Document serial number: 000004892117

libxml Reference

000004892117

PlayStation®Vita Programmer Tool Runtime Library Release 3.50

© 2011 Sony Computer Entertainment Inc.
All Rights Reserved.
SCE Confidential

Document serial number: 000004892117

Document serial number: 000004892117

Document serial number: 000004892117

Table of Contents

000004892117

Document serial number: 000004892117

©SCEI

Document serial number: 000004892117

Document serial number: 000004892117

000004892117

Document serial number: 000004892117

Document serial number: 000004892117

Document serial number: 000004892117

000004892117

Document serial number: 000004892117

Document serial number: 000004892117

Document serial number: 000004892117

000004892117

Document serial number: 000004892117

©SCEI

Document serial number: 000004892117

Document serial number: 000004892117

000004892117

Document serial number: 000004892117

©SCEI

Document serial number: 000004892117

Document serial number: 000004892117

000004892117

Document serial number: 000004892117

©SCEI

Document serial number: 000004892117

Document serial number: 000004892117

Introduction

000004892117

Document serial number: 000004892117

©SCEI

Document serial number: 000004892117

Document serial number: 000004892117

Library Summary

Library Contents

Item	Description
sce::Xml	Namespace of XML processer
sce::Xml::Attr	Attr interface class
sce::Xml::AttributeList	AttributeList interface class
sce::Xml::Dom	Namespace of DOM API
sce::Xml::Dom::Document	DOM interface class
sce::Xml::Dom::DocumentBuilder	DOM creation interface class
sce::Xml::Dom::Node	Node interface class
sce::Xml::Dom::NodeList	NodeList interface class
sce::Xml::Initializer	Object initialization interface class
sce::Xml::InitParameter	Initialization parameter class
sce::Xml::MemAllocator	Memory allocator interface class
sce::Xml::Sax	Namespace of SAX API
sce::Xml::Sax::DocumentHandler	SAX event handler interface class
sce::Xml::Sax::Parser	SAX interface class
sce::Xml::SerializeParameter	XML output parameter class
sce::Xml::SimpleData	Class which holds a pointer to and length of data
sce::Xml::String	Class which holds a pointer to and length of a character string
sce::Xml::Util	Namespace of utility

000004892117

Document serial number: 000004892117

©SCEI

Document serial number: 000004892117

Document serial number: 000004892117

Defines

000004892117

Document serial number: 000004892117

©SCEI

Document serial number: 000004892117

Document serial number: 000004892117

List of Definitions

Macro definitions

Definition

Definition	Value	Description
`SCE_XML_ATTR_NAME_SIZE_MAX`	`(128)`	Maximum size of an attribute name in DOM. If user provides attribute name bigger than this, it will be truncated to fit this size.
`SCE_XML_ELEMENT_NAME_SIZE_MAX`	`(1024)`	Maximum size of an element name in DOM. If user provides element name bigger than this, it will be truncated to fit this size.
`SCE_XML_INVALID_INDEX`	`((index_t)-1)`	Invalid index value
`SCE_XML_INVALID_NODE_ID`	`(0ULL)`	Invalid node ID value
`SCE_XML_INVALID_SIZE`	`((size_t)-1)`	Invalid size value
`SCE_XML_SERIALIZE_OPT_USE_XML_DECLARATION`	`(1)`	Option which outputs XML declaration

000004892117

Document serial number: 000004892117

©SCEI

Document serial number: 000004892117

Document serial number: 000004892117

sce::Xml

000004892117

Document serial number: 000004892117

©SCEI

Document serial number: 000004892117

Document serial number: 000004892117

Summary

sce::Xml

Namespace of XML processer

Definition

```
namespace Xml {}
```

Description

Namespace of XML processer

Variables

Public Variables

const index_t invalidIndex	Value which represents invalid index
const size_t invalidSize	Value which represents invalid size

Internal classes, Structures, Namespaces

Item	Description
sce::Xml::Attr	Attr interface class
sce::Xml::AttributeList	AttributeList interface class
sce::Xml::Dom	Namespace of DOM API
sce::Xml::Initializer	Object initialization interface class
sce::Xml::InitParameter	Initialization parameter class
sce::Xml::MemAllocator	Memory allocator interface class
sce::Xml::Sax	Namespace of SAX API
sce::Xml::SerializeParameter	XML output parameter class
sce::Xml::SimpleData	Class which holds a pointer to and length of data
sce::Xml::String	Class which holds a pointer to and length of a character string
sce::Xml::Util	Namespace of utility

000004892117

Document serial number: 000004892117

©SCEI

Document serial number: 000004892117

Document serial number: 000004892117

Enumeration Type

EntityType

Types of entity

Definition

```
#include <xml/xml_types.h>
namespace sce {
        namespace Xml {
              enum EntityType {
                     entityUnknown = 0,
                     entityCharReference,
                     entityBuiltInAmp,
                     entityBuiltInQuot,
                     entityBuiltInLt,
                     entityBuiltInGt,
                     entityBuiltInApos
              };
        }
}
```

Enumeration Values

Macro	Value	Description
`entityUnknown`	0	Entity type is unknown
`entityCharReference`	1	Character reference entity
`entityBuiltInAmp`	2	Predefined entity: &
`entityBuiltInQuot`	3	Predefined entity: "
`entityBuiltInLt`	4	Predefined entity: <
`entityBuiltInGt`	5	Predefined entity: >
`entityBuiltInApos`	6	Predefined entity: '

Description

These are the types of entity.

000004892117

Document serial number: 000004892117

Document serial number: 000004892117

Document serial number: 000004892117

result_t

Result codes

Definition

```
#include <xml/xml_result.h>
namespace sce {
        namespace Xml {
              enum result_t {
                     resultSuccess = 0,
                     resultGenericError = SCE_XML_PARSER_ERROR_OFFSET,
                     resultNoMemory,
                     resultNotInitialized,
                     resultInvalidArgument,
                     resultNotSupported,
                     resultInitializeFailed,
                     resultInvalidBinXml,
                     resultParserBusy,
                     resultXmlUnexpextedEoF,
                     resultXmlSyntaxError,
                     resultXmlEndTagMismatch,
                     resultXmlInvalidChar,
                     resultXmlInvalidDecValue,
                     resultXmlInvalidHexValue,
                     resultXmlClosingAngleBracketCharNotFound,
                     resultXmlEqualityCharNotFound,
                     resultXmlSemiColonCharNotFound,
                     resultXmlQuoteCharNotFound,
                     resultXmlEndOfCommentNotFound,
                     resultXmlEndOfCDSectNotFound,
                     resultXmlEndOfDtdNotFound,
                     resultXmlUnknownEncoding,
                     resultXmlHandlerNotSet,
                     resultXmlInvalidPi,
                     resultXmlInvalidDocumentElement,
                     resultXmlDocumentElementNotFound,
                     resultXmlDuplicateAttrName,
                     resultDomError = SCE_XML_PARSER_ERROR_OFFSET + 0x200,
                     resultDomNodeNotFound,
                     resultDomReadOnlyError,
                     resultDomMaxUniqueElementError,
                     resultDomMaxUniqueAttrError,
                     resultDomMaxNumOfAttrError,
                     resultDomMaxSizeOfElementNameError,
                     resultDomMaxSizeOfAttrNameError,
                     resultDomMaxSizeOfAttrValueError,
                     resultDomInvalidEnitity,
                     resultDomInvalidNodeType,
                     resultGenericMessage = SCE_XML_PARSER_INFO_OFFSET,
                     resultXmlParseInProgress,
                     resultXmlParseAborted,
                     resultMultipleInitialized
              };
        }
}
```

000004892117

Document serial number: 000004892117

Document serial number: 000004892117

Document serial number: 000004892117

Enumeration Values

Macro	Value	Description
`resultSuccess`	0x00000000	Success
`resultGenericError`	0x80850000	General error
`resultNoMemory`	0x80850001	Insufficient memory
`resultNotInitialized`	0x80850002	Not initialized
`resultInvalidArgument`	0x80850003	Invalid argument
`resultNotSupported`	0x80850004	Non-supported function
`resultInitializeFailed`	0x80850005	Initialization error
`resultInvalidBinXml`	0x80850006	Invalid binary XML data
`resultParserBusy`	0x80850007	Parser is busy right now
`resultXmlUnexpextedEoF`	0x80850008	Unexpected end of file of the XML document
`resultXmlSyntaxError`	0x80850009	Syntax error exists in the XML document
`resultXmlEndTagMismatch`	0x8085000a	End tag in the XML document is not matched
`resultXmlInvalidChar`	0x8085000b	Invalid character exists in the XML document
`resultXmlInvalidDecValue`	0x8085000c	Decimal character in the XML document is invalid
`resultXmlInvalidHexValue`	0x8085000d	Hexadecimal character in the XML document is invalid
`resultXmlClosingAngleBracketCharNotFound`	0x8085000e	Closing parenthesis is missing in the XML document
`resultXmlEqualityCharNotFound`	0x8085000f	No equal character is found in the XML document
`resultXmlSemiColonCharNotFound`	0x80850010	No semi colon is found in the XML document
`resultXmlQuoteCharNotFound`	0x80850011	No quote character is found in the XML document
`resultXmlEndOfCommentNotFound`	0x80850012	End of comment does not exist in the XML document
`resultXmlEndOfCDSectNotFound`	0x80850013	End of CDATA does not exist in the XML document
`resultXmlEndOfDtdNotFound`	0x80850014	No end of DTD is found in the XML document
`resultXmlUnknownEncoding`	0x80850015	Unknown Encoding
`resultXmlHandlerNotSet`	0x80850016	No handler is set
`resultXmlInvalidPi`	0x80850017	PI in the XML document is invalid
`resultXmlInvalidDocumentElement`	0x80850018	Elements in the XML document are invalid
`resultXmlDocumentElementNotFound`	0x80850019	No elements are found in the XML document
`resultXmlDuplicateAttrName`	0x8085001a	Duplicate attribute names are found in the XML document
`resultDomError`	0x80850200	DOM general error
`resultDomNodeNotFound`	0x80850201	DOM operation target is not found

000004892117

Document serial number: 000004892117

Document serial number: 000004892117

Document serial number: 000004892117

Macro	Value	Description
`resultDomReadOnlyError`	0x80850202	DOM operation target is read only
`resultDomMaxUniqueElementError`	0x80850203	The number of elements of node exceeds the maximum value in DOM
`resultDomMaxUniqueAttrError`	0x80850204	The number of attributes of node exceeds the maximum number in DOM
`resultDomMaxNumOfAttrError`	0x80850205	The number of attributes of 1 node exceeds the maximum number in DOM
`resultDomMaxSizeOfElementNameError`	0x80850206	The element name of 1 node exceeds the maximum length in DOM
`resultDomMaxSizeOfAttrNameError`	0x80850207	The attribute name of 1 node exceeds the maximum length in DOM
`resultDomMaxSizeOfAttrValueError`	0x80850208	The attribute value of 1 node exceeds the maximum length in DOM
`resultDomInvalidEnitity`	0x80850209	Invalid entity in DOM
`resultDomInvalidNodeType`	0x8085020a	Invalid node type in DOM
`resultGenericMessage`	0x00850000	General message
`resultXmlParseInProgress`	0x00850001	Parse processing is being performed
`resultXmlParseAborted`	0x00850002	Parse processing is aborted by user
`resultMultipleInitialized`	0x00850003	Initialization processing is called more than once

Description

These are the numeric values which represent the result whose return value type is `int`. Negative values represent the failure, and positive values represent the information.

000004892117

Document serial number: 000004892117

©SCEI

Document serial number: 000004892117

Document serial number: 000004892117

TokenType

Types of token

Definition

```
#include <xml/xml_types.h>
namespace sce {
        namespace Xml {
              enum TokenType {
                     tokenUnknown = -1,
                     tokenDtd = 4,
                     tokenDtdEnd = 5,
                     tokenPi = 6,
                     tokenPiEnd = 7,
                     tokenCdata = 8,
                     tokenCdataEnd = 9,
                     tokenComment = 10,
                     tokenCommentEnd = 11,
                     tokenUnexpected = 12,
              };
        }
}
```

Enumeration Values

Macro	Value	Description
`tokenUnknown`	0xffffffff	Type is unknown
`tokenDtd`	0x00000004	Document Type Definition (DTD)
`tokenDtdEnd`	0x00000005	End of DTD
`tokenPi`	0x00000006	XML Processing Instruction (PI)
`tokenPiEnd`	0x00000007	End of PI
`tokenCdata`	0x00000008	CDATA section
`tokenCdataEnd`	0x00000009	End of CDATA section
`tokenComment`	0x0000000a	Comment
`tokenCommentEnd`	0x0000000b	End of comment
`tokenUnexpected`	0x0000000c	Unexpected token

Description

These are the types of token.

SAX event handler `skippedText` returns these values.

000004892117

Document serial number: 000004892117

Document serial number: 000004892117

Document serial number: 000004892117

Type Definition

index_t

A type which represents index

Definition

```
#include <xml/xml_types.h>
namespace sce {
        namespace Xml {
              typedef SceUInt32 index_t;
        }
}
```

Description

This is a type which represents an index.

000004892117

Document serial number: 000004892117

©SCEI

Document serial number: 000004892117

Document serial number: 000004892117

size_t

A type which represents size

Definition

```
#include <xml/xml_types.h>
namespace sce {
        namespace Xml {
              typedef SceUInt32 size_t;
        }
}
```

Description

This is a type which represents a size.

000004892117

Document serial number: 000004892117

©SCEI

Document serial number: 000004892117

Document serial number: 000004892117

XmlText

A type for handling XML document

Definition

```
#include <xml/xml_types.h>
namespace sce {
        namespace Xml {
              typedef String XmlText;
        }
}
```

Description

This is a type for handling an XML document. This type is the same as `String`.

