XBinder

XML Schema Compiler
Version 2.1
C++ XML Runtime
Reference Manual

Objective Systems, Inc. — February 2010

Author's Contact Information

Comments, suggestions, and inquiries regarding XBinder may be submitted via electronic mail to info@obj-sys.com.

Contents

Chapter 1

XBinder Main Page

C++ XML Runtime Library Classes

The **C++ XML run-time classes** are wrapper classes that provide an object-oriented interface to the common C XML run-time library functions. The categories of classes provided are as follows:

- XML encode functions.
- XML decode functions.
- SAX interfaces.
- Objects for global elements, namespaces, and message buffers.

Chapter 2

XBinder Hierarchical Index

2.1 XBinder Class Hierarchy

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

Chapter 3

XBinder Class Index

3.1 XBinder Class List

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

Chapter 4

XBinder File Index

4.1 XBinder File List

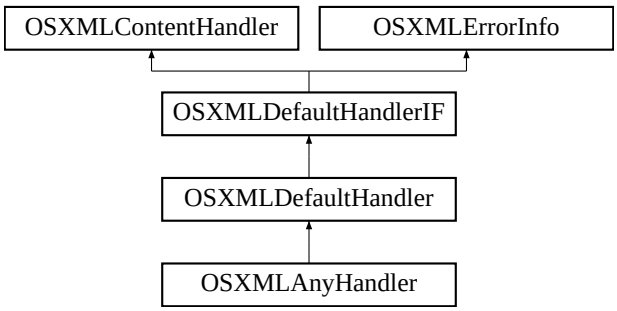
Here is a list of all documented files with brief descriptions:

Chapter 5

XBinder Class Documentation

5.1 OSXMLAnyHandler Class Reference

Inheritance diagram for OSXMLAnyHandler::



Public Member Functions

- virtual EXTXMLMETHOD int startElement (const OSUTF8CHAR ∗const uri, const OSUTF8CHAR ∗const localname, const OSUTF8CHAR ∗const qname, const OSUTF8CHAR ∗const ∗attrs)

  Receive notification of the beginning of an element.

- virtual EXTXMLMETHOD int endElement (const OSUTF8CHAR ∗const uri, const OSUTF8CHAR ∗const localname, const OSUTF8CHAR ∗const qname)

  Receive notification of the end of an element.

5.1.1 Detailed Description

Definition at line 38 of file rtSaxCppAny.h.

5.1.2 Member Function Documentation

5.1.2.1 virtual EXTXMLMETHOD int OSXMLAnyHandler::startElement (const OSUTF8CHAR *const *uri*, const OSUTF8CHAR *const *localname*, const OSUTF8CHAR *const *qname*, const OSUTF8CHAR *const * *attrs*) [virtual]

Receive notification of the beginning of an element.

The Parser will invoke this method at the beginning of every element in the XML document; there will be a corresponding endElement() event for every startElement() event (even when the element is empty). All of the element's content will be reported, in order, before the corresponding endElement() event.

Parameters:

uri The URI of the asscioated namespace for this element

localname The local part of the element name

qname The QName of this element

attrs The attributes name/value pairs attached to the element, if any.

See also:

endElement

Reimplemented from OSXMLDefaultHandler.

5.1.2.2 virtual EXTXMLMETHOD int OSXMLAnyHandler::endElement (const OSUTF8CHAR *const *uri*, const OSUTF8CHAR *const *localname*, const OSUTF8CHAR *const *qname*) [virtual]

Receive notification of the end of an element.

The SAX parser will invoke this method at the end of every element in the XML document; there will be a corresponding startElement() event for every endElement() event (even when the element is empty).

Parameters:

uri The URI of the asscioated namespace for this element

localname The local part of the element name

qname The QName of this element

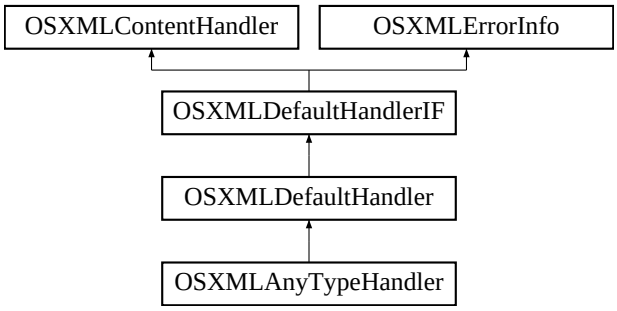
Reimplemented from OSXMLDefaultHandler.

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

- rtSaxCppAny.h

5.2 OSXMLAnyTypeHandler Class Reference

Inheritance diagram for OSXMLAnyTypeHandler::



Public Member Functions

- virtual int startElement (const OSUTF8CHAR *const uri, const OSUTF8CHAR *const localname, const OSUTF8CHAR *const qname, const OSUTF8CHAR *const *attrs)

  Receive notification of the beginning of an element.

- virtual int endElement (const OSUTF8CHAR *const uri, const OSUTF8CHAR *const localname, const OSUTF8CHAR *const qname)

  Receive notification of the end of an element.

5.2.1 Detailed Description

Definition at line 39 of file rtSaxCppAnyType.h.

5.2.2 Member Function Documentation

5.2.2.1 virtual int OSXMLAnyTypeHandler::startElement (const OSUTF8CHAR *const *uri*, const OSUTF8CHAR *const *localname*, const OSUTF8CHAR *const *qname*, const OSUTF8CHAR *const * *attrs*) [virtual]

Receive notification of the beginning of an element.

The Parser will invoke this method at the beginning of every element in the XML document; there will be a corresponding endElement() event for every startElement() event (even when the element is empty). All of the element's content will be reported, in order, before the corresponding endElement() event.

Parameters:

uri The URI of the asscioated namespace for this element

localname The local part of the element name

qname The QName of this element

attrs The attributes name/value pairs attached to the element, if any.

See also:

endElement

Reimplemented from OSXMLDefaultHandler.

5.2.2.2 virtual int OSXMLAnyTypeHandler::endElement (const OSUTF8CHAR *const *uri*, const OSUTF8CHAR *const *localname*, const OSUTF8CHAR *const *qname*) [virtual]

Receive notification of the end of an element.

The SAX parser will invoke this method at the end of every element in the XML document; there will be a corresponding startElement() event for every endElement() event (even when the element is empty).

Parameters:

uri The URI of the asscioated namespace for this element

localname The local part of the element name

qname The QName of this element

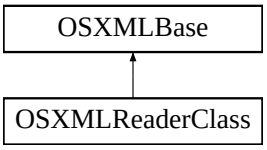
Reimplemented from OSXMLDefaultHandler.

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

- rtSaxCppAnyType.h

5.3 OSXMLBase Class Reference

Inheritance diagram for OSXMLBase::



Protected Member Functions

- OSXMLBase ()
- virtual ~OSXMLBase ()

5.3.1 Detailed Description

Definition at line 186 of file rtSaxCppParserIF.h.

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

- rtSaxCppParserIF.h

5.4 OSXMLBasePtr Class Reference

Public Member Functions

- OSXMLBasePtr ()
- OSXMLBasePtr (OSXMLBase *ptr)
- ~OSXMLBasePtr ()
- operator OSXMLBase * () const
- OSXMLBase * operator= (OSXMLBase *ptr)

5.4.1 Detailed Description

Definition at line 35 of file rtSaxCppParser.h.

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

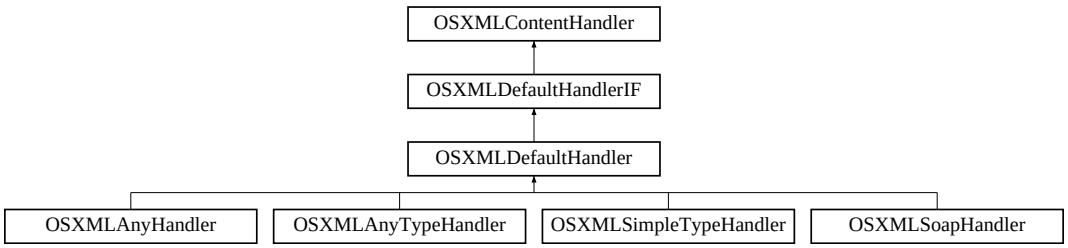
- rtSaxCppParser.h

5.5 OSXMLContentHandler Class Reference

Receive notification of general document events.

`#include <rtSaxCppParserIF.h>`

Inheritance diagram for OSXMLContentHandler::



Public Member Functions

- virtual ~OSXMLContentHandler ()

The virtual document handler interface

- virtual int characters (const OSUTF8CHAR *const chars, unsigned int length)=0
  Receive notification of character data.
- virtual int endElement (const OSUTF8CHAR *const uri, const OSUTF8CHAR *const localname, const OSUTF8CHAR *const qname)=0
  Receive notification of the end of an element.
- virtual int startElement (const OSUTF8CHAR *const uri, const OSUTF8CHAR *const localname, const OSUTF8CHAR *const qname, const OSUTF8CHAR *const *attrs)=0
  Receive notification of the beginning of an element.

5.5.1 Detailed Description

Receive notification of general document events.

Definition at line 112 of file rtSaxCppParserIF.h.

5.5.2 Member Function Documentation

5.5.2.1 virtual int OSXMLContentHandler::characters (const OSUTF8CHAR *const *chars*, unsigned int *length*) [pure virtual]

Receive notification of character data.

The Parser will call this method to report each chunk of character data. SAX parsers may return all contiguous character data in a single chunk, or they may split it into several chunks; however, all of the characters in any single event must come from the same external entity, so that the Locator provides useful information.

Parameters:

chars The characters from the XML document.

length The length of chars.

Implemented in OSXMLDefaultHandler, OSXMLSimpleTypeHandler, and OSXMLSoapHandler.

