

XBinder

XML Schema Compiler
Version 2.0
C++ XML Runtime
Reference Manual

The software described in this document is furnished under a license agreement and may be used only in accordance with the terms of this agreement.

Copyright Notice

Copyright ©1997-2008 Objective Systems, Inc. All rights reserved.

This document may be distributed in any form, electronic or otherwise, provided that it is distributed in its entirety and that the copyright and this notice are included.

Author's Contact Information

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

Contents

1	Main Page	1
2	Class Index	2
2.1	Class Hierarchy	2
3	Class Index	3
3.1	Class List	3
4	File Index	4
4.1	File List	4
5	Class Documentation	5
5.1	OSXMLContentHandler Class Reference	5
5.1.1	Detailed Description	5
5.1.2	Member Function Documentation	6
5.1.2.1	characters	6
5.1.2.2	endElement	6
5.1.2.3	startElement	6
5.2	OSXMLDefaultHandler Class Reference	8
5.2.1	Detailed Description	8
5.2.2	Member Function Documentation	8
5.2.2.1	startElement	8
5.2.2.2	characters	9
5.2.2.3	endElement	9
5.2.2.4	getState	10
5.3	OSXMLDefaultHandlerIF Class Reference	11
5.3.1	Detailed Description	11
5.4	OSXMLNamespaceClass Class Reference	12
5.4.1	Detailed Description	12
5.4.2	Constructor & Destructor Documentation	12

5.4.2.1	OSXMLNamespaceClass	12
5.4.2.2	OSXMLNamespaceClass	13
5.5	OSXMLStrListHandler Class Reference	14
5.5.1	Detailed Description	14
5.6	OSXSDGlobalElement Class Reference	15
5.6.1	Detailed Description	16
5.6.2	Constructor & Destructor Documentation	17
5.6.2.1	OSXSDGlobalElement	17
5.6.2.2	OSXSDGlobalElement	17
5.6.2.3	OSXSDGlobalElement	17
5.6.2.4	~OSXSDGlobalElement	17
5.6.3	Member Function Documentation	17
5.6.3.1	setMsgBuf	17
5.6.3.2	decodeFrom	18
5.6.3.3	encodeTo	18
5.6.3.4	getCtxtPtr	18
5.6.3.5	memAlloc	18
5.6.3.6	memFreePtr	18
5.6.3.7	setDefaultNamespace	19
5.6.3.8	setDiag	19
5.6.3.9	setEncXSINamespace	19
5.6.3.10	setNamespace	19
5.6.3.11	setNoNSSchemaLocation	19
5.6.3.12	setSchemaLocation	20
5.6.3.13	setXSIType	20
5.6.3.14	validateFrom	20
5.6.4	Member Data Documentation	20
5.6.4.1	mpContext	20
6	File Documentation	21
6.1	rtSaxCppAny.h File Reference	21
6.1.1	Detailed Description	21
6.2	rtSaxCppAnyType.h File Reference	22
6.2.1	Detailed Description	22
6.3	rtSaxCppSimpleType.h File Reference	23
6.3.1	Detailed Description	23
6.4	rtSaxCppSoap.h File Reference	24

6.4.1	Detailed Description	24
6.5	rtSaxCppStrList.h File Reference	25
6.5.1	Detailed Description	25
6.6	rtXmlCppEncFuncs.h File Reference	26
6.6.1	Detailed Description	26
6.6.2	Function Documentation	26
6.6.2.1	rtXmlCppEncAnyAttr	26
6.6.2.2	rtXmlCppEncAnyTypeValue	27
6.6.2.3	rtXmlEncAny	27
6.6.2.4	rtXmlEncString	27
6.7	rtXmlCppMsgBuf.h File Reference	29
6.7.1	Detailed Description	29
6.8	rtXmlCppNamespace.h File Reference	30
6.8.1	Detailed Description	30
6.9	rtXmlCppXSDElement.h File Reference	31
6.9.1	Detailed Description	31
6.10	rtXmlpCppDecFuncs.h File Reference	32
6.10.1	Detailed Description	32

Chapter 1

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

Class Index

2.1 Class Hierarchy

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

OSXMLContentHandler	5
OSXMLDefaultHandlerIF	11
OSXMLDefaultHandler	8
OSXMLNamespaceClass	12
OSXMLStrListHandler	14
OSXSDGlobalElement	15

Chapter 3

Class Index

3.1 Class List

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

OSXMLContentHandler (Receive notification of general document events)	5
OSXMLDefaultHandler (This class is derived from the SAX class DefaultHandler base class)	8
OSXMLDefaultHandlerIF (This class is derived from the SAX class DefaultHandler base class)	11
OSXMLNamespaceClass (This class is used to hold an XML namespace prefix to URI mapping)	12
OSXMLStrListHandler (OSXMLStrListHandler)	14
OSXSDGlobalElement (XSD global element base class)	15

