree::list_stream< wchar_t > &l, const Type &i)`
- `void xsd::operator<< (::xercesc::DOMAttr &a, const Type &i)`
- `void xsd::operator<< (::xercesc::DOMElement &e, const Type &i)`
- `void xsd::operator<< (::xsd::cxx::tree::list_stream< wchar_t > &l, const DataType &i)`
- `void xsd::operator<< (::xercesc::DOMAttr &a, const DataType &i)`
- `void xsd::operator<< (::xercesc::DOMElement &e, const DataType &i)`
- `void xsd::operator<< (::xsd::cxx::tree::list_stream< wchar_t > &l, const AxisType &i)`
- `void xsd::operator<< (::xercesc::DOMAttr &a, const AxisType &i)`
- `void xsd::operator<< (::xercesc::DOMElement &e, const AxisType &i)`
- `void xsd::operator<< (::xsd::cxx::tree::list_stream< wchar_t > &l, const FeatureType &i)`
- `void xsd::operator<< (::xercesc::DOMAttr &a, const FeatureType &i)`
- `void xsd::operator<< (::xercesc::DOMElement &e, const FeatureType &i)`
- `void xsd::operator<< (::xsd::cxx::tree::list_stream< wchar_t > &l, const RotationMatrixElementType &i)`

- void `xsd::operator<<` (::xercesc::DOMAttr &a, const RotationMatrixElement-  
Type &i)
- void `xsd::operator<<` (::xercesc::DOMElement &e, const RotationMatrix-  
ElementType &i)
- void `xsd::operator<<` (::xercesc::DOMElement &e, const RotationType &i)
- void `xsd::operator<<` (::xercesc::DOMElement &e, const MatrixDimension-  
Type &i)
- void `xsd::operator<<` (::xercesc::DOMElement &e, const DataLinkType &i)
- void `xsd::operator<<` (::xercesc::DOMElement &e, const DataListType &i)
- void `xsd::operator<<` (::xercesc::DOMElement &e, const ProbingSystemType  
&i)
- void `xsd::operator<<` (::xercesc::DOMElement &e, const InstrumentType &i)
- void `xsd::operator<<` (::xercesc::DOMElement &e, const AxisDescriptionType  
&i)
- void `xsd::operator<<` (::xercesc::DOMElement &e, const AxesType &i)
- void `xsd::operator<<` (::xercesc::DOMElement &e, const Record4Type &i)
- void `xsd::operator<<` (::xercesc::DOMElement &e, const Record3Type &i)
- void `xsd::operator<<` (::xercesc::DOMElement &e, const Record2Type &i)
- void `xsd::operator<<` (::xercesc::DOMElement &e, const ISO5436\_2Type &i)
- void `xsd::operator<<` (::xercesc::DOMElement &e, const Record1Type &i)
- ::std::wostream & `xsd::operator<<` (::std::wostream &o, const Datum &i)
- ::std::wostream & `xsd::operator<<` (::std::wostream &o, const Type &i)
- ::std::wostream & `xsd::operator<<` (::std::wostream &o, Type::value i)
- ::std::wostream & `xsd::operator<<` (::std::wostream &o, const DataType &i)
- ::std::wostream & `xsd::operator<<` (::std::wostream &o, DataType::value i)
- ::std::wostream & `xsd::operator<<` (::std::wostream &o, const AxisType &i)
- ::std::wostream & `xsd::operator<<` (::std::wostream &o, AxisType::value i)
- ::std::wostream & `xsd::operator<<` (::std::wostream &o, const FeatureType &i)
- ::std::wostream & `xsd::operator<<` (::std::wostream &o, const RotationMatrix-  
ElementType &i)
- ::std::wostream & `xsd::operator<<` (::std::wostream &o, const RotationType  
&i)
- ::std::wostream & `xsd::operator<<` (::std::wostream &o, const Matrix-  
DimensionType &i)
- ::std::wostream & `xsd::operator<<` (::std::wostream &o, const DataLinkType  
&i)
- ::std::wostream & `xsd::operator<<` (::std::wostream &o, const DataListType  
&i)
- ::std::wostream & `xsd::operator<<` (::std::wostream &o, const ProbingSystem-  
Type &i)
- ::std::wostream & `xsd::operator<<` (::std::wostream &o, const InstrumentType  
&i)
- ::std::wostream & `xsd::operator<<` (::std::wostream &o, const AxisDescription-  
Type &i)
- ::std::wostream & `xsd::operator<<` (::std::wostream &o, const AxesType &i)
- ::std::wostream & `xsd::operator<<` (::std::wostream &o, const Record4Type &i)
- ::std::wostream & `xsd::operator<<` (::std::wostream &o, const Record3Type &i)