5.5.2.2 virtual int OSXMLContentHandler::endElement (const OSUTF8CHAR *const *uri*, const OSUTF8CHAR *const *localname*, const OSUTF8CHAR *const *qname*) [pure virtual]

Receive notification of the end of an element.

The SAX parser will invoke this method at the end of every element in the XML document; there will be a corresponding startElement() event for every endElement() event (even when the element is empty).

Parameters:

uri The URI of the asscioated namespace for this element

localname The local part of the element name

qname The QName of this element

Implemented in OSXMLAnyHandler, OSXMLAnyTypeHandler, OSXMLDefaultHandler, OSXMLSimpleTypeHandler, and OSXMLSoapHandler.

5.5.2.3 virtual int OSXMLContentHandler::startElement (const OSUTF8CHAR *const *uri*, const OSUTF8CHAR *const *localname*, const OSUTF8CHAR *const *qname*, const OSUTF8CHAR *const * *attrs*) [pure virtual]

Receive notification of the beginning of an element.

The Parser will invoke this method at the beginning of every element in the XML document; there will be a corresponding endElement() event for every startElement() event (even when the element is empty). All of the element's content will be reported, in order, before the corresponding endElement() event.

Parameters:

uri The URI of the asscioated namespace for this element

localname The local part of the element name

qname The QName of this element

attrs The attributes name/value pairs attached to the element, if any.

See also:

endElement

Implemented in OSXMLAnyHandler, OSXMLAnyTypeHandler, OSXMLDefaultHandler, OSXMLSimpleTypeHandler, and OSXMLSoapHandler.

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

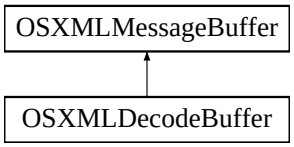
- rtSaxCppParserIF.h

5.6 OSXMLDecodeBuffer Class Reference

The OSXMLDecodeBuffer class is derived from the OSXMLMessageBuffer base class.

`#include <OSXMLDecodeBuffer.h>`

Inheritance diagram for OSXMLDecodeBuffer::



Public Member Functions

- OSXMLDecodeBuffer (const char *xmlFile)

  This version of the OSXMLDecodeBuffer constructor takes a name of a file that contains XML data to be decoded and constructs a buffer.

- OSXMLDecodeBuffer (const OSOCTET *msgbuf, size_t bufsiz)

  This version of the OSXMLDecodeBuffer constructor takes parameters describing a message in memory to be decoded and constructs a buffer.

- OSXMLDecodeBuffer (OSRTInputStream &inputStream)

  This version of the OSXMLDecodeBuffer constructor takes a reference to the OSInputStream object.

- EXTXMLMETHOD int decodeXML (OSXMLReaderClass *pReader)

  This method decodes an XML message associated with this buffer.

- virtual EXTXMLMETHOD int init ()

  This method initializes the decode message buffer.

- EXTXMLMETHOD int parseElementName (OSUTF8CHAR **ppName)

  This method parses the initial tag from an XML message.

- EXTXMLMETHOD int parseElemQName (OSXMLQName *pQName)

  This method parses the initial tag from an XML message.

- EXTXMLMETHOD OSUINT32 setMaxErrors (OSUINT32 maxErrors)

  This method sets the maximum number of errors returned by the SAX parser.

- virtual OSBOOL isA (int bufferType)

  This is a virtual method that must be overridden by derived classes to allow identification of the class.

Protected Attributes

- OSRTInputStream * mpInputStream

  Input source for message to be decoded.

- OSBOOL mbOwnStream

  This is set to true if this object creates the underlying stream object.

5.6.1 Detailed Description

The OSXMLDecodeBuffer class is derived from the OSXMLMessageBuffer base class.

It contains variables and methods specific to decoding XML messages. It is used to manage an input buffer or stream containing a message to be decoded.

Note that the XML decode buffer object does not take a message buffer argument because buffer management is handled by the XML parser.

Definition at line 45 of file OSXMLDecodeBuffer.h.

5.6.2 Constructor & Destructor Documentation

5.6.2.1 OSXMLDecodeBuffer::OSXMLDecodeBuffer (const char * *xmlFile*)

This version of the OSXMLDecodeBuffer constructor takes a name of a file that contains XML data to be decoded and constructs a buffer.

Parameters:

xmlFile A pointer to name of file to be decoded.

5.6.2.2 OSXMLDecodeBuffer::OSXMLDecodeBuffer (const OSOCTET * *msgbuf*, size_t *bufsiz*)

This version of the OSXMLDecodeBuffer constructor takes parameters describing a message in memory to be decoded and constructs a buffer.

Parameters:

msgbuf A pointer to a buffer containing an XML message.

bufsiz Size of the message buffer.

5.6.2.3 OSXMLDecodeBuffer::OSXMLDecodeBuffer (OSRTInputStream & *inputStream*)

This version of the OSXMLDecodeBuffer constructor takes a reference to the OSInputStream object.

The stream is assumed to have been previuously initialized to point at an encoded XML message.

Parameters:

inputStream reference to the OSInputStream object

5.6.3 Member Function Documentation

5.6.3.1 EXTXMLMETHOD int OSXMLDecodeBuffer::decodeXML (OSXMLReaderClass * *pReader*)

This method decodes an XML message associated with this buffer.

Returns:

stat Status of the operation. Possible values are 0 if successful or one of the negative error status codes defined in Appendix A of the C/C++ runtime Common Functions Reference Manual.

Parameters:

pReader Pointer to OSXMLReaderClass object.

Returns:

Completion status.

5.6.3.2 virtual EXTXMLMETHOD int OSXMLDecodeBuffer::init () [virtual]

This method initializes the decode message buffer.

Returns:

Completion status of operation:

- 0 (0) = success,
- negative return value is error.

5.6.3.3 EXTXMLMETHOD int OSXMLDecodeBuffer::parseElementName (OSUTF8CHAR ** *ppName*)

This method parses the initial tag from an XML message.

If the tag is a QName, only the local part of the name is returned.

Parameters:

ppName Pointer to a pointer to receive decoded UTF-8 string. Dynamic memory is allocated for the variable using the rtxMemAlloc function.

Returns:

Completion status of operation:

- 0 = success,
- negative return value is error.

5.6.3.4 EXTXMLMETHOD int OSXMLDecodeBuffer::parseElemQName (OSXMLQName * *pQName*)

This method parses the initial tag from an XML message.

Parameters:

pQName Pointer to a QName structure to receive parsed name prefix and local name. Dynamic memory is allocated for both name parts using the rtxMemAlloc function.

Returns:

Completion status of operation:

- 0 = success,
- negative return value is error.

5.6.3.5 EXTXMLMETHOD OSUINT32 OSXMLDecodeBuffer::setMaxErrors (OSUINT32 *maxErrors*)

This method sets the maximum number of errors returned by the SAX parser.

Parameters:

maxErrors The desired number of maximum errors.

Returns:

The previously set maximum number of errors.

5.6.3.6 virtual OSBOOL OSXMLDecodeBuffer::isA (int *bufferType*) `[inline, virtual]`

This is a virtual method that must be overridden by derived classes to allow identification of the class.

The base class variant is abstract. This method matches an enumerated identifier defined in the base class. One identifier is declared for each of the derived classes.

Parameters:

bufferType Enumerated identifier specifying a derived class. This type is defined as a public access type in the OSRTMessageBufferIF base interface. Possible values include BEREncode, BERDecode, PEREncode, PERDecode, XMLEncode, and XMLDecode.

Returns:

Boolean result of the match operation. True if the `bufferType` argument is `XMLDecode`. argument.

Definition at line 161 of file OSXMLDecodeBuffer.h.

5.6.4 Member Data Documentation

5.6.4.1 OSBOOL OSXMLDecodeBuffer::mbOwnStream `[protected]`

This is set to true if this object creates the underlying stream object.

In this case, the stream will be deleted in the object's destructor.

Definition at line 57 of file OSXMLDecodeBuffer.h.

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

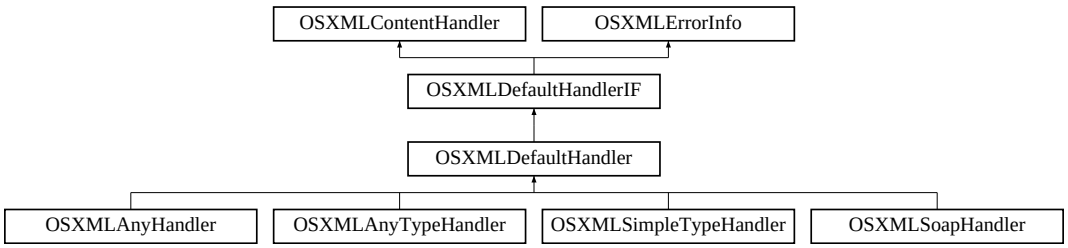
- OSXMLDecodeBuffer.h

5.7 OSXMLDefaultHandler Class Reference

This class is derived from the SAX class DefaultHandler base class.

`#include <rtSaxCppParser.h>`

Inheritance diagram for OSXMLDefaultHandler::



Public Member Functions

- OSXMLDefaultHandler (OSRTContext *pContext, const OSUTF8CHAR *elemName=0, OSINT32 level=0)
- virtual ~OSXMLDefaultHandler ()
- virtual EXTXMLMETHOD int startElement (const OSUTF8CHAR *const uri, const OSUTF8CHAR *const localname, const OSUTF8CHAR *const qname, const OSUTF8CHAR *const *attrs)

  Receive notification of the beginning of an element.

- virtual EXTXMLMETHOD int characters (const OSUTF8CHAR *const chars, unsigned int length)

  Receive notification of character data.