000004892117

Document serial number: 000004892117

©SCEI

Document serial number: 000004892117

Document serial number: 000004892117

sce::Xml::Attr

000004892117

Document serial number: 000004892117

©SCEI

Document serial number: 000004892117

Document serial number: 000004892117

Summary

sce::Xml::Attr

Attribute interface class

Definition

```
#include <xml/xml_attribute.h>
class Attr {};
```

Description

This represents the attributes.

The attribute list `AttributeList` passed from the `startElement` event is valid only during the scope of the event; once the event handler returns control to the parser, the attribute list becomes invalid. At the same time, Attr also becomes invalid.

```
int startElement(void *userData, const String *name,
                    const AttributeList* list)
{
      for(size_t k=0; k < list->getLength(); k++) {
            const Attr& att = list->getAttribute(k);

            const String& name = att.getName();
            const String& value = att.getValue();
      }
}
```

(Note that the result of `list->getLength()` will be zero if there are no attributes.)

Method List

Method	Description
`Attr`	Constructor
`~Attr`	Destructor
`getName`	Get the attribute name
`getValue`	Get the attribute value
`initialize`	Initialize
`isAvailable`	Check the availability
`setName`	Set the attribute name
`setValue`	Set the attribute value
`terminate`	Terminate

000004892117

Document serial number: 000004892117

©SCEI

Document serial number: 000004892117

Document serial number: 000004892117

Constructors and Destructors

Attr

Constructor

Definition

```
#include <xml/xml_attribute.h>
namespace sce {
        namespace Xml {
              class Attr {
                    Attr();

                    Attr(const Attr &);
              }
        }
}
```

Return Values

None

Description

Constructor

000004892117

Document serial number: 000004892117

©SCEI

Document serial number: 000004892117

Document serial number: 000004892117

~Attr

Destructor

Definition

```
#include <xml/xml_attribute.h>
namespace sce {
        namespace Xml {
              class Attr {
                     ~Attr();
              }
        }
}
```

Return Values

None

Description

Destructor

000004892117

Document serial number: 000004892117

©SCEI

Document serial number: 000004892117

Document serial number: 000004892117

Public Instance Methods

getName

Get the attribute name

Definition

```
#include <xml/xml_attribute.h>
namespace sce {
        namespace Xml {
              class Attr {
                    String getName() const;
              }
        }
}
```

Return Values

The attribute name associated to this Attr.

Description

This gets the attribute name.

000004892117

Document serial number: 000004892117

©SCEI

Document serial number: 000004892117

Document serial number: 000004892117

getValue

Get the attribute value

Definition

```
#include <xml/xml_attribute.h>
namespace sce {
        namespace Xml {
              class Attr {
                     String getValue() const;
              }
        }
}
```

Return Values

The attribute value associated to this Attr.

Description

This gets the attribute value.

000004892117

Document serial number: 000004892117

©SCEI

Document serial number: 000004892117

Document serial number: 000004892117

initialize

Initialize

Definition

```
#include <xml/xml_attribute.h>
namespace sce {
        namespace Xml {
              class Attr {
                     int initialize(
                            const Initializer *init
                     );
              }
        }
}
```

Arguments

init	(in) Pointer to Initializer object

Return Values

Value	Description
<0	Error code
SCE_OK	Success

Description

This API must be called before another interface of attr interface class is used.

000004892117

Document serial number: 000004892117

©SCEI

Document serial number: 000004892117

Document serial number: 000004892117

isAvailable

Check the availability

Definition

```
#include <xml/xml_attribute.h>
namespace sce {
        namespace Xml {
              class Attr {
                     bool isAvailable() const;
              }
        }
}
```

Return Values

Value	Description
`true`	Initialized and ready to use
`false`	Not initialized

Description

This returns `true` if this object is available.

000004892117

Document serial number: 000004892117

©SCEI

Document serial number: 000004892117

Document serial number: 000004892117

setName

Set the attribute name

Definition

```
#include <xml/xml_attribute.h>
namespace sce {
        namespace Xml {
              class Attr {
                     int setName(const String *);
              }
        }
}
```

Description

This sets the attribute name.

000004892117

Document serial number: 000004892117

©SCEI

Document serial number: 000004892117

Document serial number: 000004892117

setValue

Set the attribute value

Definition

```
#include <xml/xml_attribute.h>
namespace sce {
        namespace Xml {
              class Attr {
                     int setValue(const String *);
              }
        }
}
```

Description

This sets the attribute value.

000004892117

Document serial number: 000004892117

©SCEI

Document serial number: 000004892117

Document serial number: 000004892117

terminate

Terminate

Definition

```
#include <xml/xml_attribute.h>
namespace sce {
        namespace Xml {
              class Attr {
                     int terminate();
              }
        }
}
```

Return Values

Value	Description
SCE_OK	Success

Description

This terminates and destroys any local objects.

000004892117

Document serial number: 000004892117

©SCEI

Document serial number: 000004892117

Document serial number: 000004892117

sce::Xml::AttributeList

000004892117

Document serial number: 000004892117

©SCEI

Document serial number: 000004892117

Document serial number: 000004892117

Summary

sce::Xml::AttributeList

`AttributeList` interface class

Definition

```
#include <xml/xml_attribute_list.h>
class AttributeList {};
```

Description

This represents attribute list

Method List

Method	Description
addAttribute	Add a new attribute to the list
AttributeList	Constructor
~AttributeList	Destructor
clear	Clear the attribute list
getAttribute	Find and get the Attr object
getLength	Get the number of Attr objects
initialize	Initialize
isAvailable	Check the availability
terminate	Terminate

000004892117

Document serial number: 000004892117

©SCEI

Document serial number: 000004892117

Document serial number: 000004892117

Constructors and Destructors

AttributeList

Constructor

Definition

```
#include <xml/xml_attribute_list.h>
namespace sce {
        namespace Xml {
              class AttributeList {
                    AttributeList();

                    AttributeList(const AttributeList &);
              }
        }
}
```

Return Values

None

Description

Constructor

000004892117

Document serial number: 000004892117

©SCEI

Document serial number: 000004892117

Document serial number: 000004892117

~AttributeList

Destructor

Definition

```
#include <xml/xml_attribute_list.h>
namespace sce {
        namespace Xml {
              class AttributeList {
                     ~AttributeList();
              }
        }
}
```

Return Values

None

Description

Destructor

000004892117

Document serial number: 000004892117

©SCEI

Document serial number: 000004892117

Document serial number: 000004892117

Public Instance Methods

addAttribute

Add a new attribute to the list

Definition

```
#include <xml/xml_attribute_list.h>
namespace sce {
        namespace Xml {
              class AttributeList {
                    int addAttribute(
                          const String *name,
                          const String *value
                    );
              }
        }
}
```

Arguments

name	(in) Name set to Attr
value	(in) Value set to Attr

Return Values

Value	Description
<0	Error code
SCE_OK	Success

Description

This adds a new attribute to the list with name and value.

Notes

Only whether the attribute name *name* is not NULL is checked; the conformance to the XML specification is not checked. When using with R/W DOM APIs, the attribute name and value can be set with any string; it is user's responsibility to check for the valid XML name as needed. If invalid name or invalid value is specified, resulting XML document after serialization may not be well-formed document.

000004892117

Document serial number: 000004892117

Document serial number: 000004892117

Document serial number: 000004892117

clear

Clear the attribute list

Definition

```
#include <xml/xml_attribute_list.h>
namespace sce {
        namespace Xml {
              class AttributeList {
                     void clear();
              }
        }
}
```

Return Values

None

Description

If the attribute list has to be used very frequently, then using this function clears the attributes from the list logically while possessing the memory. The logically deleted objects can be reused; thus the performance will be improved.

000004892117

Document serial number: 000004892117

©SCEI

Document serial number: 000004892117

Document serial number: 000004892117

getAttribute

Find and get the Attr object

Definition

```
#include <xml/xml_attribute_list.h>
namespace sce {
        namespace Xml {
              class AttributeList {
                     Attr getAttribute(
                            const String *attName
                     ) const;

                     Attr getAttribute(
                            index_t index
                     ) const;
              }
        }
}
```

Arguments

attName	(in) Attr name to be searched
index	(in) Attr index to be searched

Return Values

The `Attr` object whose *attName* or *index* is matched, or invalid `Attr` if nothing is found.

Description

This gets the `Attr` object by specifying the name "`attName`" or the index "`index`".

000004892117

Document serial number: 000004892117

Document serial number: 000004892117

Document serial number: 000004892117

getLength

Get the number of Attr objects

Definition

```
#include <xml/xml_attribute_list.h>
namespace sce {
        namespace Xml {
              class AttributeList {
                     size_t getLength() const;
              }
        }
}
```

Return Values

The number of attributes in the list.

Description

This gets the number of attributes in the `AttributeList`.

The return value of `getLength()` is zero when any attributes do not exist.

000004892117

Document serial number: 000004892117

©SCEI

Document serial number: 000004892117

Document serial number: 000004892117

initialize

Initialize

Definition

```
#include <xml/xml_attribute_list.h>
namespace sce {
        namespace Xml {
              class AttributeList {
                     int initialize(
                            const Initializer *init
                     );
              }
        }
}
```

Arguments

`init` (in) Pointer to `Initializer` object

Return Values

Value	Description
<0	Error code
`SCE_OK`	Success

Description

This API must be called before another interface is used.

000004892117

Document serial number: 000004892117

©SCEI

Document serial number: 000004892117

Document serial number: 000004892117

isAvailable

Check the availability

Definition

```
#include <xml/xml_attribute_list.h>
namespace sce {
        namespace Xml {
              class AttributeList {
                     bool isAvailable() const;
              }
        }
}
```

Return Values

Value	Description
`true`	Initialized and ready to use
`false`	Not initialized

Description

This returns `true` if this object is available.

000004892117

Document serial number: 000004892117

©SCEI

Document serial number: 000004892117

Document serial number: 000004892117

terminate

Terminate

Definition

```
#include <xml/xml_attribute_list.h>
namespace sce {
        namespace Xml {
              class AttributeList {
                     int terminate();
              }
        }
}
```

Return Values

Value	Description
`SCE_OK`	Success

Description

This destroys any local objects.

000004892117

Document serial number: 000004892117

©SCEI

Document serial number: 000004892117

Document serial number: 000004892117

sce::Xml::Dom

000004892117

Document serial number: 000004892117

©SCEI

Document serial number: 000004892117

Document serial number: 000004892117

Summary

sce::Xml::Dom

Namespace of DOM API

Definition

```
namespace Dom {}
```

Description

Namespace of DOM API

Variables

Public Variables

`const NodeId invalidNodeId` Value which represents invalid node ID

Internal classes, Structures, Namespaces

Item	Description
`sce::Xml::Dom::Document`	DOM interface class
`sce::Xml::Dom::DocumentBuilder`	DOM creation interface class
`sce::Xml::Dom::Node`	`Node` interface class
`sce::Xml::Dom::NodeList`	`NodeList` interface class

000004892117

Document serial number: 000004892117

©SCEI

Document serial number: 000004892117

Document serial number: 000004892117

Enumeration Type

NodeType

Node type

Definition

```
#include <xml/xml_node_types.h>
namespace sce {
        namespace Xml {
              namespace Dom {
                     enum NodeType {
                            nodeUnknown = SCE_XML_TOKEN_DOM_UNKNOWN,
                            nodeElement = SCE_XML_TOKEN_DOM_ELEMENT,
                            nodeElementRw = SCE_XML_TOKEN_DOM_RW_ELEMENT,
                            nodeAttribute = SCE_XML_TOKEN_DOM_ATTR,
                            nodeText = SCE_XML_TOKEN_DOM_TEXT,
                            nodeTexthasChild =
SCE_XML_TOKEN_DOM_HAS_TEXT_CHILD,
                            nodeCdataSection =
SCE_XML_TOKEN_DOM_SKIP_TEXT_CDATA,
                            nodeEntity = SCE_XML_TOKEN_DOM_ENTITY,
                            nodeProcessingInstruction =
SCE_XML_TOKEN_DOM_SKIP_TEXT_PI,
                            nodeComment =
SCE_XML_TOKEN_DOM_SKIP_TEXT_COMMENT,
                            nodeDocument = SCE_XML_TOKEN_DOM_DOCUMENT
                     };
              }
        }
}
```

Enumeration Values

Macro	Value	Description
`nodeUnknown`	0xffffffff	Node type is unknown
`nodeElement`	0x00000004	Element node
`nodeElementRw`	0x00000005	Element node (Read/Write)
`nodeAttribute`	0x00000001	Attribute node
`nodeText`	0x00000008	Text node
`nodeTexthasChild`	0x00000028	Text node
`nodeCdataSection`	0x00000007	CDATA node
`nodeEntity`	0x00000002	Entity node
`nodeProcessingInstruction`	0x00000017	PI node
`nodeComment`	0x0000001f	Comment node
`nodeDocument`	0x00000000	Document node

Description

These node types are acquired using `Document::getNodeType()`.

000004892117

Document serial number: 000004892117

Document serial number: 000004892117

Document serial number: 000004892117

Type Definition

MetaNodeId

Metanode ID

Definition

```
#include <xml/xml_types.h>
namespace sce {
        namespace Xml {
              namespace Dom {
                     typedef NodeId MetaNodeId;
              }
        }
}
```

Description

This ID identifies a node acquired with `getXmlMeta()`, and is a 64-bit integer value like `NodeId`.

000004892117

Document serial number: 000004892117

©SCEI

Document serial number: 000004892117

Document serial number: 000004892117

NodeId

Node ID

Definition

```
#include <xml/xml_types.h>
namespace sce {
        namespace Xml {
             namespace Dom {
                    typedef SceUInt64 NodeId;
             }
        }
}
```

Description

This ID identifies the node and is a 64-bit integer value.