## 10.83

### S:/openGPS/ISO5436\_XML/trunk/src/ISO5436\_2\_XML/iso5436\_2\_xsd.hxx File Reference 290

- `::std::wostream & xsd::operator<< (::std::wostream &o, const Record2Type &i)`
- `::std::wostream & xsd::operator<< (::std::wostream &o, const ISO5436_2Type &i)`
- `::std::wostream & xsd::operator<< (::std::wostream &o, const Record1Type &i)`
- `bool xsd::operator== (const RotationType &x, const RotationType &y)`
- `bool xsd::operator== (const MatrixDimensionType &x, const MatrixDimensionType &y)`
- `bool xsd::operator== (const DataLinkType &x, const DataLinkType &y)`
- `bool xsd::operator== (const DataListType &x, const DataListType &y)`
- `bool xsd::operator== (const ProbingSystemType &x, const ProbingSystemType &y)`
- `bool xsd::operator== (const InstrumentType &x, const InstrumentType &y)`
- `bool xsd::operator== (const AxisDescriptionType &x, const AxisDescriptionType &y)`
- `bool xsd::operator== (const AxesType &x, const AxesType &y)`
- `bool xsd::operator== (const Record4Type &x, const Record4Type &y)`
- `bool xsd::operator== (const Record3Type &x, const Record3Type &y)`
- `bool xsd::operator== (const Record2Type &x, const Record2Type &y)`
- `bool xsd::operator== (const ISO5436_2Type &x, const ISO5436_2Type &y)`
- `bool xsd::operator== (const Record1Type &x, const Record1Type &y)`

## 10.83 S:/openGPS/ISO5436\_XML/trunk/src/ISO5436\_2\_XML/iso5436\_2\_xsd.hxx File Reference

### 10.83.1 Detailed Description

Generated from iso5436\_2.xsd.

```
#include <xsd/cxx/version.hxx>
#include <xsd/cxx/pre.hxx>
#include <xsd/cxx/tree/exceptions.hxx>
#include <xsd/cxx/tree/elements.hxx>
#include <xsd/cxx/tree/types.hxx>
#include <xsd/cxx/xml/error-handler.hxx>
#include <xsd/cxx/tree/parsing.hxx>
#include <xsd/cxx/tree/serialization.hxx>
#include <xsd/cxx/xml/dom/serialization-header.hxx>
#include <xsd/cxx/tree/std-ostream-operators.hxx>
#include <memory>
#include <algorithm>
#include <xsd/cxx/tree/containers.hxx>
#include <xsd/cxx/tree/list.hxx>
```

```
#include <xsd/cxx/xml/dom/parsing-header.hxx>
#include <iosfwd>
#include <xercesc/dom/DOMDocument.hpp>
#include <xercesc/dom/DOMInputSource.hpp>
#include <xercesc/dom/DOMErrorHandler.hpp>
#include <xercesc/framework/XMLFormatter.hpp>
#include <xsd/cxx/xml/dom/auto-ptr.hxx>
#include <xsd/cxx/post.hxx>
```

### Namespaces

- namespace [xml\\_schema](#)
- namespace [xsd](#)

### Classes

- class [xsd::AxesType](#)  
*Class corresponding to the AxesType schema type.*
- class [xsd::AxisDescriptionType](#)  
*Class corresponding to the AxisDescriptionType schema type.*
- class [xsd::AxisType](#)  
*Enumeration class corresponding to the AxisType schema type.*
- class [xsd::DataLinkType](#)  
*Class corresponding to the DataLinkType schema type.*
- class [xsd::DataListType](#)  
*Class corresponding to the DataListType schema type.*
- class [xsd::DataType](#)  
*Enumeration class corresponding to the DataType schema type.*
- class [xsd::Datum](#)  
*Class corresponding to the Datum schema type.*
- class [xsd::FeatureType](#)  
*Class corresponding to the FeatureType schema type.*
- class [xsd::InstrumentType](#)  
*Class corresponding to the InstrumentType schema type.*