Chapter 4

File Index

4.1 File List

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

rtSaxCppAny.h	21
rtSaxCppAnyType.h	22
rtSaxCppParser.h	??
rtSaxCppParserIF.h	??
rtSaxCppSimpleType.h	23
rtSaxCppSoap.h	24
rtSaxCppStrList.h	25
rtXmlCppEncFuncs.h (XML low-level C++ encode functions)	26
rtXmlCppMsgBuf.h (This file is deprecated)	29
rtXmlCppNamespace.h (XML namespace handling structures and function definitions)	30
rtXmlCppXSDElement.h (C++ run-time XML schema global element class definition)	31
rtXmlpCppDecFuncs.h (XML low-level C++ decode functions)	32

Chapter 5

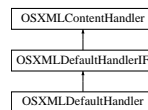
Class Documentation

5.1 OSXMLContentHandler Class Reference

Receive notification of general document events.

```
#include <rtSaxCppParserIF.h>
```

Inheritance diagram for OSXMLContentHandler::



Public Member Functions

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.

5.1.1 Detailed Description

Receive notification of general document events.

Definition at line 112 of file rtSaxCppParserIF.h.

5.1.2 Member Function Documentation

5.1.2.1 `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](#).

5.1.2.2 `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 associated namespace for this element

localname The local part of the element name

qname The QName of this element

Implemented in [OSXMLDefaultHandler](#).

5.1.2.3 `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 associated 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 [OSXMLDefaultHandler](#).

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

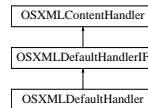
- [rtSaxCppParserIF.h](#)

5.2 OSXMLDefaultHandler Class Reference

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

```
#include <rtSaxCppParser.h>
```

Inheritance diagram for OSXMLDefaultHandler::



Public Member Functions

- virtual EXTERNALMETHOD 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 EXTERNALMETHOD int [characters](#) (const OSUTF8CHAR *const chars, unsigned int length)
Receive notification of character data.
- virtual EXTERNALMETHOD 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.

5.2.1 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.

5.2.2 Member Function Documentation

- #### 5.2.2.1 virtual EXTERNALMETHOD 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 associated 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](#).

5.2.2.2 virtual EXTERNALMETHOD 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](#).

5.2.2.3 virtual EXTERNALMETHOD 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 associated namespace for this element
- localname* The local part of the element name
- qname* The QName of this element

Implements [OSXMLContentHandler](#).

5.2.2.4 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 XML-LEND.

Definition at line 124 of file rtSaxCppParser.h.

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

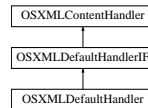
- rtSaxCppParser.h

5.3 OSXMLDefaultHandlerIF Class Reference

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

```
#include <rtSaxCppParserIF.h>
```

Inheritance diagram for OSXMLDefaultHandlerIF::



5.3.1 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

5.4 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.

5.4.1 Detailed Description

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

Definition at line 38 of file rtXmlCppNamespace.h.

5.4.2 Constructor & Destructor Documentation

5.4.2.1 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.

5.4.2.2 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](#)

5.5 OSXMLStrListHandler Class Reference

[OSXMLStrListHandler](#).

```
#include <rtSaxCppStrList.h>
```

5.5.1 Detailed Description

[OSXMLStrListHandler](#).

Definition at line 41 of file `rtSaxCppStrList.h`.

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

- [rtSaxCppStrList.h](#)

5.6 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.

5.6.1 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.

5.6.2 Constructor & Destructor Documentation

5.6.2.1 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.

5.6.2.2 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.

5.6.2.3 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.

5.6.2.4 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.

5.6.3 Member Function Documentation

5.6.3.1 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.

5.6.3.2 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`.

5.6.3.3 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`.

5.6.3.4 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`.

5.6.3.5 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`.

5.6.3.6 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`.

5.6.3.7 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 rtXmlCxxXSDElement.h.

5.6.3.8 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 rtXmlCxxXSDElement.h.

5.6.3.9 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 rtXmlCxxXSDElement.h.

5.6.3.10 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 rtXmlCxxXSDElement.h.

5.6.3.11 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 rtXmlCxxXSDElement.h.

5.6.3.12 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.

5.6.3.13 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.

5.6.3.14 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.

5.6.4 Member Data Documentation

5.6.4.1 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

File Documentation

6.1 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"
```

6.1.1 Detailed Description

Definition in file [rtSaxCppAny.h](#).

6.2 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"
```

6.2.1 Detailed Description

Definition in file [rtSaxCppAnyType.h](#).

6.3 rtSaxCppSimpleType.h File Reference

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

6.3.1 Detailed Description

Definition in file [rtSaxCppSimpleType.h](#).