000004892117

Document serial number: 000004892117

©SCEI

Document serial number: 000004892117

Document serial number: 000004892117

sce::Xml::Dom::Document

000004892117

Document serial number: 000004892117

©SCEI

Document serial number: 000004892117

Document serial number: 000004892117

Summary

sce::Xml::Dom::Document

DOM interface class

Definition

```
#include <xml/xml.h>
class Document {};
```

Description

DOM interface class

Method List

Method	Description
addElementChild	Create and add an element
createElement	Creates an element
createTextNode	Creates a Text node
Document	Constructor
~Document	Destructor
getAttribute	Get the attribute value
getAttributes	Get attributes
getAttrName	Get the attribute name
getAttrValue	Get the attribute value
getChildNodes	Get child nodes
getDocRoot	Get the document root node
getElementsByTagName	Get a NodeList by the name
getEntity	Get the entity text
getEntityType	Get the entity type
getFirstAttr	Get the first attribute
getFirstChild	Get the child nodes
getLastChild	Get the last child node
getNextAttr	Get the next attribute
getNodeName	Get the node name
getNodeType	Get the node type
getParent	Get the parent node
getRoot	Get the root node
getSibling	Get the sibling node
getSkippedText	Get the skipped text
getStatus	Get the status
getText	Get the text node value
getXmlMeta	Get the MetaNodeId
hasAttributes	Check if attribute(s) exist
hasChildNodes	Check if child node(s) exist
importNode	Import another document
importParent	Import another document
initialize	Initialize
insertNode	Insert the node

000004892117

Document serial number: 000004892117

©SCEI

Document serial number: 000004892117

Document serial number: 000004892117

Method	Description
isAvailable	Check the availability
isReadOnly	Check if the Document is read only
recurseDelete	Remove nodes recursively
removeAttribute	Remove attributes
removeAttributes	Remove attributes recursively
removeChild	Remove the child node
resetStatus	Reset the status
serialize	Output to XML text
setAttribute	Set the attribute
setAttributeList	Set the attributes
setAttrValue	Set the attribute value
setText	Set the text
setWritable	Set the DOM tree readable and writable mode
terminate	Terminate

000004892117

Document serial number: 000004892117

Document serial number: 000004892117

Document serial number: 000004892117

Constructors and Destructors

Document

Constructor

Definition

```
#include <xml/xml.h>
namespace sce {
        namespace Xml {
              namespace Dom {
                     class Document {
                           Document();

                     }
              }
        }
}
```

Return Values

None

Description

Constructor

000004892117

Document serial number: 000004892117

©SCEI

Document serial number: 000004892117

Document serial number: 000004892117

~Document

Destructor

Definition

```
#include <xml/xml.h>
namespace sce {
        namespace Xml {
              namespace Dom {
                     class Document {
                           ~Document();
                     }
              }
        }
}
```

Return Values

None

Description

Destructor

000004892117

Document serial number: 000004892117

©SCEI

Document serial number: 000004892117

Document serial number: 000004892117

Public Instance Methods

addElementChild

Create and add an element

Definition

```
#include <xml/xml.h>
namespace sce {
        namespace Xml {
              namespace Dom {
                     class Document {
                           NodeId addElementChild(
                                  NodeId parent,
                                  const String *name,
                                  const AttributeList *list = 0,
                                  const String *text = 0
                           );
                     }
              }
        }
}
```

Arguments

parent	(in) Parent node to which the given element is appended.
name	(in) The name of the element node to instantiate.
list	(in) Attribute list to be added to the element.
text	(in) Text to be added to the element as child node.

Return Values

The node added.

Description

The element with attribute list and text is added to the end of the list of children of the specified node.

Notes

The name is not checked for XML canonical naming convention.

000004892117

Document serial number: 000004892117

©SCEI

Document serial number: 000004892117

Document serial number: 000004892117

createElement

Create an element

Definition

```
#include <xml/xml.h>
namespace sce {
        namespace Xml {
              namespace Dom {
                     class Document {
                           NodeId createElement(
                                  const String *name,
                                  const AttributeList *list = 0,
                                  const String *text = 0
                           );
                     }
              }
        }
}
```

Arguments

name	(in) The name of the element node to instantiate.
list	(in) List of attributes which has to be added to the element
text	(in) Text of child to be added

Return Values

A new element node

Description

For example, to create a node such as:

```
<test>example</test>
```

the following call will create such an element node:

```
createElement(&String("test"), NULL, &String("example"));
```

Notes

The name is not checked for XML canonical naming convention

Memory Note:

If the created `Node` is inserted to R/W DOM tree using `insertNode()` then, the node will be recovered automatically when document is destroyed. If the node is not inserted as part of the tree, it has to be destroyed explicitly by using `recurseDelete()` API, failed to do so will result in memory leak.

000004892117

Document serial number: 000004892117

Document serial number: 000004892117

Document serial number: 000004892117

createTextNode

Create a Text node

Definition

```
#include <xml/xml.h>
namespace sce {
        namespace Xml {
              namespace Dom {
                     class Document {
                           NodeId createTextNode(
                                  const String *text
                           );
                     }
              }
        }
}
```

Arguments

text	(in) The text for the node.

Return Values

The new Text object

Description

This creates a text node given the specified string and length.

Notes

Memory Note:

If the created Node is inserted to R/W DOM tree using `insertNode` then, the node will be recovered automatically when document is destroyed. If the node is not inserted as part of the tree, it has to be destroyed explicitly by using `recurseDelete()` API, failed to do so will result in memory leak.

000004892117

Document serial number: 000004892117

Document serial number: 000004892117

Document serial number: 000004892117

getAttribute

Get the attribute value

Definition

```
#include <xml/xml.h>
namespace sce {
        namespace Xml {
              namespace Dom {
                     class Document {
                           String getAttribute(
                                  const NodeId node,
                                  const String *name
                           ) const;
                     }
              }
        }
}
```

Arguments

node	Target node
name	Name of attribute

Return Values

The attribute value

Description

This returns the attribute value for the given attribute.

000004892117

Document serial number: 000004892117

©SCEI

Document serial number: 000004892117

Document serial number: 000004892117

getAttributes

Get attributes

Definition

```
#include <xml/xml.h>
namespace sce {
        namespace Xml {
              namespace Dom {
                     class Document {
                            int getAttributes(
                                   NodeId node,
                                   NodeList *nodelist
                            ) const;
                     }
              }
        }
}
```

Arguments

node	(in) Specifying NodeId for which child node list is returned.
nodelist	(out) NodeList object to get the result containing all the matched nodes

Return Values

Value	Description
<0	Error code
SCE_OK	Success

Description

Input *node* must be valid node of type element, this check is not done internally, and it must be ensured before specifying the node.

nodelist is not required `NodeList::initialize()` before this API call.

nodelist requires to be deleted after an application finishes using it.

000004892117

Document serial number: 000004892117

©SCEI

Document serial number: 000004892117

Document serial number: 000004892117

getAttrName

Get the attribute name

Definition

```
#include <xml/xml.h>
namespace sce {
        namespace Xml {
              namespace Dom {
                     class Document {
                            String getAttrName(
                                   NodeId node
                            ) const;
                     }
              }
        }
}
```

Arguments

node	(in) Node for which attribute name is returned.

Return Values

String of the attribute name, or empty string if given node is not of attribute type.

Description

This returns the name of the given attribute node.

000004892117

Document serial number: 000004892117

©SCEI

Document serial number: 000004892117

Document serial number: 000004892117

getAttrValue

Get the attribute value

Definition

```
#include <xml/xml.h>
namespace sce {
        namespace Xml {
              namespace Dom {
                     class Document {
                           String getAttrValue(
                                  NodeId node
                           ) const;
                     }
              }
        }
}
```

Arguments

node	(in) Node for which attribute value is returned

Return Values

String of the attribute value, or empty string if given node is not of attribute type.

Description

This returns the attribute value for the given attribute node.

000004892117

Document serial number: 000004892117

Document serial number: 000004892117

Document serial number: 000004892117

getChildNodes

Get child nodes

Definition

```
#include <xml/xml.h>
namespace sce {
        namespace Xml {
              namespace Dom {
                     class Document {
                            int getChildNodes(
                                   NodeId node,
                                   NodeList *nodelist
                            ) const;
                     }
              }
        }
}
```

Arguments

node	(in) Specifying Node for which child node list is returned
nodelist	(out) NodeList object to get the result containing all the matched nodes

Return Values

Value	Description
<0	Error code
`SCE_OK`	Success

Description

Return the list including all the children if exists.

nodelist is not required to be initialized by `NodeList::initialize()` before this API call.

nodelist requires to be deleted after an application finishes using it.

000004892117

Document serial number: 000004892117

©SCEI

Document serial number: 000004892117

Document serial number: 000004892117

getDocRoot

Get the document root node

Definition

```
#include <xml/xml.h>
namespace sce {
        namespace Xml {
              namespace Dom {
                     class Document {
                            NodeId getDocRoot() const;
                     }
              }
        }
}
```

Return Values

`NodeId` to the document root, or `Dom::invalidNodeId` if document root node does not exist or the document not initialized.

Description

This returns the document root node (first node) of the document.

000004892117

Document serial number: 000004892117

©SCEI

Document serial number: 000004892117

Document serial number: 000004892117

getElementsByTagName

Get a NodeList by the name

Definition

```
#include <xml/xml.h>
namespace sce {
        namespace Xml {
              namespace Dom {
                     class Document {
                            int getElementsByTagName(
                                   NodeId node,
                                   const String *name,
                                   NodeList *nodelist
                            );
                     }
              }
        }
}
```

Arguments

node	(in) Root of the search tree
name	(in) Name of the elements to match on
nodelist	(out) NodeList object to get the result. The special value "*" matches all names.

Return Values

Value	Description
<0	Error code
`SCE_OK`	Success

Description

This returns a NodeList consisting of all the descendant elements with a given node name in the order in which they are encountered in a preorder traversal (depth-first search) of this element tree.

It is not necessary to call `NodeList::initialize()` for *nodelist* before calling this API. An application must delete the used *nodelist*.

000004892117

Document serial number: 000004892117

Document serial number: 000004892117

Document serial number: 000004892117

getEntity

Get the entity text

Definition

```
#include <xml/xml.h>
namespace sce {
        namespace Xml {
              namespace Dom {
                     class Document {
                            String getEntity(
                                   NodeId node
                            ) const;
                     }
              }
        }
}
```

Arguments

node	(in) Node for which entity value is returned

Return Values

String of the entity text, or empty string if given node is not of user defined entity or character entity reference.

Description

This returns the value of the given entity node, for user defined entity or character entity reference.

000004892117

Document serial number: 000004892117

©SCEI

Document serial number: 000004892117

Document serial number: 000004892117

getEntityType

Get the entity type

Definition

```
#include <xml/xml.h>
namespace sce {
        namespace Xml {
              namespace Dom {
                     class Document {
                           EntityType getEntityType(
                                  NodeId node
                           ) const;
                     }
              }
        }
}
```

Arguments

node	(in) Node for which entity type is returned.

Return Values

Entity type of the node

Description

This returns the entity type of the given entity node.

000004892117

Document serial number: 000004892117

©SCEI

Document serial number: 000004892117

Document serial number: 000004892117

getFirstAttr

Get the first attribute

Definition

```
#include <xml/xml.h>
namespace sce {
        namespace Xml {
              namespace Dom {
                     class Document {
                           NodeId getFirstAttr(
                                  NodeId node
                           ) const;
                     }
              }
        }
}
```

Arguments

node	(in) Node for which the first attribute is returned

Return Values

`NodeId` which represents the first valid attribute of the given element node or `Dom::invalidNodeId` if given node does not have any attribute.

Description

This returns the first attribute for a given element node.

000004892117

Document serial number: 000004892117

©SCEI

Document serial number: 000004892117

Document serial number: 000004892117

getFirstChild

Get the child node

Definition

```
#include <xml/xml.h>
namespace sce {
        namespace Xml {
              namespace Dom {
                     class Document {
                           NodeId getFirstChild(
                                  NodeId node
                           ) const;
                     }
              }
        }
}
```

Arguments

node	(in) Node for which child node is returned

Return Values

Valid `NodeId` of the first child node, or `Dom::invalidNodeId` if no child node or the node is leaf node.

Description

This returns the first child for the given node within the current document

000004892117

Document serial number: 000004892117

©SCEI

Document serial number: 000004892117

Document serial number: 000004892117

getLastChild

Get the last child node

Definition

```
#include <xml/xml.h>
namespace sce {
        namespace Xml {
              namespace Dom {
                     class Document {
                           NodeId getLastChild(
                                  NodeId node
                           ) const;
                     }
              }
        }
}
```

Arguments

node	(in) Node for which child node is returned

Return Values

The last child node of the given node

Description

This gets the last child node of the given node.

000004892117

Document serial number: 000004892117

©SCEI

Document serial number: 000004892117

Document serial number: 000004892117

getNextAttr

Get the next attribute

Definition

```
#include <xml/xml.h>
namespace sce {
        namespace Xml {
              namespace Dom {
                     class Document {
                           NodeId getNextAttr(
                                  NodeId node
                           ) const;
                     }
              }
        }
}
```

Arguments

node	(in) Node for which next attribute is returned

Return Values

`NodeId` which represents the next attribute of the given element node, or `Dom::invalidNodeId` if given node does not have any more attributes.

Description

This returns the next attribute for a given element node.

000004892117

Document serial number: 000004892117

©SCEI

Document serial number: 000004892117

Document serial number: 000004892117

getNodeName

Get the node name

Definition

```
#include <xml/xml.h>
namespace sce {
        namespace Xml {
              namespace Dom {
                     class Document {
                            String getNodeName(
                                   NodeId node
                            ) const;
                     }
              }
        }
}
```

Arguments

node	(in) Node for which name is returned

Return Values

String of the node name, or empty string if given node does not support node name.

Description

This returns the name of the given element node.