- class [xsd::ISO5436\\_2Type](#)  
*Class corresponding to the ISO5436\_2Type schema type.*
- class [xsd::MatrixDimensionType](#)  
*Class corresponding to the MatrixDimensionType schema type.*
- class [xsd::ProbingSystemType](#)  
*Class corresponding to the ProbingSystemType schema type.*
- class [xsd::Record1Type](#)  
*Class corresponding to the Record1Type schema type.*
- class [xsd::Record2Type](#)  
*Class corresponding to the Record2Type schema type.*
- class [xsd::Record3Type](#)  
*Class corresponding to the Record3Type schema type.*
- class [xsd::Record4Type](#)  
*Class corresponding to the Record4Type schema type.*
- class [xsd::RotationMatrixElementType](#)  
*Class corresponding to the RotationMatrixElementType schema type.*
- class [xsd::RotationType](#)  
*Class corresponding to the RotationType schema type.*
- class [xsd::Type](#)  
*Enumeration class corresponding to the Type schema type.*

#### Serialization functions for the ISO5436\_2 document root.

The only global element: The root node

- `::xsd::cxx::xml::dom::auto_ptr< ::xercesc::DOMDocument > xsd::ISO5436\_2 (const ::xsd::ISO5436\_2Type &x, const ::xml\_schema::namespace\_infomap &m, ::xml\_schema::flags f=0)`  
*Serialize to a new Xerces-C++ DOM document.*
- `void xsd::ISO5436\_2 (::xercesc::DOMDocument &d, const ::xsd::ISO5436\_2Type &x, ::xml\_schema::flags f=0)`  
*Serialize to an existing Xerces-C++ DOM document.*

- void `xsd::ISO5436_2` (`::xercesc::XMLFormatTarget` &ft, const `::xsd::ISO5436_2Type` &x, const `::xml_schema::namespace_infomap` &m, `::xercesc::DOMErrorHandler` &eh, const `::std::wstring` &e=L"UTF-8", `::xml_schema::flags` f=0)  
*Serialize to a Xerces-C++ XML format target with a Xerces-C++ DOM error handler.*
- void `xsd::ISO5436_2` (`::xercesc::XMLFormatTarget` &ft, const `::xsd::ISO5436_2Type` &x, const `::xml_schema::namespace_infomap` &m, `::xml_schema::error_handler` &eh, const `::std::wstring` &e=L"UTF-8", `::xml_schema::flags` f=0)  
*Serialize to a Xerces-C++ XML format target with an error handler.*
- void `xsd::ISO5436_2` (`::xercesc::XMLFormatTarget` &ft, const `::xsd::ISO5436_2Type` &x, const `::xml_schema::namespace_infomap` &m, const `::std::wstring` &e=L"UTF-8", `::xml_schema::flags` f=0)  
*Serialize to a Xerces-C++ XML format target.*
- void `xsd::ISO5436_2` (`::std::ostream` &os, const `::xsd::ISO5436_2Type` &x, const `::xml_schema::namespace_infomap` &m, `::xercesc::DOMErrorHandler` &eh, const `::std::wstring` &e=L"UTF-8", `::xml_schema::flags` f=0)  
*Serialize to a standard output stream with a Xerces-C++ DOM error handler.*
- void `xsd::ISO5436_2` (`::std::ostream` &os, const `::xsd::ISO5436_2Type` &x, const `::xml_schema::namespace_infomap` &m, `::xml_schema::error_handler` &eh, const `::std::wstring` &e=L"UTF-8", `::xml_schema::flags` f=0)  
*Serialize to a standard output stream with an error handler.*
- void `xsd::ISO5436_2` (`::std::ostream` &os, const `::xsd::ISO5436_2Type` &x, const `::xml_schema::namespace_infomap` &m, const `::std::wstring` &e=L"UTF-8", `::xml_schema::flags` f=0)  
*Serialize to a standard output stream.*