- virtual EXTXMLMETHOD int endElement (const OSUTF8CHAR *const uri, const OSUTF8CHAR *const localname, const OSUTF8CHAR *const qname)

  Receive notification of the end of an element.

- OSINT16 getState ()

  This method returns the current state of the decoding process.

- virtual void init (int level=0)
- void setElemName (const OSUTF8CHAR *elemName)
- OSBOOL isComplete ()

Protected Attributes

- OSRTCtxtPtr mpContext
- const OSUTF8CHAR * mpElemName
- OSINT32 mLevel
- OSINT16 mStartLevel
- OSINT16 mReqElemCnt
- OSINT16 mCurrElemIdx
- OSINT16 mState

Classes

- struct ErrorInfo

5.7.1 Detailed Description

This class is derived from the SAX class DefaultHandler base class.

It contains variables and methods specific to decoding XML messages. It is used to intercept standard SAX parser events, such as startElement, characters, endElement. This class is used as a base class for XBinder generated global element control classes (<elem>_CC).

Definition at line 58 of file rtSaxCppParser.h.

5.7.2 Member Function Documentation

5.7.2.1 virtual EXTXMLMETHOD int OSXMLDefaultHandler::startElement (const OSUTF8CHAR *const *uri*, const OSUTF8CHAR *const *localname*, const OSUTF8CHAR *const *qname*, const OSUTF8CHAR *const * *attrs*) [virtual]

Receive notification of the beginning of an element.

The Parser will invoke this method at the beginning of every element in the XML document; there will be a corresponding endElement() event for every startElement() event (even when the element is empty). All of the element's content will be reported, in order, before the corresponding endElement() event.

Parameters:

uri The URI of the asscioated namespace for this element

localname The local part of the element name

qname The QName of this element

attrs The attributes name/value pairs attached to the element, if any.

See also:

endElement

Implements OSXMLContentHandler.

Reimplemented in OSXMLAnyHandler, OSXMLAnyTypeHandler, OSXMLSimpleTypeHandler, and OSXMLSoapHandler.

5.7.2.2 virtual EXTXMLMETHOD int OSXMLDefaultHandler::characters (const OSUTF8CHAR *const *chars*, unsigned int *length*) [virtual]

Receive notification of character data.

The Parser will call this method to report each chunk of character data. SAX parsers may return all contiguous character data in a single chunk, or they may split it into several chunks; however, all of the characters in any single event must come from the same external entity, so that the Locator provides useful information.

Parameters:

chars The characters from the XML document.

length The length of chars.

Implements OSXMLContentHandler.

Reimplemented in OSXMLSimpleTypeHandler, and OSXMLSoapHandler.

5.7.2.3 virtual EXTXMLMETHOD int OSXMLDefaultHandler::endElement (const OSUTF8CHAR *const *uri*, const OSUTF8CHAR *const *localname*, const OSUTF8CHAR *const *qname*) [virtual]

Receive notification of the end of an element.

The SAX parser will invoke this method at the end of every element in the XML document; there will be a corresponding startElement() event for every endElement() event (even when the element is empty).

Parameters:

uri The URI of the asscioated namespace for this element

localname The local part of the element name

qname The QName of this element

Implements OSXMLContentHandler.

Reimplemented in OSXMLAnyHandler, OSXMLAnyTypeHandler, OSXMLSimpleTypeHandler, and OSXMLSoapHandler.

5.7.2.4 OSINT16 OSXMLDefaultHandler::getState () [inline]

This method returns the current state of the decoding process.

Returns:

The state of the decoding process as type OSXMLState. Can be XMLINIT, XMLSTART, XMLDATA, or XMLEND.

Definition at line 124 of file rtSaxCppParser.h.

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

- rtSaxCppParser.h

5.8 OSXMLDefaultHandler::ErrorInfo Struct Reference

Public Member Functions

- ErrorInfo ()

Public Attributes

- int stat
- const char * file
- int line

5.8.1 Detailed Description

Definition at line 69 of file rtSaxCppParser.h.

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

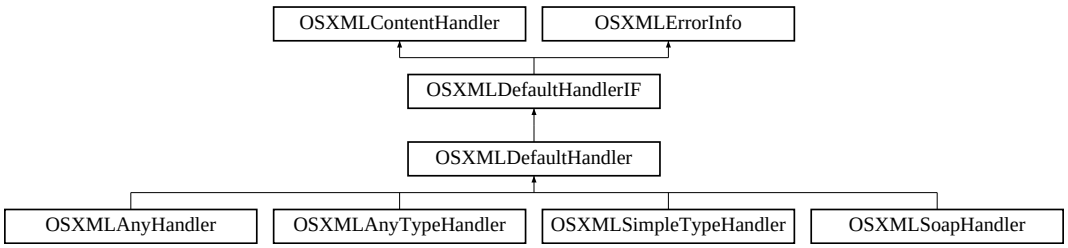
- rtSaxCppParser.h

5.9 OSXMLDefaultHandlerIF Class Reference

This class is derived from the SAX class DefaultHandler base class.

`#include <rtSaxCppParserIF.h>`

Inheritance diagram for OSXMLDefaultHandlerIF::



Public Member Functions

- virtual ~OSXMLDefaultHandlerIF ()

5.9.1 Detailed Description

This class is derived from the SAX class DefaultHandler base class.

It contains variables and methods specific to decoding XML messages. It is used to intercept standard SAX parser events, such as startElement, characters, endElement. This class is used as a base class for XBinder generated global element control classes (<elem>_CC).

Definition at line 260 of file rtSaxCppParserIF.h.

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

- rtSaxCppParserIF.h

5.10 OSXMLDefaultHandlerPtr Class Reference

Public Member Functions

- OSXMLDefaultHandlerPtr ()
- OSXMLDefaultHandlerPtr (OSXMLDefaultHandler *ptr)
- ~OSXMLDefaultHandlerPtr ()
- operator OSXMLDefaultHandler * ()
- operator const OSXMLDefaultHandler * () const
- OSXMLDefaultHandler * operator -> () const
- OSXMLDefaultHandler * operator= (OSXMLDefaultHandler *ptr)
- int operator== (const OSXMLDefaultHandler *ptr) const
- int operator!= (const OSXMLDefaultHandler *ptr) const
- int operator! () const

Friends

- int operator!= (const void *ptr, const OSXMLDefaultHandlerPtr &ptr2)
- int operator== (const void *ptr, const OSXMLDefaultHandlerPtr &ptr2)

5.10.1 Detailed Description

Definition at line 147 of file rtSaxCppParser.h.

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

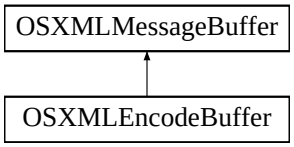
- rtSaxCppParser.h

5.11 OSXMLEncodeBuffer Class Reference

The OSXMLEncodeBuffer class is derived from the OSXMLMessageBuffer base class.

`#include <OSXMLEncodeBuffer.h>`

Inheritance diagram for OSXMLEncodeBuffer::



Public Member Functions

- EXTXMLMETHOD OSXMLEncodeBuffer ()

  Default constructor.

- EXTXMLMETHOD OSXMLEncodeBuffer (OSOCTET *pMsgBuf, size_t msgBufLen)

  This constructor allows a static message buffer to be specified to receive the encoded message.

- int addXMLHeader (const OSUTF8CHAR *version=OSUTF8("1.0"), const OSUTF8CHAR *encoding=OSUTF8(OSXMLHDRUTF8), OSBOOL newLine=TRUE)

  This method adds XML header text to the output buffer with the given version number and encoding attributes.

- EXTXMLMETHOD int addXMLText (const OSUTF8CHAR *text)

  This method adds encoded XML text to the encode buffer.

- virtual size_t getMsgLen ()

  This method returns the length of a previously encoded XML message.

- virtual EXTXMLMETHOD int init ()

  This method reinitializes the encode buffer to allow a new message to be encoded.

- virtual OSBOOL isA (int bufferType)

  This is a virtual method that must be overridden by derived classes to allow identification of the class.

- void nullTerminate ()

  This method adds a null-terminator character (") at the current buffer position.

- EXTXMLMETHOD void setFragment (OSBOOL value=TRUE)

  This method sets a flag indicating that the data is to be encoded as ax XML fragment instead of as a complete XML document (i.e.

- virtual EXTXMLMETHOD long write (const char *filename)

  This method writes the encoded message to the given file.

- virtual EXTXMLMETHOD long write (FILE *fp)

  This version of the write method writes to a file that is specified by a FILE pointer.

Protected Member Functions

- OSXMLEncodeBuffer (OSRTContext *pContext)

5.11.1 Detailed Description

The OSXMLEncodeBuffer class is derived from the OSXMLMessageBuffer base class.

It contains variables and methods specific to encoding XML messages. It is used to manage the buffer into which a message is to be encoded.

Definition at line 38 of file OSXMLEncodeBuffer.h.

5.11.2 Constructor & Destructor Documentation

5.11.2.1 EXTXMLMETHOD OSXMLEncodeBuffer::OSXMLEncodeBuffer (OSOCTET * *pMsgBuf*, size_t *msgBufLen*)

This constructor allows a static message buffer to be specified to receive the encoded message.

Parameters:

pMsgBuf A pointer to a fixed size message buffer to receive the encoded message.

msgBufLen Size of the fixed-size message buffer.

5.11.3 Member Function Documentation

5.11.3.1 int OSXMLEncodeBuffer::addXMLHeader (const OSUTF8CHAR * *version* = `OSUTF8("1.0")`, const OSUTF8CHAR * *encoding* = `OSUTF8(OSXMLHDRUTF8)`, OSBOOL *newLine* = `TRUE`)

This method adds XML header text to the output buffer with the given version number and encoding attributes.