000004892117

Document serial number: 000004892117

©SCEI

Document serial number: 000004892117

Document serial number: 000004892117

getNodeType

Get the node type

Definition

```
#include <xml/xml.h>
namespace sce {
        namespace Xml {
              namespace Dom {
                     class Document {
                           NodeType getNodeType(
                                  NodeId node
                           ) const;
                     }
              }
        }
}
```

Arguments

node	Target node

Return Values

Type of the node

Description

This returns the type of the given node, as defined in `Dom::NodeType`.

000004892117

Document serial number: 000004892117

©SCEI

Document serial number: 000004892117

Document serial number: 000004892117

getParent

Get the parent node

Definition

```
#include <xml/xml.h>
namespace sce {
        namespace Xml {
              namespace Dom {
                     class Document {
                            NodeId getParent(
                                   NodeId node
                            ) const;
                     }
              }
        }
}
```

Arguments

node (in) Node for which parent is returned

Return Values

Valid `NodeId` of the parent node, or `Dom::invalidNodeId` if given node does not belong to the current `Document`.

Description

This returns the parent node of a given node.

Notes

Performance note:

`getParent` searches the document from root for parent, and may reduce performance if used many time. Instead it is recommended to keep parent `NodeId`.

000004892117

Document serial number: 000004892117

Document serial number: 000004892117

Document serial number: 000004892117

getRoot

Get the root node

Definition

```
#include <xml/xml.h>
namespace sce {
        namespace Xml {
              namespace Dom {
                     class Document {
                            NodeId getRoot() const;
                     }
              }
        }
}
```

Return Values

Valid `NodeId` of the root node, or `Dom::invalidNodeId` if root node does not exist or the document not initialized.

Description

This returns the root node (first element node) of the document.

000004892117

Document serial number: 000004892117

©SCEI

Document serial number: 000004892117

Document serial number: 000004892117

getSibling

Get the sibling node

Definition

```
#include <xml/xml.h>
namespace sce {
        namespace Xml {
              namespace Dom {
                     class Document {
                           NodeId getSibling(
                                  NodeId node
                           ) const;
                     }
              }
        }
}
```

Arguments

node	(in) Node for which sibling node is returned

Return Values

Valid `NodeId` of the sibling node which exists, or `Dom::invalidNodeId` if no sibling node.

Description

This returns the sibling for the given node within the current document.

000004892117

Document serial number: 000004892117

©SCEI

Document serial number: 000004892117

Document serial number: 000004892117

getSkippedText

Get the skipped text

Definition

```
#include <xml/xml.h>
namespace sce {
        namespace Xml {
              namespace Dom {
                     class Document {
                           String getSkippedText(
                                  NodeId node
                           ) const;
                     }
              }
        }
}
```

Arguments

node	(in) Node for which value is returned

Return Values

String of the skipped text, or empty string if given node does not support skipped text

Description

This returns the value of the given skipped text node.

000004892117

Document serial number: 000004892117

©SCEI

Document serial number: 000004892117

Document serial number: 000004892117

getStatus

Get the status

Definition

```
#include <xml/xml.h>
namespace sce {
        namespace Xml {
              namespace Dom {
                     class Document {
                           int getStatus() const;
                     }
              }
        }
}
```

Return Values

Status. `resultSuccess` or an error.

Description

This returns status of previous operation.

000004892117

Document serial number: 000004892117

©SCEI

Document serial number: 000004892117

Document serial number: 000004892117

getText

Get the text node value

Definition

```
#include <xml/xml.h>
namespace sce {
        namespace Xml {
              namespace Dom {
                     class Document {
                           String getText(
                                  NodeId node
                           ) const;
                     }
              }
        }
}
```

Arguments

node	(in) Node for which the value is returned

Return Values

String of the text, or empty string if given node does not support text.

Description

This returns the value of the given text node.

000004892117

Document serial number: 000004892117

©SCEI

Document serial number: 000004892117

Document serial number: 000004892117

getXmlMeta

Get the MetaNodeId

Definition

```
#include <xml/xml.h>
namespace sce {
        namespace Xml {
              namespace Dom {
                     class Document {
                            MetaNodeId getXmlMeta() const;
                     }
              }
        }
}
```

Return Values

Valid MetaNodeId of the document root node

Description

This returns the meta node for the document root.

000004892117

Document serial number: 000004892117

Document serial number: 000004892117

Document serial number: 000004892117

hasAttributes

Check if attribute node(s) exist

Definition

```
#include <xml/xml.h>
namespace sce {
        namespace Xml {
              namespace Dom {
                     class Document {
                           bool hasAttributes(
                                  NodeId node
                           ) const;
                     }
              }
        }
}
```

Arguments

node	Target node

Return Values

Value	Description
`true`	Attribute(s) exist
`false`	No attribute

Description

This returns `true` if the given node has attribute node(s)

000004892117

Document serial number: 000004892117

©SCEI

Document serial number: 000004892117

Document serial number: 000004892117

hasChildNodes

Check if child node(s) exist

Definition

```
#include <xml/xml.h>
namespace sce {
        namespace Xml {
              namespace Dom {
                     class Document {
                            bool hasChildNodes(
                                   NodeId node
                            ) const;
                     }
              }
        }
}
```

Arguments

node	Target node

Return Values

Value	Description
`true`	Child node(s) exist
`false`	No child node

Description

This returns `true` if the given node has child node(s)

000004892117

Document serial number: 000004892117

Document serial number: 000004892117

Document serial number: 000004892117

importNode

Import another document

Definition

```
#include <xml/xml.h>
namespace sce {
        namespace Xml {
              namespace Dom {
                     class Document {
                           NodeId importNode(
                                  NodeId nparent,
                                  NodeId ref,
                                  const Document *doc,
                                  NodeId node
                           );
                     }
              }
        }
}
```

Arguments

nparent	(in) Starting node which has to be imported
ref	(in) The reference node. The new node will be inserted after this node
doc	(in) Source document for importing *nparent* node
node	(in) Connection destination node of a node to be imported

Description

Import the given node within the given document to the given node of this document. The import operation copies all the data and the nodes to this document.

000004892117

Document serial number: 000004892117

Document serial number: 000004892117

Document serial number: 000004892117

importParent

Import another document

Definition

```
#include <xml/xml.h>
namespace sce {
        namespace Xml {
                namespace Dom {
                        class Document {
                                int importParent(
                                        const Document *parentDoc,
                                        NodeId parent
                                );
                        }
                }
        }
}
```

Arguments

parentDoc	(in) Parent document
parent	(in) Parent node which will become parent.

Return Values

Value	Description
<0	Error code
SCE_OK	Success

Description

This imports the given document with the node specified by *parentDoc* and *parent* as parent node. The whole tree of its own object before being called will be the child node of the given node. The last node of the given parent tree will be the parent node of the child tree.

Effective execution will be possible compared to `importNode()` when the child tree is larger than the parent tree.

Example:

Parent tree:

```
<test><firstNode><impNode/></firstNode></test>
```

child tree:

```
<bigChild>Big Big Tree</bigChild>
```

If the parent tree is imported as parent of child tree, the `<impNode>` of parent tree becomes the parent node of the tree. The imported parent node must not have attributes.

000004892117

Document serial number: 000004892117

Document serial number: 000004892117

Document serial number: 000004892117

initialize

Initialize

Definition

```
#include <xml/xml.h>
namespace sce {
        namespace Xml {
              namespace Dom {
                     class Document {
                            int initialize(
                                   const Initializer *init
                            );
                     }
              }
        }
}
```

Arguments

init	(in) Pointer to initialize object

Return Values

Value	Description
<0	Error code
SCE_OK	Success

Description

This API must be called before another interface is used.

000004892117

Document serial number: 000004892117

Document serial number: 000004892117

Document serial number: 000004892117

insertNode

Insert the node

Definition

```
#include <xml/xml.h>
namespace sce {
        namespace Xml {
              namespace Dom {
                     class Document {
                           NodeId insertNode(
                                  NodeId parent,
                                  NodeId ref,
                                  NodeId child
                           );
                     }
              }
        }
}
```

Arguments

`parent`	(in) Parent of the reference node or to which given child node is inserted
`ref`	(in) The reference node, i.e., the node after which the new node must be inserted
`child`	(in) The node to insert.

Return Values

The node being inserted

Description

Insert the node *`child`* after the existing child node *`ref`*. If *`ref`* is invalid (`Dom::invalidNodeId`), insert *`child`* as the first node of the element. If the *`child`* is already in the tree, it is first removed

000004892117

Document serial number: 000004892117

Document serial number: 000004892117

Document serial number: 000004892117

isAvailable

Check the availability

Definition

```
#include <xml/xml.h>
namespace sce {
        namespace Xml {
              namespace Dom {
                     class Document {
                            bool isAvailable() const;
                     }
              }
        }
}
```

Return Values

Value	Description
`true`	Initialized and ready to use
`false`	Not initialized

Description

This returns `true` if this object is available.

000004892117

Document serial number: 000004892117

Document serial number: 000004892117

Document serial number: 000004892117

isReadOnly

Check if the Document is read only.

Definition

```
#include <xml/xml.h>
namespace sce {
        namespace Xml {
              namespace Dom {
                     class Document {
                            bool isReadOnly() const;
                     }
              }
        }
}
```

Return Values

Value	Description
`true`	Read only DOM
`false`	Read/Write DOM

Description

This returns if Document is read only.

000004892117

Document serial number: 000004892117

©SCEI

Document serial number: 000004892117

Document serial number: 000004892117

recurseDelete

Remove nodes recursively

Definition

```
#include <xml/xml.h>
namespace sce {
        namespace Xml {
              namespace Dom {
                     class Document {
                           int recurseDelete(
                                  NodeId node
                           );
                     }
              }
        }
}
```

Arguments

node	Node to start deleting

Return Values

Value	Description
<0	Error code
SCE_OK	Success

Description

This deletes the given node from the memory. If the given node is a tree, it deletes recursively.

000004892117

Document serial number: 000004892117

©SCEI

Document serial number: 000004892117

Document serial number: 000004892117

removeAttribute

Remove attributes

Definition

```
#include <xml/xml.h>
namespace sce {
        namespace Xml {
              namespace Dom {
                     class Document {
                           int removeAttribute(
                                 NodeId node,
                                 const String *name
                           );
                     }
              }
        }
}
```

Arguments

node	(in) Element node, attribute of this node is removed.
name	(in) Attribute name to be removed.

Return Values

Value	Description
<0	Error code
`SCE_OK`	Success

Description

This removes the attributes from the given element node. This API removes the attribute whose name is given from element node of an R/W DOM tree.

000004892117

Document serial number: 000004892117

©SCEI

Document serial number: 000004892117

Document serial number: 000004892117

removeAttributes

Remove attributes recursively

Definition

```
#include <xml/xml.h>
namespace sce {
        namespace Xml {
              namespace Dom {
                     class Document {
                            int removeAttributes(
                                   NodeId node
                            );
                     }
              }
        }
}
```

Arguments

node	(in) Element node

Return Values

Value	Description
<0	Error code
`SCE_OK`	Success

Description

This removes all the attributes from the given element node. This API removes the attributes of all the children existing in the given element node or below.

000004892117

Document serial number: 000004892117

©SCEI

Document serial number: 000004892117

Document serial number: 000004892117

removeChild

Remove the child node

Definition

```
#include <xml/xml.h>
namespace sce {
        namespace Xml {
             namespace Dom {
                    class Document {
                          NodeId removeChild(
                                 NodeId child,
                                 NodeId parent
                          );
                    }
             }
        }
}
```

Arguments

child	(in) The node to be removed
parent	(in) The parent node of node being removed

Return Values

The node removed

Description

This removes the child node indicated by *child* and returns it.

Notes

The NodeId is returned but it is unavailable any longer.

000004892117

Document serial number: 000004892117

©SCEI

Document serial number: 000004892117

Document serial number: 000004892117

resetStatus

Reset the status

Definition

```
#include <xml/xml.h>
namespace sce {
        namespace Xml {
              namespace Dom {
                     class Document {
                           void resetStatus();
                     }
              }
        }
}
```

Return Values

None

Description

This resets the `Document` status to `resultSuccess`. If the previously invoked API results in error such as:

- `resultDomMaxNumOfAttrError`
- `resultDomMaxUniqueElementError`

It is safe to reset, so that the `Document` can be reused.

In case of memory error, or other critical errors, in such case resetting may cause unknown behavior. In such case it is better to re-create the document, in order to remove the error.

000004892117

Document serial number: 000004892117

©SCEI

Document serial number: 000004892117

Document serial number: 000004892117

serialize

Output to XML text

Definition

```
#include <xml/xml.h>
namespace sce {
        namespace Xml {
              namespace Dom {
                     class Document {
                            int serialize(
                                   const SerializeParameter *param,
                                   XmlText *outputString
                            );
                     }
              }
        }
}
```

Arguments

param	(in) Parameter to control serialization
outputString	(out) Output string

Return Values

Value	Description
<0	Error code
`SCE_OK`	Success

Description

This serializes the `Document` to XML document.

`Document` object holds the generated XML document.

000004892117

Document serial number: 000004892117

©SCEI

Document serial number: 000004892117

Document serial number: 000004892117

setAttribute

Set the attribute

Definition

```
#include <xml/xml.h>
namespace sce {
        namespace Xml {
              namespace Dom {
                     class Document {
                           int setAttribute(
                                 NodeId node,
                                 const String *name,
                                 const String *value
                           );
                     }
              }
        }
}
```

Arguments

node	(in) Element node, attribute of this node is changed
name	(in) Name of the attribute whose value to be changed.
value	(in) Value which has to be set for the given attribute

Return Values

Value	Description
<0	Error code
`SCE_OK`	Success

Description

This adds the given attribute name, of the given element node. If the given attribute does not exist, a new attribute is added to the given element node, otherwise, the value is updated.

Notes

This API replaces the old attribute value.