### Parsing functions for the ISO5436\_2 document root.

The only global element: The root node

- `::std::auto_ptr< ::xsd::ISO5436_2Type > xsd::ISO5436_2` (`::xercesc::DOMDocument` \*d, `::xml_schema::flags` f=0, const `::xml_schema::properties` &p=`::xml_schema::properties()`)  
*Parse a Xerces-C++ DOM document.*
- `::std::auto_ptr< ::xsd::ISO5436_2Type > xsd::ISO5436_2` (const `::xercesc::DOMDocument` &d, `::xml_schema::flags` f=0, const `::xml_schema::properties` &p=`::xml_schema::properties()`)  
*Parse a Xerces-C++ DOM document.*

- `::std::auto_ptr< ::xsd::ISO5436_2Type > xsd::ISO5436_2 (const ::xercesc::DOMInputSource &is, ::xercesc::DOMErrorHandler &eh, ::xml_schema::flags f=0, const ::xml_schema::properties &p=::xml_schema::properties())`

*Parse a Xerces-C++ DOM input source with a Xerces-C++ DOM error handler.*

- `::std::auto_ptr< ::xsd::ISO5436_2Type > xsd::ISO5436_2 (const ::xercesc::DOMInputSource &is, ::xml_schema::error_handler &eh, ::xml_schema::flags f=0, const ::xml_schema::properties &p=::xml_schema::properties())`

*Parse a Xerces-C++ DOM input source with an error handler.*

- `::std::auto_ptr< ::xsd::ISO5436_2Type > xsd::ISO5436_2 (const ::xercesc::DOMInputSource &is, ::xml_schema::flags f=0, const ::xml_schema::properties &p=::xml_schema::properties())`

*Parse a Xerces-C++ DOM input source.*

- `::std::auto_ptr< ::xsd::ISO5436_2Type > xsd::ISO5436_2 (::std::istream &is, const ::std::wstring &id, ::xercesc::DOMErrorHandler &eh, ::xml_schema::flags f=0, const ::xml_schema::properties &p=::xml_schema::properties())`

*Parse a standard input stream with a resource id and a Xerces-C++ DOM error handler.*

- `::std::auto_ptr< ::xsd::ISO5436_2Type > xsd::ISO5436_2 (::std::istream &is, const ::std::wstring &id, ::xml_schema::error_handler &eh, ::xml_schema::flags f=0, const ::xml_schema::properties &p=::xml_schema::properties())`

*Parse a standard input stream with a resource id and an error handler.*

- `::std::auto_ptr< ::xsd::ISO5436_2Type > xsd::ISO5436_2 (::std::istream &is, const ::std::wstring &id, ::xml_schema::flags f=0, const ::xml_schema::properties &p=::xml_schema::properties())`

*Parse a standard input stream with a resource id.*

- `::std::auto_ptr< ::xsd::ISO5436_2Type > xsd::ISO5436_2 (::std::istream &is, ::xercesc::DOMErrorHandler &eh, ::xml_schema::flags f=0, const ::xml_schema::properties &p=::xml_schema::properties())`

*Parse a standard input stream with a Xerces-C++ DOM error handler.*

- `::std::auto_ptr< ::xsd::ISO5436_2Type > xsd::ISO5436_2 (::std::istream &is, ::xml_schema::error_handler &eh, ::xml_schema::flags f=0, const ::xml_schema::properties &p=::xml_schema::properties())`

*Parse a standard input stream with an error handler.*

- `::std::auto_ptr< ::xsd::ISO5436_2Type > xsd::ISO5436_2 (::std::istream &is, ::xml_schema::flags f=0, const ::xml_schema::properties &p=::xml_schema::properties())`

*Parse a standard input stream.*

- `::std::auto_ptr< ::xsd::ISO5436_2Type > xsd::ISO5436_2 (const ::std::wstring &uri, ::xercesc::DOMErrorHandler &eh, ::xml_schema::flags f=0, const ::xml_schema::properties &p=::xml_schema::properties())`

*Parse a URI or a local file with a Xerces-C++ DOM error handler.*

- `::std::auto_ptr< ::xsd::ISO5436_2Type > xsd::ISO5436_2 (const ::std::wstring &uri, ::xml_schema::error_handler &eh, ::xml_schema::flags f=0, const ::xml_schema::properties &p=::xml_schema::properties())`

*Parse a URI or a local file with an error handler.*

- `::std::auto_ptr< ::xsd::ISO5436_2Type > xsd::ISO5436_2 (const ::std::wstring &uri, ::xml_schema::flags f=0, const ::xml_schema::properties &p=::xml_schema::properties())`