Parameters:

version XML version (default is 1.0)

encoding Character encoding (default is UTF-8)

newLine Add newline char at end of header

Returns:

Zero if success or negative error code.

5.11.3.2 EXTXMLMETHOD int OSXMLEncodeBuffer::addXMLText (const OSUTF8CHAR * *text*)

This method adds encoded XML text to the encode buffer.

It is assumed that the user has already processed the text to do character escaping, etc.. The text is copied directly to the buffer as-is.

Parameters:

text Encoded XML text to be added to the buffer.

Returns:

Zero if success or negative error code.

5.11.3.3 virtual size_t OSXMLEncodeBuffer::getMsgLen () [inline, virtual]

This method returns the length of a previously encoded XML message.

Returns:

Length of the XML message encapsulated within this buffer object.

Definition at line 90 of file OSXMLEncodeBuffer.h.

5.11.3.4 virtual EXTXMLMETHOD int OSXMLEncodeBuffer::init () [virtual]

This method reinitializes the encode buffer to allow a new message to be encoded.

This makes it possible to reuse one message buffer object in a loop to encode multiple messages. After this method is called, any previously encoded message in the buffer will be overwritten on the next encode call.

5.11.3.5 virtual OSBOOL OSXMLEncodeBuffer::isA (int *bufferType*) [inline, virtual]

This is a virtual method that must be overridden by derived classes to allow identification of the class.

The base class variant is abstract. This method matches an enumerated identifier defined in the base class. One identifier is declared for each of the derived classes.

Parameters:

bufferType Enumerated identifier specifying a derived class. This type is defined as a public access type in the OSRTMessageBufferIF base interface. Possible values include BEREncode, BERDecode, PEREncode, PERDecode, XMLEncode, and XMLDecode.

Returns:

Boolean result of the match operation. True if the `bufferType` argument is `XMLEncode`. argument.

Definition at line 118 of file OSXMLEncodeBuffer.h.

5.11.3.6 EXTXMLMETHOD void OSXMLEncodeBuffer::setFragment (OSBOOL *value* = TRUE)

This method sets a flag indicating that the data is to be encoded as ax XML fragment instead of as a complete XML document (i.e.

an XML header will not be added).

5.11.3.7 virtual EXTXMLMETHOD long OSXMLEncodeBuffer::write (const char * *filename*) [virtual]

This method writes the encoded message to the given file.

Parameters:

filename The name of file to which the encoded message will be written.

Returns:

Number of octets actually written. This value may be less than the actual message length if an error occurs.

5.11.3.8 virtual EXTXMLMETHOD long OSXMLEncodeBuffer::write (FILE * *fp*) [virtual]

This version of the write method writes to a file that is specified by a FILE pointer.

Parameters:

fp Pointer to FILE structure to which the encoded message will be written.

Returns:

Number of octets actually written. This value may be less than the actual message length if an error occurs.

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

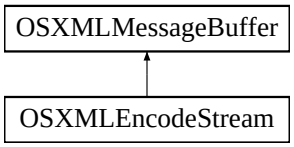
- OSXMLEncodeBuffer.h

5.12 OSXMLEncodeStream Class Reference

The OSXMLEncodeStream class is derived from the OSXMLMessageBuffer base class.

`#include <OSXMLEncodeStream.h>`

Inheritance diagram for OSXMLEncodeStream::



Public Member Functions

- EXTXMLMETHOD OSXMLEncodeStream (OSRTOutputStream &outputStream)

  This version of the OSXMLEncodeStream constructor takes a reference to the OSOutputStream object.

- OSXMLEncodeStream (OSRTOutputStream *pOutputStream, OSBOOL ownStream=TRUE)

  This version of the OSXMLEncodeStream constructor takes a pointer to the OSRTOutputStream object.

- virtual EXTXMLMETHOD int init ()

  This method reinitializes the encode stream to allow a new message to be encoded.

- virtual OSBOOL isA (int bufferType)

  This is a virtual method that must be overridden by derived classes to allow identification of the class.

- virtual const OSOCTET * getMsgPtr ()

  This is a virtual method that must be overridden by derived classes to allow access to the stored message.

Protected Attributes

- OSRTOutputStream * mpStream

  A pointer to an OSRTOutputStream object.

- OSBOOL mbOwnStream

  TRUE if the OSXMLEncodeStream object will close and free the stream in the destructor.

5.12.1 Detailed Description

The OSXMLEncodeStream class is derived from the OSXMLMessageBuffer base class.

It contains variables and methods specific to streaming encoding XML messages. It is used to manage the stream into which a message is to be encoded.

Definition at line 40 of file OSXMLEncodeStream.h.

5.12.2 Constructor & Destructor Documentation

5.12.2.1 EXTXMLMETHOD OSXMLEncodeStream::OSXMLEncodeStream (OSRTOutputStream & *outputStream*)

This version of the OSXMLEncodeStream constructor takes a reference to the OSOutputStream object.

The stream is assumed to have been previously initialized.

Parameters:

outputStream reference to the OSOutputStream object

5.12.2.2 OSXMLEncodeStream::OSXMLEncodeStream (OSRTOutputStream * *pOutputStream*, OSBOOL *ownStream* = `TRUE`)

This version of the OSXMLEncodeStream constructor takes a pointer to the OSRTOutputStream object.

The stream is assumed to have been previously initialized. If ownStream is set to TRUE, then stream will be closed and freed in the destructor.

Parameters:

pOutputStream reference to the OSOutputStream object

ownStream set ownership for the passed stream object.

5.12.3 Member Function Documentation

5.12.3.1 virtual EXTXMLMETHOD int OSXMLEncodeStream::init () `[virtual]`

This method reinitializes the encode stream to allow a new message to be encoded.

This makes it possible to reuse one stream object in a loop to encode multiple messages.

5.12.3.2 virtual OSBOOL OSXMLEncodeStream::isA (int *bufferType*) `[inline, virtual]`

This is a virtual method that must be overridden by derived classes to allow identification of the class.

The base class variant is abstract. This method matches an enumerated identifier defined in the base class. One identifier is declared for each of the derived classes.

Parameters:

bufferType Enumerated identifier specifying a derived class. This type is defined as a public access type in the OSRTMessageBufferIF base interface. Possible values include BEREncode, BERDecode, PEREncode, PERDecode, XMLEncode, and XMLDecode.

Returns:

Boolean result of the match operation. True if the `bufferType` argument is `XMLEncode`. argument.

Definition at line 95 of file OSXMLEncodeStream.h.

5.12.3.3 virtual const OSOCTET* OSXMLEncodeStream::getMsgPtr () [inline, virtual]

This is a virtual method that must be overridden by derived classes to allow access to the stored message.

The base class implementation returns a null value.

Returns:

A pointer to the stored message.

Definition at line 105 of file OSXMLEncodeStream.h.

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

- OSXMLEncodeStream.h

5.13 OSXMLErrorHandler Class Reference

Public Member Functions

- virtual ~OSXMLErrorHandler ()

The error handler interface

- virtual void warning ()=0
  Receive notification of a warning.
- virtual void error ()=0
  Receive notification of a recoverable error.
- virtual void fatalError ()=0
  Receive notification of a non-recoverable error.
- virtual void resetErrors ()=0
  Reset the Error handler object on its reuse.

5.13.1 Detailed Description

Definition at line 42 of file rtSaxCppParserIF.h.

5.13.2 Member Function Documentation

5.13.2.1 virtual void OSXMLErrorHandler::warning () [pure virtual]

Receive notification of a warning.

SAX parsers will use this method to report conditions that are not errors or fatal errors as defined by the XML 1.0 recommendation. The default behaviour is to take no action.

The SAX parser must continue to provide normal parsing events after invoking this method: it should still be possible for the application to process the document through to the end.

5.13.2.2 virtual void OSXMLErrorHandler::error () [pure virtual]

Receive notification of a recoverable error.

This corresponds to the definition of "error" in section 1.2 of the W3C XML 1.0 Recommendation. For example, a validating parser would use this callback to report the violation of a validity constraint. The default behaviour is to take no action.

The SAX parser must continue to provide normal parsing events after invoking this method: it should still be possible for the application to process the document through to the end. If the application cannot do so, then the parser should report a fatal error even if the XML 1.0 recommendation does not require it to do so.

5.13.2.3 virtual void OSXMLErrorHandler::fatalError () [pure virtual]

Receive notification of a non-recoverable error.

This corresponds to the definition of "fatal error" in section 1.2 of the W3C XML 1.0 Recommendation. For example, a parser would use this callback to report the violation of a well-formedness constraint.

The application must assume that the document is unusable after the parser has invoked this method, and should continue (if at all) only for the sake of collecting addition error messages: in fact, SAX parsers are free to stop reporting any other events once this method has been invoked.

5.13.2.4 virtual void OSXMLErrorHandler::resetErrors () [pure virtual]

Reset the Error handler object on its reuse.

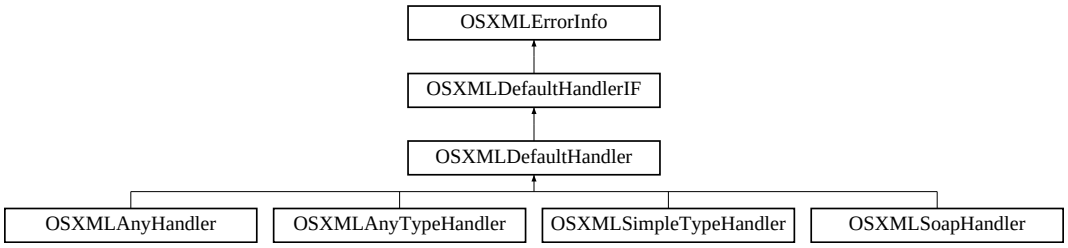
This method helps in reseting the Error handler object implementational defaults each time the Error handler is begun.