000004892117

Document serial number: 000004892117

Document serial number: 000004892117

Document serial number: 000004892117

setAttributeList

Set the attributes

Definition

```
#include <xml/xml.h>
namespace sce {
        namespace Xml {
              namespace Dom {
                     class Document {
                           int setAttributeList(
                                 NodeId node,
                                 const AttributeList *list
                           );
                     }
              }
        }
}
```

Arguments

node	(in) The element node to which attribute list to be added
list	(in) Attribute list to be added to the element

Return Values

Value	Description
<0	Error code
`SCE_OK`	Success

Description

This adds attribute list to the node. This API removes all the attributes from the given element if they exist.

000004892117

Document serial number: 000004892117

©SCEI

Document serial number: 000004892117

Document serial number: 000004892117

setAttrValue

Set the attribute value

Definition

```
#include <xml/xml.h>
namespace sce {
        namespace Xml {
              namespace Dom {
                     class Document {
                           int setAttrValue(
                                 NodeId node,
                                 const String *name,
                                 const String *value
                           );
                     }
              }
        }
}
```

Arguments

node	(in) Element node, attribute of this node is changed.
name	(in) Name of the attribute whose value to be changed
value	(in) Value which has to be set for the given attribute

Return Values

Value	Description
<0	Error code
`SCE_OK`	Success

Description

Set value for the given attribute name of the given element node. If the given attribute does not exist, a new attribute is added to the given element node.

Notes

This API replaces the old attribute value.

000004892117

Document serial number: 000004892117

©SCEI

Document serial number: 000004892117

Document serial number: 000004892117

setText

Set the text

Definition

```
#include <xml/xml.h>
namespace sce {
        namespace Xml {
              namespace Dom {
                     class Document {
                           int setText(
                                 NodeId node,
                                 const String *text
                           );
                     }
              }
        }
}
```

Arguments

node	(in) Text node whose value has to be changed
text	(in) Value which has to be set.

Return Values

Value	Description
<0	Error code
`SCE_OK`	Success

Description

This sets text value for text node.

Notes

This API replaces the old attribute value.

000004892117

Document serial number: 000004892117

©SCEI

Document serial number: 000004892117

Document serial number: 000004892117

setWritable

Set the DOM tree to readable and writable mode

Definition

```
#include <xml/xml.h>
namespace sce {
        namespace Xml {
              namespace Dom {
                     class Document {
                           NodeId setWritable();
                     }
              }
        }
}
```

Return Values

`NodeId` for `DocumentRoot` (refer to `getDocRoot()`).

`Dom::invalidNodeId` if failed.

Description

This sets the DOM tree to read/write mode.

000004892117

Document serial number: 000004892117

©SCEI

Document serial number: 000004892117

Document serial number: 000004892117

terminate

Terminate

Definition

```
#include <xml/xml.h>
namespace sce {
        namespace Xml {
              namespace Dom {
                     class Document {
                            int terminate();
                     }
              }
        }
}
```

Return Values

Value	Description
SCE_OK	Success

Description

Terminates the Document and recovers the memory occupied.

000004892117

Document serial number: 000004892117

©SCEI

Document serial number: 000004892117

Document serial number: 000004892117

sce::Xml::Dom::DocumentBuilder

000004892117

Document serial number: 000004892117

©SCEI

Document serial number: 000004892117

Document serial number: 000004892117

Summary

sce::Xml::Dom::DocumentBuilder

DOM creation interface class

Definition

```
#include <xml/xml.h>
class DocumentBuilder {};
```

Description

DOM creation interface class

Method List

Method	Description
DocumentBuilder	Constructor
~DocumentBuilder	Destructor
getDocument	Get the created Document
initialize	Initialize
parse	Parse an XML document and create the Document tree structure
setResolveEntity	Set entity resolving behavior
setSkipIgnorableText	Set text skip behavior
setSkipIgnorableWhiteSpace	Set white space skip behavior
terminate	Terminate

000004892117

Document serial number: 000004892117

©SCEI

Document serial number: 000004892117

Document serial number: 000004892117

Constructors and Destructors

DocumentBuilder

Constructor

Definition

```
#include <xml/xml.h>
namespace sce {
        namespace Xml {
              namespace Dom {
                     class DocumentBuilder {
                           DocumentBuilder();
                     }
              }
        }
}
```

Return Values

None

Description

Constructor

000004892117

Document serial number: 000004892117

Document serial number: 000004892117

Document serial number: 000004892117

~DocumentBuilder

Destructor

Definition

```
#include <xml/xml.h>
namespace sce {
        namespace Xml {
              namespace Dom {
                     class DocumentBuilder {
                           ~DocumentBuilder();
                     }
              }
        }
}
```

Return Values

None

Description

Destructor

000004892117

Document serial number: 000004892117

©SCEI

Document serial number: 000004892117

Document serial number: 000004892117

Public Instance Methods

getDocument

Get the created Document

Definition

```
#include <xml/xml.h>
namespace sce {
        namespace Xml {
              namespace Dom {
                     class DocumentBuilder {
                            Document getDocument();
                     }
              }
        }
}
```

Return Values

The Document, created by parsing the XML document in the last parse() call

Description

This gets the Document created by the last parse() call. This object has a reference to the Document held in this DocumentBuilder instance. The Document object is kept by the DocumentBuilder until the next parse() or the DocumentBuilder deleted. DocumentBuilder should be remained while using the Document.

000004892117

Document serial number: 000004892117

©SCEI

Document serial number: 000004892117

Document serial number: 000004892117

initialize

Initialize

Definition

```
#include <xml/xml.h>
namespace sce {
        namespace Xml {
              namespace Dom {
                     class DocumentBuilder {
                            int initialize(
                                   const Initializer *init
                            );
                     }
              }
        }
}
```

Arguments

init	(in) Pointer to Initializer object

Return Values

Value	Description
<0	Error code
SCE_OK	Success

Description

This API must be called before another interface is used.

000004892117

Document serial number: 000004892117

©SCEI

Document serial number: 000004892117

Document serial number: 000004892117

parse

Parse an XML document and create the Document tree structure

Definition

```
#include <xml/xml.h>
namespace sce {
        namespace Xml {
              namespace Dom {
                     class DocumentBuilder {
                            int parse(
                                   const XmlText *text,
                                   bool isFinal = true
                            );
                     }
              }
        }
}
```

Arguments

text	(in) The chunk of XML data to be parsed
isFinal	(in) Whether the input chunk is the final chunk to be parsed in the current XML stream. Default value is `true`.

Return Values

Value	Description
<0	Error code
`SCE_OK`	Success in case *isFinal* is `true`
`resultXmlParseInProgress`	In case of chunk parsing, the current parse with the given chunk is successful

Description

Parse the XML document from the buffer and create `Document`.

If the *isFinal* is `true`, then the parser will be reset before returning from the function call.

000004892117

Document serial number: 000004892117

Document serial number: 000004892117

Document serial number: 000004892117

setResolveEntity

Set entity resolving behavior

Definition

```
#include <xml/xml.h>
namespace sce {
        namespace Xml {
              namespace Dom {
                     class DocumentBuilder {
                            void setResolveEntity(
                                   bool isResolved
                            );
                     }
              }
        }
}
```

Arguments

isResolved	(in) `true` in case the entities have to be resolved while parsing the XML

Return Values

None

Description

Configure if the parser resolves the built in entities and character references.
In case of user defined entities, it will be treated like entity without being resolved.
Default is `false`.

000004892117

Document serial number: 000004892117

©SCEI

Document serial number: 000004892117

Document serial number: 000004892117

setSkipIgnorableText

Set text skip behavior

Definition

```
#include <xml/xml.h>
namespace sce {
        namespace Xml {
              namespace Dom {
                     class DocumentBuilder {
                            void **setSkipIgnorableText**(
                                   bool *isSkipped*
                            );
                     }
              }
        }
}
```

Arguments

isSkipped	(in) `true` if it has to be skipped. `false` otherwise.

Return Values

None

Description

This ignores the skip text (CDATA, Programming Instruction (PI), XML Declaration, Comment and Document Type Definition (DTD)) not to be included in the DOM structure. When the parse is in progress in case of chunked parsing, it cannot be changed.

Default is `false`.

000004892117

Document serial number: 000004892117

©SCEI

Document serial number: 000004892117

Document serial number: 000004892117

setSkipIgnorableWhiteSpace

Set white space skip behavior

Definition

```
#include <xml/xml.h>
namespace sce {
        namespace Xml {
              namespace Dom {
                     class DocumentBuilder {
                            void setSkipIgnorableWhiteSpace(
                                   bool isSkipped
                            );
                     }
              }
        }
}
```

Arguments

isSkipped	(in) Set `true` in case the ignorable white spaces have to be skipped.

Return Values

None

Description

This skips the ignorable white spaces, if the skip flag is set to `true`.
Default is `true`.

000004892117

Document serial number: 000004892117

©SCEI

Document serial number: 000004892117

Document serial number: 000004892117

terminate

Terminate

Definition

```
#include <xml/xml.h>
namespace sce {
        namespace Xml {
              namespace Dom {
                     class DocumentBuilder {
                           int terminate();
                     }
              }
        }
}
```

Return Values

Value	Description
SCE_OK	Success

Description

This terminates and destroys any local objects including the generated Document.

000004892117

Document serial number: 000004892117

©SCEI

Document serial number: 000004892117

Document serial number: 000004892117

sce::Xml::Dom::Node

000004892117

Document serial number: 000004892117

©SCEI

Document serial number: 000004892117

Document serial number: 000004892117

Summary

sce::Xml::Dom::Node

Interface class of Node

Definition

```
#include <xml/xml_node.h>
class Node {};
```

Description

A Class represents Node with reference to a Document object.

Note: Class Node is implemented as the wrapper to Document in this library.

Please refer to Document class.

Method List

Method	Description
appendChild	Insert a node
getAttributes	Get the attribute nodes
getChildNodes	Get the child nodes
getFirstChild	Get the first child node
getLastChild	Get the last child node
getNextSibling	Get the sibling node
getNodeName	Get the node name
getNodeType	Get the node type
getNodeValue	Get the node value
getOwnerDocument	Get the owner document
getParentNode	Get the parent node
hasAttributes	Check if attribute(s) exist
hasChildNodes	Check if child node(s) exist
insertBefore	Insert a node
isAvailable	Check the availability
Node	Constructor
Node	Copy constructor
~Node	Destructor
operator=	Assignment operator
removeChild	Remove the child node

000004892117

Document serial number: 000004892117

Document serial number: 000004892117

Document serial number: 000004892117

Constructors and Destructors

Node

Constructor

Definition

```
#include <xml_node.h>
namespace sce {
        namespace Xml {
              namespace Dom {
                     class Node {
                            Node(
                                   NodeId id
                            );
                     }
              }
        }
}
```

Arguments

id	`Node` id to set

Return Values

None

Description

Constructs the object and initializes it with the given node id.

000004892117

Document serial number: 000004892117

©SCEI

Document serial number: 000004892117

Document serial number: 000004892117

Node

Copy constructor

Definition

```
#include <xml_node.h>
namespace sce {
        namespace Xml {
              namespace Dom {
                     class Node {
                           Node(
                                  const Node &src
                           );
                     }
              }
        }
}
```

Arguments

src	`Node` to be copied

Return Values

None

Description

Copy Constructor.

000004892117

Document serial number: 000004892117

©SCEI

Document serial number: 000004892117

Document serial number: 000004892117

~Node

Destructor

Definition

```
#include <xml_node.h>
namespace sce {
        namespace Xml {
              namespace Dom {
                     class Node {
                           ~Node();
                     }
              }
        }
}
```

Return Values

None

Description

Destructor.

000004892117

Document serial number: 000004892117

©SCEI

Document serial number: 000004892117

Document serial number: 000004892117

Operator Methods

operator=

Assignment operator

Definition

```
#include <xml/xml_node.h>
namespace sce {
        namespace Xml {
              namespace Dom {
                     class Node {
                            Node &operator=(
                                  const Node &src
                            );
                     }
              }
        }
}
```

Arguments

src (in) Node to be copied

Return Values

Reference to the copied Node.

Description

Returns the copy of the given node.

000004892117

Document serial number: 000004892117

©SCEI

Document serial number: 000004892117

Document serial number: 000004892117

Public Instance Methods

appendChild

Insert a child node

Definition

```
#include <xml/xml_node.h>
namespace sce {
        namespace Xml {
              namespace Dom {
                     class Node {
                            NodeId appendChild(
                                   NodeId newChild
                            );
                     }
              }
        }
}
```

Arguments

newChild	(in) the node to be inserted

Return Values

NodeId of the inserted node.

Description

Inserts a new node to the current owner document.

000004892117

Document serial number: 000004892117

©SCEI

Document serial number: 000004892117

Document serial number: 000004892117

getAttributes

Get the attribute nodes

Definition

```
#include <xml/xml_node.h>
namespace sce {
        namespace Xml {
              namespace Dom {
                     class Node {
                           NodeList *getAttributes() const;
                     }
              }
        }
}
```

Return Values

Pointer to the list of the attribute nodes.

Description

Returns list of the attribute nodes.

See Also: **`Document`**`::getAttributes()`

000004892117

Document serial number: 000004892117

©SCEI

Document serial number: 000004892117

Document serial number: 000004892117

getChildNodes

Get the child nodes

Definition

```
#include <xml/xml_node.h>
namespace sce {
        namespace Xml {
              namespace Dom {
                     class Node {
                           NodeList *getChildNodes() const;
                     }
              }
        }
}
```

Return Values

Pointer to the list of the child nodes.

Description

Returns list of the child nodes.

See also: **Document**::getChildNodes()

000004892117

Document serial number: 000004892117

Document serial number: 000004892117

Document serial number: 000004892117

getFirstChild

Get the first child node

Definition

```
#include <xml/xml_node.h>
namespace sce {
        namespace Xml {
              namespace Dom {
                     class Node {
                           NodeId getFirstChild() const;
                     }
              }
        }
}
```

Return Values

NodeId of the first child node.

Description

Returns the first child node.