*Parse a URI or a local file.*

## Typedefs

- `typedef ::xsd::cxx::tree::base64_binary< wchar_t, simple_type > xml_schema::base64_binary`
- `typedef bool xml_schema::boolean`
- `typedef ::xsd::cxx::tree::bounds< wchar_t > xml_schema::bounds`
- `typedef ::xsd::cxx::tree::buffer< wchar_t > xml_schema::buffer`
- `typedef signed char xml_schema::byte`
- `typedef ::xsd::cxx::tree::date< wchar_t, simple_type > xml_schema::date`
- `typedef ::xsd::cxx::tree::date_time< wchar_t, simple_type > xml_schema::date_time`
- `typedef ::xsd::cxx::tree::day< wchar_t, simple_type > xml_schema::day`
- `typedef double xml_schema::decimal`
- `typedef ::xsd::cxx::tree::diagnostics< wchar_t > xml_schema::diagnostics`
- `typedef double xml_schema::double_`
- `typedef ::xsd::cxx::tree::duplicate_id< wchar_t > xml_schema::duplicate_id`
- `typedef ::xsd::cxx::tree::duration< wchar_t, simple_type > xml_schema::duration`
- `typedef ::xsd::cxx::tree::entities< wchar_t, simple_type, entity > xml_schema::entities`
- `typedef ::xsd::cxx::tree::entity< wchar_t, ncname > xml_schema::entity`
- `typedef ::xsd::cxx::tree::error< wchar_t > xml_schema::error`
- `typedef ::xsd::cxx::xml::error_handler< wchar_t > xml_schema::error_handler`
- `typedef ::xsd::cxx::tree::exception< wchar_t > xml_schema::exception`
- `typedef ::xsd::cxx::tree::expected_attribute< wchar_t > xml_schema::expected_attribute`
- `typedef ::xsd::cxx::tree::expected_element< wchar_t > xml_schema::expected_element`
- `typedef ::xsd::cxx::tree::expected_text_content< wchar_t > xml_schema::expected_text_content`
- `typedef ::xsd::cxx::tree::flags xml_schema::flags`