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

- rtSaxCppParserIF.h

5.14 OSXMLErrorInfo Class Reference

Inheritance diagram for OSXMLErrorInfo::



Public Member Functions

- virtual ~OSXMLErrorInfo ()

5.14.1 Detailed Description

Definition at line 34 of file rtSaxCppParserIF.h.

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

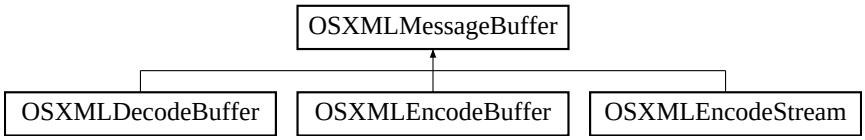
- rtSaxCppParserIF.h

5.15 OSXMLMessageBuffer Class Reference

The XML message buffer class is derived from the OSMessageBuffer base class.

`#include <OSXMLMessageBuffer.h>`

Inheritance diagram for OSXMLMessageBuffer::



Public Member Functions

- virtual EXTXMLMETHOD void * getAppInfo ()

  The getAppInfo method returns the pointer to application context information.

- EXTXMLMETHOD int getIndent ()

  This method returns current XML output indent value.

- EXTXMLMETHOD int getIndentChar ()

  This method returns current XML output indent character value (default is space).

- EXTXMLMETHOD OSBOOL getWriteBOM ()

  This function returns whether writing the Unicode BOM is currently enabled or disabled.

- virtual EXTXMLMETHOD void setNamespace (const OSUTF8CHAR *prefix, const OSUTF8CHAR *uri, OSRTDList *pNSAttrs=0)

  This method sets a namespace in the context namespace list.

- virtual EXTXMLMETHOD void setAppInfo (void *pXMLInfo)

  This method sets application specific context information within the common context structure.

- EXTXMLMETHOD void setFormatting (OSBOOL doFormatting)

  This method sets XML output formatting to the given value.

- EXTXMLMETHOD void setIndent (OSUINT8 indent)

  This method sets XML output indent to the given value.

- EXTXMLMETHOD void setIndentChar (char indentChar)

  This method sets XML output indent character to the given value.

- EXTXMLMETHOD void setWriteBOM (OSBOOL write)

  This method sets whether to write the Unicode byte order mark before the XML header.

Protected Member Functions

- EXTXMLMETHOD OSXMLMessageBuffer (Type bufferType, OSRTContext *pContext=0)

  The protected constructor creates a new context and sets the buffer class type.

5.15.1 Detailed Description

The XML message buffer class is derived from the OSMessageBuffer base class.

It is the base class for the OSXMLEncodeBuffer and OSXMLDecodeBuffer classes. It contains variables and methods specific to encoding or decoding XML messages. It is used to manage the buffer into which a message is to be encoded or decoded.

Definition at line 42 of file OSXMLMessageBuffer.h.

5.15.2 Constructor & Destructor Documentation

5.15.2.1 EXTXMLMETHOD OSXMLMessageBuffer::OSXMLMessageBuffer (Type *bufferType*, OSRTContext * *pContext* = 0) [protected]

The protected constructor creates a new context and sets the buffer class type.

Parameters:

bufferType Type of message buffer that is being created (for example, XMLEncode or XMLDecode).

pContext Pointer to a context to use. If NULL, new context will be allocated.

5.15.3 Member Function Documentation

5.15.3.1 EXTXMLMETHOD int OSXMLMessageBuffer::getIndent ()

This method returns current XML output indent value.

Returns:

Current indent value (>= 0) if OK, negative status code if error.

5.15.3.2 EXTXMLMETHOD int OSXMLMessageBuffer::getIndentChar ()

This method returns current XML output indent character value (default is space).

Returns:

Current indent character (> 0) if OK, negative status code if error.

5.15.3.3 EXTXMLMETHOD OSBOOL OSXMLMessageBuffer::getWriteBOM ()

This function returns whether writing the Unicode BOM is currently enabled or disabled.

Returns:

TRUE if writing BOM is enabled, FALSE otherwise.

5.15.3.4 virtual EXTXMLMETHOD void OSXMLMessageBuffer::setNamespace (const OSUTF8CHAR * *prefix*, const OSUTF8CHAR * *uri*, OSRTDList * *pNSAttrs* = 0) [virtual]

This method sets a namespace in the context namespace list.

If the given namespace URI does not exist in the list, the namespace is added. If the URI is found, the value of the namespace prefix will be changed to the given prefix.

Parameters:

prefix Namespace prefix

uri Namespace URI

pNSAttrs Namespace list to which namespace is to be added

5.15.3.5 virtual EXTXMLMETHOD void OSXMLMessageBuffer::setAppInfo (void * *pXMLInfo*) [virtual]

This method sets application specific context information within the common context structure.

For XML encoding/decoding, this is a structure of type *OSXMLCtxtInfo*.

Parameters:

pXMLInfo Pointer to context information.

5.15.3.6 EXTXMLMETHOD void OSXMLMessageBuffer::setFormatting (OSBOOL *doFormatting*)

This method sets XML output formatting to the given value.

If TRUE (the default), the XML document is formatted with indentation and newlines. If FALSE, all whitespace between elements is suppressed. Turning formatting off can provide more compressed documents and also a more canonical representation which is important for security applications.

Parameters:

doFormatting Boolean value indicating if formatting is to be done

Returns:

Status of operation: 0 if OK, negative status code if error.

5.15.3.7 EXTXMLMETHOD void OSXMLMessageBuffer::setIndent (OSUINT8 *indent*)

This method sets XML output indent to the given value.

Parameters:

indent Number of spaces per indent. Default is 3.

5.15.3.8 EXTXMLMETHOD void OSXMLMessageBuffer::setIndentChar (char *indentChar*)

This method sets XML output indent character to the given value.

Parameters:

indentChar Indent character. Default is space.

5.15.3.9 EXTXMLMETHOD void OSXMLMessageBuffer::setWriteBOM (OSBOOL *write*)

This method sets whether to write the Unicode byte order mark before the XML header.

Parameters:

write TRUE if BOM should be written, FALSE otherwise.

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

- OSXMLMessageBuffer.h

5.16 OSXMLNamespaceClass Class Reference

This class is used to hold an XML namespace prefix to URI mapping.

`#include <rtXmlCppNamespace.h>`

Public Member Functions

- OSXMLNamespaceClass ()

  The default constructor sets the namespace prefix and URI values to empty values.

- ~OSXMLNamespaceClass ()

  The destructor deletes the prefix and uri string variables.

- OSXMLNamespaceClass (const OSUTF8CHAR *nsPrefix, const OSUTF8CHAR *nsURI)

  The parameterized constructor sets the namespace prefix and URI values to the given values.

- OSXMLNamespaceClass (const OSUTF8CHAR *nsPrefix, size_t nsPrefixBytes, const OSUTF8CHAR *nsURI, size_t nsURIBytes)

  The parameterized constructor sets the namespace prefix and URI values to the given values.

- OSXMLNamespaceClass (const OSXMLNamespaceClass &o)

  The copy constructor make a deep-copy of the prefix and URI values.

- const OSUTF8CHAR * getPrefix () const

  This method is used to get the namespace prefix value.

- const OSUTF8CHAR * getURI () const

  This method is used to get the namespace URI value.

- void setPrefix (const OSUTF8CHAR *nsPrefix)

  This method is used to set the namespace prefix value.

- void setURI (const OSUTF8CHAR *nsURI)

  This method is used to set the namespace URI value.

5.16.1 Detailed Description

This class is used to hold an XML namespace prefix to URI mapping.

Definition at line 38 of file rtXmlCppNamespace.h.

5.16.2 Constructor & Destructor Documentation

5.16.2.1 OSXMLNamespaceClass::OSXMLNamespaceClass (const OSUTF8CHAR * *nsPrefix*, const OSUTF8CHAR * *nsURI*)

The parameterized constructor sets the namespace prefix and URI values to the given values.

A deep copy of the values is done.

Parameters:

nsPrefix Namespace prefix value.

nsURI Namespace URI value.

5.16.2.2 OSXMLNamespaceClass::OSXMLNamespaceClass (const OSUTF8CHAR * *nsPrefix*, size_t *nsPrefixBytes*, const OSUTF8CHAR * *nsURI*, size_t *nsURIBytes*)

The parameterized constructor sets the namespace prefix and URI values to the given values.

A deep copy of the values is done.

Parameters:

nsPrefix Namespace prefix value.

nsPrefixBytes Namespace prefix value size in bytes.

nsURI Namespace URI value.

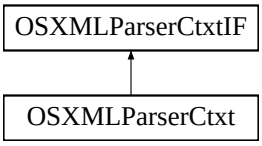
nsURIBytes Namespace URI value size in bytes.

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

- rtXmlCppNamespace.h

5.17 OSXMLParserCtxt Class Reference

Inheritance diagram for OSXMLParserCtxt::



5.17.1 Detailed Description

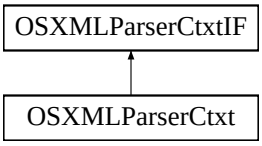
Definition at line 179 of file rtSaxCppParser.h.

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

- rtSaxCppParser.h

5.18 OSXMLParserCtxtIF Class Reference

Inheritance diagram for OSXMLParserCtxtIF::



Public Member Functions

- virtual ~OSXMLParserCtxtIF ()

5.18.1 Detailed Description

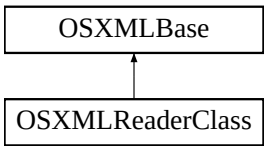
Definition at line 287 of file rtSaxCppParserIF.h.