6.4 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"
```

6.4.1 Detailed Description

Definition in file [rtSaxCppSoap.h](#).

6.5 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.

6.5.1 Detailed Description

Definition in file [rtSaxCppStrList.h](#).

6.6 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.

6.6.1 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](#).

6.6.2 Function Documentation

6.6.2.1 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.

6.6.2.2 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.

6.6.2.3 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.

6.6.2.4 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.

6.7 rtXmlCppMsgBuf.h File Reference

This file is deprecated.

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

6.7.1 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](#).

6.8 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.

6.8.1 Detailed Description

XML namespace handling structures and function definitions.

Definition in file [rtXmlCppNamespace.h](#).

6.9 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.

6.9.1 Detailed Description

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

Definition in file [rtXmlCppXSDElement.h](#).

6.10 rtXmlpCppDecFuncs.h File Reference

XML low-level C++ decode functions.

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

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

6.10.1 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

- ~OSXSDGlobalElement
 - [OSXSDGlobalElement](#), [17](#)
 - characters
 - [OSXMLContentHandler](#), [6](#)
 - [OSXMLDefaultHandler](#), [9](#)
 - decodeFrom
 - [OSXSDGlobalElement](#), [17](#)
 - encodeTo
 - [OSXSDGlobalElement](#), [18](#)
 - endElement
 - [OSXMLContentHandler](#), [6](#)
 - [OSXMLDefaultHandler](#), [9](#)
 - getCtxtPtr
 - [OSXSDGlobalElement](#), [18](#)
 - getState
 - [OSXMLDefaultHandler](#), [9](#)
 - memAlloc
 - [OSXSDGlobalElement](#), [18](#)
 - memFreePtr
 - [OSXSDGlobalElement](#), [18](#)
 - mpContext
 - [OSXSDGlobalElement](#), [20](#)
 - OSXMLContentHandler, [5](#)
 - [characters](#), [6](#)
 - [endElement](#), [6](#)
 - [startElement](#), [6](#)
 - OSXMLDefaultHandler, [8](#)
 - [characters](#), [9](#)
 - [endElement](#), [9](#)
 - [getState](#), [9](#)
 - [startElement](#), [8](#)
 - OSXMLDefaultHandlerIF, [11](#)
 - OSXMLNamespaceClass, [12](#)
 - [OSXMLNamespaceClass](#), [12](#), [13](#)
 - OSXMLStrListHandler, [14](#)
 - OSXSDGlobalElement, [15](#)
 - [~OSXSDGlobalElement](#), [17](#)
 - [decodeFrom](#), [17](#)
 - [encodeTo](#), [18](#)
 - [getCtxtPtr](#), [18](#)
 - [memAlloc](#), [18](#)
 - [memFreePtr](#), [18](#)
 - [mpContext](#), [20](#)
 - [OSXSDGlobalElement](#), [17](#)
 - [setDefaultNamespace](#), [18](#)
 - [setDiag](#), [19](#)
 - [setEncXSINamespace](#), [19](#)
 - [setMsgBuf](#), [17](#)
 - [setNamespace](#), [19](#)
 - [setNoNSSchemaLocation](#), [19](#)
 - [setSchemaLocation](#), [19](#)
 - [setXSIType](#), [20](#)
 - [validateFrom](#), [20](#)
 - [rtSaxCppAny.h](#), [21](#)
 - [rtSaxCppAnyType.h](#), [22](#)
 - [rtSaxCppSimpleType.h](#), [23](#)
 - [rtSaxCppSoap.h](#), [24](#)
 - [rtSaxCppStrList.h](#), [25](#)
 - [rtXmlCppEncAnyAttr](#)
 - [rtXmlCppEncFuncs.h](#), [26](#)
 - [rtXmlCppEncAnyTypeValue](#)
 - [rtXmlCppEncFuncs.h](#), [26](#)
 - [rtXmlCppEncFuncs.h](#), [26](#)
 - [rtXmlCppEncAnyAttr](#), [26](#)
 - [rtXmlCppEncAnyTypeValue](#), [26](#)
 - [rtXmlEncAny](#), [27](#)
 - [rtXmlEncString](#), [27](#)
 - [rtXmlCppMsgBuf.h](#), [29](#)
 - [rtXmlCppNamespace.h](#), [30](#)
 - [rtXmlCppXSDElement.h](#), [31](#)
 - [rtXmlEncAny](#)
 - [rtXmlCppEncFuncs.h](#), [27](#)
 - [rtXmlEncString](#)
 - [rtXmlCppEncFuncs.h](#), [27](#)
 - [rtXmlpCppDecFuncs.h](#), [32](#)
- [setDefaultNamespace](#)
 - [OSXSDGlobalElement](#), [18](#)
 - [setDiag](#)
 - [OSXSDGlobalElement](#), [19](#)
 - [setEncXSINamespace](#)
 - [OSXSDGlobalElement](#), [19](#)
 - [setMsgBuf](#)
 - [OSXSDGlobalElement](#), [17](#)

setNamespace
 OSXSDGlobalElement, 19

setNoNSSchemaLocation
 OSXSDGlobalElement, 19

setSchemaLocation
 OSXSDGlobalElement, 19

setXSIType
 OSXSDGlobalElement, 20

startElement
 OSXMLContentHandler, 6
 OSXMLDefaultHandler, 8

validateFrom
 OSXSDGlobalElement, 20