See also: **Document**::getFirstChild()

000004892117

Document serial number: 000004892117

©SCEI

Document serial number: 000004892117

Document serial number: 000004892117

getLastChild

Get the last child node

Definition

```
#include <xml/xml_node.h>
namespace sce {
        namespace Xml {
              namespace Dom {
                     class Node {
                           NodeId getLastChild() const;
                     }
              }
        }
}
```

Return Values

NodeId of the last child node.

Description

Returns the last child node.

See also: **Document**::getLastChild()

000004892117

Document serial number: 000004892117

©SCEI

Document serial number: 000004892117

Document serial number: 000004892117

getNextSibling

Get the sibling node

Definition

```
#include <xml/xml_node.h>
namespace sce {
        namespace Xml {
              namespace Dom {
                     class Node {
                           NodeId getNextSibling() const;
                     }
              }
        }
}
```

Return Values

NodeId of the sibling node.

Description

Returns the sibling node.

000004892117

Document serial number: 000004892117

©SCEI

Document serial number: 000004892117

Document serial number: 000004892117

getNodeName

Get the node name

Definition

```
#include <xml/xml_node.h>
namespace sce {
        namespace Xml {
              namespace Dom {
                     class Node {
                           String getNodeName() const;
                     }
              }
        }
}
```

Return Values

String of the node name, or empty string if given node does not support node name.

Description

Returns the name of this node.

See also: **Document**::getNodeName()

000004892117

Document serial number: 000004892117

©SCEI

Document serial number: 000004892117

Document serial number: 000004892117

getNodeType

Get the node type

Definition

```
#include <xml/xml_node.h>
namespace sce {
        namespace Xml {
              namespace Dom {
                     class Node {
                           NodeType getNodeType() const;
                     }
              }
        }
}
```

Return Values

Type of the node.

Description

Returns the type of this node.

See also: **Document**::getNodeType()

000004892117

Document serial number: 000004892117

©SCEI

Document serial number: 000004892117

Document serial number: 000004892117

getNodeValue

Get the node value

Definition

```
#include <xml/xml_node.h>
namespace sce {
        namespace Xml {
              namespace Dom {
                     class Node {
                           String getNodeValue() const;
                     }
              }
        }
}
```

Return Values

String of the node value, or empty string if given node does not support node value.

Description

Returns the value of this node.

000004892117

Document serial number: 000004892117

©SCEI

Document serial number: 000004892117

Document serial number: 000004892117

getOwnerDocument

Get the owner document

Definition

```
#include <xml/xml_node.h>
namespace sce {
        namespace Xml {
              namespace Dom {
                     class Node {
                           Document *getOwnerDocument() const;
                     }
              }
        }
}
```

Return Values

Pointer to the owner document.

Description

Returns the owner document.

000004892117

Document serial number: 000004892117

©SCEI

Document serial number: 000004892117

Document serial number: 000004892117

getParentNode

Get the parent node

Definition

```
#include <xml/xml_node.h>
namespace sce {
        namespace Xml {
              namespace Dom {
                     class Node {
                           NodeId getParentNode() const;
                     }
              }
        }
}
```

Return Values

NodeId of the parent node.

Description

Returns the parent node.

000004892117

Document serial number: 000004892117

©SCEI

Document serial number: 000004892117

Document serial number: 000004892117

hasAttributes

Check if any attribute(s) exist

Definition

```
#include <xml/xml_node.h>
namespace sce {
        namespace Xml {
              namespace Dom {
                     class Node {
                            bool hasAttributes() const;
                     }
              }
        }
}
```

Return Values

Value	Description
`true`	The node has an attribute at least one.
`false`	The node has no attribute.

Description

Returns `true` if the node has attribute(s)

See also: **`Document`**`::hasAttributes()`

000004892117

Document serial number: 000004892117

©SCEI

Document serial number: 000004892117

Document serial number: 000004892117

hasChildNodes

Check if any child node(s) exist

Definition

```
#include <xml/xml_node.h>
namespace sce {
        namespace Xml {
              namespace Dom {
                     class Node {
                           bool hasChildNodes();
                     }
              }
        }
}
```

Return Values

Value	Description
`true`	The node has a child at least one.
`false`	The node has no child.

Description

Returns `true` if the node has child node(s)

See also: **`Document`**`::hasChildNodes()`

000004892117

Document serial number: 000004892117

©SCEI

Document serial number: 000004892117

Document serial number: 000004892117

insertBefore

Insert a node

Definition

```
#include <xml/xml_node.h>
namespace sce {
        namespace Xml {
              namespace Dom {
                     class Node {
                            NodeId insertBefore(
                                   NodeId newChild,
                                   NodeId refChild
                            );
                     }
              }
        }
}
```

Arguments

newChild	(in) the node to be inserted
refChild	(in) the node of the position of the insertion

Return Values

NodeId of the inserted node.

Description

Inserts a node.

000004892117

Document serial number: 000004892117

©SCEI

Document serial number: 000004892117

Document serial number: 000004892117

isAvailable

Check the availability

Definition

```
#include <xml/xml_node.h>
namespace sce {
        namespace Xml {
              namespace Dom {
                     class Node {
                           bool isAvailable() const;
                     }
              }
        }
}
```

Return Values

Value	Description
`true`	Initialized and ready to use
`false`	Not initialized

Description

This returns `true` if this object is available.

000004892117

Document serial number: 000004892117

©SCEI

Document serial number: 000004892117

Document serial number: 000004892117

removeChild

Remove the child node

Definition

```
#include <xml/xml_node.h>
namespace sce {
        namespace Xml {
              namespace Dom {
                     class Node {
                           NodeId removeChild(
                                  NodeId oldChild
                           );
                     }
              }
        }
}
```

Arguments

oldChild	(in) the node to be removed

Return Values

NodeId of the removed node.

Description

Remove the child node.

See also: **Document**::removeChild()

Notes

The NodeId is returned but it is unavailable any longer.

000004892117

Document serial number: 000004892117

©SCEI

Document serial number: 000004892117

Document serial number: 000004892117

sce::Xml::Dom::NodeList

000004892117

Document serial number: 000004892117

©SCEI

Document serial number: 000004892117

Document serial number: 000004892117

Summary

sce::Xml::Dom::NodeList

NodeList interface class

Definition

```
#include <xml/xml_node_list.h>
class NodeList {};
```

Description

A class represents a linked list of Node held in the Document.

Mixing of nodes from different Document objects are not supported, which may cause unknown behavior.

Method List

Method	Description
clear	Clear the list
findItem	Find a node by the name
getLength	Get the number of nodes
initialize	Initialize the list
insertFirst	Insert a new Node at the beginning of the list
insertLast	Insert a new Node at the end of the list
isAvailable	Check the availability
item	Get a node by the index
NodeList	Constructor
~NodeList	Destructor
operator[]	Get a node by the index
removeItem	Remove the node from the list
terminate	Terminate

000004892117

Document serial number: 000004892117

©SCEI

Document serial number: 000004892117

Document serial number: 000004892117

Constructors and Destructors

NodeList

Constructor

Definition

```
#include <xml/xml_node_list.h>
namespace sce {
        namespace Xml {
              namespace Dom {
                     class NodeList {
                            NodeList();

                            NodeList(const NodeList &);
                     }
              }
        }
}
```

Return Values

None

Description

Constructor

000004892117

Document serial number: 000004892117

©SCEI

Document serial number: 000004892117

Document serial number: 000004892117

~NodeList

Destructor

Definition

```
#include <xml/xml_node_list.h>
namespace sce {
        namespace Xml {
              namespace Dom {
                     class NodeList {
                           ~NodeList();
                     }
              }
        }
}
```

Return Values

None

Description

Destructor

000004892117

Document serial number: 000004892117

©SCEI

Document serial number: 000004892117

Document serial number: 000004892117

Operator Methods

operator[]

Get a node by the index

Definition

```
#include <xml/xml_node_list.h>
namespace sce {
        namespace Xml {
              namespace Dom {
                     class NodeList {
                            NodeId operator[](
                                   index_t index
                            );
                     }
              }
        }
}
```

Arguments

index	index number that starts with 0

Return Values

A node located at *index*th position in NodeList.

`Dom::invalidNodeId`: In case NodeList is not available, or *index* is invalid.

Description

This is same as `item()`.

000004892117

Document serial number: 000004892117

Document serial number: 000004892117

Document serial number: 000004892117

Public Instance Methods

clear

Clear the list

Definition

```
#include <xml/xml_node_list.h>
namespace sce {
        namespace Xml {
              namespace Dom {
                     class NodeList {
                           void clear();
                     }
              }
        }
}
```

Return Values

None

Description

This clears the list by removing all the nodes.

000004892117

Document serial number: 000004892117

©SCEI

Document serial number: 000004892117

Document serial number: 000004892117

findItem

Find a node by the name

Definition

```
#include <xml/xml_node_list.h>
namespace sce {
        namespace Xml {
              namespace Dom {
                     class NodeList {
                           NodeId findItem(
                                  const String *name
                           );

                           NodeId findItem(
                                  NodeId node
                           );
                     }
              }
        }
}
```

Arguments

name	(in) The node name to be searched
node	(in) The node to be searched

Return Values

The found node

`Dom::invalidNodeId` if not found.

Description

If there are more than one, the first matched node is returned.

000004892117

Document serial number: 000004892117

Document serial number: 000004892117

Document serial number: 000004892117

getLength

Get the number of nodes

Definition

```
#include <xml/xml_node_list.h>
namespace sce {
        namespace Xml {
              namespace Dom {
                     class NodeList {
                           size_t getLength();
                     }
              }
        }
}
```

Return Values

Length of the list

0 : In case the NodeList is unavailable or invalid.

Description

This returns the number of nodes in the list. The range of valid child node indices is 0 to (length-1) inclusive.

000004892117

Document serial number: 000004892117

©SCEI

Document serial number: 000004892117

Document serial number: 000004892117

initialize

Initialize the list

Definition

```
#include <xml/xml_node_list.h>
namespace sce {
        namespace Xml {
              namespace Dom {
                     class NodeList {
                            int initialize(
                                   const Initializer *init
                            );
                     }
              }
        }
}
```

Arguments

init	(in) Pointer to `Initializer` object

Return Values

Value	Description
<0	Error code
`SCE_OK`	Success

Description

This API must be called before another interface of this class is used.

000004892117

Document serial number: 000004892117

Document serial number: 000004892117

Document serial number: 000004892117

insertFirst

Insert a new Node at the beginning of the list

Definition

```
#include <xml/xml_node_list.h>
namespace sce {
        namespace Xml {
              namespace Dom {
                     class NodeList {
                           NodeId insertFirst(
                                  NodeId newNode
                           );
                     }
              }
        }
}
```

Arguments

newNode	(in) Node to be inserted

Return Values

The inserted node

`Dom::invalidNodeId`: In case the `NodeList` is unavailable or invalid

Description

This inserts a new Node at the beginning of the list.

000004892117

Document serial number: 000004892117

©SCEI

Document serial number: 000004892117

Document serial number: 000004892117

insertLast

Insert a new Node at the end of the list

Definition

```
#include <xml/xml_node_list.h>
namespace sce {
        namespace Xml {
              namespace Dom {
                     class NodeList {
                           NodeId insertLast(
                                  NodeId newNode
                           );
                     }
              }
        }
}
```

Arguments

newNode	(in) Node to be inserted

Return Values

The inserted node

`Dom::invalidNodeId`: In case the NodeList is unavailable or invalid

Description

This inserts a new Node at the end of the list.

000004892117

Document serial number: 000004892117

©SCEI

Document serial number: 000004892117

Document serial number: 000004892117

isAvailable

Check the availability

Definition

```
#include <xml/xml_node_list.h>
namespace sce {
        namespace Xml {
              namespace Dom {
                     class NodeList {
                           bool isAvailable() const;
                     }
              }
        }
}
```

Return Values

Value	Description
`true`	Initialized and ready to use
`false`	Not initialized

Description

This returns `true` if this object is available.

000004892117

Document serial number: 000004892117

©SCEI

Document serial number: 000004892117

Document serial number: 000004892117

item

Get a node by the index

Definition

```
#include <xml/xml_node_list.h>
namespace sce {
        namespace Xml {
              namespace Dom {
                     class NodeList {
                           NodeId item(
                                  index_t index
                           );
                     }
              }
        }
}
```

Arguments

index	(in) Index into the collection

Return Values

The node at the *index*th position in the `NodeList`.

`Dom::invalidNodeId`: In case the `NodeList` is unavailable or *index* is invalid.

Description

If index is greater than or equal to the number of nodes in the list, this returns `invalidNodeId`.

000004892117

Document serial number: 000004892117

©SCEI

Document serial number: 000004892117

Document serial number: 000004892117

removeItem

Remove the node from the list

Definition

```
#include <xml/xml_node_list.h>
namespace sce {
        namespace Xml {
              namespace Dom {
                     class NodeList {
                           NodeId removeItem(
                                  NodeId node
                           );
                     }
              }
        }
}
```

Arguments

node	(in) The node to be deleted

Return Values

The removed node

`Dom::invalidNodeId`: In case the `NodeList` is unavailable or invalid.

Description

This removes the node from the list.

000004892117

Document serial number: 000004892117

©SCEI

Document serial number: 000004892117

Document serial number: 000004892117

terminate

Terminate

Definition

```
#include <xml/xml_node_list.h>
namespace sce {
        namespace Xml {
                namespace Dom {
                        class NodeList {
                                int terminate();
                        }
                }
        }
}
```

Return Values

Value	Description
SCE_OK	Success

Description

This performs Termination.

000004892117

Document serial number: 000004892117

Document serial number: 000004892117

Document serial number: 000004892117

sce::Xml::Initializer

000004892117

Document serial number: 000004892117

©SCEI

Document serial number: 000004892117

Document serial number: 000004892117

Summary

sce::Xml::Initializer

Object initialization interface class

Definition

```
#include <xml/xml_types.h>
class Initializer {};
```

Description

The instance of this class must be created before other objects of `sce::Xml` are created.