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

- rtSaxCppParserIF.h

5.19 OSXMLReaderClass Class Reference

Inheritance diagram for OSXMLReaderClass::



Public Member Functions

- virtual int parse (OSRTInputStreamIF &source)=0
  Parse an XML document.
- virtual int parse (const char *const pBuffer, size_t bufSize)=0
  Parse an XML document from memory buffer.
- virtual int parse (const char *const systemId)=0
  Parse an XML document from a system identifier (URI).

5.19.1 Detailed Description

Definition at line 198 of file rtSaxCppParserIF.h.

5.19.2 Member Function Documentation

5.19.2.1 virtual int OSXMLReaderClass::parse (OSRTInputStreamIF & *source*) [pure virtual]

Parse an XML document.

The application can use this method to instruct the SAX parser to begin parsing an XML document from any valid input source (a character stream, a byte stream, or a URI).

Applications may not invoke this method while a parse is in progress (they should create a new Parser instead for each additional XML document). Once a parse is complete, an application may reuse the same Parser object, possibly with a different input source.

Parameters:

source The input source for the top-level of the XML document.

5.19.2.2 virtual int OSXMLReaderClass::parse (const char *const *pBuffer*, size_t *bufSize*) [pure virtual]

Parse an XML document from memory buffer.

Parameters:

pBuffer Buffer containg the XML data to be parsed.

bufSize Buffer size, in octets.

5.19.2.3 virtual int OSXMLReaderClass::parse (const char *const *systemId*) [pure virtual]

Parse an XML document from a system identifier (URI).

This method is a shortcut for the common case of reading a document from a system identifier. It is the exact equivalent of the following:

parse(new URLInputSource(systemId));

If the system identifier is a URL, it must be fully resolved by the application before it is passed to the parser.

Parameters:

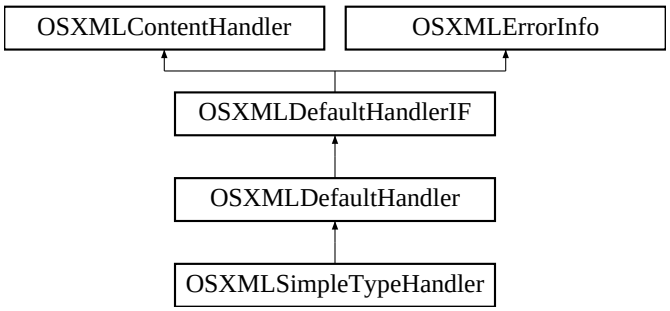
systemId The system identifier (URI).

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

- rtSaxCppParserIF.h

5.20 OSXMLSimpleTypeHandler Class Reference

Inheritance diagram for OSXMLSimpleTypeHandler::



Public Member Functions

- virtual EXTXMLMETHOD int startElement (const OSUTF8CHAR *const uri, const OSUTF8CHAR *const localname, const OSUTF8CHAR *const qname, const OSUTF8CHAR *const *attrs)

  Receive notification of the beginning of an element.

- virtual EXTXMLMETHOD int characters (const OSUTF8CHAR *const chars, unsigned int length)

  Receive notification of character data.

- virtual EXTXMLMETHOD int endElement (const OSUTF8CHAR *const uri, const OSUTF8CHAR *const localname, const OSUTF8CHAR *const qname)

  Receive notification of the end of an element.

5.20.1 Detailed Description

Definition at line 39 of file rtSaxCppSimpleType.h.

5.20.2 Member Function Documentation

5.20.2.1 virtual EXTXMLMETHOD int OSXMLSimpleTypeHandler::startElement (const OSUTF8CHAR *const *uri*, const OSUTF8CHAR *const *localname*, const OSUTF8CHAR *const *qname*, const OSUTF8CHAR *const * *attrs*) [virtual]

Receive notification of the beginning of an element.

The Parser will invoke this method at the beginning of every element in the XML document; there will be a corresponding endElement() event for every startElement() event (even when the element is empty). All of the element's content will be reported, in order, before the corresponding endElement() event.

Parameters:

uri The URI of the asscioated namespace for this element

localname The local part of the element name

qname The QName of this element

attrs The attributes name/value pairs attached to the element, if any.

See also:

endElement

Reimplemented from OSXMLDefaultHandler.

5.20.2.2 virtual EXTXMLMETHOD int OSXMLSimpleTypeHandler::characters (const OSUTF8CHAR *const *chars*, unsigned int *length*) [virtual]

Receive notification of character data.

The Parser will call this method to report each chunk of character data. SAX parsers may return all contiguous character data in a single chunk, or they may split it into several chunks; however, all of the characters in any single event must come from the same external entity, so that the Locator provides useful information.

Parameters:

chars The characters from the XML document.

length The length of chars.

Reimplemented from OSXMLDefaultHandler.

5.20.2.3 virtual EXTXMLMETHOD int OSXMLSimpleTypeHandler::endElement (const OSUTF8CHAR *const *uri*, const OSUTF8CHAR *const *localname*, const OSUTF8CHAR *const *qname*) [virtual]

Receive notification of the end of an element.

The SAX parser will invoke this method at the end of every element in the XML document; there will be a corresponding startElement() event for every endElement() event (even when the element is empty).

Parameters:

uri The URI of the asscioated namespace for this element

localname The local part of the element name

qname The QName of this element

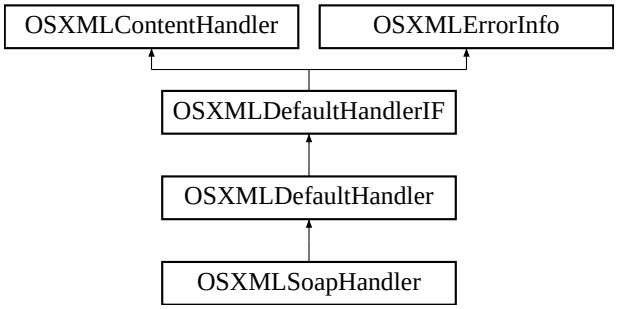
Reimplemented from OSXMLDefaultHandler.

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

- rtSaxCppSimpleType.h

5.21 OSXMLSoapHandler Class Reference

Inheritance diagram for OSXMLSoapHandler::



Public Member Functions

- virtual int startElement (const OSUTF8CHAR *const uri, const OSUTF8CHAR *const localname, const OSUTF8CHAR *const qname, const OSUTF8CHAR *const *attrs)

  Receive notification of the beginning of an element.

- virtual int characters (const OSUTF8CHAR *const chars, unsigned int length)

  Receive notification of character data.

- virtual int endElement (const OSUTF8CHAR *const uri, const OSUTF8CHAR *const localname, const OSUTF8CHAR *const qname)

  Receive notification of the end of an element.

5.21.1 Detailed Description

Definition at line 40 of file rtSaxCppSoap.h.

5.21.2 Member Function Documentation

5.21.2.1 virtual int OSXMLSoapHandler::startElement (const OSUTF8CHAR *const *uri*, const OSUTF8CHAR *const *localname*, const OSUTF8CHAR *const *qname*, const OSUTF8CHAR *const * *attrs*) [virtual]

Receive notification of the beginning of an element.

The Parser will invoke this method at the beginning of every element in the XML document; there will be a corresponding endElement() event for every startElement() event (even when the element is empty). All of the element's content will be reported, in order, before the corresponding endElement() event.

Parameters:

uri The URI of the asscioated namespace for this element

localname The local part of the element name

qname The QName of this element

attrs The attributes name/value pairs attached to the element, if any.

See also:

endElement

Reimplemented from OSXMLDefaultHandler.

5.21.2.2 virtual int OSXMLSoapHandler::characters (const OSUTF8CHAR *const *chars*, unsigned int *length*) [virtual]

Receive notification of character data.

The Parser will call this method to report each chunk of character data. SAX parsers may return all contiguous character data in a single chunk, or they may split it into several chunks; however, all of the characters in any single event must come from the same external entity, so that the Locator provides useful information.

Parameters:

chars The characters from the XML document.

length The length of chars.

Reimplemented from OSXMLDefaultHandler.

5.21.2.3 virtual int OSXMLSoapHandler::endElement (const OSUTF8CHAR *const *uri*, const OSUTF8CHAR *const *localname*, const OSUTF8CHAR *const *qname*) [virtual]

Receive notification of the end of an element.

The SAX parser will invoke this method at the end of every element in the XML document; there will be a corresponding startElement() event for every endElement() event (even when the element is empty).

Parameters:

uri The URI of the asscioated namespace for this element

localname The local part of the element name

qname The QName of this element

Reimplemented from OSXMLDefaultHandler.

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

- rtSaxCppSoap.h

5.22 OSXMLStrListHandler Class Reference

OSXMLStrListHandler.

`#include <rtSaxCppStrList.h>`

Static Public Member Functions

- static int parseSTL (OSCTXT *pctxt, OSRTMEMBUF *pMemBuf, OSRTObjListClass *pStrList)
- static int match (OSCTXT *)

5.22.1 Detailed Description

OSXMLStrListHandler.

Definition at line 39 of file rtSaxCppStrList.h.

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

- rtSaxCppStrList.h

5.23 OSXSDGlobalElement Class Reference

XSD global element base class.

`#include <rtXmlCppXSDElement.h>`

Public Member Functions

- OSXSDGlobalElement (OSRTMessageBufferIF &msgBuf)

  This constructor sets the internal message buffer pointer to point at the given message buffer or stream object.

- OSXSDGlobalElement (const OSXSDGlobalElement &o)

  The copy constructor sets the internal message buffer pointer and context to point at the message buffer and context from the original OSCType object.

- virtual ~OSXSDGlobalElement ()

  The virtual destructor does nothing.