- typedef float [xml\\_schema::float\\_](#)
- typedef [::xsd::cxx::tree::hex\\_binary](#)< [wchar\\_t](#), [simple\\_type](#) > [xml\\_schema::hex\\_binary](#)
- typedef [::xsd::cxx::tree::id](#)< [wchar\\_t](#), [ncname](#) > [xml\\_schema::id](#)
- typedef [::xsd::cxx::tree::idref](#)< [type](#), [wchar\\_t](#), [ncname](#) > [xml\\_schema::idref](#)
- typedef [::xsd::cxx::tree::idrefs](#)< [wchar\\_t](#), [simple\\_type](#), [idref](#) > [xml\\_schema::idrefs](#)
- typedef int [xml\\_schema::int\\_](#)
- typedef long long [xml\\_schema::integer](#)
- typedef [::xsd::cxx::tree::language](#)< [wchar\\_t](#), [token](#) > [xml\\_schema::language](#)
- typedef long long [xml\\_schema::long\\_](#)
- typedef [::xsd::cxx::tree::month](#)< [wchar\\_t](#), [simple\\_type](#) > [xml\\_schema::month](#)
- typedef [::xsd::cxx::tree::month\\_day](#)< [wchar\\_t](#), [simple\\_type](#) > [xml\\_schema::month\\_day](#)
- typedef [::xsd::cxx::tree::name](#)< [wchar\\_t](#), [token](#) > [xml\\_schema::name](#)
- typedef [::xsd::cxx::xml::dom::namespace\\_info](#)< [wchar\\_t](#) > [xml\\_schema::namespace\\_info](#)
- typedef [::xsd::cxx::xml::dom::namespace\\_infomap](#)< [wchar\\_t](#) > [xml\\_schema::namespace\\_infomap](#)
- typedef [::xsd::cxx::tree::ncname](#)< [wchar\\_t](#), [name](#) > [xml\\_schema::ncname](#)
- typedef integer [xml\\_schema::negative\\_integer](#)
- typedef [::xsd::cxx::tree::nmtoken](#)< [wchar\\_t](#), [token](#) > [xml\\_schema::nmtoken](#)
- typedef [::xsd::cxx::tree::nmtokens](#)< [wchar\\_t](#), [simple\\_type](#), [nmtoken](#) > [xml\\_schema::nmtokens](#)
- typedef [::xsd::cxx::tree::no\\_namespace\\_mapping](#)< [wchar\\_t](#) > [xml\\_schema::no\\_namespace\\_mapping](#)
- typedef [::xsd::cxx::tree::no\\_prefix\\_mapping](#)< [wchar\\_t](#) > [xml\\_schema::no\\_prefix\\_mapping](#)
- typedef [::xsd::cxx::tree::no\\_type\\_info](#)< [wchar\\_t](#) > [xml\\_schema::no\\_type\\_info](#)
- typedef integer [xml\\_schema::non\\_negative\\_integer](#)
- typedef integer [xml\\_schema::non\\_positive\\_integer](#)
- typedef [::xsd::cxx::tree::normalized\\_string](#)< [wchar\\_t](#), [string](#) > [xml\\_schema::normalized\\_string](#)
- typedef [::xsd::cxx::tree::not\\_derived](#)< [wchar\\_t](#) > [xml\\_schema::not\\_derived](#)
- typedef [::xsd::cxx::tree::parsing](#)< [wchar\\_t](#) > [xml\\_schema::parsing](#)
- typedef integer [xml\\_schema::positive\\_integer](#)
- typedef [::xsd::cxx::tree::properties](#)< [wchar\\_t](#) > [xml\\_schema::properties](#)
- typedef [::xsd::cxx::tree::qname](#)< [wchar\\_t](#), [simple\\_type](#), [uri](#), [ncname](#) > [xml\\_schema::qname](#)
- typedef [::xsd::cxx::tree::serialization](#)< [wchar\\_t](#) > [xml\\_schema::serialization](#)
- typedef [::xsd::cxx::tree::severity](#) [xml\\_schema::severity](#)
- typedef short [xml\\_schema::short\\_](#)
- typedef [::xsd::cxx::tree::simple\\_type](#)< [type](#) > [xml\\_schema::simple\\_type](#)
- typedef [::xsd::cxx::tree::string](#)< [wchar\\_t](#), [simple\\_type](#) > [xml\\_schema::string](#)
- typedef [::xsd::cxx::tree::time](#)< [wchar\\_t](#), [simple\\_type](#) > [xml\\_schema::time](#)
- typedef [::xsd::cxx::tree::token](#)< [wchar\\_t](#), [normalized\\_string](#) > [xml\\_schema::token](#)

- typedef ::xsd::cxx::tree::type xml\_schema::type
- typedef ::xsd::cxx::tree::unexpected\_element< wchar\_t > xml\_schema::unexpected\_element
- typedef ::xsd::cxx::tree::unexpected\_enumerator< wchar\_t > xml\_schema::unexpected\_enumerator
- typedef unsigned char xml\_schema::unsigned\_byte
- typedef unsigned int xml\_schema::unsigned\_int
- typedef unsigned long long xml\_schema::unsigned\_long
- typedef unsigned short xml\_schema::unsigned\_short
- typedef ::xsd::cxx::tree::uri< wchar\_t, simple\_type > xml\_schema::uri
- typedef ::xsd::cxx::tree::xsi\_already\_in\_use< wchar\_t > xml\_schema::xsi\_already\_in\_use
- typedef ::xsd::cxx::tree::year< wchar\_t, simple\_type > xml\_schema::year
- typedef ::xsd::cxx::tree::year\_month< wchar\_t, simple\_type > xml\_schema::year\_month