000004892117

Method List

Method	Description
`initialize`	Initialize
`Initializer`	Constructor
`~Initializer`	Destructor
`terminate`	Terminate

Document serial number: 000004892117

©SCEI

Document serial number: 000004892117

Document serial number: 000004892117

Constructors and Destructors

Initializer

Constructor

Definition

```
#include <xml/xml_types.h>
namespace sce {
        namespace Xml {
              class Initializer {
                     Initializer();
              }
        }
}
```

Return Values

None

Description

Constructor

000004892117

Document serial number: 000004892117

©SCEI

Document serial number: 000004892117

Document serial number: 000004892117

~Initializer

Destructor

Definition

```
#include <xml/xml_types.h>
namespace sce {
        namespace Xml {
              class Initializer {
                     ~Initializer();
              }
        }
}
```

Return Values

None

Description

Destructor

000004892117

Document serial number: 000004892117

©SCEI

Document serial number: 000004892117

Document serial number: 000004892117

Public Instance Methods

initialize

Initialize

Definition

```
#include <xml/xml_types.h>
namespace sce {
        namespace Xml {
              class Initializer {
                     int initialize(
                            const InitParameter *initParam
                     );
              }
        }
}
```

Arguments

initParam	(in) Initialize parameter

Return Values

Value	Description
<0	Error code
`SCE_OK`	Success

Description

This performs initialization and must be called before the `Initializer` object is used.

000004892117

Document serial number: 000004892117

Document serial number: 000004892117

Document serial number: 000004892117

terminate

Terminate

Definition

```
#include <xml/xml_types.h>
namespace sce {
        namespace Xml {
              class Initializer {
                     int terminate();
              }
        }
}
```

Return Values

Value	Description
SCE_OK	Success

Description

This performs termination.

000004892117

Document serial number: 000004892117

©SCEI

Document serial number: 000004892117

Document serial number: 000004892117

sce::Xml::InitParameter

000004892117

Document serial number: 000004892117

©SCEI

Document serial number: 000004892117

Document serial number: 000004892117

Summary

sce::Xml::InitParameter

Initialization parameter class

Definition

```
#include <xml/xml_types.h>
class InitParameter {};
```

Description

Parameters passed to the `Initializer` class

Field

Public Instance Field

`MemAllocator *allocator`	(in) Memory allocator
`void *userData`	(in) Arbitrary user data This field is passed to an argument of an implementation function of the memory allocator.

Method List

Method	Description
`InitParameter`	Constructor

000004892117

Document serial number: 000004892117

©SCEI

Document serial number: 000004892117

Document serial number: 000004892117

Constructors and Destructors

InitParameter

Constructor

Definition

```
#include <xml/xml_types.h>
namespace sce {
        namespace Xml {
              class InitParameter {
                     inline InitParameter();
              }
        }
}
```

Return Values

None

Description

This is a default constructor that initializes a member with zero.

000004892117

Document serial number: 000004892117

Document serial number: 000004892117

Document serial number: 000004892117

sce::Xml::MemAllocator

000004892117

Document serial number: 000004892117

©SCEI

Document serial number: 000004892117

Document serial number: 000004892117

Summary

sce::Xml::MemAllocator

Memory allocator interface class

Definition

```
#include <xml/xml_types.h>
class MemAllocator {};
```

Description

Memory allocator interface class

Implement each function with a class which inherits this class according to user operation. This class is called when memory allocation/deallocation is required in the library.

Method List

Method	Description
allocate	Memory allocation function
deallocate	Memory deallocation function
MemAllocator	Constructor
~MemAllocator	Destructor

000004892117

Document serial number: 000004892117

©SCEI

Document serial number: 000004892117

Document serial number: 000004892117

Constructors and Destructors

MemAllocator

Constructor

Definition

```
#include <xml/xml_types.h>
namespace sce {
        namespace Xml {
              class MemAllocator {
                     MemAllocator();
              }
        }
}
```

Return Values

None

Description

Constructor

000004892117

Document serial number: 000004892117

©SCEI

Document serial number: 000004892117

Document serial number: 000004892117

~MemAllocator

Destructor

Definition

```
#include <xml/xml_types.h>
namespace sce {
         namespace Xml {
               class MemAllocator {
                      virtual ~MemAllocator();
               }
         }
}
```

Return Values

None

Description

Destructor

000004892117

Document serial number: 000004892117

©SCEI

Document serial number: 000004892117

Document serial number: 000004892117

Public Instance Methods

allocate

Memory allocation function

Definition

```
#include <xml/xml_types.h>
namespace sce {
        namespace Xml {
              class MemAllocator {
                     virtual void *allocate(
                            size_t size,
                            void *userData
                     )=0;
              }
        }
}
```

Arguments

size	(in) Size
userData	(in) Arbitrary data passed by user The returned data is the one passed to `InitParameter::userData`

Return Values

Value	Description
Non-NULL	The beginning pointer of allocated memory area
NULL	Failed to memory allocation

Description

Memory allocation function

Allocate memory and return the pointer when this function is called. Return NULL if failed to allocate the memory.

000004892117

Document serial number: 000004892117

Document serial number: 000004892117

Document serial number: 000004892117

deallocate

Memory deallocation function

Definition

```
#include <xml/xml_types.h>
namespace sce {
        namespace Xml {
              class MemAllocator {
                     virtual void deallocate(
                            void *ptr,
                            void *userData
                     )=0;
              }
        }
}
```

Arguments

ptr	(in) Pointer to be deallocated
userData	(in) Arbitrary data passed by user The returned data is the one passed to `InitParameter::userData`

Return Values

None

Description

Memory deallocation function

000004892117

Document serial number: 000004892117

©SCEI

Document serial number: 000004892117

Document serial number: 000004892117

sce::Xml::Sax

000004892117

Document serial number: 000004892117

©SCEI

Document serial number: 000004892117

Document serial number: 000004892117

Summary

sce::Xml::Sax

Namespace of SAX API

Definition

```
namespace Sax {}
```

Description

Namespace of SAX API

Internal classes, Structures, Namespaces

Item	Description
sce::Xml::Sax::DocumentHandler	SAX event handler interface class
sce::Xml::Sax::Parser	SAX interface class

000004892117

Document serial number: 000004892117

©SCEI

Document serial number: 000004892117

Document serial number: 000004892117

sce::Xml::Sax::DocumentHandler

000004892117

Document serial number: 000004892117

©SCEI

Document serial number: 000004892117

Document serial number: 000004892117

Summary

sce::Xml::Sax::DocumentHandler

SAX event handler interface class

Definition

```
#include <xml/xml_sax_document_handler.h>
class DocumentHandler {};
```

Description

SAX event handler interface class

Implement each function with a class which inherits this class according to user operation.

Method List

Method	Description
`characters`	Receive notification of character data
`DocumentHandler`	Constructor
`~DocumentHandler`	Destructor
`endDocument`	Receive notification of the end of a document
`endElement`	Receive notification of the end of an element
`entityData`	Receive notification of entity information
`fatalError`	Receive notification of a non-recoverable error
`skippedText`	Receive notification of the escaped text
`startDocument`	Receive notification of the beginning of a document
`startElement`	Receive notification of the beginning of an element

000004892117

Document serial number: 000004892117

©SCEI

Document serial number: 000004892117

Document serial number: 000004892117

Constructors and Destructors

DocumentHandler

Constructor

Definition

```
#include <xml/xml_sax_document_handler.h>
namespace sce {
        namespace Xml {
              namespace Sax {
                     class DocumentHandler {
                            inline DocumentHandler();
                     }
              }
        }
}
```

Return Values

None

Description

Constructor

000004892117

Document serial number: 000004892117

©SCEI

Document serial number: 000004892117

Document serial number: 000004892117

~DocumentHandler

Destructor

Definition

```
#include <xml/xml_sax_document_handler.h>
namespace sce {
        namespace Xml {
              namespace Sax {
                     class DocumentHandler {
                            virtual inline ~DocumentHandler();
                     }
              }
        }
}
```

Return Values

None

Description

Destructor

000004892117

Document serial number: 000004892117

©SCEI

Document serial number: 000004892117

Document serial number: 000004892117

Public Instance Methods

characters

Receive notification of character data

Definition

```
#include <xml/xml_sax_document_handler.h>
namespace sce {
        namespace Xml {
              namespace Sax {
                     class DocumentHandler {
                            virtual inline int characters(
                                   void *userData,
                                   const String *ch
                            );
                     }
              }
        }
}
```

Arguments

userData	(out) User defined data set to the parser from the application
ch	(out) Character data

Return Values

Value	Description
Non-zero	Value to abort the parsing

Description

Parser will call this method to report each chunk of character data. SAX parser may return all continuous character data in a single chunk, or they may split it into several chunks; The application must not read from the outside area of the specified range of array. Character data will not be null terminated.

000004892117

Document serial number: 000004892117

©SCEI

Document serial number: 000004892117

Document serial number: 000004892117

endDocument

Receive notification of the end of a document

Definition

```
#include <xml/xml_sax_document_handler.h>
namespace sce {
        namespace Xml {
              namespace Sax {
                     class DocumentHandler {
                            virtual inline int endDocument(
                                   void *userData
                            );
                     }
              }
        }
}
```

Arguments

userData	(out) User defined data set to the parser from the application

Return Values

Value	Description
Non-zero	Value to abort the parsing

Description

The SAX parser will invoke this method only once, and it will be the last method invoked during the parse. The parser shall not invoke this method until it has either abandoned parsing (because of an unrecoverable error) or reached the end of input.

000004892117

Document serial number: 000004892117

Document serial number: 000004892117

Document serial number: 000004892117

endElement

Receive notification of the end of an element

Definition

```
#include <xml/xml_sax_document_handler.h>
namespace sce {
        namespace Xml {
              namespace Sax {
                     class DocumentHandler {
                            virtual inline int endElement(
                                   void *userData,
                                   const String *name
                            );
                     }
              }
        }
}
```

Arguments

userData	(out) User defined data set to the parser from the application
name	(out) Name of the end element.

Return Values

Value	Description
Non-zero	Value to abort the parsing

Description

The SAX parser will call this method at the end of every element (end tag) in the XML document, even when the element is empty (empty element tag). Each `endElement()` event has a corresponding `startElement()` event. End element name will be terminated with NULL.

000004892117

Document serial number: 000004892117

©SCEI

Document serial number: 000004892117

Document serial number: 000004892117

entityData

Receive notification of entity information

Definition

```
#include <xml/xml_sax_document_handler.h>
namespace sce {
        namespace Xml {
              namespace Sax {
                     class DocumentHandler {
                            virtual inline int entityData(
                                   void *userData,
                                   EntityType entityType,
                                   const String *name
                            );
                     }
              }
        }
}
```

Arguments

userData	(out) User defined data set to the parser from the application
entityType	(out) Entity type
name	(out) User defined entity name or built in entity name or resolved entity data.

Return Values

Value	Description
Non-zero	Value to abort the parsing

Description

This event is notified only in the case of `Parser::setResolveEntity(true)`.

000004892117

Document serial number: 000004892117

Document serial number: 000004892117

Document serial number: 000004892117

fatalError

Receive notification of a non-recoverable error

Definition

```
#include <xml/xml_sax_document_handler.h>
namespace sce {
        namespace Xml {
              namespace Sax {
                     class DocumentHandler {
                           virtual inline void fatalError(
                                  void *userData,
                                  int errCode
                           );
                     }
              }
        }
}
```

Arguments

userData	(out) User defined data set to the parser from the application
errCode	(out) Error while scanning the XML document

Return Values

None

Description

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 the parsing will be stopped after this callback.

000004892117

Document serial number: 000004892117

©SCEI

Document serial number: 000004892117

Document serial number: 000004892117

skippedText

Receive notification of the escaped text

Definition

```
#include <xml/xml_sax_document_handler.h>
namespace sce {
        namespace Xml {
              namespace Sax {
                     class DocumentHandler {
                            virtual inline int skippedText(
                                   void *userData,
                                   TokenType tokenType,
                                   const String *text
                            );
                     }
              }
        }
}
```

Arguments

userData	(out) User defined data set to the parser from the application
tokenType	(out) The kind of token that represents the text.
text	(out) Escaped text by the parser

Return Values

Value	Description
Non-zero	Value to abort the parsing

Description

The escaped text is part of XML specifications but not supported by the parser. This unsupported data will be sent to escapeText event, so that the application can extend this functionality for support. Escaped text will not be null terminated.

The possible *tokenType* are as follows

- `tokenPi`
- `tokenDtd`
- `tokenComment`
- `tokenCdata`
- `tokenPiEnd`
- `tokenDtdEnd`
- `tokenCommentEnd`
- `tokenCdataEnd`

The tokens ends with `End` are the ends of escaped text. Otherwise there is still some data following the text in the following events.

000004892117

Document serial number: 000004892117

Document serial number: 000004892117

Document serial number: 000004892117

startDocument

Receive notification of the beginning of a document

Definition

```
#include <xml/xml_sax_document_handler.h>
namespace sce {
        namespace Xml {
              namespace Sax {
                     class DocumentHandler {
                            virtual inline int startDocument(
                                   void *userData
                            );
                     }
              }
        }
}
```

Arguments

userData	(out) User defined data set to the parser from the application

Return Values

Value	Description
Non-zero	Value to abort the parsing

Description

The SAX parser will invoke this method only once, before any other methods in this interface are invoked.

000004892117

Document serial number: 000004892117

©SCEI

Document serial number: 000004892117

Document serial number: 000004892117

startElement

Receive notification of the beginning of an element

Definition

```
#include <xml/xml_sax_document_handler.h>
namespace sce {
        namespace Xml {
              namespace Sax {
                     class DocumentHandler {
                            virtual inline int startElement(
                                   void *userData,
                                   const String *name,
                                   const AttributeList *attrList
                            );
                     }
              }
        }
}
```

Arguments

userData	(out) User defined data set to the parser from the application
name	(out) Name of the start element.
attrList	(out) Attribute list of the start element.