- int decode ()

  The `decode` *method decodes the message described by the encapsulated message buffer object.*

- virtual int decodeFrom (OSRTMessageBufferIF &)

  The `decodeFrom` *method decodes a message from the given message buffer or stream argument.*

- int encode ()

  The `encode` *method encodes a message using the encoding rules specified by the derived message buffer object.*

- virtual int encodeTo (OSRTMessageBufferIF &)

  The `encodeTo` *method encodes a message into the given message buffer or stream argument.*

- OSCTXT * getCtxtPtr ()

  The getCtxtPtr method returns the underlying C runtime context.

- void * memAlloc (size_t numocts)

  The memAlloc method allocates memory using the C runtime memory management functions.

- void memFreePtr (void *ptr)

  The memFreePtr method frees the memory at a specific location.

- void setDefaultNamespace (const OSUTF8CHAR *uri)

  The setDefaultNamespace method sets the default namespace for the element to the given value.

- void setDiag (OSBOOL value=TRUE)

  The setDiag method turns diagnostic tracing on or off.

- void setEncXSINamespace (OSBOOL value=TRUE)

  The setEncXSINamespace method sets a flag in the internal context that indicates the xsi namespace attribute must be encoded.

- void setNamespace (const OSUTF8CHAR *prefix, const OSUTF8CHAR *uri)

The setNamespace method adds or modifies the namespace with the given URI in the namespace list to contain the given prefix.

- void setNoNSSchemaLocation (const OSUTF8CHAR *uri)

  The setNoNSSchemaLocation method adds an xsi:noNamespaceSchemaLocation attribute to the document.

- void setSchemaLocation (const OSUTF8CHAR *uri)

  The setSchemaLocation method adds an xsi:schemaLocation attribute to the document.

- void setXSIType (const OSUTF8CHAR *typeName)

  The setXSIType method sets a type name to be used in the xsi:type attribute in the top-level module element declaration.

- int validate ()

  The `validate` *method validates the message described by the encapsulated message buffer object.*

- virtual int validateFrom (OSRTMessageBufferIF &)

  The `validateFrom` *method validates a message from the given message buffer or stream argument.*

Protected Member Functions

- OSXSDGlobalElement ()

  The default constructor sets the message pointer member variable to NULL and creates a new context object.

- OSXSDGlobalElement (OSRTContext &ctxt)

  This constructor sets the message pointer member variable to NULL and initializes the context object to point at the given context value.

- void setMsgBuf (OSRTMessageBufferIF &msgBuf)

  The setMsgBuf method is used to set the internal message buffer pointer to point at the given message buffer or stream object.

Protected Attributes

- OSRTCtxtPtr mpContext

  The mpContext member variable holds a reference-counted C runtime variable.

- OSRTMessageBufferIF * mpMsgBuf

  The mpMsgBuf member variable is a pointer to a derived message buffer or stream class that will manage the message being encoded or decoded.

5.23.1 Detailed Description

XSD global element base class.

This is the main base class for all generated global element control classes. It holds a variable of a generated data type as well as the associated message buffer or stream class to which a message will be encoded or from which a message will be decoded.

Definition at line 57 of file rtXmlCppXSDElement.h.

5.23.2 Constructor & Destructor Documentation

5.23.2.1 OSXSDGlobalElement::OSXSDGlobalElement (OSRTContext & *ctxt*) [inline, protected]

This constructor sets the message pointer member variable to NULL and initializes the context object to point at the given context value.

Parameters:

ctxt - Reference to a context object.

Definition at line 85 of file rtXmlCppXSDElement.h.

5.23.2.2 OSXSDGlobalElement::OSXSDGlobalElement (OSRTMessageBufferIF & *msgBuf*) [inline]

This constructor sets the internal message buffer pointer to point at the given message buffer or stream object.

The context is set to point at the context contained within the message buffer object. Thus, the message buffer and control class object share the context. It will not be released until both objects are destroyed.

Parameters:

msgBuf - Reference to a message buffer or stream object.

Definition at line 105 of file rtXmlCppXSDElement.h.

5.23.2.3 OSXSDGlobalElement::OSXSDGlobalElement (const OSXSDGlobalElement & *o*) [inline]

The copy constructor sets the internal message buffer pointer and context to point at the message buffer and context from the original OSCType object.

Parameters:

o - Reference to a global element object.

Definition at line 116 of file rtXmlCppXSDElement.h.

5.23.2.4 virtual OSXSDGlobalElement::~OSXSDGlobalElement () [inline, virtual]

The virtual destructor does nothing.

It is overridden by derived versions of this class.

Definition at line 123 of file rtXmlCppXSDElement.h.

5.23.3 Member Function Documentation

5.23.3.1 void OSXSDGlobalElement::setMsgBuf (OSRTMessageBufferIF & *msgBuf*) [protected]

The setMsgBuf method is used to set the internal message buffer pointer to point at the given message buffer or stream object.

Parameters:

msgBuf - Reference to a message buffer or stream object.

5.23.3.2 virtual int OSXSDGlobalElement::decodeFrom (OSRTMessageBufferIF &) [inline, virtual]

The decodeFrom method decodes a message from the given message buffer or stream argument.

Parameters:

- Message buffer or stream containing message to decode.

Definition at line 138 of file rtXmlCppXSDElement.h.

5.23.3.3 virtual int OSXSDGlobalElement::encodeTo (OSRTMessageBufferIF &) [inline, virtual]

The encodeTo method encodes a message into the given message buffer or stream argument.

Parameters:

- Message buffer or stream to which the message is to be encoded.

Definition at line 153 of file rtXmlCppXSDElement.h.

5.23.3.4 OSCTXT* OSXSDGlobalElement::getCtxtPtr () [inline]

The getCtxtPtr method returns the underlying C runtime context.

This context can be used in calls to C runtime functions.

Definition at line 159 of file rtXmlCppXSDElement.h.

5.23.3.5 void* OSXSDGlobalElement::memAlloc (size_t *numocts*) [inline]

The memAlloc method allocates memory using the C runtime memory management functions.

The memory is tracked in the underlying context structure. When both this OSXSDGlobalElement derived control class object and the message buffer object are destroyed, this memory will be freed.

Parameters:

numocts - Number of bytes of memory to allocate

Definition at line 172 of file rtXmlCppXSDElement.h.

5.23.3.6 void OSXSDGlobalElement::memFreePtr (void * *ptr*) [inline]

The memFreePtr method frees the memory at a specific location.

This memory must have been allocated using the memAlloc method described earlier.

Parameters:

ptr - Pointer to a block of memory allocated with memAlloc

Definition at line 200 of file rtXmlCppXSDElement.h.

5.23.3.7 void OSXSDGlobalElement::setDefaultNamespace (const OSUTF8CHAR * *uri*) [inline]

The setDefaultNamespace method sets the default namespace for the element to the given value.

Parameters:

uri - Default namespace URI

Definition at line 210 of file rtXmlCppXSDElement.h.

5.23.3.8 void OSXSDGlobalElement::setDiag (OSBOOL *value* = TRUE) [inline]

The setDiag method turns diagnostic tracing on or off.

Parameters:

value - Boolean on/off value (default = on)

Definition at line 219 of file rtXmlCppXSDElement.h.

5.23.3.9 void OSXSDGlobalElement::setEncXSINamespace (OSBOOL *value* = TRUE) [inline]

The setEncXSINamespace method sets a flag in the internal context that indicates the xsi namespace attribute must be encoded.

Parameters:

value - Boolean on/off value (default = on)

Definition at line 229 of file rtXmlCppXSDElement.h.

5.23.3.10 void OSXSDGlobalElement::setNamespace (const OSUTF8CHAR * *prefix*, const OSUTF8CHAR * *uri*) [inline]

The setNamespace method adds or modifies the namespace with the given URI in the namespace list to contain the given prefix.

Parameters:

prefix - Namespace prefix

uri - Namespace URI

Definition at line 240 of file rtXmlCppXSDElement.h.

5.23.3.11 void OSXSDGlobalElement::setNoNSSchemaLocation (const OSUTF8CHAR * *uri*) [inline]

The setNoNSSchemaLocation method adds an xsi:noNamespaceSchemaLocation attribute to the document.

Parameters:

uri - URI for noNamespaceSchemaLocation.

Definition at line 250 of file rtXmlCppXSDElement.h.

5.23.3.12 void OSXSDGlobalElement::setSchemaLocation (const OSUTF8CHAR * *uri*) [inline]

The setSchemaLocation method adds an xsi:schemaLocation attribute to the document.

Parameters:

uri - URI for schemaLocation.

Definition at line 260 of file rtXmlCppXSDElement.h.

5.23.3.13 void OSXSDGlobalElement::setXSIType (const OSUTF8CHAR * *typeName*) [inline]

The setXSIType method sets a type name to be used in the xsi:type attribute in the top-level module element declaration.

Parameters:

typeName - XSI type name

Definition at line 270 of file rtXmlCppXSDElement.h.

5.23.3.14 virtual int OSXSDGlobalElement::validateFrom (OSRTMessageBufferIF &) [inline, virtual]

The `validateFrom` method validates a message from the given message buffer or stream argument.

Parameters:

- Message buffer or stream containing message to validate.

Definition at line 287 of file rtXmlCppXSDElement.h.

5.23.4 Member Data Documentation

5.23.4.1 OSRTCtxtPtr OSXSDGlobalElement::mpContext [protected]

The mpContext member variable holds a reference-counted C runtime variable.

This context is used in calls to all C run-time functions. The context pointed at by this smart-pointer object is shared with the message buffer object contained within this class.

Definition at line 65 of file rtXmlCppXSDElement.h.