## Functions

- bool xsd::operator!= (const RotationType &x, const RotationType &y)
- bool xsd::operator!= (const MatrixDimensionType &x, const MatrixDimensionType &y)
- bool xsd::operator!= (const DataLinkType &x, const DataLinkType &y)
- bool xsd::operator!= (const DataListType &x, const DataListType &y)
- bool xsd::operator!= (const ProbingSystemType &x, const ProbingSystemType &y)
- bool xsd::operator!= (const InstrumentType &x, const InstrumentType &y)
- bool xsd::operator!= (const AxisDescriptionType &x, const AxisDescriptionType &y)
- bool xsd::operator!= (const AxesType &x, const AxesType &y)
- bool xsd::operator!= (const Record4Type &x, const Record4Type &y)
- bool xsd::operator!= (const Record3Type &x, const Record3Type &y)
- bool xsd::operator!= (const Record2Type &x, const Record2Type &y)
- bool xsd::operator!= (const ISO5436\_2Type &x, const ISO5436\_2Type &y)
- bool xsd::operator!= (const Record1Type &x, const Record1Type &y)
- void xsd::operator<< (::xsd::cxx::tree::list\_stream< wchar\_t > &l, const Datum &i)
- void xsd::operator<< (::xercesc::DOMAttr &a, const Datum &i)
- void xsd::operator<< (::xercesc::DOMElement &e, const Datum &i)
- void xsd::operator<< (::xsd::cxx::tree::list\_stream< wchar\_t > &l, const Type &i)
- void xsd::operator<< (::xercesc::DOMAttr &a, const Type &i)
- void xsd::operator<< (::xercesc::DOMElement &e, const Type &i)
- void xsd::operator<< (::xsd::cxx::tree::list\_stream< wchar\_t > &l, const DataType &i)
- void xsd::operator<< (::xercesc::DOMAttr &a, const DataType &i)
- void xsd::operator<< (::xercesc::DOMElement &e, const DataType &i)

- void `xsd::operator<<` (::xsd::cxx::tree::list\_stream< wchar\_t > &l, const AxisType &i)
- void `xsd::operator<<` (::xercesc::DOMAttr &a, const AxisType &i)
- void `xsd::operator<<` (::xercesc::DOMElement &e, const AxisType &i)
- void `xsd::operator<<` (::xsd::cxx::tree::list\_stream< wchar\_t > &l, const FeatureType &i)
- void `xsd::operator<<` (::xercesc::DOMAttr &a, const FeatureType &i)
- void `xsd::operator<<` (::xercesc::DOMElement &e, const FeatureType &i)
- void `xsd::operator<<` (::xsd::cxx::tree::list\_stream< wchar\_t > &l, const RotationMatrixElementType &i)
- void `xsd::operator<<` (::xercesc::DOMAttr &a, const RotationMatrixElementType &i)
- void `xsd::operator<<` (::xercesc::DOMElement &e, const RotationMatrixElementType &i)
- void `xsd::operator<<` (::xercesc::DOMElement &e, const RotationType &i)
- void `xsd::operator<<` (::xercesc::DOMElement &e, const MatrixDimensionType &i)
- void `xsd::operator<<` (::xercesc::DOMElement &e, const DataLinkType &i)
- void `xsd::operator<<` (::xercesc::DOMElement &e, const DataListType &i)
- void `xsd::operator<<` (::xercesc::DOMElement &e, const ProbingSystemType &i)
- void `xsd::operator<<` (::xercesc::DOMElement &e, const InstrumentType &i)
- void `xsd::operator<<` (::xercesc::DOMElement &e, const AxisDescriptionType &i)
- void `xsd::operator<<` (::xercesc::DOMElement &e, const AxesType &i)
- void `xsd::operator<<` (::xercesc::DOMElement &e, const Record4Type &i)
- void `xsd::operator<<` (::xercesc::DOMElement &e, const Record3Type &i)
- void `xsd::operator<<` (::xercesc::DOMElement &e, const Record2Type &i)
- void `xsd::operator<<` (::xercesc::DOMElement &e, const ISO5436\_2Type &i)
- void `xsd::operator<<` (::xercesc::DOMElement &e, const Record1Type &i)
- ::std::wostream & `xsd::operator<<` (::std::wostream &o, const Datum &i)
- ::std::wostream & `xsd::operator<<` (::std::wostream &o, const Type &i)
- ::std::wostream & `xsd::operator<<` (::std::wostream &o, Type::value i)
- ::std::wostream & `xsd::operator<<` (::std::wostream &o, const DataType &i)
- ::std::wostream & `xsd::operator<<` (::std::wostream &o, DataType::value i)
- ::std::wostream & `xsd::operator<<` (::std::wostream &o, const AxisType &i)
- ::std::wostream & `xsd::operator<<` (::std::wostream &o, AxisType::value i)
- ::std::wostream & `xsd::operator<<` (::std::wostream &o, const FeatureType &i)
- ::std::wostream & `xsd::operator<<` (::std::wostream &o, const RotationMatrixElementType &i)
- ::std::wostream & `xsd::operator<<` (::std::wostream &o, const RotationType &i)
- ::std::wostream & `xsd::operator<<` (::std::wostream &o, const MatrixDimensionType &i)
- ::std::wostream & `xsd::operator<<` (::std::wostream &o, const DataLinkType &i)