Return Values

Value	Description
Non-zero	Value to abort the parsing

Description

The `Parser` will invoke this method at the beginning of each element (start tag) 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. If the element name has a namespace prefix, the prefix will still be attached. Note that the attribute list provided will contain only attributes with explicit values specified.

000004892117

Document serial number: 000004892117

Document serial number: 000004892117

Document serial number: 000004892117

SCE CONFIDENTIAL

sce::Xml::Sax::Parser

000004892117

Document serial number: 000004892117

©SCEI

Document serial number: 000004892117

Document serial number: 000004892117

Summary

sce::Xml::Sax::Parser

SAX interface class

Definition

```
#include <xml/xml.h>
class Parser {};
```

Description

SAX interface class

Method List

Method	Description
initialize	Initialize
parse	Parse the XML document and let SAX event occur
Parser	Constructor
~Parser	Destructor
reset	Reset
setDocumentHandler	Set the SAX event handler
setResolveEntity	Set entity resolving mode
setSkipIgnorableWhiteSpace	Set text skip operation
setUserData	Set user data
terminate	Terminate

000004892117

Document serial number: 000004892117

©SCEI

Document serial number: 000004892117

Document serial number: 000004892117

Constructors and Destructors

Parser

Constructor

Definition

```
#include <xml/xml.h>
namespace sce {
        namespace Xml {
              namespace Sax {
                     class Parser {
                           Parser();
                     }
              }
        }
}
```

Return Values

None

Description

Constructor

000004892117

Document serial number: 000004892117

©SCEI

Document serial number: 000004892117

Document serial number: 000004892117

~Parser

Destructor

Definition

```
#include <xml/xml.h>
namespace sce {
        namespace Xml {
              namespace Sax {
                     class Parser {
                           ~Parser();
                     }
              }
        }
}
```

Return Values

None

Description

Destructor

000004892117

Document serial number: 000004892117

©SCEI

Document serial number: 000004892117

Document serial number: 000004892117

Public Instance Methods

initialize

Initialize

Definition

```
#include <xml/xml.h>
namespace sce {
        namespace Xml {
              namespace Sax {
                     class Parser {
                           int initialize(
                                  const Initializer *init
                           );
                     }
              }
        }
}
```

Arguments

init (in) Pointer to Initializer object

Return Values

Value	Description
<0	Error code
SCE_OK	Success

Description

This API must be called before another interface of this class is used.

000004892117

Document serial number: 000004892117

©SCEI

Document serial number: 000004892117

Document serial number: 000004892117

parse

Parse the XML document and let SAX event occur

Definition

```
#include <xml/xml.h>
namespace sce {
        namespace Xml {
              namespace Sax {
                     class Parser {
                            int parse(
                                   const XmlText *text,
                                   bool isFinal = true
                            );
                     }
              }
        }
}
```

Arguments

text	(in) The chunk of XML data to be parsed.
isFinal	(in) Is the chunk is the final chunk to be parsed in the current XML stream. Default value is `true`.

Return Values

Value	Description
<0	Error code
`SCE_OK`	Success in case *isFinal* is `true`
`resultXmlParseInProgress`	In case of chunk parsing, the current parse with the given chunk is successful

Description

This starts parsing the XML document and notifies a SAX event to a set event handler sequentially.

If the *isFinal* is `true`, then the parser will be reset before returning from the function call.

000004892117

Document serial number: 000004892117

Document serial number: 000004892117

Document serial number: 000004892117

reset

Reset

Definition

```
#include <xml/xml.h>
namespace sce {
        namespace Xml {
              namespace Sax {
                     class Parser {
                           int reset();
                     }
              }
        }
}
```

Return Values

Value	Description
<0	Error code
`SCE_OK`	Success in case `isFinal` is `true`

Description

This reinitializes all the stacks and local variables.

000004892117

Document serial number: 000004892117

Document serial number: 000004892117

Document serial number: 000004892117

setDocumentHandler

Set the SAX event handler

Definition

```
#include <xml/xml.h>
namespace sce {
        namespace Xml {
              namespace Sax {
                     class Parser {
                            void setDocumentHandler(
                                   DocumentHandler *processor
                            );
                     }
              }
        }
}
```

Arguments

processor	(in) SAX event handler which notifies a SAX event

Return Values

None

Description

This sets an object for which `DocumentHandler` is implemented by a user to `Parser`.

000004892117

Document serial number: 000004892117

©SCEI

Document serial number: 000004892117

Document serial number: 000004892117

setResolveEntity

Set entity resolving mode

Definition

```
#include <xml/xml.h>
namespace sce {
        namespace Xml {
              namespace Sax {
                     class Parser {
                            void setResolveEntity(
                                   bool isResolved
                            );
                     }
              }
        }
}
```

Arguments

isResolved	(in) `true` in case the entities have to be resolved while parsing the XML document

Return Values

None

Description

This sets resolving mode for embedded entity reference and character reference.

In the case of `true`, the behavior complies with the XML specification, and `entityData()` of SAX event handler is called for entity reference and character reference.

In the case of `false`, `entityData()` is not called since the entity reference and character reference are recognized as a normal character string.

The default value of libxml is `false`.

000004892117

Document serial number: 000004892117

©SCEI

Document serial number: 000004892117

Document serial number: 000004892117

setSkipIgnorableWhiteSpace

Set ignorable white spaces treatment mode

Definition

```
#include <xml/xml.h>
namespace sce {
        namespace Xml {
              namespace Sax {
                     class Parser {
                           void setSkipIgnorableWhiteSpace(
                                  bool isSkipped
                           );
                     }
              }
        }
}
```

Arguments

isSkipped	(in) `true` to skip ignorable white spaces

Return Values

None

Description

This sets the ignorable white spaces handling mode.

The default value of libxml is `true`.

000004892117

Document serial number: 000004892117

Document serial number: 000004892117

Document serial number: 000004892117

setUserData

Set user data

Definition

```
#include <xml/xml.h>
namespace sce {
        namespace Xml {
              namespace Sax {
                     class Parser {
                            void setUserData(
                                   void *data
                            );
                     }
              }
        }
}
```

Arguments

data	(in) User defined data set by the application to the parser

Return Values

None

Description

This sets a user defined data to be passed to a SAX event handler.
This data is passed to all the SAX event handlers.

000004892117

Document serial number: 000004892117

©SCEI

Document serial number: 000004892117

Document serial number: 000004892117

terminate

Terminate

Definition

```
#include <xml/xml.h>
namespace sce {
        namespace Xml {
              namespace Sax {
                     class Parser {
                           int terminate();
                     }
              }
        }
}
```

Return Values

Value	Description
SCE_OK	Success

Description

This terminates and destroys any local objects.

000004892117

Document serial number: 000004892117

©SCEI

Document serial number: 000004892117

Document serial number: 000004892117

sce::Xml::SerializeParameter

000004892117

Document serial number: 000004892117

©SCEI

Document serial number: 000004892117

Document serial number: 000004892117

Summary

sce::Xml::SerializeParameter

XML output parameter class

Definition

```
#include <xml/xml_types.h>
class SerializeParameter {};
```

Description

This class controls the XML document output.

Give `SCE_XML_SERIALIZE_OPT_xxxx` to *serializeOption*.

Field

Public Instance Field

`int` *`serializeOption`*	(in) Output control option

Method List

Method	Description
`SerializeParameter`	Constructor

000004892117

Document serial number: 000004892117

Document serial number: 000004892117

Document serial number: 000004892117

Constructors and Destructors

SerializeParameter

Constructor

Definition

```
#include <xml/xml_types.h>
namespace sce {
        namespace Xml {
              class SerializeParameter {
                    SerializeParameter();
              }
        }
}
```

Return Values

None

Description

This is a default constructor that initializes a member with zero.

000004892117

Document serial number: 000004892117

Document serial number: 000004892117

Document serial number: 000004892117

sce::Xml::SimpleData

000004892117

Document serial number: 000004892117

©SCEI

Document serial number: 000004892117

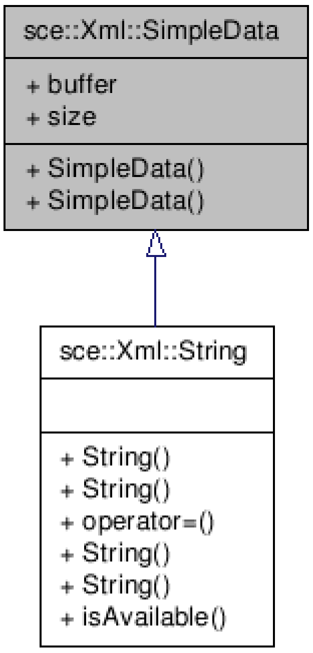
Document serial number: 000004892117

Summary

sce::Xml::SimpleData

Class which holds a pointer to and length of data

Inheritance graph



Definition

```
#include <xml/xml_types.h>
class SimpleData {};
```

Description

This class holds a pointer to and length of data.
It is possible to directly access members.

Field

Public Instance Field

`const char *buffer`	Pointer to data
`size_t size`	Length of data

Method List

Method	Description
SimpleData	Constructor

000004892117

Document serial number: 000004892117

©SCEI

Document serial number: 000004892117

Document serial number: 000004892117

Constructors and Destructors

SimpleData

Constructor

Definition

```
#include <xml/xml_types.h>
namespace sce {
        namespace Xml {
              class SimpleData {
                    SimpleData();

                    SimpleData(
                           const char *data,
                           size_t size
                    );
              }
        }
}
```

Arguments

data	(in) Pointer to data to be set
size	(in) Length of data to be set (Unit: byte)

Return Values

None

Description

This is a default constructor that initializes a member with zero.
This constructor receives the data string, which does not necessarily end with '\0', and the length.

000004892117

Document serial number: 000004892117

©SCEI

Document serial number: 000004892117

Document serial number: 000004892117

sce::Xml::String

000004892117

Document serial number: 000004892117

©SCEI

Document serial number: 000004892117

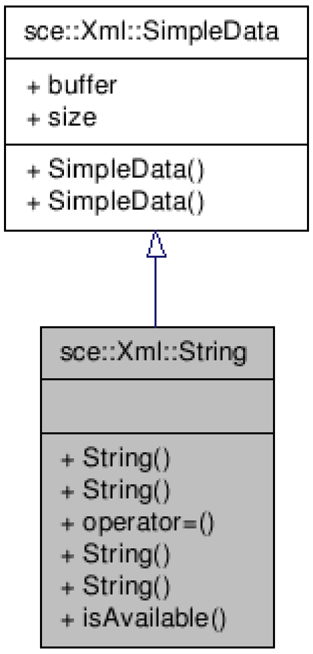
Document serial number: 000004892117

Summary

sce::Xml::String

Class which holds a pointer to and length of character string

Inheritance graph



Definition

```
#include <xml/xml_types.h>
class String : public sce::Xml::SimpleData {};
```

Description

A copied class will point to the same area of this class (copy constructor or substitution operation is used for copying), but does not relate to the pointed-to buffer.

Method List

Method	Description
isAvailable	Check the availability
operator=	This copies the pointer
String	Constructor

000004892117

Document serial number: 000004892117

Document serial number: 000004892117

Document serial number: 000004892117

Constructors and Destructors

String

Constructor

Definition

```
#include <xml/xml_types.h>
namespace sce {
        namespace Xml {
              class String {
                     String(); // (1)

                     String(const String &); // (2)

                     String(
                            const char *str,
                            size_t size
                     ); // (3)

                     String(
                            const char *str
                     ); // (4)
              }
        }
}
```

Arguments

str	(in) Character string to be set
size	(in) Length of character string to be set
str	(in) Character string to be set

Return Values

None

Description

The first one (1) is a default constructor that initializes member variables with zero.

The second (2) is a copy constructor.

The third constructor (3) receives the character string (which does not necessarily end with '\0') and the length.

The forth one (4) is also receives the character string (end with '\0') and the size is measured and set internally.

000004892117

Document serial number: 000004892117

©SCEI

Document serial number: 000004892117

Document serial number: 000004892117

Operator Methods

operator=

Copy the pointed pointer

Definition

```
#include <xml/xml_types.h>
namespace sce {
         namespace Xml {
               class String {
                      String &operator=(const String &);
               }
         }
}
```

Description

Both the source of substitution and the destination of substitution will point to the same area. This is not concerned with the pointed-to buffer.

000004892117

Document serial number: 000004892117

©SCEI

Document serial number: 000004892117

Document serial number: 000004892117

Public Instance Methods

isAvailable

Check the availability

Definition

```
#include <xml/xml_types.h>
namespace sce {
         namespace Xml {
               class String {
                      inline bool isAvailable() const;
               }
         }
}
```

Return Values

Value	Description
`true`	Available
`false`	Unavailable

Description

This returns `true` if the holding content is valid.

000004892117

Document serial number: 000004892117

Document serial number: 000004892117

Document serial number: 000004892117

sce::Xml::Util

000004892117

Document serial number: 000004892117

©SCEI

Document serial number: 000004892117

Document serial number: 000004892117

Summary

sce::Xml::Util

Namespace of utility

Definition

```
namespace Util {}
```

Description

Namespace of utility

List of Functions

Function	Description
strResult	Convert result error number to string

000004892117

Document serial number: 000004892117

©SCEI

Document serial number: 000004892117

Document serial number: 000004892117

Functions

strResult

Convert result error number to string

Definition

```
#include <xml/xml.h>
namespace sce {
        namespace Xml {
              namespace Util {
                     const char *strResult(
                            int result
                     );
              }
        }
}
```

Arguments

result	(in) Result number

Return Values

String expression of the given result

Description

This converts result error number to string.

000004892117

Document serial number: 000004892117

Document serial number: 000004892117