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

- rtXmlCppXSDElement.h

Chapter 6

XBinder File Documentation

6.1 OSXMLDecodeBuffer.h File Reference

XML decode buffer or stream class definition.

```
#include "rtxsrc/OSRTInputStream.h"
#include "rtxmlsrc/OSXMLMessageBuffer.h"
#include "rtxmlsrc/rtSaxCppParserIF.h"
```

Classes

- class OSXMLDecodeBuffer

  The OSXMLDecodeBuffer class is derived from the OSXMLMessageBuffer base class.

6.1.1 Detailed Description

XML decode buffer or stream class definition.

Definition in file OSXMLDecodeBuffer.h.

6.2 OSXMLEncodeBuffer.h File Reference

XML encode message buffer class definition.

```
#include "rtxmlsrc/OSXMLMessageBuffer.h"
```

Classes

- class OSXMLEncodeBuffer

  The OSXMLEncodeBuffer class is derived from the OSXMLMessageBuffer base class.

6.2.1 Detailed Description

XML encode message buffer class definition.

Definition in file OSXMLEncodeBuffer.h.

6.3 OSXMLEncodeStream.h File Reference

XML encode stream class definition.

```
#include "rtxsrc/OSRTOutputStream.h"
#include "rtxmlsrc/OSXMLMessageBuffer.h"
```

Classes

- class OSXMLEncodeStream

  The OSXMLEncodeStream class is derived from the OSXMLMessageBuffer base class.

6.3.1 Detailed Description

XML encode stream class definition.

Definition in file OSXMLEncodeStream.h.

6.4 OSXMLMessageBuffer.h File Reference

XML encode/decode buffer and stream base class.

```
#include "rtxsrc/OSRTMsgBuf.h"
#include "rtxmlsrc/osrtxml.h"
```

Classes

- class OSXMLMessageBuffer

  The XML message buffer class is derived from the OSMessageBuffer base class.

6.4.1 Detailed Description

XML encode/decode buffer and stream base class.

Definition in file OSXMLMessageBuffer.h.

6.5 rtSaxCppAny.h File Reference

```
#include "rtxsrc/OSRTContext.h"
#include "rtxmlsrc/osrtxml.h"
#include "rtxmlsrc/rtSaxCppParser.h"
#include "rtxmlsrc/rtXmlCppMsgBuf.h"
#include "rtxsrc/rtxCppXmlString.h"
#include "rtxmlsrc/OSXSDAnyTypeClass.h"
#include "rtxmlsrc/rtSaxCppAnyType.h"
```

Classes

- class OSXMLAnyHandler

6.5.1 Detailed Description

Definition in file rtSaxCppAny.h.

6.6 rtSaxCppAnyType.h File Reference

```
#include "rtxsrc/OSRTContext.h"
#include "rtxmlsrc/osrtxml.h"
#include "rtxmlsrc/rtSaxCppParser.h"
#include "rtxmlsrc/rtXmlCppMsgBuf.h"
#include "rtxsrc/rtxCppXmlString.h"
#include "rtxmlsrc/OSXSDAnyTypeClass.h"
```

Classes

- class OSXMLAnyTypeHandler

6.6.1 Detailed Description

Definition in file rtSaxCppAnyType.h.

6.7 rtSaxCppSimpleType.h File Reference

```
#include "rtxmlsrc/osrtxml.h"
#include "rtxmlsrc/rtSaxCppParser.h"
```

Classes

- class OSXMLSimpleTypeHandler

6.7.1 Detailed Description

Definition in file rtSaxCppSimpleType.h.

6.8 rtSaxCppSoap.h File Reference

```
#include "rtxsrc/rtxCppDynOctStr.h"
#include "rtxsrc/OSRTContext.h"
#include "rtxsrc/OSRTMemBuf.h"
#include "rtxsrc/rtxCppXmlString.h"
#include "rtxmlsrc/osrtxml.h"
#include "rtxmlsrc/rtSaxCppParser.h"
#include "rtxmlsrc/rtXmlCppMsgBuf.h"
```

Classes

- class OSXMLSoapHandler

6.8.1 Detailed Description

Definition in file rtSaxCppSoap.h.

6.9 rtSaxCppStrList.h File Reference

```
#include "rtxsrc/rtxToken.h"
#include "rtxsrc/rtxDList.h"
#include "rtxmlsrc/osrtxml.h"
#include "rtxmlsrc/rtSaxCppParser.h"
#include "rtxsrc/rtxCppDList.h"
```

Classes

- class OSXMLStrListHandler
  OSXMLStrListHandler.

6.9.1 Detailed Description

Definition in file rtSaxCppStrList.h.

6.10 rtXmlCppEncFuncs.h File Reference

XML low-level C++ encode functions.

`#include "rtxmlsrc/osrtxml.h"`

Functions

- int rtXmlCppEncAnyAttr (OSCTXT *pctxt, OSRTObjListClass *pAnyAttrList)

  This function encodes a variable of the XSD any attribute type.

- int rtXmlEncAny (OSCTXT *pctxt, OSXMLStringClass *pxmlstr, const OSUTF8CHAR *elemName, OSXMLNamespace *pNS)

  This function encodes a variable of the XSD any type.

- int rtXmlCppEncAnyTypeValue (OSCTXT *pctxt, OSXSDAnyTypeClass *pvalue)

  This function encodes a variable of the XSD anyType type.

- int rtXmlEncString (OSCTXT *pctxt, OSXMLStringClass *pxmlstr, const OSUTF8CHAR *elemName, OSXMLNamespace *pNS)

  This function encodes a variable of the XSD string type.

6.10.1 Detailed Description

XML low-level C++ encode functions.

These are overloaded versions of C XML encode functions for use with C++.

Definition in file rtXmlCppEncFuncs.h.

6.10.2 Function Documentation

6.10.2.1 int rtXmlCppEncAnyAttr (OSCTXT * *pctxt*, OSRTObjListClass * *pAnyAttrList*)

This function encodes a variable of the XSD any attribute type.

This is expressed as list of name/value pairs.

Parameters:

pctxt Pointer to context block structure.

pAnyAttrList List of name/value pair objects.

Returns:

Completion status of operation:

- 0 = success,
- negative return value is error.

6.10.2.2 int rtXmlCppEncAnyTypeValue (OSCTXT * *pctxt*, OSXSDAnyTypeClass * *pvalue*)

This function encodes a variable of the XSD anyType type.

This is considered to be a fully-wrapped element of anyType type (for example: * <myType>myData</myType>)

Parameters:

pctxt Pointer to context block structure.

pvalue Value to be encoded. This is a pointer to a OSXSDAnyTypeClass containing the fully-encoded XML text to be copied to the output stream.

Returns:

Completion status of operation:

- 0 = success,
- negative return value is error.

6.10.2.3 int rtXmlEncAny (OSCTXT * *pctxt*, OSXMLStringClass * *pxmlstr*, const OSUTF8CHAR * *elemName*, OSXMLNamespace * *pNS*)

This function encodes a variable of the XSD any type.

This is considered to be a fully-wrapped element of any type (for example: <myType>myData</myType>)

Parameters:

pctxt Pointer to context block structure.

pxmlstr Value to be encoded. This is a string containing the fully-encoded XML text to be copied to the output stream.

elemName XML element name. A name must be provided. If an empty string is passed (""), no element tag is added to the encoded value.

pNS Pointer to namespace structure.

Returns:

Completion status of operation:

- 0 = success,
- negative return value is error.

6.10.2.4 int rtXmlEncString (OSCTXT * *pctxt*, OSXMLStringClass * *pxmlstr*, const OSUTF8CHAR * *elemName*, OSXMLNamespace * *pNS*)

This function encodes a variable of the XSD string type.

Parameters:

pctxt Pointer to context block structure.

pxmlstr XML string value to be encoded.

elemName XML element name. A name must be provided. If an empty string is passed (""), no element tag is added to the encoded value.

pNS Pointer to namespace structure.

Returns:

Completion status of operation:

- 0 = success,
- negative return value is error.

6.11 rtXmlCppMsgBuf.h File Reference

This file is deprecated.

```
#include "rtxmlsrc/OSXMLEncodeBuffer.h"
#include "rtxmlsrc/OSXMLEncodeStream.h"
#include "rtxmlsrc/OSXMLDecodeBuffer.h"
```

6.11.1 Detailed Description

This file is deprecated.

Users should use one or more of the individual headers files defined in the include statements below.

Definition in file rtXmlCppMsgBuf.h.

6.12 rtXmlCppNamespace.h File Reference

XML namespace handling structures and function definitions.

```
#include "rtxmlsrc/osrtxml.h"
#include "rtxsrc/OSRTBaseType.h"
#include "rtxsrc/OSRTString.h"
```

Classes

- class OSXMLNamespaceClass

  This class is used to hold an XML namespace prefix to URI mapping.

6.12.1 Detailed Description

XML namespace handling structures and function definitions.

Definition in file rtXmlCppNamespace.h.

6.13 rtXmlCppXSDElement.h File Reference

C++ run-time XML schema global element class definition.

```
#include "rtxsrc/OSRTContext.h"
#include "rtxsrc/OSRTMsgBufIF.h"
#include "rtxsrc/rtxDiag.h"
#include "rtxsrc/rtxErrCodes.h"
#include "rtxmlsrc/osrtxml.h"
```

Classes

- class OSXSDGlobalElement

  XSD global element base class.

6.13.1 Detailed Description

C++ run-time XML schema global element class definition.

Definition in file rtXmlCppXSDElement.h.

6.14 rtXmlpCppDecFuncs.h File Reference

XML low-level C++ decode functions.

```
#include "rtxmlsrc/osrtxml.h"
#include "rtxmlsrc/OSXSDComplexType.h"
```

6.14.1 Detailed Description

XML low-level C++ decode functions.

These are overloaded versions of C XML encode functions for use with C++.

Definition in file rtXmlpCppDecFuncs.h.

Index