- `::std::wostream & xsd::operator<< (::std::wostream &o, const DataListType &i)`
- `::std::wostream & xsd::operator<< (::std::wostream &o, const ProbingSystemType &i)`
- `::std::wostream & xsd::operator<< (::std::wostream &o, const InstrumentType &i)`
- `::std::wostream & xsd::operator<< (::std::wostream &o, const AxisDescriptionType &i)`
- `::std::wostream & xsd::operator<< (::std::wostream &o, const AxesType &i)`
- `::std::wostream & xsd::operator<< (::std::wostream &o, const Record4Type &i)`
- `::std::wostream & xsd::operator<< (::std::wostream &o, const Record3Type &i)`
- `::std::wostream & xsd::operator<< (::std::wostream &o, const Record2Type &i)`
- `::std::wostream & xsd::operator<< (::std::wostream &o, const ISO5436_2Type &i)`
- `::std::wostream & xsd::operator<< (::std::wostream &o, const Record1Type &i)`
- `bool xsd::operator== (const RotationType &x, const RotationType &y)`
- `bool xsd::operator== (const MatrixDimensionType &x, const MatrixDimensionType &y)`
- `bool xsd::operator== (const DataLinkType &x, const DataLinkType &y)`
- `bool xsd::operator== (const DataListType &x, const DataListType &y)`
- `bool xsd::operator== (const ProbingSystemType &x, const ProbingSystemType &y)`
- `bool xsd::operator== (const InstrumentType &x, const InstrumentType &y)`
- `bool xsd::operator== (const AxisDescriptionType &x, const AxisDescriptionType &y)`
- `bool xsd::operator== (const AxesType &x, const AxesType &y)`
- `bool xsd::operator== (const Record4Type &x, const Record4Type &y)`
- `bool xsd::operator== (const Record3Type &x, const Record3Type &y)`
- `bool xsd::operator== (const Record2Type &x, const Record2Type &y)`
- `bool xsd::operator== (const ISO5436_2Type &x, const ISO5436_2Type &y)`
- `bool xsd::operator== (const Record1Type &x, const Record1Type &y)`

#### Variables

- `const XMLCh *const xml_schema::tree_node_key = ::xsd::cxx::tree::user_data_keys::node`

#### 10.84 S:/openGPS/ISO5436\_XML/trunk/src/ISO5436\_2\_XML\_- Demo/ISO5436\_2\_XML\_Demo.cpp File Reference

```
#include <opengps/iso5436_2.h>
#include <opengps/iso5436_2_handle.hxx>
#include <opengps/iso5436_2.hxx>
#include <opengps/iso5436_2_xsd.hxx>
#include <opengps/point_iterator.hxx>
```

```
#include <opengps/point_vector.hxx>
#include <opengps/data_point.hxx>
#include <string>
#include <iostream>
#include <ostream>
#include <tchar.h>
```

## Functions

- `int _tmain (int argc, _TCHAR *argv[ ])`
- `void mediumComplexExample (OpenGPS::String fileName)`
- `void readonlyExample (OpenGPS::String fileName)`
- `void readonlyExample2 (OpenGPS::String fileName)`
- `void readonlyExample3 (OpenGPS::String fileName)`
- `void simpleExample (OpenGPS::String fileName)`

### 10.84.1 Function Documentation

**10.84.1.1** `int _tmain (int argc, _TCHAR * argv[ ])`

**10.84.1.2** `void mediumComplexExample (OpenGPS::String fileName)`

**10.84.1.3** `void readonlyExample (OpenGPS::String fileName)`

**10.84.1.4** `void readonlyExample2 (OpenGPS::String fileName)`

**10.84.1.5** `void readonlyExample3 (OpenGPS::String fileName)`

**10.84.1.6** `void simpleExample (OpenGPS::String fileName)`